

JPC ROM for the HP-71B

Assembly Listings

Version X

JPC ROM Source File Notes

The JPC ROM is a great extension of the HP-71B system. It was developed between 1986 and 1989 by the PPC-Paris User Club. This release is based to the JPC ROM version X (note that VER\$ says Ex), which is an unofficial version made by Rodger Rosenbaum (probably between 1989 and 1992) based on the regular version D. Note that an official revision E did exist (released by PPC-Paris on end of 1989), but was not widely distributed. The so-called version 'Ex' is not related to this late official version E.

This version includes the extended HPIL messages and the unrestricted PEEK\$/POKE functions, but not the KEYWAIT\$, SCROLL and MSG\$ functions because there are available from HP application ROM or from the HP User's Lib. They were included in JPC ROM rev X for convenience only but are not part of the JPC LEX.

List of JPC LEX modules, in order of appearance in the binary code:

Name	Description (see notes below)
-----	-----
tables	headers, main table, text table (1) (6)
msg	message table (1)
decomp	decompile routines (1)
parse	parse routines (1) (6)
jumper	HPIL ROM jumper routine (2)
ppoll	NLOOP, PPOLL, SRQ, SLEEP
mmlex	MAXD, MEMM, MEMD, MAXM
rw	WREC, RREC
ddir	DDIR, PDIR (3) (6)
files	FILESIZE
polhnd	poll handler (1) (6)
finput	FINPUT (4)
adfcn	ADSIZE, ADFIND, ADCREATE, ADDELETE, ADGET, ADPUT (3)
attn	ATTN
blist	PBLIST, DBLIST, RENUMREM (6)
combarr	PERM, COMB
date	DATESTR\$, DATEADD, DDAYS, DOW\$, DOW, DMY, MDY (6)
des	INVERSE, PAINT, CONTRAST
desal	HTA\$, ADBUF\$, ASC\$, ATH\$, RED\$ (6)
div	PHI, LCM, GCD, FPRIM, NPRIM, PRIM (6)
drive	LXON, LXOFF (6)
endup	ENDUP, EXECUTE, STARTUP\$, ENDUP\$
exit	EXIT
fkey	FKEY
find	FIND (6)
forma	SPACE\$, REDUCE\$, CENTER\$, FORMAT\$, CESURE\$ (6)
frac	FRAC\$
graph	GPSET, GLINE
hms	HR, HMS, HMSSUB, HMSADD
ka	KA (5) (6)
maplex	MAP\$, MAP
marge	MARGIN
menu	MENU
print	BELL, BOLD, PCR, ESC\$, PFF, PLF, MODE, PERF, PAGELEN, UNDERLINE, WRAP (6)
rep	REPLACE\$
roman	ROMAN (3)
rplc	alternate replace (former RPLC\$)
scan	TOKEN, ENTRY\$
stk	STACK
struc2	SELECT, CASE, LEAVE, END, WHILE, REPEAT, LOOP, UNTIL, IF, ELSE
swap	VARSWAP

shrink	SHRINK
sysop	NEXTOP\$, OPCODE\$ (5)
sysed	SYSEEDIT (5) (6)
posi	POSTI (6)
edit	extended EDIT
pk	unrestricted PEEK\$/POKE functions (was removed in original rev. X)
ilmsg	HPIL extended messages

Notes:

1. The tables, msg, decomp, parse and polhnd modules were rebuilt from the JPC LEX binary and from the various JPC source files.
2. The jumper module is the original public code from HP to access the HPIL ROM.
3. The adfcn, roman and ddir modules come from the JPC LEX disassembly made by Rodger Rosenbaum, as no source files are available.
4. The finput module source was recovered from the JPC Journal #43 and checked versus the actual binary code present in the JPC LEX.
5. The source file of the ka module is based on a personal private archive file of an early version. The source files of the sys modules are based on a private file kept by Jan Buitenhuis. The ka and sys modules were unpublished up to now.
6. Changes made by Roger Rosenbaum.

All the other source files are based on published material. Note that most comments are in French, I wrote new comments in English when needed but I didn't attempt to translate the original ones, it would be too much work for too little benefit.

Many modules actually include bug fixes or enhancements that were never published in the JPC Journal nor made available in the Swap Disks. The source files were updated accordingly by comparing the original published source files with the actual code present in JPC ROM rev X. When possible, credit was given to the author(s) but the changes between the original published listings and actual JPC LEX code are not documented, except for the rev X changes made by R. Rosenbaum and a few special items.

Minor changes were done in order to assemble the JPC LEX with the SASM/SLOAD tools from HP. The build batch file (bldjpc.bat) and the loader command file (jpclex.m) are provided. I didn't bother with a makefile because the complete build just takes a few seconds on modern PC. You have also to assemble the HP71 entry points definition file, see the comment lines in bldjpc.bat file.

The JPC ROM source files were collected and edited from various sources (Swap Disks, JPC Journals, personal archives, private communications, JPC ROM disassembly from Rodger Rosenbaum) by Jean-Francois Garnier during the year 2005, and the JPC ROM version X was finally rebuilt in May 2006.

For additional information on the JPC ROM, please visit:
<http://membres.lycos.fr/jeffcal/jpcrom.html>

Token	Keyword	JPC #	File	Authors
E1/3E	NLOOP(?#)	37	ppoll.a	Jean-Francois Garnier
E1/3F	PARPOLL(?#)	37	ppoll.a	Jean-Francois Garnier
E1/40	SLEEP	37	ppoll.a	Jean-Francois Garnier
E1/41	SRQ(?#)	37	ppoll.a	Jean-Francois Garnier
E1/42	END	52	struc2.a	Pierre David, Janick Taillandier
E1/43	WHILE	52	struc2.a	Pierre David, Janick Taillandier
E1/44	REPEAT	52	struc2.a	Pierre David, Janick Taillandier
E1/45	UNTIL	52	struc2.a	Pierre David, Janick Taillandier
E1/46	LEAVE	52	struc2.a	Pierre David, Janick Taillandier
E1/47	VARSWAP	31	swap.a	Jean-Jacques Moreau
E1/48	ENTRY\$(,\$,?#)	31	scan.a	Jean-Jacques Moreau
E1/49	TOKEN\$(,\$,?#)	31	scan.a	Jean-Jacques Moreau
E1/4B	FIND *P *I	45	find.a	Jean-Jacques Moreau, Janick Taillandier
E1/4C	MAP\$(,\$,\$,?)	46	maplex.a	Tapani Tarvainen
E1/4D	MAP	46	maplex.a	Tapani Tarvainen
E1/4E	GLINE	35	graph.a	Pierre David
E1/4F	GPSET	35	graph.a	Pierre David
E1/51	SHRINK	35	shrink.a	Jean-Jacques Moreau
E1/52	FPRIM(#,?#)	48	divi.a	Guy Toublanc
E1/53	NPRIM(#,?)	48	divi.a	Guy Toublanc
E1/54	GCD(##,##,?#,?#,?#,?#,?#,?#,?#,?#)	48	divi.a	Guy Toublanc
E1/55	PHI(#)	48	divi.a	Guy Toublanc
E1/56	LCM(##,##,?#,?#,?#,?#,?#,?#,?#,?#)	48	divi.a	Guy Toublanc
E1/57	PRIM(#,?#)	48	divi.a	Olivier Arbey, Guy Toublanc
E1/58	FRACS(#,?#)	42	frac.a	Guy Toublanc
E1/59	POSI(\$,##/\$,?#/\$)	37	posi.a	Pierre David, Jean-Pierre Bondu, Janick Taillandier
E1/5A	DBLIST	53	blist.a	Jean-Pierre Bondu, Pierre David, Janick Taillandier
E1/5B	PBLIST	53	blist.a	Jean-Pierre Bondu, Pierre David, Janick Taillandier
E1/5C	RENUMREM	53	blist.a	Jean-Pierre Bondu
E1/5D	FINPUT	43	finput.a	Pierre David, Janick Taillandier
E1/5E	RREC\$(#,\$,##/\$)	45	rw.a	Michel Martinet
E1/5F	WREC	45	rw.a	Michel Martinet
E1/60	LOOP	52	struc2.a	Pierre David, Janick Taillandier
E1/61	SELECT	52	struc2.a	Pierre David, Janick Taillandier
E1/62	CASE	52	struc2.a	Pierre David, Janick Taillandier
E1/63	IF *I	52	struc2.a	Pierre David, Janick Taillandier
E1/64	ELSE	52	struc2.a	Pierre David, Janick Taillandier
E1/65	KA		ka.a	Pierre David
E1/66	ADCREATE		adfcn.a	Pierre David
E1/67	ADDELETE		adfcn.a	Pierre David
E1/68	ADFIND(\$,\$,?#)		adfcn.a	Pierre David
E1/69	ADGET		adfcn.a	Pierre David
E1/6A	ADPUT		adfcn.a	Pierre David
E1/6B	ADSIZE(\$,?#)		adfcn.a	Pierre David
E1/6C	ROMAN		roman.a	Pierre David, Janick Taillandier
E1/6E	DDIR		ddir.a	Jean-Jacques Dhenin
E1/6F	PDIR		ddir.a	Jean-Jacques Dhenin

Tools for JPC ROM Assembly

The JPC ROM was originally developed with the Areuh tools (Assembler, Linker, and misc tools) made by Pierre David and Janick Taillandier on 1986-1987. These tools were running on a HP Integral PC (using the HP-UX OS and external hard drive units, a quite large and expensive configuration), later on Pierre and Janick ported their tools to the PC world. The source of these tools in C language are still freely available on the Web, but the binaries for DOS are no longer available. I tried to compile the Areuh tools with a contemporary compiler (the Borland Turbo C that I intensively used at the time). It worked well, and I assembled small HP-71 assembly language LEX but I got severe problems when trying to assemble and link large files such as the JPC LEX modules. I got “Not Enough Memory” or “Stack Overflow” errors that are probably due to the segmented memory management in 16-bit DOS applications.

It could be possible to assemble Areuh tools in 32 bit mode to get rid of these problems, but first it was not my goal to debug the Areuh implementation for DOS and then it would have not been in line with the spirit of my JPC ROM rebuild project (i.e. rebuild the JPC ROM with the tools of the time).

So I decided to use the HP SASM/SLOAD tools. They were contemporary with the end of the JPC ROM project (1989) although there were only freely available in 1991 with release of the free HP-48 development tools. Furthermore, SASM and SLOAD are very close to the tools that HP used for the HP-71 development and give listing files similar to the HP-71 IDS documents. With these tools, I successfully assembled the whole JPC ROM without major problems.

It should be noted that SASM/SLOAD have some syntax limitations relative to the Areuh tools, so I had to introduce minor changes in the source code for correct assembly (but the final binary code is unchanged). One of these differences is that SASM accepts the ‘*’ character as a comment line delimiter only in column 1 or 2, whereas Areuh accepts the ‘*’ character in any column. So I had to reformat many comment lines in various source files.

The HP tools, as originally released for the HP-48, are still freely available on the Web. For your convenience, I included the SASM/SLOAD subset for use with the HP-71, with relevant documentation, in the TOOLS71.ZIP file on my site.

The build batch file and the linker command file are listed in following pages.

bldjpc.bat batch file

set SASM_LIB=.

rem to do once:

rem sasm -x hp71ep.a

rem ren hp71ep.o hp71ep.b

sasm -x -n8 tables.a
sasm -x -n8 msg.a
sasm -x -n8 decomp.a
sasm -x -n8 parse.a
sasm -x -n8 jumper.a
sasm -x -n8 ppoll.a
sasm -x -n8 mmlex.a
sasm -x -n8 rw.a
sasm -x -n8 ddir.a
sasm -x -n8 files.a
sasm -x -n8 polhnd.a
sasm -x -n8 finput.a
sasm -x -n8 adfcn.a
sasm -x -n8 attn.a
sasm -x -n8 blist.a
sasm -x -n8 combarr.a
sasm -x -n8 date.a
sasm -x -n8 des.a
sasm -x -n8 desal.a
sasm -x -n8 div.a
sasm -x -n8 drive.a
sasm -x -n8 endup.a
sasm -x -n8 exit.a
sasm -x -n8 fkey.a
sasm -x -n8 find.a
sasm -x -n8 forma.a
sasm -x -n8 frac.a
sasm -x -n8 graph.a
sasm -x -n8 hms.a
sasm -x -n8 ka.a
sasm -x -n8 maplex.a
sasm -x -n8 marge.a
sasm -x -n8 menu.a
sasm -x -n8 print.a
sasm -x -n8 rep.a
sasm -x -n8 roman.a
sasm -x -n8 rplc.a
sasm -x -n8 scan.a
sasm -x -n8 stk.a
sasm -x -n8 struc2.a
sasm -x -n8 swap.a
sasm -x -n8 shrink.a
sasm -x -n8 sysop.a
sasm -x -n8 sysed.a
sasm -x -n8 posi.a
sasm -x -n8 edit.a
sasm -x -n8 pk.a
sasm -x -n8 ilmsg.a

sload -H jpclex.m

jpclex.m linker command file

OU JPCLEX
LL JPCLEX.LR
SU XR
RE tables.o
RE msg.o
RE decomp.o
RE parse.o
RE jumper.o
RE ppoll.o
RE mmlex.o
RE rw.o
RE ddir.o
RE files.o
RE polhnd.o
RE finput.o
RE adfcn.o
RE attn.o
RE blist.o
RE combarr.o
RE date.o
RE des.o
RE desal.o
RE div.o
RE drive.o
RE endup.o
RE exit.o
RE fkey.o
RE find.o
RE forma.o
RE frac.o
RE graph.o
RE hms.o
RE ka.o
RE maplex.o
RE marge.o
RE menu.o
RE print.o
RE rep.o
RE roman.o
RE rplc.o
RE scan.o
RE stk.o
RE struc2.o
RE swap.o
RE shrink.o
RE sysop.o
RE sysed.o
RE posi.o
RE edit.o
RE pk.o
RE ilmsg.o
SE hp71ep.b

Saturn Loader, Ver. %I%, %G%

Output Module:

Module=JPCLEX

Start=00000 End=0CF8A Length=0CF8B Symbols=2055 References=1819
Date=Tue May 23 09:21:18 2006 Title=

Source modules:

Module=tables.o

Start=00000 End=00A17 Length=00A18
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE TABLES

Module=msg.o

Start=00A18 End=00BFC Length=001E5
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE MSG

Module=decomp.o

Start=00BFD End=00F62 Length=00366
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE DECOMP

Module=parse.o

Start=00F63 End=01637 Length=006D5
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE PARSE

Module=jumper.o

Start=01638 End=0172C Length=000F5
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE JUMPER

Module=ppoll.o

Start=0172D End=0182A Length=000FE
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE PPOLL

Module=mmlex.o

Start=0182B End=01A7B Length=00251
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE MMLEX

Module=rw.o

Start=01A7C End=01CAF Length=00234
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE RW

Module=ddir.o

Start=01CB0 End=024A8 Length=007F9
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE DDIR

Module=files.o

Start=024A9 End=02639 Length=00191
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE FILES

Module=polhnd.o

Start=0263A End=031B7 Length=00B7E
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE POLHND

Module=finput.o

Start=031B8 End=03F8D Length=00DD6
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE FINPUT

Module=adfcfn.o

Start=03F8E End=04969 Length=009DC
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE ADFCN

Module=attn.o
Start=0496A End=04988 Length=0001F
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE ATTN

Module=blis.o
Start=04989 End=054F2 Length=00B6A
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE BLIST

Module=combarr.o
Start=054F3 End=05651 Length=0015F
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE COMBARR

Module=date.o
Start=05652 End=05AB4 Length=00463
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE DATE

Module=des.o
Start=05AB5 End=05C91 Length=001DD
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE DES

Module=desal.o
Start=05C92 End=05F89 Length=002F8
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE DESAL

Module=div.o
Start=05F8A End=063D7 Length=0044E
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE DIV

Module=drive.o
Start=063D8 End=064B0 Length=000D9
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE DRIVE

Module=endup.o
Start=064B1 End=065EE Length=0013E
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE ENDUP

Module=exit.o
Start=065EF End=066E5 Length=000F7
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE EXIT

Module=fkey.o
Start=066E6 End=06748 Length=00063
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE FKEY

Module=find.o
Start=06749 End=06991 Length=00249
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE FIND

Module=forma.o
Start=06992 End=06D0C Length=0037B
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE FORMA

Module=frac.o
Start=06D0D End=06F64 Length=00258
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE FRAC

Module=graph.o
Start=06F65 End=0712F Length=001CB
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE GRAPH

Module=hms.o
Start=07130 End=07333 Length=00204
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE HMS

Module=ka.o
Start=07334 End=0843E Length=0110B
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE KA

Module=maplex.o
Start=0843F End=086F0 Length=002B2
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE MAPLEX

Module=marge.o
Start=086F1 End=0878C Length=0009C
Date=Tue May 23 09:21:17 2006 Title=JPC LEX: MODULE MARGE

Module=menu.o
Start=0878D End=089F5 Length=00269
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE MENU

Module=print.o
Start=089F6 End=08D9E Length=003A9
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE PRINT

Module=rep.o
Start=08D9F End=09003 Length=00265
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE REP

Module=roman.o
Start=09004 End=09733 Length=00730
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE ROMAN

Module=rplc.o
Start=09734 End=09DBC Length=00689
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE RPLC

Module=scan.o
Start=09DBD End=09F18 Length=0015C
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE SCAN

Module=stk.o
Start=09F19 End=09F99 Length=00081
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE STK

Module=struc2.o
Start=09F9A End=0A7BA Length=00821
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE STRUC2

Module=swap.o
Start=0A7BB End=0A888 Length=000CE
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE SWAP

Module=shrink.o
Start=0A889 End=0A9E1 Length=00159
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE SHRINK

Module=sysop.o
Start=0A9E2 End=0B744 Length=00D63
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE SYSOP

Module=sysed.o
Start=0B745 End=0C690 Length=00F4C
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE SYSED

Module=posi.o
Start=0C691 End=0C795 Length=00105
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE POSI

Module=edit.o
Start=0C796 End=0C98E Length=001F9
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE EDIT

Module=pk.o
Start=0C98F End=0CB2D Length=0019F
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE PK

Module=ilmsg.o
Start=0CB2E End=0CF8A Length=0045D
Date=Tue May 23 09:21:18 2006 Title=JPC LEX: MODULE ILMMSG

Module=hp71ep.b
Start=0CF8B Module Contains No Code
Date=Thu Jan 26 09:39:56 2006 Title=HP71 Entry Points

Saturn Cross Reference Listing

```

#CK                = 03356 - 00F65

-CHAR              = 1527D -
-LINE              = 15275 - 07B2F 08865

1/X15              = 0C33E - 06DE7

?PRFI+            = 17380 - 0854A
?PRFIL            = 1737E -

A-MULT            = 1B349 - 01A5D 09E86
ACBFND            = 090E7 - 02D6F
ACCEPT            = 0450F -
ACOS12            = 0DBD3 -
ACOS15            = 0DBD7 -
ACTIVE            = 2F5A8 -
AD15M             = 0C366 -
AD15S             = 0E19D -
AD15s             = 0C369 -
AD2-12            = 0C35F - 06F1E 0732B
AD2-15            = 0C363 - 05562 072D5
ADBUFe            = 05D60 - 00098
ADCRed            = 00DF7 - 04597
ADCRee            = 045A1 - 00425
ADCREp            = 013FF - 0459C
ADDELd            = 00E11 - 0461F
ADDELe            = 04629 - 0042E
ADDELp            = 01420 - 04624
ADDF              = 0C372 -
ADDONE            = 0C330 - 0731B
ADDP              = 03A03 -
ADDRCK            = 1C5A5 - 01AE0 01BE7 0AA06 0AB79 0B959 0C9E8 0CAC5
ADDRSS            = 0F527 - 0335A 041F5
ADFINE            = 0449B - 00437
ADGETd            = 00E21 - 046A7
ADGETe            = 046B1 - 00440
ADGETp            = 01430 - 046AC
ADHEAD            = 181B7 - 05909 05DC3 05EF8 065EA 06AEA 06B49 06EA3 08B04 09980
                  + 0B6F9 0BEFF

ADJA              = 1289A -
ADJN              = 12825 -
ADPUTd            = 00E39 - 04760
ADPUTe            = 0476A - 00449
ADPUTp            = 01448 - 04765
ADRS40            = 0F52B -
ADRS50            = 0F551 - 06F8C
ADRS80            = 0F567 -
ADRSUB            = 0F4CF -
ADSIZE            = 043AB - 00452
ALINFO            = 01531 -
ALLDUN            = 04BEF -
ALMSRV            = 1257D - 02D00
ALRM1             = 2F719 -
ALRM2             = 2F725 -
ALRM3             = 2F731 -
ALRM4             = 2F73D -
ALRM5             = 2F749 -
ALRM6             = 2F755 -

```

ANN1.5 = 2E101 -
 ANNAD1 = 2E100 -
 ANNAD2 = 2E102 -
 ANNAD3 = 2E34C -
 ANNAD4 = 2E34E -
 ARG12 = 0D67B -
 ARG15 = 0D67F -
 ARGERR = 0BF19 - 01B34 0564D 0588F 05B66 05D14 060B7 06AF1 06F76 085BC
 + 08788 08C59 08DD5 09911 09E16 09F36 0CA56
 ARGF = 0D6A4 -
 ARGPR+ = 0E8EB -
 ARGPRP = 0E8EF - 07154
 ARGST- = 0E910 -
 ARGSTA = 0E90C -
 ARITH = 061E0 - 00D94
 ARRYCK = 0366A -
 ARRe = 054F7 - 00143
 ARYDC = 05178 -
 ARYELM = 0B5A7 -
 ARYSIZ = 0B61B -
 ASCICK = 0514E -
 ASCII = 0079B -
 ASCe = 05DD9 - 000A1
 ASIN12 = 0DBC8 -
 ASIN15 = 0DBCC -
 ASLW3 = 0ED21 - 08BEB
 ASLW4 = 0ED1E - 08BE4
 ASLW5 = 0ED1B - 088C8 088D2
 ASNMNT = 0F5E0 -
 ASRW3 = 0ED10 -
 ASRW4 = 0ED0D -
 ASRW5 = 0ED0A - 089C5 089CF
 ATAN15 = 0DBBE -
 ATHe = 05E4F - 000AA
 ATNCLR = 00510 - 03D78 03F7B 0784A 0BADD 0C4ED
 ATNDIS = 2F441 - 04976
 ATNFLG = 2F442 - 01C91 01FD9 02368 0600D 06330
 ATTNe = 04974 - 000D7
 AUTINC = 2F6CB -
 AVE=C = 18BBB - 06531
 AVE=D1 = 18BB8 - 024AB 03541 03A88 03F6D 0448A 06779 0A823
 AVM2DS = 01C01 -
 AVMEME = 2F599 - 0225A 02312 06665
 AVMEMS = 2F594 - 01F64 0213D 023D5 0240E 0242B 0357D 047A9 047DF 0482C
 + 04885 04CFC 04F13 04FF1 0501B 05037 06BA9 074A7 074BD
 + 0788D 07A70 0831B 08368 08E5B 09801 0983A 098BE 0994A
 + 0BC6C
 AVS2DS = 09708 -
 AVS=D0 = 01494 -
 BACK = 1BA4F -
 BACK1B = 13B0C -
 BACK2B = 13B0A -
 BACK3B = 13B08 -
 BASCHA = 07741 - 04B00
 BASCHK = 0773E -
 BASE = 0F953 -
 BASICs = 000B5 -
 BCKSPC = 151E7 -
 BEEP = 0EA6E -

BELLe = 08A7D - 001EE
 BF2DS+ = 01C08 -
 BF2DSP = 01C0E - 06906 088EF 0BEAC
 BF2STK = 18663 -
 BIASA+ = 0D52D -
 BIASC+ = 0D540 -
 BIG = 0B747 -
 BLANKC = 07818 -
 BLDBIT = 019BC -
 BLDCON = 16279 -
 BLDDSP = 01898 - 02CCA 06922
 BLDLCD = 0189C -
 BLNC+ = 07810 -
 BLNKCK = 051C1 - 00F31
 BOLDe = 08A8E - 001F7
 BOPNM- = 1B864 -
 BP+C = 0EB40 -
 BRT30 = 0DBE3 -
 BRTF = 0DC15 -
 BSCEX2 = 0743A -
 BSCEXC = 07437 -
 BSCEXT = 075CF -
 BSERR = 0939A - 00A13 016C9 01BB3 01D32 01EE2 02559 0314D 0328F 0336A
 + 037FA 03B04 04547 0499D 06357 064A5 067B1 07477 0787B
 + 07B8B 0813D 0900A 0A374 0BC5C 0BF56 0C937

 BitsOK = 00001 -
 BldIM+ = 1BA6A -
 BldIMA = 1BA66 -
 BldIMG = 1BA68 -

 C+A2D1 = 1C053 -
 CALBIN = 18D8C - 03009
 CALL = 18DAE -
 CALLP = 0389C -
 CALSTK = 2F5AD -
 CASEd = 00D6E - 0A1E2
 CASEe = 0A1EC - 00401
 CASEp = 01113 - 0A1E7
 CAT\$20 = 06746 - 01F56
 CAT\$83 = 068D4 -
 CAT\$90 = 068D7 -
 CATC++ = 03F66 -
 CATCH+ = 03F69 -
 CATCHR = 03F70 -
 CATEDT = 06435 -
 CENTEe = 06AFA - 001C1
 CESURE = 06C55 - 001CA
 CHAIN+ = 07C12 -
 CHAIN- = 07C1C -
 CHEDIT = 14C99 - 03C41 07ADF 0C4A1
 CHIRP = 0EC5A - 02CA7 07F70
 CHKEOL = 13D6D -
 CHKSUM = 1CDEE -
 CHKmem = 012C7 - 085D9 09A35
 CHN#SV = 2F96F -
 CHNHED = 0F579 -
 CHNLST = 2F5BE -
 CK"ON" = 076AD - 02BF3 04C08 06952
 CKINF- = 18534 -
 CKINFO = 18542 - 023CE 04FEA

CKSREQ = 00721 - 01813 02D1A 03CAD 07714 0B9C9
CKSUM2 = 0AA81 -
CKSUM3 = 153A9 -
CKSUM4 = 1DBA6 -
CLASSA = 0D590 -
CLCBFR = 2F576 -
CLCSTK = 2F585 -
CLLINK = 1A72C -
CLOSEA = 120E4 -
CLOSEF = 12087 -
CLRFRC = 0C6F4 - 05643 060E4 06ED0 072A7
CLRPRM = 04827 -
CMD1ST = 01654 -
CMDFND = 01693 -
CMDINI = 016D1 -
CMDPR" = 01627 -
CMDPTR = 2F6D4 -
CMDS20 = 01672 -
CMOSTV = 0168F -
CMOSTW = 2F438 -
CMPT = 125B2 - 059F6
CNFFND = 109AC - 01FF2
CNFLCT = 0BD15 - 0A123
CNTADR = 2F67E -
CNVUCR = 152A7 - 037D1 04012 0755C
CNVWUC = 03FB8 - 01247
COLDST = 00000 -
COLLAP = 091FB - 01B26 03446 048B2 07518 07A94 0B778
COMBe = 05502 - 0014C
COMCK = 036CD - 00FCE 0109E 014A1 0153F 015A4
COMCK+ = 032AE - 00F8E 00FFE 01180 012BF 012F9 01333 013F0
COMCKO = 032AA -
COMPL# = 07870 -
CONCOM = 0467E -
CONF = 10212 -
CONFST = 2F9E6 -
CONTRe = 05C6B - 000FB
CONVUC = 152AA - 04091 048D6 05D2B 07778 07AB8 0C58F 0CA75
COPYu = 08269 - 07BED 0C8F0
CORUPT = 09083 - 0C91B 0CB61
COS12 = 0D721 -
COS15 = 0D725 -
COUNTC = 1C346 -
CPL#10 = 07887 - 05351 054D9
CR = 2C000 -
CRDFIL = 1D21D -
CREATE = 115A7 -
CRETf+ = 084C4 - 01DE5 045CE 04B87 07C34 0BF4C 0C8A0
CRFSB- = 11664 -
CRLFND = 0229E - 08888 089A7 0B922
CRLFOF = 02296 - 03D6D 0C55E
CRLFSD = 022A2 - 07F77
CRTF = 116C1 -
CSL9R0 = 1BA0D -
CSLC1 = 1B441 -
CSLC10 = 1B418 -
CSLC11 = 1B41B -
CSLC12 = 1B41E -
CSLC13 = 1B421 -
CSLC14 = 1B424 -

CSLC15 = 1B427 -
 CSLC2 = 1B43E -
 CSLC3 = 1B43B -
 CSLC4 = 1B438 -
 CSLC5 = 1B435 - 022BB 028E6 028EF 05F0C 069A0 08E80 0C6E0
 CSLC6 = 1B432 - 022C2
 CSLC7 = 1B42F -
 CSLC8 = 1B42C -
 CSLC9 = 1B415 -
 CSLW3 = 0ED43 -
 CSLW4 = 0ED40 -
 CSLW5 = 0ED3D - 067EA 09DF2 0ABD5
 CSPEED = 2F977 -
 CSRC1 = 1B427 -
 CSRC10 = 1B432 -
 CSRC11 = 1B435 -
 CSRC12 = 1B438 -
 CSRC13 = 1B43B -
 CSRC14 = 1B43E -
 CSRC15 = 1B441 -
 CSRC2 = 1B424 -
 CSRC3 = 1B421 -
 CSRC4 = 1B41E -
 CSRC5 = 1B41B - 022B0 028FC 02905 05F80 06AD6 08DCE 0B299 0C714 0C72A
 CSRC6 = 1B418 - 022C9
 CSRC7 = 1B415 -
 CSRC8 = 1B42C -
 CSRC9 = 1B42F -
 CSRW3 = 0ED32 -
 CSRW4 = 0ED2F -
 CSRW5 = 0ED2C - 01880 06930
 CURBOT = 10059 -
 CURDVC = 0A60B - 0C93E
 CURREN = 2F56C - 0536C 0546B 0A402
 CURRL = 2F7E8 - 054E8 067B8
 CURRST = 2F55D - 030AF 049EA 054B1 0A3D8 0C94C
 CURSFL = 151DF - 03C14 07945 07B28 0884E 0C46D 0C68C
 CURSFR = 151D7 -
 CURSOR = 2F47E - 023F9 027F1 0287E 0292D 02A19 02C93
 CURSRD = 100A4 -
 CURSRL = 151CF -
 CURSRR = 151C7 -
 CURSRT = 096C1 - 0691B
 CURSRU = 1009A -
 CURTOP = 10063 -
 CVUCW = 03FBC -
 CkLoop = 1B669 -
 CkLpNC = 1B66D -
 Clear = 00005 -
 CurOff = 00006 -

 D0+2RD = 13A32 -
 D0=AVS = 09B2C - 05A6B 08C02 0978B
 D0=FIB = 13AC5 -
 D0=OBS = 05067 -
 D0=PCA = 09B37 - 054D2
 D0ASC+ = 0982C -
 D0ASCI = 09833 -
 D12R0A = 1BA3C -
 D1=AVE = 18651 - 07B00 0A765

D1@AVS = 01299 -
D1C=R3 = 03047 - 09EA3 09ECA
D1FSTK = 1955D -
D1MST+ = 13E21 -
D1MSTK = 1954E - 02521 0257C 0348F 03E97 03F4A 04491 0C4FA
D=AVME = 1A476 -
D=AVMS = 1A460 - 01904 01C0C 034C9 048C6 058CF 05DBC 05E6A 065B6 06A8B
+ 06AE3 06B12 07AA8 08AFA 0B641 0BEF5
D=WORD = 04C0E -
DATEAe = 05912 - 0025A
DATESe = 058AF - 00251
DATLEN = 0B584 -
DATPTR = 2F692 - 087B6
DAY2JD = 13407 -
DAYYMD = 13335 - 058B5 05938
DBLISe = 049AC - 003B9
DBLPI4 = 0DAFC -
DBLSUB = 0DADD -
DCHX=C = 1B2D0 -
DCHXF = 1B223 -
DCHXW = 0ECD C - 05FE5
DCONTR = 2E3FE - 05C73
DCPLIN = 10108 -
DCRMNT = 1C177 -
DD1CTL = 2E3FF -
DD1END = 2E34C -
DD1ST = 2E300 - 05B3B 05B87
DD2CTL = 2E2FF -
DD2END = 2E260 - 05B1C
DD2ST = 2E200 - 05B31 05B97
DD3CTL = 2E1FF -
DD3END = 2E160 - 05B10
DD3ST = 2E104 - 05BA1
DDAYSe = 05995 - 00263
DDIRd = 00F0F - 01CB0 01CC1
DDIRe = 01CBA - 0046D
DDIRp = 015DB - 01CB5 01CC6
DEBNCE = 00CF7 -
DECHEX = 1B2D2 - 02B7A 02B87
DECP = 0328F -
DEFADC = 052FC -
DEFADR = 2F967 - 02819 02ABB
DELAYT = 2F948 -
DELAYp = 02AC6 - 0129C
DEST = 0F7B0 - 03F43 0A83E
DISINT = 2F470 -
DISPDC = 05450 -
DISPP = 035A4 -
DISPt = 00000 -
DIVF = 0C4B8 -
DMNSN = 0AE39 -
DMYe = 05A8B - 0026C
DONNA = 09656 -
DOPEI\$ = 2F871 - 0342B
DOWe = 059DD - 0027E
DOWse = 059D3 - 00275
DPART2 = 17EA3 -
DPART3 = 17EF8 -
DPOS = 2F94D -
DPVCTR = 0AC50 -

DRANGE = 1B076 - 037EA 0586D 05D21 0C3F8 0C585 0CA5D
 DROPDC = 05470 - 00C0F 00C9B
 DSLEEP = 0056D - 083F2
 DSP\$00 = 185DB - 03EA1 07522 07B0A 0C504
 DSPBFE = 2F540 - 029C8 03E53
 DSPBFS = 2F480 - 027BE 03DAA
 DSPBUF = 09723 - 0885E
 DSPCHA = 01C3E - 03BAB 0B912 0C672
 DSPCHC = 01C3C - 0B8AA
 DSPCHX = 2F674 - 02A57 02AAD 07603 07663 0C523
 DSPCL? = 020B6 -
 DSPCNA = 09721 - 073DE 0B89F
 DSPCNB = 0971F -
 DSPCNO = 09716 - 06911
 DSPDGT = 2F6DD -
 DSPFMT = 2F6DC - 06F33 06F54
 DSPLI+ = 1010F -
 DSPLIN = 10127 -
 DSPMSK = 2F540 - 02944 029D2 03B5F 03BD3 08843 088FD 0C448
 DSPRST = 02443 -
 DSPSET = 2F7B1 -
 DSPSTA = 2F475 - 02A3D
 DSPUPD = 01ADA -
 DSTRDC = 05280 -
 DV15M = 0C4AC -
 DV15S = 0C4B2 -
 DV2-12 = 0C4A8 - 06E3A
 DV2-15 = 0C4AC - 055C3 07299
 DVZNIB = 2F6FC -
 DWIDTH = 2F94F -
 DXP100 = 0CF7F -
 DZP = 00003 -

 EDIT80 = 0A5A5 - 0C8DE
 EDITWF = 0A533 - 0C7E5
 EDITe = 0C79B - 00179
 EDITp = 012AF - 0C796
 EFIELD = 00000 -
 ELSE2e = 0A360 - 00413
 ELSE2p = 01229 - 0A35B
 END2d = 00C14 - 0A26A
 END2e = 0A274 - 002E1
 END2p = 010C3 - 0A26F
 ENDALL = 0769A -
 ENDBIN = 0764B -
 ENDIMG = 1C040 -
 ENDSUB = 195A8 -
 ENDUPe = 064BB - 00128
 ENDUse = 0659A - 0011F
 ENTRYe = 09EC4 - 00317
 EOLCK = 02A7E - 01047 0106D 01278 0128E 01405 0154F 015DD
 EOLCK8 = 02A92 -
 EOLCKR = 02A7A -
 EOLDC = 05402 -
 EOLLEN = 2F95A -
 EOLSCN = 08AA7 - 089EA
 EOLSN5 = 08AB1 -
 EOLSTR = 2F95B -
 EOLXC* = 052EC - 00D36 00DCC 00DFD 00E68 00F11 00F38
 EOLXCK = 05405 - 00D69 01D05 042CB 049FF 051BF 05AC4 07B62 08700 08B7B

```

+ 08CD0 0A37B
ERR#           = 2F7E4 -
ERRADR        = 2F688 -
ERRL#         = 2F7EC -
ERRLCH        = 2F97C -
ERRM$f        = 09806 - 05A7C
ERRRTN        = 074ED -
ERRSUB        = 2F683 -
ESCSEQ        = 023C1 - 07930
ESCSTA        = 2F47B - 02A74 02A8C 0C532 0C550
ESCe          = 08ADE - 00209
EX-115        = 0CF48 -
EX12          = 0D5C6 -
EX15M         = 0D5CA -
EX15S         = 0D5CE -
EXAB1         = 0D3E7 - 055EB 055F9
EXAB2         = 0D40E - 0557C 055D1 055E4 072AE
EXACT         = 128B0 -
EXCAD+        = 08631 -
EXCHRe        = 02E81 -
EXCPAR        = 187E8 -
EXDCLP        = 0592E -
EXECUe        = 064DC - 0013A
EXF           = 0D5DF -
EXITd         = 00CA0 - 065EF
EXITe         = 065F9 - 002B4
EXITp         = 012A8 - 065F4
EXP15         = 0CF5A -
EXPEX+        = 0F182 -
EXPEX-        = 0F178 - 01A8C 04297 042D5 06F7D 0847D 084EA 0870A 08B8C 08C39
               + 09FB0 0A436 0A7CE 0B952 0C9E1 0C9FE
EXPEXC        = 0F186 - 01A96 031E7 03A81 04A81 05AD6 064BD 064DE 06750 09F25
               + 0A514 0A7EA
EXPP10        = 03FE3 -
EXPPAR        = 03FD9 - 01022 01191 011A0 012ED 013B2
EXPPLS        = 03FDC -
EXPR          = 0F23C - 01C6C 043E9 05DD1 05E24 05EE0 06B57 06EB1 086EC 08FFF
               + 0989E 0999B 09F14 0AA55 0CB29
EXPRDC        = 05922 - 00CE3 00CF5 00D4C 00EF6
EXPSKP        = 1A9AC -
EndNum        = 000E6 -
Except        = 0000C -

F-R0-0        = 2F89B - 0416D 0432D 04385 044D6 08ED6 08FB7 0CAA7 0CB1C
F-R0-1        = 2F8A0 - 04136
F-R0-2        = 2F8A5 - 044F3
F-R0-3        = 2F8AA - 01118 011BC
F-R1-0        = 2F8AB - 01E54 022D8 022F2 03AA9 044A7
F-R1-1        = 2F8B0 -
F-R1-2        = 2F8B5 -
F-R1-3        = 2F8BA - 03F15
FAC15S        = 0E72B -
FASCFD        = 110C3 -
FCHLBL        = 0782C -
FCSTRT        = 0E757 -
FGTBL         = 00C9B -
FIBAD-        = 11478 -
FIBADR        = 11457 - 08473
FIBOFF        = 12132 -
FILCRD        = 1C879 -

```

FILDC* = 05759 - 00C08 00CB0 00CC5 00DEB 00E87 00ECB 00F2A 00F49
 FILEF = 09FB0 -
 FILEP = 03E9C -
 FILEP! = 03F0F -
 FILEP+ = 03F07 -
 FILEP- = 03F00 -
 FILEP1 = 03EFC -
 FILESe = 024B3 - 000CE
 FILFIL = 011CE -
 FILSK+ = 06F1D - 02B94 0A930
 FILXQ\$ = 09B95 - 024C7 04427 0BF06
 FILXQ^ = 09B76 - 06447 084C0
 FIND = 0F563 -
 FINDA = 023E3 - 00E6F 0150E 03C64 03CC1 04A0D 04C93 050FC 0515A 07851
 + 0893B 0A22D 0A5C4 0BA24 0C331 0C4CD
 FINDD0 = 023E0 -
 FINDF = 09F77 - 024D5 04568 06458 07C13 0C911
 FINDF+ = 09F63 - 01DB2 01E7D 04469 04A2F 04AC2 0850D 0A8A6 0BF2E 0C882
 FINDL = 0FFE4 - 052F7 05327 067C5
 FINDL0 = 0FFFD - 04B42
 FINDLB = 07786 -
 FINDLR = 0FFDE -
 FINDe = 0674E - 00332
 FINITA = 0CD03 -
 FINITC = 0CD0F -
 FINLIN = 18A3A - 07B21
 FINPUD = 00D1D - 0334E
 FINPUe = 03358 - 003D4
 FINPUp = 01342 - 03353
 FIRSTC = 2F47C - 02A36
 FIXDC = 05493 -
 FIXP = 02A6E - 012A3
 FIXp = 012A1 - 06FC3 08BC5 09F1E 0A2C0 0A2FD
 FKEYe = 066F0 - 000F2
 FLADDR = 0126B - 081FF
 FLDEVX = 01154 - 0C945
 FLGREG = 2F6E9 - 02756
 FLIP10 = 0DB9C -
 FLIP11 = 0DBAB -
 FLIP8 = 0DB8D -
 FLOAT = 1B322 - 0250A 059C1 06000
 FLTDH = 1B223 - 0857C 09756 0CAB1
 FLTYPp = 03E71 -
 FNDCLR = 1DAEF -
 FNDFCN = 1A0A1 -
 FNPWDS = 0D3C0 -
 FNRTN1 = 0F216 - 0179E 0252B 05A46 05C64 0601B 0C74A
 FNRTN2 = 0F219 - 05C8D 089D6 09EBB
 FNRTN3 = 0F235 -
 FNRTN4 = 0F238 - 018B8 0561F 05D0D 06D08 071C8
 FORMAe = 06B7E - 001D3
 FORSTK = 2F59E - 06620 08D02
 FORUPD = 0A6AE -
 FPOLL = 1250A - 05A62
 FPRIME = 0611C - 00371
 FRAC15 = 0C70E -
 FRACe = 06D11 - 003A7
 FRange = 0B46A -
 FSPECe = 02F02 - 00FDC 0105F 013C3 0149A 0160C
 FSPECp = 03CC5 - 00F7D 01055 013B9 0147F 014DD 015EB 0162E

```

FSPECx      = 09F2D - 01D40 01D93 04517 04A25 04AA3 07B81 0A895 0C80F
FTBSCH      = 11093 - 02FCC
FTYPDC      = 06902 -
FTYPF#      = 11059 - 025B8
FUNCD0      = 2F8BB - 0178E 04026 04054 0411A 048A6 05586 0560B 06D16 06E56
              + 07420 0745A 07488 074DB 07570 0773B 07A88 07EE5 0AB8E
              + 0B6E9 0B8B8 0B8C8 0C640
FUNCD1      = 2F8C0 - 040F0 048B9 056D7 05765 07535 07A9B 09739 09891 0998E
              + 0B4EC 0B551 0B648 0C679
FUNCR0      = 2F89B - 01460 014CB 014EA 0286A 02897 028BF 0814E 08162 08240
              + 0883C 088F6 09768 09792 09810 0984B 098E0 09918 09924
              + 09940 09967 099AB 099DD 099F0 09A00 09A2B 09A45 0B88D
              + 0BF19 0BF38
FUNCR1      = 2F8AB - 0BF6A
FiLeNd      * 0CF8B - 00028

GDISP$      = 1C3C7 -
GETAVM      = 1864D -
GETCH#      = 11427 - 0846C
GETCNT      = 1A076 -
GETCON      = 0DAA3 -
GETDIM      = 0AD6B -
GETMSK      = 01BBA -
GETNAM      = 1A085 -
GETPR+      = 06BF5 -
GETPR1      = 06BFB - 04B0A
GETPRO      = 06BEE - 0548E 06780
GETSA      = 0E551 -
GETST*      = 07716 -
GETST-      = 07728 -
GETSTC      = 07726 - 054A0 0678A 0891B
GETVAL      = 0DAB2 -
GETx$I      = 032E4 - 031BB 031CA
GLINEe      = 0704A - 0034D
GLINEp      = 01080 - 07045
GNXTCR      = 03064 - 012DF
GOSUB      = 079E9 -
GOSUBp      = 029F6 -
GOTO        = 079FA -
GOTODC      = 0552E -
GOTOp      = 029F6 -
GPSETe      = 06FC8 - 00356
GSBSTK      = 2F5A3 -
GTEXT       = 05079 -
GTEXT+      = 05199 -
GTEXT1      = 051A5 - 00F42 00F5A
GTFLAG      = 1365E -
GTKY54      = 08E9A -
GTKYC+      = 08D9B -
GTKYCD      = 08D92 - 066F2
GTPTRS      = 14636 -
GTPTRX      = 14670 -
GTXT++      = 05192 -
GetEXP      = 1C086 -

HASH1       = 1B0A1 -
HASH2       = 1B0A3 -
HDFLT      = 1B31B - 01787 018A8 03F5D 043C5 05C80 06CFE 089B6 09E99 0C73D
HEXASC      = 17148 - 05DA0 05EAF 09EE2 0AA23 0B834 0B886 0BB80 0C09F 0C258
              + 0C2FF 0CAFA

```

HEXDEC	=	0ECAF	-	034B4	076C9	0C231
HMSADe	=	07236	-	00155		
HMSSEC	=	13274	-			
HMSsUe	=	07224	-	0015E		
HMSe	=	0713C	-	00167		
HNDLFL	=	0CBC9	-			
HPSCRH	=	2F97F	-			
HRe	=	07133	-	00170		
HTAe	=	05C96	-	000B3		
HTRAP	=	0CB2F	-			
HUGE	=	0B75D	-			
HXDASC	=	05FF4	-			
HXDCW	=	0ECB4	-	02503	0288B	059BA
I/OAL+	=	1197B	-			
I/OALL	=	1197D	-	0656D	08738	09023
I/OCOL	=	11979	-			
I/OCON	=	11920	-	090BD		
I/ODAL	=	11A41	-	06546	0877A	090DC
I/OEX2	=	11A0F	-			
I/OEXP	=	11A11	-	0903B		
I/OFND	=	118BA	-	01687	02C86	02DA9 064F6 065A7
I/ORES	=	118FF	-	02D89		
IDIV	=	0EC7B	-	05A32	061DB	
IDIVA	=	0EC6E	-	06BE8	081DC	08218
IF12A	=	0C739	-			
IF2d	=	00D51	-	0A33B		
IF2e	=	0A345	-	0040A		
IF2p	=	011E2	-	0A340		
ILCNTe	=	02E70	-			
ILFART	=	02A55	-	03D67		
ILMSG	*	0CB2E	-	0C995		
IMD0+2	=	1BA2D	-			
IMD0-2	=	1BA21	-			
IMGxq1	=	1BAAB	-			
IMentr	=	1B535	-			
IMerr	=	1B989	-			
IMinit	=	1B88F	-			
IMoffs	=	1BA58	-			
IMxq27	=	1BB9C	-			
IN/REP	=	15255	-			
INADDR	=	2F6D4	-			
INBS	=	2F6C6	-			
INF*0	=	0C607	-			
INFR15	=	0C73D	-			
INPOFF	=	18B49	-			
INTA	=	2F410	-			
INTB	=	2F420	-			
INTGR	=	0F99B	-			
INTM	=	2F430	-			
INTR4	=	2F400	-			
INTR50	=	000DB	-			
INTRPT	=	0000F	-			
INVERe	=	05ABF	-	00104		
INVERp	=	0106B	-	05ABA		
INVNaN	=	0C65F	-			
INXNIB	=	2F6F9	-			
IOBFEN	=	2F576	-	09F5A		
IOBFST	=	2F571	-			
IOFND0	=	118C1	-	05D87	09015	09090 090E9 0A39F 0A564

IOFSCR = 1188E -
 IS-DSP = 2F78D -
 IS-INP = 2F79B -
 IS-PLT = 2F7A2 -
 IS-PRT = 2F794 -
 IS-TBL = 2F78D -
 ISRAM? = 10192 -
 IVAERR = 0E920 - 04A77 087A3 0C772
 IVARG = 0D749 -
 IVEXPe = 02E35 - 011D4 01313
 IVLNIB = 2F6FD -
 IVP = 00004 -
 IVPARe = 02E3F - 01321 015CF
 IVVARe = 02E66 - 013A4 013E2
 InhEOL = 00004 - 0500A
 Insert = 00007 -

 JPCLEX = 00008 -
 JUMPER = 01638 - 00D07 01303 01732 017A9 017BD 017CB 01834 0184D 0186F
 + 0188D 0191E 019B5 01AA8 01ABB 01B18 01B42 01B5B 01B78
 + 01B8A 01C29 0208A 0209C 020AE 020E1 0210C 02171 0217C
 + 021B3 021E4 021EF 0220F 0226D 02291 02353 02383 02599

 KAd = 00CDA - 07B53
 KAe = 07B5D - 0041C
 KAp = 01045 - 07B58
 KCOL0 = 2F46F -
 KCOL1 = 2F46E -
 KCOL2 = 2F46D -
 KCOL3 = 2F46C -
 KCOL4 = 2F46B -
 KCOL5 = 2F46A -
 KCOL6 = 2F469 -
 KCOL7 = 2F468 -
 KCOL8 = 2F467 -
 KCOL9 = 2F466 -
 KCOLA = 2F465 -
 KCOLB = 2F464 -
 KCOLC = 2F463 -
 KCOLD = 2F462 -
 KEYS = 1ACA8 -
 KEYBUF = 2F444 - 06707 06722 075CA 075E2 0B759
 KEYCOD = 1FD22 -
 KEYDEL = 08D2C -
 KEYFND = 08CB8 -
 KEYMRG = 08B8F -
 KEYNAM = 1AC04 - 07E28
 KEYPTR = 2F443 - 07B38 0B747
 KEYRD = 14E11 -
 KEYSAV = 2F462 -
 KEYSN = 00D4D -
 KYDN? = 00774 -

 LABELP = 03E9F -
 LABLDC = 05702 -
 LASTFN = 000B4 -
 LBLIN# = 2F871 -
 LBLINP = 02A04 -
 LBLNAM = 077E7 -
 LBLNIF = 02A0D -

LCDINI	=	00665	-
LDCEXT	=	04F5E	-
LDCM10	=	04F6F	- 04E22 067FA
LDCOMP	=	04F69	- 068BF
LDCSET	=	05060	- 07858
LDCSPC	=	2F6C1	- 02AE8 06810 068D3 0796B 0A4F2 0A501 0A7F1 0A7FC
LDSST1	=	04F72	-
LDSST2	=	04F9E	-
LEAVE	=	04C01	-
LEAVEe	=	0A207	- 00305
LEEWAY	=	000D4	- 09A22
LEXBF+	=	10DDF	- 06432
LEXPIL	=	000FF	- 01691
LEXPTR	=	2F6CF	- 0A7D5 0A80B
LGT15	=	0D1AE	-
LIMITS	=	0AC3E	-
LIN#AU	=	05122	-
LIN#D+	=	05112	-
LIN#DC	=	05115	- 00EA8
LINEP	=	02620	-
LINEP*	=	02634	-
LINEP+	=	02626	- 02DE2 06500
LINP	=	02A07	-
LINSKP	=	08A05	-
LISTDC	=	05839	- 00CDC
LN1+15	=	0CD44	-
LN1+XF	=	0CD51	-
LN12	=	0CD7D	-
LN15	=	0CD81	-
LN30	=	0CD9C	-
LNPE66	=	027EA	-
LNPEXT	=	02617	-
LNSKP-	=	089FF	-
LOCADR	=	0A611	- 0435D 07C77 081F8
LOCFIL	=	1721D	-
LOCKWD	=	2F7B2	-
LOOPST	=	2F7AC	-
LOOPe	=	0A2F0	- 003EF
LSLEEP	=	006CD	-
LSTLEN	=	06C27	-
LXDRVd	=	00CDA	- 063D8 06404
LXDRVp	=	01053	- 063DD 06409
LXFND	=	0979D	- 0CB57
LXOFFd	=	00CC3	-
LXOFFe	=	063E2	- 000E0
LXONd	=	00CAE	-
LXONe	=	0640E	- 000E9
LXTXTT	=	1EE9F	-
MAIN05	=	00338	-
MAIN30	=	0037E	- 06929
MAINEN	=	2F571	-
MAINLP	=	002FD	-
MAINST	=	2F558	- 01CE8
MAKE1	=	0DACE	-
MAKEBF	=	01751	-
MANSTK	=	196D3	-
MAPd	=	00BFD	- 08451
MAPe	=	0845B	- 00344
MAPp	=	00F63	- 08456

MAPse = 08444 - 0033B
MARGIe = 086FB - 00194
MARGIp = 01276 - 086F6
MAXCMD = 2F976 - 09F4F
MAXDe = 01860 - 00290
MAXMe = 018E3 - 00299
MBOX^ = 2F7A9 -
MDYe = 05AA3 - 00287
MEMBER = 1B098 - 0A5BA 0A6B1
MEMCKL = 012A5 - 047D1 04D93 0746D 07871 0BB40
MEMDe = 018D7 - 002A2
MEMER* = 0945B -
MEMERR = 0944D - 018C3 01C77 02154 03594 05ECF 06577 06B75 08FB0 09B8F
+ 0A788
MEMERX = 0944F -
MEMMe = 018CB - 002AB
MENUe = 08898 - 001B8
MESSG = 0CC17 -
MFER42 = 0962C -
MFERR = 09393 - 0206E 032A1 0372D 03F07 054AA 06498 06560 066B7 06798
+ 06C18 06FB9 08535 087FD 08925 08A18 0A89F 0BFC0 0BFF0
MFERR* = 093F1 -
MFERRS = 0939E -
MFERsp = 0940D -
MFLG=0 = 13DA1 -
MFWRN = 093BC -
MFWRNQ = 093C5 -
MFWRQ8 = 093C3 -
MGOSUB = 1AF01 - 0313E 0466E 04934 07BE8 08106 082AB 083A4 083ED 0A911
+ 0BC8F
MLFFLG = 2F870 - 01CD1 023C0 049B2
MOD15 = 0C796 -
MODEe = 08BCA - 00224
MOVE*M = 01308 - 02334 031E0 06718 0785F 09830
MOVED0 = 1B0F4 -
MOVED1 = 1B101 -
MOVED2 = 1B104 - 02424 05030 0904D
MOVED3 = 1B109 -
MOVEDA = 1B0FA -
MOVEDD = 1B106 - 08FD6
MOVEDM = 1B0EE - 099CC 09A72
MOVEU0 = 1B162 - 08B2D 090B2
MOVEU1 = 1B16F -
MOVEU2 = 1B172 -
MOVEU3 = 1B177 - 03181 03EFC 06587
MOVEU4 = 1B174 -
MOVEUA = 1B168 - 04F0C
MOVEUM = 1B15C - 09A5E
MP1-12 = 0C436 -
MP15S = 0C440 -
MP2-12 = 0C432 - 06EDC
MP2-15 = 0C43A - 055B0 07290
MPOP1N = 0BD8D - 0A43D
MPOP2N = 0BD54 - 0A0FE
MPY = 0ECBB - 02628 0359B 04D83 05FF1
MSGTB1 = 00A18 - 0008C
MSIZE = 10407 -
MSIZE+ = 1040A -
MSN12 = 0D553 -
MSN15 = 0D557 -

MSPARe = 02E5C - 010B5 0133D 013FA
MTADDR = 08195 -
MTADR+ = 081A1 -
MTHSTK = 2F599 - 03265 0A50B
MULTF = 0C446 -
MVMEM+ = 0133C - 03143 0A916

NEEDSC = 2F94A - 0224A
NEXTOe = 0AA04 - 0019D
NLOOPe = 01730 - 002BD
NOARGd = 00C78 - 017F4 08A73 08AB3 08B53 0A1FD 0A2E6 0A356
NOARGp = 010BA - 017F9 05A86 05A9E 08A78 08AB8 08B58 0A202 0A2EB
NORDIM = 0AE2D -
NOSCRL = 14C8A - 01F0B 03F82 07FA8 0BAE4
NPRIME = 06126 - 0037A
NRMCON = 161AF -
NTOKEN = 0493B - 00F87 00FF7 0113C 01201 0132C 013E9 014AF 01507 01613
+ 09E05
NTOKNL = 048E6 - 015B1
NULLP = 07999 - 0679F
NUMC++ = 03690 -
NUMC+O = 03696 -
NUMCK = 0369D - 00FD5 0108B 011E4 013CA 014FC
NUMSCN = 04D18 -
NXPR10 = 0B3B0 -
NXPRMP = 0B3A5 -
NXTADR = 147E8 -
NXTELM = 148AC -
NXTEXP = 1C2F7 -
NXTIRQ = 2F70D -
NXTLIN = 10031 - 04F97 067D7 06939
NXTP = 03455 - 012AA
NXTSTM = 08A48 - 01826 01B2D 01EDB 02C09 03F89 04616 04698 04753 04961
+ 04984 04C15 05AB0 05B47 06439 064CD 06744 06962 0712B
+ 07FB2 0866E 08781 08A5F 0906F 09F95 0A2A6 0A816 0A920
+ 0BAEB 0C8E5 0CA4F

NXTVA- = 13E58 -
NoCont = 0000E -
NwOFFS = 1C02D -

OAGNXT = 03060 -
OBCOLL = 01435 - 01F5D 0230B 03556 0465D 04799 04FD3 07868 080D9 08345
+ 0995E 0A534 0BB4A

OBEDIT = 17687 -
OBSOLD = 00E49 - 00A01
OFFFLG = 2F442 -
OKP = 00000 -
ONDC20 = 05501 -
ONINTR = 2F68D -
ONOFFd = 00C92 - 0496A 08A84 08C14 08D70 09074
ONOFFp = 01258 - 0496F 08A89 08C19 08D75 09079
ONP40 = 02B7B -
ONTIMR = 08008 -
OPCODe = 0AB77 - 001A6
OPENF = 11B06 -
ORGSB = 0D65B -
ORSB = 0D63C -
ORXM = 0D633 -
OUT1T+ = 02CDF - 00F6F
OUT1TK = 02CEB - 014D6 01627

OUT2TC = 02CFD - 00C8A
 OUT2TK = 02CFF -
 OUT3TC = 02D12 -
 OUT3TK = 02D15 - 014F5 015C8
 OUTBS = 2F58F - 02BD8 04CCA 04DA1 04E18 04E57 04EB5 06806 068C6 099D3
 OUTBY+ = 02CE5 - 00CEE 00D00 00D18
 OUTBYT = 02CE8 - 00E9E 00F0A 0114C 01175
 OUTC15 = 05421 -
 OUTEL1 = 05300 - 00CA9 00D45 00E0C 00F50
 OUTELA = 05303 - 00C7A 00E5E
 OUTLI1 = 03709 -
 OUTLIT = 036F3 -
 OUTNBC = 05423 - 00DE4 00EC1 00EEC
 OUTNBS = 05426 -
 OUTNIB = 02D28 - 01109 0136A
 OUTRES = 0BC84 -
 OUTVAR = 0373E -
 OVFL = 0CA73 -
 OVFNIB = 2F6FB -
 OVP = 00002 -

 P1-10 = 041C1 -
 PAGELe = 08CCB - 00236
 PAGELp = 0128C - 08CC6
 PAINTe = 05BB4 - 00116
 PARERR = 02F08 -
 PART3 = 18097 -
 PBLISd = 00E63 - 04989 049A2
 PBLISe = 04993 - 003C2
 PBLISp = 01458 - 0498E 049A7
 PCADDR = 2F679 - 05299 089DD
 PCEXPRe = 06828 -
 PCRe = 08ABD - 00200
 PDEV = 09E9E -
 PDIRE = 01CCB - 00476
 PEDIT = 0FF5F -
 PEDITD = 0FF62 -
 PEDITM = 0FF70 -
 PERFe = 08C1E - 0022D
 PFFe = 08B5D - 00212
 PFINDL = 078DF -
 PFNDZL = 078E2 -
 PGCDe = 0603E - 00383
 PHIE = 05F8D - 0038C
 PI/2 = 0DB77 -
 PI/2D = 0DB7A -
 PI/4 = 0DAA1 -
 PILERR = 01B67 - 017A4 0185A 02573
 PKLEX * 0C98F - 00033
 PLFe = 08B76 - 0021B
 PLFp = 01276 - 08B71
 PNDALM = 2F761 -
 POKEp = 012CE - 0C9DA
 POLHND = 02656 - 00090
 POLL = 12337 - 02542 08A06
 POLLD+ = 1232D -
 POP1N = 0BD1C -
 POP1N+ = 0BD91 -
 POP1R = 0E8FD - 05626 05672 05E37 060C1 06D3E 06D67 08D1A 0974C 0CA9A
 POP1S = 0BD38 - 02583 03231 03EA8 040D6 042DC 04435 044B4 056CB 05C9C

```

+ 05DE3 05E55 05EEA 06996 06A66 06C5D 07529 07B11 0859C
+ 085B0 08B10 08E08 08E73 08E92 08EA4 09FC8 0A12D 0C50B
+ 0C6C5 0C779 0CA05
POP2N          = 0BC8C -
POP2N+        = 0BD58 -
POPBUF        = 010EE - 02CE2 03CB8 0771F 0888F 0B9D4
POPMTH        = 1B3DB -
POPSTK        = 08F55 -
POPSTR        = 1B405 -
POPUPD        = 08F3E - 02BCE 0301E 03029 03034 0315E 082D3 0A20D 0A47F
POSIE         = 0C696 - 003B0
PPCMe         = 0602C - 00395
PPOLLe        = 01766 - 002C6
PPOS          = 2F956 - 08AC7
PREP          = 0ADAF -
PRESCN        = 04A49 -
PRGFMF        = 0A146 -
PRGMEN        = 2F567 - 06675 087AA 0A550
PRGMST        = 2F562 - 087C4
PRIME         = 061F5 - 0039E
PRINT*        = 17F37 -
PRINTt        = 00001 -
PRMCHN        = 0B375 -
PRMCNT        = 2F94B -
PRMPTR        = 2F5B7 -
PRNEXe        = 02E95 -
PRNTDC        = 05450 -
PRPSND        = 06B17 - 02BEC
PRSC00        = 07B93 - 05282
PRSSc+        = 1BA84 -
PRSScn        = 1BA88 -
PRT#DC        = 06841 -
PSHGSB        = 08F13 - 0A46A
PSHMCR        = 08F0B -
PSHSTK        = 08C7F -
PSHSTL        = 08C85 -
PSHUPD        = 08F0D - 02FE9 02FF4 02FFF 030F3 082DA
PUGFIB        = 12198 -
PURGDC        = 05745 -
PURGEF        = 17359 -
PUTRES        = 18115 -
PWIDTH        = 2F958 -
PWROFF        = 00526 -
PgmRun        = 0000D - 03006 0A395 0A3BA 0A55A

QUOEXe        = 02E8B -
QUOTCK        = 0623D -

R1REV         = 00785 -
R2REV         = 0AA83 -
R3=D10        = 03526 - 012E6 05A5B
R3REV         = 153AB -
R4REV         = 1DBA8 -
R<RST2        = 014DB - 020DA
R<RSTK        = 014DD - 01A4A 01C80 024BC 02A6D 02CB5 043FD 04420 073FF 07AD8
+ 0C49A
RAMEND        = 2F5B2 -
RAMROM        = 0A5F7 - 0853C
RANGE         = 1B07C - 05D38 05E0A 06608 07E3C 07E50 09F43 0C3A5 0C59C 0C6F7
+ 0CA82

```

RAWBFR = 2F580 -
 RCCD1 = 0D3F5 - 055A9 072CE
 RCCD2 = 0D41C - 05525 05569 055B7 071A4
 RCL* = 0E983 -
 RCLALL = 1AFBF -
 RCLW1 = 0E981 - 055A2 05604 06E11
 RCLW2 = 0E9BE - 06DCE 06F14
 RCLW3 = 0E9C4 -
 RCSCR = 0E954 - 06DF2 06F02
 RCURON = 14C80 -
 RCVOFS = 1C050 -
 RDATTY = 17CC6 - 03296
 RDBAS = 173FF -
 RDBYTA = 13A2F -
 RDCHD+ = 076EE -
 RDCHDR = 076F0 -
 RDHDR1 = 076FD - 0A8B9
 RDINFO = 0846B - 0C872
 RDLNAS = 13A1F -
 RDTEXT = 17489 -
 READIN = 0F484 -
 READNB = 17518 -
 READP5 = 0323B - 01347
 RECADR = 0F4B7 -
 RECALL = 0F281 -
 REDIT = 05EFD - 06B07
 REDUCE = 15977 -
 REDUCe = 06ACD - 001DC
 REDe = 05EE8 - 000BC
 RELJMP = 05047 -
 RENSUB = 1A753 - 0528F 05444
 RENUMd = 00CDA - 05268
 RENUMe = 05272 - 003CB
 RENUMp = 0152D - 0526D
 REPEAe = 0A2F0 - 002F3
 REPLAe = 08DE0 - 000C5 0035F
 REPROM = 18A1E -
 RESCAN = 04A4C - 09E34
 RESERV = 2F986 -
 RESJPC = 2F991 - 02772
 RESPTR = 03172 - 00FC7 01039 010AE 0118A 013AB 013D1 014C4 01559 015D6
 + 09E28
 RESREG = 2F7C2 -
 REST* = 03035 - 0126A
 REV\$ = 1B38E - 01C65 05DCA 06B50 06EAA 06EBC 09987 09F0D 0AA4E 0CB15
 REVPOP = 0BD31 - 01AFD 05D62 06522 06757 06B00 0977A 0981A 098D9 09DE1
 REWIND = 11365 -
 RFAD++ = 0A6FB -
 RFAD+I = 0A702 -
 RFAD-- = 0A652 -
 RFAD-I = 0A659 -
 RFNBFRR = 2F57B - 09F7E
 RFUPD+ = 0A66E -
 RJUST = 12AE2 - 060F4 08C40
 RND-12 = 1B01F -
 RND12+ = 0C9D5 -
 RNDAHX = 136CB - 01AEC 01BF1 0429E 04A88 05914 05B51 05BEB 06A21 06A49
 + 06B5E 06F67 08711 0878F 08B93 09DCA 09F2C 0C75B
 RNDNRM = 0CAB1 -
 RNSEED = 2F6FE -

ROMANe = 0907E - 0045B
 ROMCHK = 10FDE -
 ROMCID = 00BFE -
 ROMCK5 = 10FE4 -
 ROMFND = 1102F - 02063
 ROWDVR = 2E350 -
 RPLCe = 09734 - 08DED
 RPLLIN = 013F7 - 0248F 0509B 0810B 082B0 083A9 0BC94
 RPLSBH = 1799B -
 RPTKY = 152BA - 03C99 07DBE 0B9B5
 RRECe = 01BBF - 003DD
 RST2<R = 014A6 - 02105
 RSTD0 = 06832 -
 RSTD1 = 1C596 -
 RSTK<R = 014A8 - 0189C 01C53 02515 02A9C 02CF0 043DD 04483 07508 07AED
 + 0C4C6
 RSTKBF = 2F820 -
 RSTKBp = 2F81F -
 RSTST = 0F5C5 -
 RTNX10 = 08FF2 -
 RUNRT1 = 074E7 - 089F1 0A42F
 RUNRTN = 074EA - 066E1
 ResetC = 00008 -

 S-R0-0 = 2F871 - 0419D 041D9 04202 0427A 04553 045DE 0A855 0C9EF 0CA16
 S-R0-1 = 2F876 - 041BE 046E6 0A13C
 S-R0-2 = 2F87B -
 S-R0-3 = 2F880 - 0122E 0478C 0485D 0A03E 0A05A
 S-R1-0 = 2F881 - 045BA 07612 07659 07962 07A52 07F29
 S-R1-1 = 2F886 - 02BC4 075B0 08166
 S-R1-2 = 2F88B - 03F51 06763 07336 077DA 0781D 07F82
 S-R1-3 = 2F890 - 06831 073BF 07FD9 07FED 07FFC
 SALLOC = 0153B - 07BC5 0C7A6
 SAVD0 = 1C587 -
 SAVD1 = 1C578 -
 SAVESB = 0D66E -
 SAVEXM = 0D663 -
 SAVGSB = 0D64E -
 SAVSTK = 2F59E -
 SB15S = 0E19A -
 SCAN = 04C40 -
 SCNRT = 022B9 -
 SCOPCK = 0915B -
 SCREX0 = 2F941 -
 SCREX1 = 2F951 -
 SCREX2 = 2F961 -
 SCREX3 = 2F971 -
 SCRLLR = 0212E - 03CA3 0770A 08934 0B9BF
 SCROLT = 2F946 -
 SCRPtr = 2F966 -
 SCRST0 = 2F901 -
 SCRTCH = 2F901 -
 SE1-10 = 04468 -
 SECHMS = 13252 -
 SELECe = 09FA4 - 003F8
 SELECp = 0118F - 09F9F
 SEND20 = 17DFA -
 SENDEL = 17DC1 - 02407 05014
 SENDIT = 17DE3 -
 SENDWD = 17E15 -

SETALM = 1290D -
 SETALR = 12917 -
 SETFMT = 0F01F -
 SETSB = 0D641 -
 SETTMO = 13158 - 02CD1
 SFLAG? = 1364C - 0285C 02D25 02DBC 02DCA 05658 08911
 SFLAGC = 13601 - 05AA9
 SFLAGS = 135FA - 02DD8 05A91
 SFLAGT = 13608 -
 SHF10 = 0C486 -
 SHFLAC = 0DB46 -
 SHFRAC = 0DB51 -
 SHFRBD = 0DB5F -
 SHRIND = 00CDA - 0A889
 SHRINE = 0A893 - 00368
 SHRINp = 01053 - 0A88E
 SHRT = 0F96C -
 SHUTDN = 005E2 -
 SIGCHK = 0BD98 -
 SIGTST = 0E636 -
 SIN12 = 0D716 -
 SIN15 = 0D71A -
 SKIPDC = 057F6 -
 SLEEP = 006C2 - 0180C 02CD8
 SLEEPe = 017FE - 002CF
 SNAPBF = 2F7F0 - 0163F 01649 01653 0165D 016EC
 SNAPR* = 01578 -
 SNAPRS = 01571 -
 SNAPSV = 015A7 -
 SNDWD+ = 17E1F - 023F2 0500D
 SPACE = 0AD9D -
 SPACEe = 06A10 - 001E5
 SPLITA = 0C6BF - 0563C 060DD 06DE0 06EEA 072E5 072F8 07314
 SPLITC = 0C940 -
 SPLTAC = 0C934 -
 SPLTAX = 0E62B -
 SQR15 = 0C534 -
 SQR17 = 0C553 -
 SQR70 = 0C5C3 -
 QRSVAV = 0D629 -
 SRLEAS = 015EC -
 SRQe = 01773 - 002D8
 STAB1 = 0D3D9 - 0551E 0714D 0716E 0724E
 STAB2 = 0D400 - 0550B 072A0
 STACKe = 09F23 - 00182
 STARTe = 06590 - 00131
 STATAR = 2F7AD -
 STATRS = 172F3 -
 STATSV = 1732F -
 STCD2 = 0D427 -
 STDd = 00C99 - 05A81 05A99 05AB5 064B1 064D2 066E6 06FBE 07040 086F1
 + 08B6C 08BC0 08CC1 09F19 09F9A 0A2BB 0A2F8 0A7BB 0B92E
 + 0C9D5
 STKCHR = 18504 - 03482 048E0 058A7 065D5 06AB3 06B3A 07AC2 0B740
 STKCMD = 155ED -
 STKVCT = 1470C -
 STMBCL = 090E7 -
 STMBUF = 090DF -
 STMTD0 = 2F891 - 03FC0 042AF 0467D 07363 07394 077BD 077EA 0783E 07D8B
 + 07EFD 080E0 081EE 082E7 08303 0837C 083D1 0840F

STMTD1 = 2F896 - 01D4A 0207B 021A7 04586 04637 04664 046C3 04778 048EE
 + 04923 04943 07345 0759E 0766C 0780A 07CC2 07F8C 0802B
 + 083BD 083DF 083FF 0841F 0848A 0A0E0 0A1D5 0A307 0A319
 STMTR0 = 2F871 - 04AAD 04B73 0705C 07091 07447 0747E 074CC 07BFE 07C20
 + 084D1 084F7 08C65 09FD3
 STMTR1 = 2F881 - 04B9E 073E5 0740A 07899 078A3 079D5 079F7 07A16 07C41
 + 08322 0834C 084DC 08502
 STORE = 0F5F8 - 03F74 0A82E
 STR\$00 = 1815C -
 STR\$SB = 18149 - 06F4A
 STRALL = 1AF5D -
 STRASN = 0F6B3 -
 STREQI = 1B1EF -
 STRGCK = 036BA - 012B8 01417
 STRHDR = 0F09A - 09EF4 0AA35 0CAE1
 STRHED = 14C2E -
 STRNGP = 0379D - 00F98 00FA6 01066 012D4
 STRNGp = 01064 - 064B6 064D7 066EB 06749
 STRTST = 1B1C7 - 0A17F
 STSAVE = 2F6BE - 03C09 03C1B
 STSCR = 0E92C - 0559B 055CA 06EF1
 STUFF = 1B0B2 - 04CF2 04DC6
 SUBONE = 0C327 - 055D8 055F2
 SVINF+ = 08457 - 07BD2
 SVINFO = 0845A - 07BE1 0C84B 0C868
 SVTRC = 0FA35 - 0A7C7 0A7E3
 SWPBYT = 17A24 - 02445 02C39 05051 05CEC 05EBB 0A9A5 0C5CF 0C625
 SYNTXe = 02E2B - 00FE3 0101B 01271 012C9 0131A 01578
 SYSEDe = 0B938 - 001AF
 SYSEN = 2F58A -
 SYSFLG = 2F6D9 - 028D5
 SYSGNX = 0AA5A - 0B7DD
 SYSGOP = 0AB69 - 0B78E
 SYSINI = 0A9E2 - 0B7D7
 SavLvl = 00005 -
 SetAVM = 1B9FA -

 TAN12 = 0D72F -
 TAN15 = 0D733 -
 TASTK = 2F599 -
 TBLJMC = 02426 - 00C25 03811 0A5F1 0AF98 0B9F2
 TBLJMP = 0242A -
 TBMSG\$ = 099AB - 05A75
 TERCHR = 2F97D -
 TFHDLR = 1702F -
 TFORN = 2F59E - 0CA23
 TGSBS = 2F5A3 -
 TIMAF = 2F787 -
 TIMER1 = 2E3F8 -
 TIMER2 = 2E2F8 -
 TIMER3 = 2E1F8 -
 TIMLAF = 2F77B -
 TIMLST = 2F76F -
 TIMOFS = 2F763 -
 TKSCN+ = 08A6B - 087EB
 TKSCN7 = 08A99 - 066A9 0A576
 TMRAD1 = 2F697 -
 TMRAD2 = 2F69C -
 TMRAD3 = 2F6A1 -
 TMRIN1 = 2F6A6 -

TMRIN2	=	2F6AE	-	
TMRIN3	=	2F6B6	-	
TODT	=	13229	-	05A18
TOKENe	=	09E41	-	00320
TONE	=	0EBEB	-	
TRACDC	=	052FC	-	00C94
TRACEM	=	2F7B0	-	0A3BF
TRC90	=	0DA11	-	
TRFMBF	=	2F8C5	-	01F2A 01F34 01FA6 034A3 03508 0A85C 0C819 0C8FA
TRFROM	=	0FE59	-	0A415
TRKDON	=	1CFAC	-	
TRMNTR	=	0F1DD	-	
TRPREG	=	2F6F9	-	
TRSFMu	=	16B84	-	
TRTO+	=	0FE7B	-	0A422
TST12A	=	0D476	-	06DC0
TST15	=	0D47A	-	0552E 0554C 05572
TWO*	=	0DB38	-	
Trace	=	0000F	-	0A3B5
TstEnd	=	1C0FF	-	
UNDERd	=	00C7F	-	08D45
UNDERe	=	08D4F	-	0023F
UNDERp	=	01245	-	08D4A
UNFNIB	=	2F6FA	-	
UNP	=	00001	-	
UNTILe	=	0A302	-	002FC
UPCPOS	=	13C67	-	
UPD1EN	=	2F599	-	
UPD1ST	=	2F55D	-	
UPD2EN	=	2F6A6	-	
UPD2ST	=	2F674	-	
UPDANN	=	13571	-	
UPDPCC	=	07B65	-	
USG*10	=	1B508	-	
USGch+	=	1BC15	-	
USGch-	=	1BC0B	-	
USGrst	=	1BC63	-	
USING	=	1B446	-	
USINGp	=	03628	-	
USRSTA	=	01B80	-	02CC3 02D13
USloop	=	1C14B	-	
USnm05	=	1BD12	-	
USst03	=	1BBCE	-	
USst05	=	1BBD4	-	
VAL00	=	1AD8F	-	
VALCHK	=	1AE61	-	
VARDC	=	0537C	-	00CA2 00DF2
VARNB-	=	0E28D	-	
VARNBR	=	0E289	-	
VARP	=	0350E	-	013D8
VARSWe	=	0A7C5	-	0030E
VARSWp	=	00FE8	-	0A7C0
VECTOR	=	2F43C	-	
VIEWD1	=	15147	-	028C6 08244
VRIABL	=	04BC4	-	
ValSub	=	0000A	-	
WFTMDT	=	085DD	-	

WHILe	=	0A2C5	-	002EA
WINDLN	=	2F473	-	02A26
WINDST	=	2F471	-	
WIPOUT	=	1B0AF	-	02877 0815B
WRAPe	=	08D7A	-	00248
WRBYTC	=	13A73	-	
WRDSC+	=	02C26	-	
WRDSCN	=	02C2A	-	010BE
WREcd	=	00CE1	-	01A80
WRECe	=	01A8A	-	003E6
WREcp	=	012D9	-	01A85
WRITNB	=	1752B	-	
WRTBLA	=	09126	-	02D78
WRTFIB	=	11CEE	-	
WRTNUM	=	139C4	-	
WRTSTR	=	1396F	-	
WSTRFX	=	138B5	-	
XDelay	=	00009	-	
XMTADR	=	08133	-	
XROM01	=	00001	-	
XXHEAD	=	1A44E	-	0650A 06EC5
XYEX	=	0C697	-	
YMDDAY	=	13304	-	0584A
YMDH01	=	130E5	-	
YMDHMS	=	130DB	-	
YX2-12	=	0D274	-	
YX2-15	=	0D27A	-	
ZERBUF	=	18B20	-	
a!	=	00021	-	
a"	=	00022	-	
a\$	=	00024	-	
a'	=	00027	-	
a.	=	0002E	-	
a0	=	00030	-	
a1	=	00031	-	
a2	=	00032	-	
a3	=	00033	-	
a4	=	00034	-	
a5	=	00035	-	
a6	=	00036	-	
a7	=	00037	-	
a8	=	00038	-	
a9	=	00039	-	
bALTCH	=	00BFB	-	02D7E 09115
bASSGN	=	00804	-	
bCARD	=	00807	-	
bCHARS	=	00BFB	-	
bECOMD	=	00809	-	
bFIB	=	00803	-	
bFILE	=	00805	-	
bIEXKY	=	00802	-	
bLEX	=	00BFC	-	01682
bPILAI	=	00810	-	
bPILSV	=	0080F	-	
bROMTB	=	00BFE	-	

bSCRTC	= 00E00 -
bSTART	= 00808 - 06592
bSTAT	= 00806 -
bSTMT	= 00801 - 0A39A 0A55F
bSTMXQ	= 00811 -
cC->C	= 00068 -
cR->C	= 00069 -
cRCL	= 00067 -
dCARD	= 00007 - 0C7F5
dIRAM	= 00001 -
dMAIN	= 00000 -
dPCRD	= 00007 -
dPORT	= 00001 -
e#of#	= 000F7 -
e0^0	= 00006 -
e0^NEG	= 00005 -
e1^INF	= 00011 -
e2MROM	= 0001A -
eAF	= 0001B -
eALGN	= 000F0 -
eCALGN	= 00060 -
eCHNL#	= 00029 -
eDATTY	= 0001F -
eDVCNF	= 00040 - 08A14
eEOFIL	= 00036 - 08640 0A8EC 0A950 0BFEC
eEXCHR	= 0004E -
eEXP0	= 00003 -
eEXPCT	= 000E7 -
eF2BIG	= 0004A -
eFACCS	= 0003C - 064AC 08543
eFEXST	= 0003B - 04ACC
eFILE	= 000EA -
eFNNTf	= 00021 -
eFOPEN	= 0003E -
eFPROT	= 0003D - 030DA
eFSPEC	= 0003A - 084CA 0C931
eFTYPE	= 0003F - 06440 06794 08531 0A8CA 0BFBC
eFnFND	= 00039 - 06451 06494
eFwoNX	= 0002A - 066B3
eIF*ZR	= 00010 -
eIF-IF	= 0000F -
eIF/IF	= 0000E -
eILCNT	= 0004F -
eILEXP	= 00050 -
eILKEY	= 00055 -
eILLEG	= 000E6 -
eILPAR	= 00051 -
eILTFM	= 00037 -
eILVAR	= 00053 -
eIMGOV	= 0002F -
eINF	= 000F3 -
eINF^0	= 00012 -
eINPUT	= 000F4 -
eINVIM	= 0002D -
eINVLD	= 000EC -
eINVST	= 000ED -
eINVUS	= 0002E -

eINX	=	00015	-	
eIVARG	=	0000B	-	02B69
eIVSAR	=	00033	-	
eIVSOP	=	00035	-	
eIVSTA	=	00034	-	
eIVTAB	=	00030	-	
eL2LNG	=	00041	-	
eLN0	=	0000C	-	
eLOBAT	=	00016	-	
eLOG-	=	0000D	-	
eMEM	=	00018	-	
eMMCOR	=	00017	-	
eMPI	=	00019	-	
eMSPAR	=	00052	-	
eNEG^X	=	00009	-	
eNFOUN	=	000E8	-	
eNODAT	=	00020	-	
eNOTIN	=	00043	-	
eNSVAR	=	00033	-	
eNUMIN	=	00026	-	
eNVSTA	=	00033	-	
eNXwoF	=	0002B	-	
eOVFL*	=	000F5	-	
eOVFLW	=	00002	-	
ePALGN	=	0005E	-	
ePLLC	=	0005A	-	
ePLLC#	=	00059	-	
ePRCER	=	00054	-	
ePRMIS	=	00024	-	
ePRNEX	=	0004C	-	
ePROTD	=	00042	-	
ePRTCT	=	000F8	-	
ePULL	=	000F6	-	
eQUOEX	=	0004D	-	
eR0WRN	=	00056	-	
eR1WRN	=	00057	-	
eRALGN	=	0005D	-	
eRECOR	=	0001D	-	
eRWERR	=	00046	-	
eRwoGS	=	0002C	-	
eSIGOP	=	00013	-	
eSPGNF	=	00031	-	
eSQR-	=	0000A	-	
eSTMNF	=	0001E	-	052AA
eSTROV	=	00025	-	0329D 03729 03F03 0655C
eSUBSC	=	0001C	-	
eSYNTAX	=	0004B	-	
eYSER	=	00017	-	
eTFFLD	=	00038	-	
eTFM	=	000F1	-	
eTFWRN	=	00058	-	
eTNINF	=	00004	-	
eTOO	=	000EF	-	
eTOOFI	=	00028	-	
eTOOMI	=	00027	-	
eTRKDN	=	00061	-	
eTRKOF	=	000E5	-	
eTUFAS	=	00047	-	
eTUSLO	=	00048	-	
eUALGN	=	0005F	-	

eUNFLW	= 00001 -
eUNKCD	= 00045 -
eUNORC	= 00014 -
eVALGN	= 0005C -
eVARTY	= 00032 -
eVFYER	= 00044 -
eWALGN	= 0005B -
eWRGNM	= 00049 -
eXFNNF	= 00022 -
eXWORD	= 00023 - 016C3
eZRDIV	= 00008 -
eZRO/0	= 00007 -
enull	= 00000 -
ew/o	= 000EB -
faos	= 000DF -
fASCII	= 00001 -
fBASIC	= 0E214 - 054C6 0A3ED 0C8BA
fBIN	= 0E204 -
fDATA	= 0E0F0 -
fEOF	= 000FF -
fEOR	= 000EF -
fEOS	= 0006F -
fKEY	= 0E20C -
fLEX	= 0E208 - 00018 03071 063F2 06414
flLIF1	= 00001 -
fMOS	= 0007F -
FROM	= 0E21C -
fSDATA	= 0E0D0 -
fsos	= 000CF -
fTEXT	= 00001 - 02B0C
fTMOUT	= 00003 - 02D07
flAC	= FFFC7 -
flALRM	= FFFC4 -
flBASE	= FFFF0 -
flBAT	= FFFC3 -
flBEEP	= FFFFE -
flBPLD	= FFFE7 -
flCALC	= FFFC0 -
flCLOC	= FFFD3 -
flCMDS	= FFFD1 -
flCTON	= FFFFD -
flCTRL	= FFFD0 -
flDATE	= FFFFFFFCB - 05A8D 05AA5
flDG0	= FFFEF -
flDG1	= FFFEE -
flDG2	= FFFED -
flDG3	= FFFEC -
flDORM	= FFFD5 -
flDVZ	= FFFF9 -
flEOT	= FFFE9 -
flEXAC	= FFFD2 -
flEXTD	= FFFEA -
flFXEN	= FFFF3 -
flINFR	= FFFF5 -
flINX	= FFFFC -
flIVL	= FFFF8 -
flLC	= FFFF1 -
flMKOF	= FFFCE -
flNEGR	= FFFF4 -

flNOFN	=	FFFD6	-
flNOPR	=	FFFE6	-
flNZ4	=	FFFE8	-
flNZ5	=	FFFCB	-
flNZ6	=	FFPCA	-
flNZ7	=	FFFC9	-
flNZ8	=	FFFC8	-
fLOVF	=	FFFFA	-
flPDWN	=	FFFEB	-
flPRGM	=	FFFC2	-
flPWDN	=	FFFCF	- 02DB8
flQIET	=	FFFFF	-
flRAD	=	FFFF6	-
flRPTD	=	FFFC5	- 02D21
flRTN	=	FFFD4	-
flSCEN	=	FFFF2	-
flSUSP	=	FFFC1	-
flTNOF	=	FFLCD	- 02DC6 02DD4
flUNF	=	FFFFB	-
flUSER	=	FFFF7	- 02858
flUSRX	=	FFFC6	-
flVIEW	=	FFFC	-
id	=	000E1	- 0514B
k#-CHR	=	00068	-
k#-LIN	=	0006B	-
k#1	=	00027	-
k#2	=	00028	-
k#3	=	00029	-
k#ATTN	=	0002B	- 07D4E 07E0E 08940 0BA29
k#BKSP	=	00067	- 02922
k#BOT	=	000A3	- 08954
k#CALC	=	0006F	-
k#CONT	=	00070	-
k#CTRL	=	0009E	-
k#DOWN	=	00033	- 07DD7 08945
k#EOL	=	00026	- 08959
k#FLFT	=	0009F	-
k#FRT	=	000A0	-
k#GON	=	0009B	-
k#I/R	=	00069	-
k#LAST	=	000A4	-
k#LC	=	0006A	-
k#LERR	=	000A1	-
k#LFT	=	0002F	-
k#OFF	=	00063	- 02D0C 07D5D 07E18 0BA8D
k#RT	=	00030	-
k#RUN	=	0002E	-
k#SST	=	00066	-
k#TOP	=	000A2	- 0894F
k#UP	=	00032	- 07DD2 0894A
k#USER	=	0006D	-
k#USEX	=	000A5	-
k#VIEW	=	0006E	-
kc-CHR	=	00000	-
kc-LIN	=	00004	-
kcATTN	=	0000E	- 0799E 07E96
kcBKSP	=	00007	-
kcBOT	=	00015	- 079B7

kcCALC = 00017 - 026E5 02769
 kcCONT = 00010 -
 kcCTRL = 0000A -
 kcDOWN = 00013 - 079AD
 kcEOL = 0000D - 0270A 07999 07E9B
 kcFLFT = 00005 -
 kcFRT = 00006 - 0272A
 kcGON = 00016 -
 kcI/R = 00002 -
 kcLAST = 00019 -
 kcLC = 00001 -
 kcLERR = 0001A -
 kcLFT = 00008 - 026DC
 kcOFF = 00018 - 079BC 07EA0
 kcRT = 00009 - 02721
 kcRUN = 0000F - 02733 079A3
 kcSST = 00011 -
 kcTOP = 00014 - 079B2
 kcUP = 00012 - 079A8
 kcUSER = 00003 -
 kcUSEX = 0000C -
 kcVIEW = 0000B - 026D3
 kwt10 = 0771D - 07710

lACCSb = 00001 -
 lAp = 00010 -
 lBPOSp = 00005 -
 lCOPYb = 00001 -
 lCPOsb = 00006 -
 lD0p = 00005 -
 lD1p = 00005 -
 lDATEh = 00006 -
 lDBEGb = 0000B -
 lDEVC = 00005 -
 lDEVcb = 00001 -
 lDLENb = 00006 -
 lDp = 00010 -
 lEOL = 00002 - 04B3D
 lFBEGb = 00006 -
 lFBF#b = 00003 -
 lFIB = 0003F -
 lFIL#b = 00002 -
 lFILSV = 00032 - 0C79D
 lFLAGh = 00002 -
 lFLENh = 00005 - 0C8C7
 lFNAM+ = 00004 -
 lFNAM8 = 00010 -
 lFNAMh = 00010 - 0C8B4 0C8B7
 lFSIZb = 00006 -
 lFTYPb = 00004 -
 lFTYPb = 00004 - 0C8C1
 lLXADR = 00005 -
 lLXENT = 0000B -
 lLXFAD = 00005 -
 lLXID = 00002 -
 lLXTKR = 00004 -
 lMSGp = 00004 -
 lPOL#p = 00005 -
 lPOLLp = 00005 -
 lPOLSV = 0003E -

lPOLra	=	00006	-
lPROTb	=	00001	-
lREC#b	=	00004	-
lRECLb	=	00004	-
lRELn	=	00005	-
lRTN1p	=	00005	-
lRTN2p	=	00005	-
lRTN3p	=	00005	-
lSHLNb	=	00002	-
lSPDTB	=	0004E	-
lSPDn	=	00001	-
lSPDn2	=	00001	-
lTEXTp	=	00004	-
lTIMEh	=	00004	-
o41sod	=	00005	-
oACCSb	=	0000B	-
oAp	=	0003E	-
oBNsod	=	00011	-
oBPOSp	=	00005	-
oBSsod	=	00011	-
oCOPYb	=	0000A	-
oCPOsb	=	00028	-
oD0p	=	00019	-
oD1p	=	0001E	-
oDATEh	=	0001A	-
oDAsod	=	0000D	-
oDBEGb	=	00015	-
oDEVcb	=	0000C	-
oDLEnb	=	0002E	-
oDp	=	0002E	-
oFBEGb	=	0000D	-
oFBF#b	=	00002	-
oFIL#b	=	00000	-
oFLAGh	=	00014	-
oFLENh	=	00020	-
oFLSTr	=	00031	- 0547F
oFNAMh	=	00000	-
oFSIZb	=	00039	-
oFT-FL	=	00010	-
oFTYPb	=	00005	-
oFTYPb	=	00010	- 054BE 0A3E5 0A92A
oIMPLh	=	00025	-
oKYsod	=	00005	-
oLXsod	=	00005	-
oMAINT	=	0005D	-
oMSGPT	=	00009	-
oPOL#p	=	0000A	-
oPROTb	=	00009	-
oREC#b	=	00020	-
oRECLb	=	00024	-
oRELn	=	00034	-
oRTN1p	=	0000A	-
oRTN2p	=	0000F	-
oRTN3p	=	00014	-
oSHLNb	=	00013	-
oSPDTB	=	00111	-
oSPDn2	=	0000E	-
oSUBLn	=	00025	-
oTIMEh	=	00016	-

oTXsod	=	00005	-
pBSCen	=	000F5	-
pBSCex	=	000F6	-
pCALRS	=	00036	-
pCALSV	=	00037	-
pCAT	=	00006	-
pCAT\$	=	00007	-
pCLDST	=	000FF	- 0267A
pCMPLX	=	00038	-
pCONFG	=	000FB	- 02669 0CB4C
pCOPYx	=	00008	-
pCRDAB	=	00033	-
pCREAT	=	00009	-
pCRT=8	=	00023	-
pCURSR	=	00029	-
pDATLN	=	0002A	-
pDEVCP	=	00001	-
pDIDST	=	0000A	-
pDSWKY	=	000FD	-
pDSWNK	=	000FE	-
pEDIT	=	0002B	- 02694
pENTER	=	00012	-
pEOFIL	=	00025	-
pERROR	=	000F2	-
pExcpt	=	000F8	-
pFASCH	=	0002C	-
pFILDC	=	00002	-
pFILXQ	=	00003	-
pFINDF	=	00017	-
pFNIN	=	0003D	-
pFNOUT	=	0003E	-
pFPROT	=	0000B	-
pFSPCp	=	00004	-
pFSPCx	=	00005	-
pFTYPE	=	0002D	- 02683
pIMCHR	=	0001E	-
pIMXCH	=	0001F	-
pIMXQT	=	0001D	-
pIMbck	=	00020	-
pIMcpi	=	00021	-
pIMcpw	=	00022	-
pKYDF	=	0001B	- 02658
pLIST	=	0000C	-
pLIST2	=	0002E	-
pMEM	=	000F1	-
pMERGE	=	0000D	-
pMNLP	=	000FA	-
pMRGE2	=	0002F	- 0269D
pPARSE	=	000F4	-
pPRGPR	=	00032	-
pPRIN#	=	00026	-
pPRTCL	=	0000E	-
pPRTIS	=	0000F	- 08A0B
pPURGE	=	00010	-
pPWROF	=	000FC	-
pRCRD	=	00034	-
pRDCBF	=	00018	-
pRDNBF	=	00019	-
pREAD#	=	00027	-

pREN = 00039 -
pRNAME = 00011 -
pRTNTp = 0003A -
pRUNft = 00030 -
pRUNnB = 00031 -
pSREC# = 00028 -
pSREQ = 000F9 -
pTEST = 000F0 -
pTMR# = 0003B -
pTRANS = 000EF - 05A67
pTRFMx = 0003C -
pVER\$ = 00000 -
pWARN = 000F3 -
pWCRD = 00035 -
pWCRD8 = 00024 -
pWRCBF = 0001A -
pWTKY = 0001C -
pZERPG = 000F7 -

qENDI = 00002 - 010FD
qENDS = 00001 - 010F7 051C9

sARITH = 00007 -
sBADRC = 00008 - 0A8E7 0A94B 0A964 0A981 0A9D3 0A9DF
sBYEx = 00000 -
sC/P = 00001 -
sCARD = 00002 -
sCARDC = 00008 -
sCHAIN = 0000B -
sCONT = 0000A -
sCONTK = 00009 -
sCURBT = 00003 -
sCURUD = 00004 -
sCURUP = 00002 -
sCntg = 00002 -
sCplxP = 00007 -
sDEST = 00003 - 0C7CB 0C7E0 0C852 0C857 0C86F
sENDx = 00001 -
sEOF = 00007 - 0A961 0A9D8
sERROR = 00000 -
sEXTDV = 00000 -
sEXTGS = 00005 -
sFOUND = 0000A -
sGOSUB = 00003 -
sI/OBF = 0000A -
sINFRD = 0000A -
sINX = 00005 -
sIRAM = 00002 -
sIX = 00007 -
sInit = 00003 -
sKEYS = 00005 -
sMAINC = 00005 -
sMULT = 00008 -
sNEGRD = 0000B -
sNoChn = 00002 -
sONERR = 00004 -
sONTMR = 00006 -
sPCRD = 00008 -
sPRGCF = 0000B -
sRAD = 00009 -

sRDX	=	0000B	-
sREADI	=	00004	-
sRENAM	=	00006	-
sRENUM	=	00008	-
sRESTR	=	0000A	-
sRETRN	=	00000	-
sRFILE	=	00008	-
sRUNBn	=	00004	-
sRUNDC	=	00007	-
sSIGN	=	00009	-
sSST	=	00002	-
sSSTdc	=	00001	-
sSTAT	=	00006	-
sSTOP	=	00005	-
sSpec1	=	00006	-
sUNDEF	=	00001	-
sXCPT	=	00004	-
sXQT	=	00000	-
sXWORD	=	00009	-
t!	=	000FC	- 0151D 05133 053DB
t%	=	00085	-
t&	=	00089	-
t*	=	00083	-
t+	=	00087	-
t-	=	00082	-
t/	=	00084	-
t@	=	000F4	- 01518
tABS	=	000A2	-
tACOS	=	0009A	-
tADD	=	000D5	-
tADIG0	=	00060	-
tADIG1	=	00061	-
tADIG2	=	00062	-
tADIG3	=	00063	-
tADIG4	=	00064	-
tADIG5	=	00065	-
tADIG6	=	00066	-
tADIG7	=	00067	-
tADIG8	=	00068	-
tADIG9	=	00069	-
tALL	=	000F8	- 015FA
tAND	=	0008B	-
tANGLE	=	601B3	-
tARRAY	=	0007D	-
tASIN	=	00099	-
tATAN	=	0009B	-
tAUTO	=	000EE	-
tBASE	=	000E9	-
tBEEP	=	000E8	-
tBIG	=	00010	-
tCALL	=	000F9	- 03010
tCARD	=	000D0	- 0C7EC
tCASE	=	00062	- 05182
tCAT	=	000EC	-
tCEIL	=	00072	-
tCFLAG	=	000FA	-
tCHR\$	=	000A4	-
tCLOCK	=	501EF	-
tCMLPX	=	0007A	-

```

tCOLON      = 000E2 -
tCOMMA     = 000F1 - 00E74 015C1 02B34 02B4E 04A12 050B8 05457 0A0B8
tCOPY      = 000B5 -
tCOS       = 00097 -
tCVAL     = 000E1 -
tDATA     = 000C6 - 0511A
tDATE     = 00077 -
tDATE$    = 00078 -
tDEF      = 000B9 - 05115
tDEG      = 0006F -
tDEGRE    = 000D3 -
tDELAY    = 000D6 -
tDELET    = 000B7 -
tDIM      = 000CC -
tDISP     = 000C5 -
tDIV      = 00086 -
tDMYAR    = 0007E -
tDSTRY    = 000BE -
tDVZ     = 000B1 -
tEDIT     = 000B8 -
tELSE     = 000F5 - 01522
tELSE2    = 00064 - 05178
tEND      = 000DA - 05124
tEND2     = 00042 - 05187
tENDDF    = 000BA - 05129
tENDSB    = 000C2 - 0512E
tENTER    = 4FFEF -
tEOL     = 000F0 - 01208 01513 0301A 04D11 04F72 050DB 07B78 0B949 0C8D4
tEPS      = 00071 -
tERRL     = 00075 -
tERRN     = 00076 -
tERROR    = 000E3 -
tEXOR     = 0008C -
tEXP      = 00094 -
tEXTIF    = 000F4 -
tEXTND    = 601EF -
tFACT     = 000A8 -
tFETCH    = 000C8 -
tFFN     = 000B4 -
tFLOW     = 901EF -
tFLT1     = 0001D -
tFLT10    = 00014 -
tFLT11    = 00013 -
tFLT12    = 00012 -
tFLT2     = 0001C -
tFLT3     = 0001B -
tFLT4     = 0001A -
tFLT5     = 00019 -
tFLT6     = 00018 -
tFLT7     = 00017 -
tFLT8     = 00016 -
tFLT9     = 00015 -
tFN       = 0007C -
tFOR      = 000C3 - 05106
tFP       = 0006B -
tGOSUB    = 000DC -
tGOTO     = 000DD -
tIF       = 000DF - 05138
tIF2     = 00063 - 05173
tIMAGE    = 000FF -

```

tIN	=	000F2	-
tINF	=	00070	-
tINPUT	=	000C9	-
tINT	=	0009C	-
tINT10	=	00004	-
tINT11	=	00003	-
tINT12	=	00002	-
tINT2	=	0000C	-
tINT3	=	0000B	-
tINT4	=	0000A	-
tINT5	=	00009	-
tINT6	=	00008	-
tINT7	=	00007	-
tINT8	=	00006	-
tINT9	=	00005	-
tINTEG	=	000CA	-
tINTO	=	E01EF	-
tINTR	=	015FF	-
tINX	=	000B2	-
tIP	=	0006A	-
tIS	=	000E7	-
tISUB\$	=	000A7	-
tIVL	=	000AE	-
tKEY	=	000E5	-
tKEY\$	=	00073	-
tKEYS	=	000CF	- 0C96B
tLBLRF	=	0000E	-
tLBLST	=	000F6	- 0511F
tLEN	=	000A9	-
tLET	=	000C0	-
tLINE#	=	0000F	- 015B8
tLINPT	=	000BF	-
tLIST	=	000BB	-
tLITRL	=	000C4	- 03012 07B6E
tLN	=	00091	-
tLOG	=	00090	-
tLOG10	=	00093	-
tLOOP	=	00060	- 05164
tLPRP	=	000AA	-
tLR	=	000B6	-
tMAIN	=	000D2	-
tMATH	=	601EF	-
tMAX	=	000AD	-
tMAXRL	=	0006C	-
tMEAN	=	0009D	-
tMIN	=	000AC	-
tMOD	=	00074	-
tNAME	=	000BD	-
tNEAR	=	C01EF	-
tNEG	=	D01EF	-
tNEXT	=	000C4	- 0510B 066A5
tNOT	=	00081	-
tNUM	=	000A3	-
tOFF	=	000E1	-
tON	=	000E0	- 0125C 01261 09083
tOPT'N	=	000ED	-
tOR	=	0008D	-
tOVF	=	000AF	-
tPAUSE	=	000D7	-
tPCRD	=	E01EF	-

tPI	=	00079	-
tPORT	=	000D1	-
tPOS	=	201B3	-
tPREDV	=	0009F	-
tPRINT	=	000CD	-
tPRMEN	=	000F8	- 03018
tPRMST	=	000F3	-
tPURGE	=	000EB	-
tRAD	=	0006E	-
tRDIAN	=	000D4	-
tREAD	=	000C7	-
tREAL	=	000BC	-
tRELOP	=	0008A	- 00D8B 01143 0A035
tREM	=	000E6	- 053E4
tREPEAT	=	00044	- 05169
tRES	=	0007F	-
tRESTR	=	000DE	-
tRETRN	=	000DB	-
tRFILE	=	000DE	-
tRMD	=	0006D	-
tRND	=	000A0	-
tROUND	=	C01EF	-
tRUN	=	000FE	-
tSDEV	=	0009E	-
tSELECT	=	00061	- 0517D
tSEMIC	=	000F2	-
tSFLAG	=	000FB	-
tSGN	=	000A1	-
tSHORT	=	000CB	-
tSIN	=	00096	-
tSMALL	=	00011	-
tSQR	=	00092	-
tSTAT	=	000CE	-
tSTEP	=	000F6	-
tSTOP	=	000D9	-
tSTR\$	=	000A6	-
tSUB	=	000C1	- 05110
tSVAR	=	0002D	-
tTAB	=	000F7	-
tTAN	=	00098	-
tTHEN	=	000F4	- 011F1
tTIME	=	0007B	-
tTIME\$	=	00095	-
tTIMER	=	000E4	-
tTO	=	000F3	- 00DA9 00E79 0116C 01491 0157F 01603 0161A 04A1C 0A07D + 0C7BF
tTRACE	=	000EA	-
tUNF	=	000B0	-
tUNTIL	=	00045	- 0516E
tUPRC\$	=	000AB	-
tUSER	=	000E2	-
tUSING	=	000FD	-
tVAL	=	000A5	-
tVARS	=	B01EF	-
tWAIT	=	000D8	-
tWHILE	=	00043	- 0515F
tXFN	=	000B3	- 09E53
tXWORD	=	000EF	- 00E7E 010C7 010D0 010D9 010E2 0112B 04A17 05101 09E4A + 0A4BF 0A572
tZ	=	0005A	-

tZERO	=	C01EF	-
t^	=	00080	-
uALit	=	000F7	-
uCPLXC	=	000EE	-
uDELIM	=	000F4	-
uHKB^	=	000F6	-
uIMXCH	=	000D4	-
uIMbck	=	000DC	-
uIMend	=	000F0	-
uIMsta	=	000DE	-
uJMPd1	=	000DB	-
uJMPst	=	000DA	-
uJMP{ }	=	000D9	-
uLOOPB	=	000D2	-
uLOOPP	=	000EF	-
uLOOPS	=	000D3	-
uMODES	=	0BDB1	-
uMULT	=	000D1	-
uNUMEn	=	000FC	-
uNUMEs	=	000FD	-
uNUMFn	=	000FA	-
uNUMFs	=	000FB	-
uNUMNn	=	000F8	-
uNUMNs	=	000F9	-
uOPNM-	=	000DF	-
uOPNNM	=	000D8	-
uOPNWM	=	000E0	-
uRES12	=	0C994	- 05618 06EF8 071BC
uRESD1	=	0E1EE	-
uRESNX	=	0C9BD	-
uRESTP	=	000F1	-
uRESXT	=	0C9C1	-
uRND>P	=	0C9CF	-
uSTRPT	=	000D0	-
uTEST	=	0D435	- 0A114
xANGLE	=	00006	-
xCLOCK	=	00015	-
xEXTND	=	00026	-
xFLOW	=	00029	-
xINTO	=	0002E	-
xMATH	=	00036	-
xNEAR	=	0003C	-
xNEG	=	0003D	-
xPCRD	=	0003E	-
xPOS	=	00042	-
xROUND	=	0004C	-
xVARS	=	0005B	-
xZERO	=	0001C	-

/SLOAD: End of Saturn Loader Execution


```
1          TITLE JPC LEX: MODULE TABLES
2
3          * *****
4          * tables module
5          *
6          * ROM header, LEX header
7          * main table, text table
8          *
9          * rebuilt from JPC rev D binary, J-F G. 2005
10         * updated for rev X (R. Rosenbaum), J-F G. 2006:
11         *   LEX ON/OFF replaced by LXON / LXOFF,
12         *   ARR, PGCD and PPCM renamed to PERM, GCD, LCM.
13         *
14         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
15         * *****
16
17 00000          ABS      #0
18
19
20         * Enable the HP-71 ROM header for listing building only,
21         * to get correct generation of absolute address references.
22         * Use no header for LEX generation and subsequent addition
23         * of LIF header for transfer to the HP-71.
24         * Check for the corresponding change in drive.a.
25
26         * *****
27         * -- HP-71 ROM header--
28         * *****
29 00000          BSS      8          8 nibble ROM ID
30         * -- HP 71 file header, 37 (#25) nibbles --
31 00008 A40534C4 =JPCLEX NIBASC 'JPCLEX ' File Name
          54850202
32 00018 0000          CON(4) =fLEX      File Type = Lex
33 0001C 00          CON(2) 0          Copy Code = 0
34 0001E 0000          CON(4) 0          Time = 0
35 00022 0000          CON(4) 0          Date = 0
36 00026 00          CON(2) 0
37 00028 00000          REL(5) =FiLeNd      End of File Chain
38
39
40         * *****
41         * --LEX header--
42         * *****
43 0002D 1E          CON(2) #E1          JPC LEX ID
44 0002F 10          CON(2) #01          LOWEST TOKEN
45 00031 F6          CON(2) #6F          HIGHEST TOKEN
46 00033 00000          REL(5) =PKLEX      NEXT LINKED LEX
47 00038 0          NIBHEX 0          SPEED TABLE EXISTS
48 00039 000          CON(3) (TxT1A)-(TxTSt1)
49 0003C E80          CON(3) (TxT1B)-(TxTSt1)
50 0003F 4A0          CON(3) (TxT1C)-(TxTSt1)
51 00042 DE0          CON(3) (TxT1D)-(TxTSt1)
52 00045 551          CON(3) (TxT1E)-(TxTSt1)
53 00048 6C1          CON(3) (TxT1F)-(TxTSt1)
54 0004B 922          CON(3) (TxT1G)-(TxTSt1)
55 0004E C42          CON(3) (TxT1H)-(TxTSt1)
56 00051 582          CON(3) (TxT1I)-(TxTSt1)
```

```
57 00054 285          CON(3) (TxT1J)-(TxTSt1)
58 00057 CA2         CON(3) (TxT1K)-(TxTSt1)
59 0005A 3B2         CON(3) (TxT1L)-(TxTSt1)
60 0005D CE2         CON(3) (TxT1M)-(TxTSt1)
61 00060 A53         CON(3) (TxT1N)-(TxTSt1)
62 00063 583         CON(3) (TxT1O)-(TxTSt1)
63 00066 693         CON(3) (TxT1P)-(TxTSt1)
64 00069 285         CON(3) (TxT1Q)-(TxTSt1)
65 0006C F24         CON(3) (TxT1R)-(TxTSt1)
66 0006F A94         CON(3) (TxT1S)-(TxTSt1)
67 00072 E05         CON(3) (TxT1T)-(TxTSt1)
68 00075 B15         CON(3) (TxT1U)-(TxTSt1)
69 00078 B35         CON(3) (TxT1V)-(TxTSt1)
70 0007B C45         CON(3) (TxT1W)-(TxTSt1)
71 0007E 285         CON(3) (TxT1X)-(TxTSt1)
72 00081 285         CON(3) (TxT1Y)-(TxTSt1)
73 00084 285         CON(3) (TxT1Z)-(TxTSt1)
74 00087 0           NIBHEX 0
75 00088 5F30        REL(4) (TxTSt1)+1 OFFSET TO TEXT TABLE
76 0008C 0000        REL(4) =MSGTB1    OFFSET TO MESSAGE TABLE
77 00090 00000       REL(5) =POLHND   OFFSET TO POLL HANDLER
78
79                   STITLE Main Table
```

```
80
81 * *****
82 * -----MAIN TABLE-----
83 * *****
84 00095 000 CON(3) (ADBUft) - (TxTSt1)
85 00098 00000 REL(5) =ADBUFe
86 0009D F NIBHEX F
87 0009E D60 CON(3) (ASct) - (TxTSt1)
88 000A1 00000 REL(5) =ASce
89 000A6 F NIBHEX F
90 000A7 870 CON(3) (ATHt) - (TxTSt1)
91 000AA 00000 REL(5) =ATHe
92 000AF F NIBHEX F
93 000B0 A72 CON(3) (HTAt) - (TxTSt1)
94 000B3 00000 REL(5) =HTAe
95 000B8 F NIBHEX F
96 000B9 F24 CON(3) (REDt) - (TxTSt1)
97 000BC 00000 REL(5) =REDe
98 000C1 F NIBHEX F
99 000C2 D64 CON(3) (REPLAt) - (TxTSt1)
100 000C5 00000 REL(5) =REPLAe
101 000CA F NIBHEX F
102 000CB 1D1 CON(3) (FILESt) - (TxTSt1)
103 000CE 00000 REL(5) =FILESe
104 000D3 F NIBHEX F
105 000D4 380 CON(3) (ATTNt) - (TxTSt1)
106 000D7 00000 REL(5) =ATTNe
107 000DC D NIBHEX D
108 000DD FD2 CON(3) (LXOFFt) - (TxTSt1) *** rev X
109 000E0 00000 REL(5) =LXOFFe
110 000E5 D NIBHEX D
111 000E6 9C2 CON(3) (LXONt) - (TxTSt1) *** rev X
112 000E9 00000 REL(5) =LXONe
113 000EE D NIBHEX D
114 000EF 6C1 CON(3) (FKEYt) - (TxTSt1)
115 000F2 00000 REL(5) =FKEYe
116 000F7 D NIBHEX D
117 000F8 AD0 CON(3) (CONTRt) - (TxTSt1)
118 000FB 00000 REL(5) =CONTRe
119 00100 F NIBHEX F
120 00101 B92 CON(3) (INVERt) - (TxTSt1)
121 00104 00000 REL(5) =INVERe
122 00109 D NIBHEX D
123 0010A F65 CON(3) (OBSOLt) - (TxTSt1)
124 0010D EF800 REL(5) OBSOLe
125 00112 0 NIBHEX 0
126 00113 7A3 CON(3) (PAINTt) - (TxTSt1)
127 00116 00000 REL(5) =PAINTe
128 0011B F NIBHEX F
129 0011C B61 CON(3) (ENDUst) - (TxTSt1)
130 0011F 00000 REL(5) =ENDUSe
131 00124 F NIBHEX F
132 00125 A71 CON(3) (ENDUPt) - (TxTSt1)
133 00128 00000 REL(5) =ENDUPe
134 0012D D NIBHEX D
135 0012E AE4 CON(3) (STARTt) - (TxTSt1)
136 00131 00000 REL(5) =STARTe
```

137	00136	F	NIBHEX	F	
138	00137	AA1	CON(3)	(EXECUt) - (TxTSt1)	
139	0013A	00000	REL(5)	=EXECUe	
140	0013F	D	NIBHEX	D	
141	00140	3F3	CON(3)	(PERMt) - (TxTSt1)	*** rev X
142	00143	00000	REL(5)	=ARRe	
143	00148	F	NIBHEX	F	
144	00149	FC0	CON(3)	(COMBt) - (TxTSt1)	
145	0014C	00000	REL(5)	=COMBe	
146	00151	F	NIBHEX	F	
147	00152	C42	CON(3)	(HMSADt) - (TxTSt1)	
148	00155	00000	REL(5)	=HMSADe	
149	0015A	F	NIBHEX	F	
150	0015B	B52	CON(3)	(HMSSUt) - (TxTSt1)	
151	0015E	00000	REL(5)	=HMSSUe	
152	00163	F	NIBHEX	F	
153	00164	A62	CON(3)	(HMSt) - (TxTSt1)	
154	00167	00000	REL(5)	=HMSe	
155	0016C	F	NIBHEX	F	
156	0016D	372	CON(3)	(HRt) - (TxTSt1)	
157	00170	00000	REL(5)	=HRe	
158	00175	F	NIBHEX	F	
159	00176	551	CON(3)	(EDITt) - (TxTSt1)	
160	00179	00000	REL(5)	=EDITe	
161	0017E	5	NIBHEX	5	
162	0017F	DD4	CON(3)	(STACKt) - (TxTSt1)	
163	00182	00000	REL(5)	=STACKe	
164	00187	D	NIBHEX	D	
165	00188	F65	CON(3)	(OBSOLt) - (TxTSt1)	
166	0018B	08800	REL(5)	OBSOLe	
167	00190	0	NIBHEX	0	
168	00191	003	CON(3)	(MARGIt) - (TxTSt1)	
169	00194	00000	REL(5)	=MARGIe	
170	00199	D	NIBHEX	D	
171	0019A	A53	CON(3)	(NEXTOt) - (TxTSt1)	
172	0019D	00000	REL(5)	=NEXTOe	
173	001A2	F	NIBHEX	F	
174	001A3	583	CON(3)	(OPCODt) - (TxTSt1)	
175	001A6	00000	REL(5)	=OPCODe	
176	001AB	F	NIBHEX	F	
177	001AC	DF4	CON(3)	(SYSEDt) - (TxTSt1)	
178	001AF	00000	REL(5)	=SYSEDe	
179	001B4	D	NIBHEX	D	
180	001B5	443	CON(3)	(MENUt) - (TxTSt1)	
181	001B8	00000	REL(5)	=MENUe	
182	001BD	F	NIBHEX	F	
183	001BE	FA0	CON(3)	(CENTEt) - (TxTSt1)	
184	001C1	00000	REL(5)	=CENTEe	
185	001C6	F	NIBHEX	F	
186	001C7	0C0	CON(3)	(CESURt) - (TxTSt1)	
187	001CA	00000	REL(5)	=CESURe	
188	001CF	F	NIBHEX	F	
189	001D0	EF1	CON(3)	(FORMAt) - (TxTSt1)	
190	001D3	00000	REL(5)	=FORMAe	
191	001D8	F	NIBHEX	F	
192	001D9	A34	CON(3)	(REDUCt) - (TxTSt1)	
193	001DC	00000	REL(5)	=REDUCe	

194	001E1	F	NIBHEX	F
195	001E2	5C4	CON(3)	(SPACEt) - (TxTSt1)
196	001E5	00000	REL(5)	=SPACEe
197	001EA	F	NIBHEX	F
198	001EB	E80	CON(3)	(BELLt) - (TxTSt1)
199	001EE	00000	REL(5)	=BELLe
200	001F3	D	NIBHEX	D
201	001F4	990	CON(3)	(BOLDt) - (TxTSt1)
202	001F7	00000	REL(5)	=BOLDe
203	001FC	D	NIBHEX	D
204	001FD	4D3	CON(3)	(PCRT) - (TxTSt1)
205	00200	00000	REL(5)	=PCRe
206	00205	D	NIBHEX	D
207	00206	F91	CON(3)	(ESCt) - (TxTSt1)
208	00209	00000	REL(5)	=ESCe
209	0020E	F	NIBHEX	F
210	0020F	EF3	CON(3)	(PFFt) - (TxTSt1)
211	00212	00000	REL(5)	=PFFe
212	00217	D	NIBHEX	D
213	00218	014	CON(3)	(PLFt) - (TxTSt1)
214	0021B	00000	REL(5)	=PLFe
215	00220	D	NIBHEX	D
216	00221	F43	CON(3)	(MODEt) - (TxTSt1)
217	00224	00000	REL(5)	=MODEe
218	00229	D	NIBHEX	D
219	0022A	8E3	CON(3)	(PERFt) - (TxTSt1)
220	0022D	00000	REL(5)	=PERFe
221	00232	D	NIBHEX	D
222	00233	693	CON(3)	(PAGElt) - (TxTSt1)
223	00236	00000	REL(5)	=PAGEle
224	0023B	D	NIBHEX	D
225	0023C	B15	CON(3)	(UNDERt) - (TxTSt1)
226	0023F	00000	REL(5)	=UNDERe
227	00244	D	NIBHEX	D
228	00245	955	CON(3)	(WRAPt) - (TxTSt1)
229	00248	00000	REL(5)	=WRAPe
230	0024D	D	NIBHEX	D
231	0024E	EF0	CON(3)	(DATESt) - (TxTSt1)
232	00251	00000	REL(5)	=DATESe
233	00256	F	NIBHEX	F
234	00257	DE0	CON(3)	(DATEAt) - (TxTSt1)
235	0025A	00000	REL(5)	=DATEAe
236	0025F	F	NIBHEX	F
237	00260	021	CON(3)	(DDAYSt) - (TxTSt1)
238	00263	00000	REL(5)	=DDAYSe
239	00268	F	NIBHEX	F
240	00269	831	CON(3)	(DMYt) - (TxTSt1)
241	0026C	00000	REL(5)	=DMYe
242	00271	D	NIBHEX	D
243	00272	141	CON(3)	(DOWst) - (TxTSt1)
244	00275	00000	REL(5)	=DOWse
245	0027A	F	NIBHEX	F
246	0027B	C41	CON(3)	(DOWt) - (TxTSt1)
247	0027E	00000	REL(5)	=DOWe
248	00283	F	NIBHEX	F
249	00284	523	CON(3)	(MDYt) - (TxTSt1)
250	00287	00000	REL(5)	=MDYe

251	0028C	D	NIBHEX	D
252	0028D	F03	CON(3)	(MAXDt) - (TxTSt1)
253	00290	00000	REL(5)	=MAXDe
254	00295	F	NIBHEX	F
255	00296	A13	CON(3)	(MAXMt) - (TxTSt1)
256	00299	00000	REL(5)	=MAXMe
257	0029E	F	NIBHEX	F
258	0029F	E23	CON(3)	(MEMDt) - (TxTSt1)
259	002A2	00000	REL(5)	=MEMDe
260	002A7	F	NIBHEX	F
261	002A8	933	CON(3)	(MEMMt) - (TxTSt1)
262	002AB	00000	REL(5)	=MEMMe
263	002B0	F	NIBHEX	F
264	002B1	BB1	CON(3)	(EXITt) - (TxTSt1)
265	002B4	00000	REL(5)	=EXITe
266	002B9	C	NIBHEX	C
267	002BA	B63	CON(3)	(NLOOPt) - (TxTSt1)
268	002BD	00000	REL(5)	=NLOOPe
269	002C2	F	NIBHEX	F
270	002C3	4B3	CON(3)	(PPOLLt) - (TxTSt1)
271	002C6	00000	REL(5)	=PPOLLe
272	002CB	F	NIBHEX	F
273	002CC	8B4	CON(3)	(SLEEPt) - (TxTSt1)
274	002CF	00000	REL(5)	=SLEEPe
275	002D4	D	NIBHEX	D
276	002D5	4D4	CON(3)	(SRQt) - (TxTSt1)
277	002D8	00000	REL(5)	=SRQe
278	002DD	F	NIBHEX	F
279	002DE	781	CON(3)	(ENDt) - (TxTSt1)
280	002E1	00000	REL(5)	=END2e
281	002E6	D	NIBHEX	D
282	002E7	C45	CON(3)	(WHILEt) - (TxTSt1)
283	002EA	00000	REL(5)	=WHILEe
284	002EF	D	NIBHEX	D
285	002F0	E54	CON(3)	(REPEAt) - (TxTSt1)
286	002F3	00000	REL(5)	=REPEAe
287	002F8	D	NIBHEX	D
288	002F9	E25	CON(3)	(UNTILt) - (TxTSt1)
289	002FC	00000	REL(5)	=UNTILe
290	00301	D	NIBHEX	D
291	00302	CB2	CON(3)	(LEAVEt) - (TxTSt1)
292	00305	00000	REL(5)	=LEAVEe
293	0030A	D	NIBHEX	D
294	0030B	B35	CON(3)	(VARSWt) - (TxTSt1)
295	0030E	00000	REL(5)	=VARSWe
296	00313	D	NIBHEX	D
297	00314	091	CON(3)	(ENTRYt) - (TxTSt1)
298	00317	00000	REL(5)	=ENTRYe
299	0031C	F	NIBHEX	F
300	0031D	E05	CON(3)	(TOKENt) - (TxTSt1)
301	00320	00000	REL(5)	=TOKENe
302	00325	F	NIBHEX	F
303	00326	F65	CON(3)	(OBSOLt) - (TxTSt1)
304	00329	2E600	REL(5)	OBSOLe
305	0032E	0	NIBHEX	0
306	0032F	4E1	CON(3)	(FINDt) - (TxTSt1)
307	00332	00000	REL(5)	=FINDe

308	00337	1	NIBHEX	1	
309	00338	CE2	CON(3)	(MAPst) - (TxTSt1)	
310	0033B	00000	REL(5)	=MAPse	
311	00340	F	NIBHEX	F	
312	00341	7F2	CON(3)	(MAPt) - (TxTSt1)	
313	00344	00000	REL(5)	=MAPe	
314	00349	D	NIBHEX	D	
315	0034A	232	CON(3)	(GLINet) - (TxTSt1)	
316	0034D	00000	REL(5)	=GLINEe	
317	00352	D	NIBHEX	D	
318	00353	F32	CON(3)	(GPSETt) - (TxTSt1)	
319	00356	00000	REL(5)	=GPSETe	
320	0035B	D	NIBHEX	D	
321	0035C	D64	CON(3)	(REPLAt) - (TxTSt1)	
322	0035F	00000	REL(5)	=REPLAe	
323	00364	F	NIBHEX	F	
324	00365	9A4	CON(3)	(SHRInt) - (TxTSt1)	
325	00368	00000	REL(5)	=SHRIne	
326	0036D	D	NIBHEX	D	
327	0036E	F02	CON(3)	(FPRIMt) - (TxTSt1)	
328	00371	00000	REL(5)	=FPRIME	
329	00376	F	NIBHEX	F	
330	00377	873	CON(3)	(NPRIMt) - (TxTSt1)	
331	0037A	00000	REL(5)	=NPRIME	
332	0037F	F	NIBHEX	F	
333	00380	922	CON(3)	(GCDt) - (TxTSt1)	*** rev X
334	00383	00000	REL(5)	=PGCDe	
335	00388	F	NIBHEX	F	
336	00389	704	CON(3)	(PHIt) - (TxTSt1)	
337	0038C	00000	REL(5)	=PHIe	
338	00391	F	NIBHEX	F	
339	00392	3B2	CON(3)	(LCMt) - (TxTSt1)	*** rev X
340	00395	00000	REL(5)	=PPCMe	
341	0039A	F	NIBHEX	F	
342	0039B	424	CON(3)	(PRIMt) - (TxTSt1)	
343	0039E	00000	REL(5)	=PRIME	
344	003A3	F	NIBHEX	F	
345	003A4	C12	CON(3)	(FRAct) - (TxTSt1)	
346	003A7	00000	REL(5)	=FRACe	
347	003AC	F	NIBHEX	F	
348	003AD	914	CON(3)	(POSIt) - (TxTSt1)	
349	003B0	00000	REL(5)	=POSIE	
350	003B5	F	NIBHEX	F	
351	003B6	111	CON(3)	(DBLISt) - (TxTSt1)	
352	003B9	00000	REL(5)	=DBLISe	
353	003BE	D	NIBHEX	D	
354	003BF	5C3	CON(3)	(PBLISt) - (TxTSt1)	
355	003C2	00000	REL(5)	=PBLISe	
356	003C7	D	NIBHEX	D	
357	003C8	B44	CON(3)	(RENUMt) - (TxTSt1)	
358	003CB	00000	REL(5)	=RENUMe	
359	003D0	D	NIBHEX	D	
360	003D1	FE1	CON(3)	(FINPUt) - (TxTSt1)	
361	003D4	00000	REL(5)	=FINPUe	
362	003D9	D	NIBHEX	D	
363	003DA	D84	CON(3)	(RREct) - (TxTSt1)	
364	003DD	00000	REL(5)	=RRECe	

```
365 003E2 F          NIBHEX F
366 003E3 465       CON(3) (WREct) - (TxTSt1)
367 003E6 00000     REL(5) =WRECe
368 003EB D          NIBHEX D
369 003EC 4D2       CON(3) (LOOPt) - (TxTSt1)
370 003EF 00000     REL(5) =LOOPe
371 003F4 D          NIBHEX D
372 003F5 A94       CON(3) (SELEct) - (TxTSt1)
373 003F8 00000     REL(5) =SELECe
374 003FD D          NIBHEX D
375 003FE 4A0       CON(3) (CASEt) - (TxTSt1)
376 00401 00000     REL(5) =CASEe
377 00406 D          NIBHEX D
378 00407 582       CON(3) (IFt) - (TxTSt1)
379 0040A 00000     REL(5) =IF2e
380 0040F 9          NIBHEX 9
381 00410 061       CON(3) (ELSEt) - (TxTSt1)
382 00413 00000     REL(5) =ELSE2e
383 00418 D          NIBHEX D
384 00419 CA2       CON(3) (KAt) - (TxTSt1)
385 0041C 00000     REL(5) =KAe
386 00421 D          NIBHEX D
387 00422 F00       CON(3) (ADCREt) - (TxTSt1)
388 00425 00000     REL(5) =ADCREe
389 0042A D          NIBHEX D
390 0042B 220       CON(3) (ADDELt) - (TxTSt1)
391 0042E 00000     REL(5) =ADDELe
392 00433 D          NIBHEX D
393 00434 530       CON(3) (ADFINt) - (TxTSt1)
394 00437 00000     REL(5) =ADFINe
395 0043C F          NIBHEX F
396 0043D 440       CON(3) (ADGETt) - (TxTSt1)
397 00440 00000     REL(5) =ADGETe
398 00445 D          NIBHEX D
399 00446 150       CON(3) (ADPUTt) - (TxTSt1)
400 00449 00000     REL(5) =ADPUTe
401 0044E D          NIBHEX D
402 0044F E50       CON(3) (ADSIzT) - (TxTSt1)
403 00452 00000     REL(5) =ADSIze
404 00457 F          NIBHEX F
405 00458 084       CON(3) (ROMANt) - (TxTSt1)
406 0045B 00000     REL(5) =ROMANe
407 00460 D          NIBHEX D
408 00461 C82       CON(3) (INDENT) - (TxTSt1)
409 00464 00000     CON(5) 0          IDENT NOT AN ENTRY
410 00469 0          NIBHEX 0
411 0046A D21       CON(3) (DDIRt) - (TxTSt1)
412 0046D 00000     REL(5) =DDIRe
413 00472 D          NIBHEX D
414 00473 DD3       CON(3) (PDIRt) - (TxTSt1)
415 00476 00000     REL(5) =PDIRe
416 0047B D          NIBHEX D
417 * ---MAIN TABLE END---
418
419 STITLE Text Table
```



```

420
421      * *****
422      * ---Text table---
423      * *****
424 0047C      TxTSt1
425 0047C      TxT1A
426 0047C B    ADBUFt  CON(1) #B
427 0047D 14442455  NIBASC 'ADBUF$'
      6442
428 00489 10          CON(2) #01
429 0048B F          ADCREt  CON(1) #F
430 0048C 14443425  NIBASC 'ADCREATE'
      541444554
431 0049C 66          CON(2) #66
432 0049E F          ADDELt  CON(1) #F
433 0049F 14444454  NIBASC 'ADDELETE'
      C45444554
434 004AF 76          CON(2) #67
435 004B1 B          ADFINT  CON(1) #B
436 004B2 14446494  NIBASC 'ADFIND'
      E444
437 004BE 86          CON(2) #68
438 004C0 9          ADGETt  CON(1) #9
439 004C1 14447454  NIBASC 'ADGET'
      45
440 004CB 96          CON(2) #69
441 004CD 9          ADPUTt  CON(1) #9
442 004CE 14440555  NIBASC 'ADPUT'
      45
443 004D8 A6          CON(2) #6A
444 004DA B          ADSIZt  CON(1) #B
445 004DB 14443594  NIBASC 'ADSIZE'
      A554
446 004E7 B6          CON(2) #6B
447      *** rev X: renamed to COMB
448      *ARRt  CON(1) 5
449      *      NIBASC 'ARR'
450      *      CON(2) #14
451      ***
452 004E9 7          ASCt  CON(1) #7
453 004EA 14353442  NIBASC 'ASC$'
454 004F2 20          CON(2) #02
455 004F4 7          ATHt  CON(1) #7
456 004F5 14458442  NIBASC 'ATH$'
457 004FD 30          CON(2) #03
458 004FF 7          ATTNt  CON(1) #7
459 00500 144545E4  NIBASC 'ATTN'
460 00508 80          CON(2) #08
461 0050A      TxT1B
462 0050A 7          BELLt  CON(1) #7
463 0050B 2454C4C4  NIBASC 'BELL'
464 00513 72          CON(2) #27
465 00515 7          BOLDt  CON(1) #7
466 00516 24F4C444  NIBASC 'BOLD'
467 0051E 82          CON(2) #28
468 00520      TxT1C
469 00520 7          CASEt  CON(1) #7

```

470	00521	34143554		NIBASC	'CASE'
471	00529	26		CON(2)	#62
472	0052B	D	CENTEt	CON(1)	#D
473	0052C	3454E445 542542		NIBASC	'CENTER\$'
474	0053A	22		CON(2)	#22
475	0053C	B	CESURt	CON(1)	#B
476	0053D	34543555 2554		NIBASC	'CESURE'
477	00549	32		CON(2)	#23
478	0054B	7	COMBt	CON(1)	#7
479	0054C	34F4D424		NIBASC	'COMB'
480	00554	51		CON(2)	#15
481	00556	F	CONTRt	CON(1)	#F
482	00557	34F4E445 25143545		NIBASC	'CONTRAST'
483	00567	C0		CON(2)	#0C
484	00569		TxT1D		
485	00569	D	DATEAt	CON(1)	#D
486	0056A	44144554 144444		NIBASC	'DATEADD'
487	00578	33		CON(2)	#33
488	0057A	F	DATESt	CON(1)	#F
489	0057B	44144554 35452542		NIBASC	'DATESTR\$'
490	0058B	23		CON(2)	#32
491	0058D	B	DBLIST	CON(1)	#B
492	0058E	4424C494 3545		NIBASC	'DBLIST'
493	0059A	A5		CON(2)	#5A
494	0059C	9	DDAYSt	CON(1)	#9
495	0059D	44441495 35		NIBASC	'DDAYS'
496	005A7	43		CON(2)	#34
497	005A9	7	DDIRt	CON(1)	#7
498	005AA	44449425		NIBASC	'DDIR'
499	005B2	E6		CON(2)	#6E
500	005B4	5	DMYt	CON(1)	#5
501	005B5	44D495		NIBASC	'DMY'
502	005BB	53		CON(2)	#35
503	005BD	7	DOWSt	CON(1)	#7
504	005BE	44F47542		NIBASC	'DOW\$'
505	005C6	63		CON(2)	#36
506	005C8	5	DOWt	CON(1)	#5
507	005C9	44F475		NIBASC	'DOW'
508	005CF	73		CON(2)	#37
509	005D1		TxT1E		
510	005D1	7	EDITt	CON(1)	#7
511	005D2	54449445		NIBASC	'EDIT'
512	005DA	A1		CON(2)	#1A
513	005DC	7	ELSEt	CON(1)	#7
514	005DD	54C43554		NIBASC	'ELSE'
515	005E5	46		CON(2)	#64
516	005E7	B	ENDUSt	CON(1)	#B
517	005E8	54E44455 0542		NIBASC	'ENDUP\$'
518	005F4	01		CON(2)	#10

519	005F6	9	ENDUPt	CON(1)	#9
520	005F7	54E44455		NIBASC	'ENDUP'
		05			
521	00601	11		CON(2)	#11
522	00603	5	ENDt	CON(1)	#5
523	00604	54E444		NIBASC	'END'
524	0060A	24		CON(2)	#42
525	0060C	B	ENTRYt	CON(1)	#B
526	0060D	54E44525		NIBASC	'ENTRY\$'
		9542			
527	00619	84		CON(2)	#48
528	0061B	7	ESCt	CON(1)	#7
529	0061C	54353442		NIBASC	'ESC\$'
530	00624	A2		CON(2)	#2A
531	00626	D	EXECUt	CON(1)	#D
532	00627	54855434		NIBASC	'EXECUTE'
		554554			
533	00635	31		CON(2)	#13
534	00637	7	EXITt	CON(1)	#7
535	00638	54859445		NIBASC	'EXIT'
536	00640	D3		CON(2)	#3D
537	00642		TxT1F		
538	00642	7	FKEYt	CON(1)	#7
539	00643	64B45495		NIBASC	'FKEY'
540	0064B	B0		CON(2)	#0B
541	0064D	F	FILESt	CON(1)	#F
542	0064E	6494C454		NIBASC	'FILESIZE'
		3594A554			
543	0065E	70		CON(2)	#07
544	00660	7	FINDt	CON(1)	#7
545	00661	6494E444		NIBASC	'FIND'
546	00669	B4		CON(2)	#4B
547	0066B	B	FINPUt	CON(1)	#B
548	0066C	6494E405		NIBASC	'FINPUT'
		5545			
549	00678	D5		CON(2)	#5D
550	0067A	D	FORMAt	CON(1)	#D
551	0067B	64F425D4		NIBASC	'FORMAT\$'
		144542			
552	00689	42		CON(2)	#24
553	0068B	9	FPRIMt	CON(1)	#9
554	0068C	64052594		NIBASC	'FPRIM'
		D4			
555	00696	25		CON(2)	#52
556	00698	9	FRACt	CON(1)	#9
557	00699	64251434		NIBASC	'FRAC\$'
		42			
558	006A3	85		CON(2)	#58
559	006A5		TxT1G		
560			*** rev X: new name for PGCD		
561	006A5	5	GCDt	CON(1)	5
562	006A6	743444		NIBASC	'GCD'
563	006AC	45		CON(2)	#54
564			***		
565	006AE	9	GLINEt	CON(1)	#9
566	006AF	74C494E4		NIBASC	'GLINE'
		54			

567	006B9	E4		CON(2)	#4E
568	006BB	9	GPSETt	CON(1)	#9
569	006BC	74053554		NIBASC	'GPSET'
		45			
570	006C6	F4		CON(2)	#4F
571	006C8		TxT1H		
572	006C8	B	HMSADt	CON(1)	#B
573	006C9	84D43514		NIBASC	'HMSADD'
		4444			
574	006D5	61		CON(2)	#16
575	006D7	B	HMSSUt	CON(1)	#B
576	006D8	84D43535		NIBASC	'HMSSUB'
		5524			
577	006E4	71		CON(2)	#17
578	006E6	5	HMSt	CON(1)	#5
579	006E7	84D435		NIBASC	'HMS'
580	006ED	81		CON(2)	#18
581	006EF	3	HRt	CON(1)	#3
582	006F0	8425		NIBASC	'HR'
583	006F4	91		CON(2)	#19
584	006F6	7	HTAt	CON(1)	#7
585	006F7	84451442		NIBASC	'HTA\$'
586	006FF	40		CON(2)	#04
587	00701		TxT1I		
588	00701	3	IFt	CON(1)	#3
589	00702	9464		NIBASC	'IF'
590	00706	36		CON(2)	#63
591	00708	B	INDENT	CON(1)	#B
592	00709	94E44454		NIBASC	'INDENT'
		E445			
593	00715	D6		CON(2)	#6D
594	00717	D	INVERT	CON(1)	#D
595	00718	94E46554		NIBASC	'INVERSE'
		253554			
596	00726	D0		CON(2)	#0D
597	00728		TxT1K		
598	00728	3	KAt	CON(1)	#3
599	00729	B414		NIBASC	'KA'
600	0072D	56		CON(2)	#65
601	0072F		TxT1L		
602			*** rev X: new name for PPCM		
603	0072F	5	LCMt	CON(1)	5
604	00730	C434D4		NIBASC	'LCM'
605	00736	65		CON(2)	#56
606			***		
607	00738	9	LEAVEt	CON(1)	#9
608	00739	C4541465		NIBASC	'LEAVE'
		54			
609	00743	64		CON(2)	#46
610			*** rev X: LEX ON/OFF replaced by LXON / LXOFF		
611			*LEXt	CON(1)	5
612			*	NIBASC	'LEX'
613			*	CON(2)	#9*
614	00745	7	LXONt	CON(1)	7
615	00746	C485F4E4		NIBASC	'LXON'
616	0074E	A0		CON(2)	#A
617			***		

618	00750	7	LOOPt	CON(1)	#7
619	00751	C4F4F405		NIBASC	'LOOP'
620	00759	06		CON(2)	#60
621			***		
622	0075B	9	LXOFFt	CON(1)	9
623	0075C	C485F464		NIBASC	'LXOFF'
		64			
624	00766	90		CON(2)	#9
625			***		
626	00768		TxT1M		
627	00768	7	MAPst	CON(1)	#7
628	00769	D4140542		NIBASC	'MAP\$'
629	00771	C4		CON(2)	#4C
630	00773	5	MAPt	CON(1)	#5
631	00774	D41405		NIBASC	'MAP'
632	0077A	D4		CON(2)	#4D
633	0077C	B	MARGIt	CON(1)	#B
634	0077D	D4142574		NIBASC	'MARGIN'
		94E4			
635	00789	D1		CON(2)	#1D
636	0078B	7	MAXDt	CON(1)	#7
637	0078C	D4148544		NIBASC	'MAXD'
638	00794	93		CON(2)	#39
639	00796	7	MAXMt	CON(1)	#7
640	00797	D41485D4		NIBASC	'MAXM'
641	0079F	A3		CON(2)	#3A
642	007A1	5	MDYt	CON(1)	#5
643	007A2	D44495		NIBASC	'MDY'
644	007A8	83		CON(2)	#38
645	007AA	7	MEMDt	CON(1)	#7
646	007AB	D454D444		NIBASC	'MEMD'
647	007B3	B3		CON(2)	#3B
648	007B5	7	MEMMt	CON(1)	#7
649	007B6	D454D4D4		NIBASC	'MEMM'
650	007BE	C3		CON(2)	#3C
651	007C0	7	MENUt	CON(1)	#7
652	007C1	D454E455		NIBASC	'MENU'
653	007C9	12		CON(2)	#21
654	007CB	7	MODEt	CON(1)	#7
655	007CC	D4F44454		NIBASC	'MODE'
656	007D4	D2		CON(2)	#2D
657	007D6		TxT1N		
658	007D6	D	NEXTOt	CON(1)	#D
659	007D7	E4548545		NIBASC	'NEXTOP\$'
		F40542			
660	007E5	E1		CON(2)	#1E
661	007E7	9	NLOOPt	CON(1)	#9
662	007E8	E4C4F4F4		NIBASC	'NLOOP'
		05			
663	007F2	E3		CON(2)	#3E
664	007F4	9	NPRIMt	CON(1)	#9
665	007F5	E4052594		NIBASC	'NPRIM'
		D4			
666	007FF	35		CON(2)	#53
667	00801		TxT1O		
668	00801	D	OPCODt	CON(1)	#D
669	00802	F40534F4		NIBASC	'OPCODE\$'

```
445442
670 00810 F1          CON(2) #1F
671 00812          TxT1P
672 00812 D          PAGELt  CON(1) #D
673 00813 05147454  NIBASC 'PAGELEN'
      C454E4
674 00821 F2          CON(2) #2F
675 00823 9          PAINTt  CON(1) #9
676 00824 051494E4  NIBASC 'PAINT'
      45
677 0082E F0          CON(2) #0F
678 00830 D          PPOLLt  CON(1) #D
679 00831 05142505  NIBASC 'PARPOLL'
      F4C4C4
680 0083F F3          CON(2) #3F
681 00841 B          PBLISt  CON(1) #B
682 00842 0524C494  NIBASC 'PBLIST'
      3545
683 0084E B5          CON(2) #5B
684 00850 5          PCRt    CON(1) #5
685 00851 053425    NIBASC 'PCR'
686 00857 92        CON(2) #29
687 00859 7          PDIRt  CON(1) #7
688 0085A 05449425  NIBASC 'PDIR'
689 00862 F6          CON(2) #6F
690 00864 7          PERFt  CON(1) 7
691 00865 05542564  NIBASC 'PERF'
692 0086D E2        CON(2) #2E
693          *** rev X: new name for ARR
694 0086F 7          PERMt  CON(1) 7
695 00870 055425D4  NIBASC 'PERM'
696 00878 41        CON(2) #14
697          ***
698 0087A 5          PFFt   CON(1) #5
699 0087B 056464    NIBASC 'PFF'
700 00881 B2        CON(2) #2B
701          *** rev X: renamed to GCD
702          *PGCDt  CON(1) 7
703          *        NIBASC 'PGCD'
704          *        CON(2) #54
705          ***
706 00883 5          PHIt   CON(1) #5
707 00884 058494    NIBASC 'PHI'
708 0088A 55        CON(2) #55
709 0088C 5          PLFt   CON(1) #5
710 0088D 05C464    NIBASC 'PLF'
711 00893 C2        CON(2) #2C
712 00895 7          POSIt  CON(1) #7
713 00896 05F43594  NIBASC 'POSI'
714 0089E 95        CON(2) #59
715          *** rev X: renamed to LCM
716          *PPCMt  CON(1) 7
717          *        NIBASC 'PPCM'
718          *        CON(2) #56
719          ***
720 008A0 7          PRIMt  CON(1) #7
721 008A1 052594D4  NIBASC 'PRIM'
```

722	008A9	75		CON(2)	#57
723	008AB		TxT1R		
724	008AB	7	REDt	CON(1)	#7
725	008AC	25544442		NIBASC	'RED\$'
726	008B4	50		CON(2)	#05
727	008B6	D	REDUct	CON(1)	#D
728	008B7	25544455 345442		NIBASC	'REDUCE\$'
729	008C5	52		CON(2)	#25
730	008C7	F	RENUMt	CON(1)	#F
731	008C8	2554E455 D42554D4		NIBASC	'RENUMREM'
732	008D8	C5		CON(2)	#5C
733	008DA	B	REPEAt	CON(1)	#B
734	008DB	25540554 1445		NIBASC	'REPEAT'
735	008E7	44		CON(2)	#44
736	008E9	F	REPLAt	CON(1)	#F
737	008EA	255405C4 14345442		NIBASC	'REPLACE\$'
738	008FA	60		CON(2)	#06
739	008FC	9	ROMANt	CON(1)	#9
740	008FD	25F4D414 E4		NIBASC	'ROMAN'
741	00907	C6		CON(2)	#6C
742	00909	9	RRECT	CON(1)	#9
743	0090A	25255434 42		NIBASC	'RREC\$'
744	00914	E5		CON(2)	#5E
745	00916		TxT1S		
746	00916	B	SELEct	CON(1)	#B
747	00917	3554C454 3445		NIBASC	'SELECT'
748	00923	16		CON(2)	#61
749	00925	B	SHRINT	CON(1)	#B
750	00926	35842594 E4B4		NIBASC	'SHRINK'
751	00932	15		CON(2)	#51
752	00934	9	SLEEpt	CON(1)	#9
753	00935	35C45454 05		NIBASC	'SLEEP'
754	0093F	04		CON(2)	#40
755	00941	B	SPACEt	CON(1)	#B
756	00942	35051434 5442		NIBASC	'SPACE\$'
757	0094E	62		CON(2)	#26
758	00950	5	SRQt	CON(1)	#5
759	00951	352515		NIBASC	'SRQ'
760	00957	14		CON(2)	#41
761	00959	9	STACKt	CON(1)	#9
762	0095A	35451434 B4		NIBASC	'STACK'
763	00964	B1		CON(2)	#1B
764	00966	F	STARTt	CON(1)	#F
765	00967	35451425 45550542		NIBASC	'STARTUP\$'
766	00977	21		CON(2)	#12

```
767 00979 D SYSEDt CON(1) #D
768 0097A 35953554 NIBASC 'SYSEdit'
      449445
769 00988 02 CON(2) #20
770 0098A TxT1T
771 0098A 9 TOKENt CON(1) #9
772 0098B 45F4B454 NIBASC 'TOKEN'
      E4
773 00995 94 CON(2) #49
774 00997 TxT1U
775 00997 F UNDERt CON(1) #F
776 00998 55E44454 NIBASC 'UNDERLIN'
      25C494E4
777 009A8 03 CON(2) #30
778 009AA 9 UNTILt CON(1) #9
779 009AB 55E44594 NIBASC 'UNTIL'
      C4
780 009B5 54 CON(2) #45
781 009B7 TxT1V
782 009B7 D VARSWt CON(1) #D
783 009B8 65142535 NIBASC 'VARSWAP'
      751405
784 009C6 74 CON(2) #47
785 009C8 TxT1W
786 009C8 9 WHILEt CON(1) #9
787 009C9 758494C4 NIBASC 'WHILE'
      54
788 009D3 34 CON(2) #43
789 009D5 7 WRAPt CON(1) #7
790 009D6 75251405 NIBASC 'WRAP'
791 009DE 13 CON(2) #31
792 009E0 7 WREct CON(1) #7
793 009E1 75255434 NIBASC 'WREC'
794 009E9 F5 CON(2) #5F
795 009EB F OBSOLt CON(1) #F
796 009EC F62637F6 NIBASC 'obsolete'
      C6564756
797 009FC 01 CON(2) #10
798 009FE TxT1J
799 009FE TxT1Q
800 009FE TxT1X
801 009FE TxT1Y
802 009FE TxT1Z
803 009FE 1FF TxTEn1 NIBHEX 1FF
804 *
805
806 * *****
807 * Obsolete keyword entry
808 * *****
809 00A01 00000 REL(5) =OBSOLd
810 00A06 00000 CON(5) 0
811 00A0B 33011E OBSOLe LC(4) #E110
812 00A11 8D00000 GOVLNG =BSERR
813
814 00A18 END
```


ADBUFe	Ext		-	85	
ADBUFt	Abs	1148	#0000047C	- 426	84
ADCREe	Ext		-	388	
ADCREt	Abs	1163	#0000048B	- 429	387
ADDELe	Ext		-	391	
ADDELt	Abs	1182	#0000049E	- 432	390
ADFINE	Ext		-	394	
ADFINt	Abs	1201	#000004B1	- 435	393
ADGETe	Ext		-	397	
ADGETt	Abs	1216	#000004C0	- 438	396
ADPUTe	Ext		-	400	
ADPUTt	Abs	1229	#000004CD	- 441	399
ADSIze	Ext		-	403	
ADSIzT	Abs	1242	#000004DA	- 444	402
ARRe	Ext		-	142	
ASCe	Ext		-	88	
ASCT	Abs	1257	#000004E9	- 452	87
ATHe	Ext		-	91	
ATHT	Abs	1268	#000004F4	- 455	90
ATTNe	Ext		-	106	
ATTNt	Abs	1279	#000004FF	- 458	105
BELLe	Ext		-	199	
BELLt	Abs	1290	#0000050A	- 462	198
BOLDe	Ext		-	202	
BOLDt	Abs	1301	#00000515	- 465	201
BSERR	Ext		-	812	
CASEe	Ext		-	376	
CASeT	Abs	1312	#00000520	- 469	375
CENTEe	Ext		-	184	
CENTEt	Abs	1323	#0000052B	- 472	183
CESURe	Ext		-	187	
CESURt	Abs	1340	#0000053C	- 475	186
COMBe	Ext		-	145	
COMBT	Abs	1355	#0000054B	- 478	144
CONTRe	Ext		-	118	
CONTRt	Abs	1366	#00000556	- 481	117
DATEAe	Ext		-	235	
DATEAt	Abs	1385	#00000569	- 485	234
DATESe	Ext		-	232	
DATESt	Abs	1402	#0000057A	- 488	231
DBLISe	Ext		-	352	
DBLISt	Abs	1421	#0000058D	- 491	351
DDAYSe	Ext		-	238	
DDAYSt	Abs	1436	#0000059C	- 494	237
DDIRe	Ext		-	412	
DDIRt	Abs	1449	#000005A9	- 497	411
DMYe	Ext		-	241	
DMYt	Abs	1460	#000005B4	- 500	240
DOWe	Ext		-	247	
DOWse	Ext		-	244	
DOWst	Abs	1469	#000005BD	- 503	243
DOWt	Abs	1480	#000005C8	- 506	246
EDITe	Ext		-	160	
EDITt	Abs	1489	#000005D1	- 510	159
ELSE2e	Ext		-	382	
ELSEt	Abs	1500	#000005DC	- 513	381
END2e	Ext		-	280	

ENDUPe	Ext		-	133	
ENDUPt	Abs	1526	#000005F6	-	519 132
ENDUSe	Ext		-	130	
ENDUSt	Abs	1511	#000005E7	-	516 129
ENDt	Abs	1539	#00000603	-	522 279
ENTRYe	Ext		-	298	
ENTRYt	Abs	1548	#0000060C	-	525 297
ESCe	Ext		-	208	
ESct	Abs	1563	#0000061B	-	528 207
EXECUe	Ext		-	139	
EXECUt	Abs	1574	#00000626	-	531 138
EXITe	Ext		-	265	
EXITt	Abs	1591	#00000637	-	534 264
FILESe	Ext		-	103	
FILESt	Abs	1613	#0000064D	-	541 102
FINDe	Ext		-	307	
FINDt	Abs	1632	#00000660	-	544 306
FINPUe	Ext		-	361	
FINPUt	Abs	1643	#0000066B	-	547 360
FKEYe	Ext		-	115	
FKEYt	Abs	1602	#00000642	-	538 114
FORMAe	Ext		-	190	
FORMAt	Abs	1658	#0000067A	-	550 189
FPRIME	Ext		-	328	
FPRIMt	Abs	1675	#0000068B	-	553 327
FRACe	Ext		-	346	
FRAct	Abs	1688	#00000698	-	556 345
FiLeNd	Ext		-	37	
GCDt	Abs	1701	#000006A5	-	561 333
GLINEe	Ext		-	316	
GLINet	Abs	1710	#000006AE	-	565 315
GPSETe	Ext		-	319	
GPSETt	Abs	1723	#000006BB	-	568 318
HMSADe	Ext		-	148	
HMSADt	Abs	1736	#000006C8	-	572 147
HMSSUe	Ext		-	151	
HMSSUt	Abs	1751	#000006D7	-	575 150
HMSe	Ext		-	154	
HMSt	Abs	1766	#000006E6	-	578 153
HRe	Ext		-	157	
HRT	Abs	1775	#000006EF	-	581 156
HTAe	Ext		-	94	
HTAt	Abs	1782	#000006F6	-	584 93
IF2e	Ext		-	379	
IFt	Abs	1793	#00000701	-	588 378
INDENT	Abs	1800	#00000708	-	591 408
INVERe	Ext		-	121	
INVERT	Abs	1815	#00000717	-	594 120
=JPCLEX	Abs	8	#00000008	-	31
KAe	Ext		-	385	
KAt	Abs	1832	#00000728	-	598 384
LCMt	Abs	1839	#0000072F	-	603 339
LEAVEe	Ext		-	292	
LEAVEt	Abs	1848	#00000738	-	607 291
LOOPe	Ext		-	370	
LOOPt	Abs	1872	#00000750	-	618 369
LXOFFe	Ext		-	109	

LXOFFt	Abs	1883 #0000075B	-	622	108		
LXONe	Ext		-	112			
LXONt	Abs	1861 #00000745	-	614	111		
MAPe	Ext		-	313			
MAPse	Ext		-	310			
MAPst	Abs	1896 #00000768	-	627	309		
MAPt	Abs	1907 #00000773	-	630	312		
MARGIe	Ext		-	169			
MARGIt	Abs	1916 #0000077C	-	633	168		
MAXDe	Ext		-	253			
MAXDt	Abs	1931 #0000078B	-	636	252		
MAXMe	Ext		-	256			
MAXMt	Abs	1942 #00000796	-	639	255		
MDYe	Ext		-	250			
MDYt	Abs	1953 #000007A1	-	642	249		
MEMDe	Ext		-	259			
MEMDt	Abs	1962 #000007AA	-	645	258		
MEMMe	Ext		-	262			
MEMMt	Abs	1973 #000007B5	-	648	261		
MENUe	Ext		-	181			
MENUt	Abs	1984 #000007C0	-	651	180		
MODEe	Ext		-	217			
MODEt	Abs	1995 #000007CB	-	654	216		
MSGTB1	Ext		-	76			
NEXTOe	Ext		-	172			
NEXTOt	Abs	2006 #000007D6	-	658	171		
NLOOPe	Ext		-	268			
NLOOPt	Abs	2023 #000007E7	-	661	267		
NPRIMe	Ext		-	331			
NPRIMt	Abs	2036 #000007F4	-	664	330		
OBSOLd	Ext		-	809			
OBSOLe	Abs	2571 #00000A0B	-	811	124	166	304
OBSOLt	Abs	2539 #000009EB	-	795	123	165	303
OPCODE	Ext		-	175			
OPCODt	Abs	2049 #00000801	-	668	174		
PAGELe	Ext		-	223			
PAGELt	Abs	2066 #00000812	-	672	222		
PAINTe	Ext		-	127			
PAINTt	Abs	2083 #00000823	-	675	126		
PBLISe	Ext		-	355			
PBLISt	Abs	2113 #00000841	-	681	354		
PCRe	Ext		-	205			
PCRt	Abs	2128 #00000850	-	684	204		
PDIRe	Ext		-	415			
PDIRt	Abs	2137 #00000859	-	687	414		
PERFe	Ext		-	220			
PERFt	Abs	2148 #00000864	-	690	219		
PERMt	Abs	2159 #0000086F	-	694	141		
PFFe	Ext		-	211			
PFFt	Abs	2170 #0000087A	-	698	210		
PGCDe	Ext		-	334			
PHIe	Ext		-	337			
PHIt	Abs	2179 #00000883	-	706	336		
PKLEX	Ext		-	46			
PLFe	Ext		-	214			
PLFt	Abs	2188 #0000088C	-	709	213		
POLHND	Ext		-	77			

POSie	Ext		-	349	
POSit	Abs	2197 #00000895	-	712	348
PPCMe	Ext		-	340	
PPOLLe	Ext		-	271	
PPOLLt	Abs	2096 #00000830	-	678	270
PRIME	Ext		-	343	
PRIMt	Abs	2208 #000008A0	-	720	342
REDUCe	Ext		-	193	
REDUCt	Abs	2230 #000008B6	-	727	192
REDe	Ext		-	97	
REDt	Abs	2219 #000008AB	-	724	96
RENUMe	Ext		-	358	
RENUMt	Abs	2247 #000008C7	-	730	357
REPEAe	Ext		-	286	
REPEAT	Abs	2266 #000008DA	-	733	285
REPLAe	Ext		-	100	322
REPLAt	Abs	2281 #000008E9	-	736	99 321
ROMANe	Ext		-	406	
ROMANt	Abs	2300 #000008FC	-	739	405
RRECe	Ext		-	364	
RRECT	Abs	2313 #00000909	-	742	363
SELECe	Ext		-	373	
SELEct	Abs	2326 #00000916	-	746	372
SHRINe	Ext		-	325	
SHRINt	Abs	2341 #00000925	-	749	324
SLEEPe	Ext		-	274	
SLEEpt	Abs	2356 #00000934	-	752	273
SPACEe	Ext		-	196	
SPACEt	Abs	2369 #00000941	-	755	195
SRQe	Ext		-	277	
SRQt	Abs	2384 #00000950	-	758	276
STACKe	Ext		-	163	
STACKt	Abs	2393 #00000959	-	761	162
STARTe	Ext		-	136	
STARTt	Abs	2406 #00000966	-	764	135
SYSEDe	Ext		-	178	
SYSEDt	Abs	2425 #00000979	-	767	177
TOKENe	Ext		-	301	
TOKENt	Abs	2442 #0000098A	-	771	300
TxT1A	Abs	1148 #0000047C	-	425	48
TxT1B	Abs	1290 #0000050A	-	461	49
TxT1C	Abs	1312 #00000520	-	468	50
TxT1D	Abs	1385 #00000569	-	484	51
TxT1E	Abs	1489 #000005D1	-	509	52
TxT1F	Abs	1602 #00000642	-	537	53
TxT1G	Abs	1701 #000006A5	-	559	54
TxT1H	Abs	1736 #000006C8	-	571	55
TxT1I	Abs	1793 #00000701	-	587	56
TxT1J	Abs	2558 #000009FE	-	798	57
TxT1K	Abs	1832 #00000728	-	597	58
TxT1L	Abs	1839 #0000072F	-	601	59
TxT1M	Abs	1896 #00000768	-	626	60
TxT1N	Abs	2006 #000007D6	-	657	61
TxT1O	Abs	2049 #00000801	-	667	62
TxT1P	Abs	2066 #00000812	-	671	63
TxT1Q	Abs	2558 #000009FE	-	799	64
TxT1R	Abs	2219 #000008AB	-	723	65

TxT1S	Abs	2326 #00000916 -	745	66						
TxT1T	Abs	2442 #0000098A -	770	67						
TxT1U	Abs	2455 #00000997 -	774	68						
TxT1V	Abs	2487 #000009B7 -	781	69						
TxT1W	Abs	2504 #000009C8 -	785	70						
TxT1X	Abs	2558 #000009FE -	800	71						
TxT1Y	Abs	2558 #000009FE -	801	72						
TxT1Z	Abs	2558 #000009FE -	802	73						
TxTEn1	Abs	2558 #000009FE -	803							
TxTSt1	Abs	1148 #0000047C -	424	48	49	50	51	52	53	
			54	55	56	57	58	59	60	
			61	62	63	64	65	66	67	
			68	69	70	71	72	73	75	
			84	87	90	93	96	99	102	
			105	108	111	114	117	120	123	
			126	129	132	135	138	141	144	
			147	150	153	156	159	162	165	
			168	171	174	177	180	183	186	
			189	192	195	198	201	204	207	
			210	213	216	219	222	225	228	
			231	234	237	240	243	246	249	
			252	255	258	261	264	267	270	
			273	276	279	282	285	288	291	
			294	297	300	303	306	309	312	
			315	318	321	324	327	330	333	
			336	339	342	345	348	351	354	
			357	360	363	366	369	372	375	
			378	381	384	387	390	393	396	
			399	402	405	408	411	414		
UNDERe	Ext	-	226							
UNDERT	Abs	2455 #00000997 -	775	225						
UNTILe	Ext	-	289							
UNTILt	Abs	2474 #000009AA -	778	288						
VARSWe	Ext	-	295							
VARSWt	Abs	2487 #000009B7 -	782	294						
WHILEe	Ext	-	283							
WHILEt	Abs	2504 #000009C8 -	786	282						
WRAPe	Ext	-	229							
WRAPt	Abs	2517 #000009D5 -	789	228						
WRECe	Ext	-	367							
WRECT	Abs	2528 #000009E0 -	792	366						
fLEX	Ext	-	32							

Input Parameters

Source file name is tables.a
Listing file name is tables.l
Object file name is tables.o
Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE MSG
2
3          * *****
4          * Message module
5          *
6          * Rebuilt from JPC rev D binary, J-F G. 2005
7          *
8          * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
9          * *****
10
11 00A18          ABS      #0A18
12
13          * *****
14          * Message table
15          * msg table structure:
16          * - entry len (1st entry must be multiple of
17          *   16 nibble long, e.g. 5-char msg)
18          * - msg #
19          * - tokens:
20          *   - 0-A: literal string, is the string length -1
21          *   - B: prefix for literal string>A up to F
22          *   - C: end of msg
23          *   - D: current LEX msg number
24          *   - E: mainframe msg number
25          *   - F: # output (e.g. mainframe msg "trk ### of ###" )
26          * *****
27
28 00A18 00      =MSGTB1 CON(2) #00      msg #0 ...
29 00A1A 11      CON(2) #11          ... to #17
30
31          * msg 6: "#Dims"
32 00A1C 01      CON(2) #10
33 00A1E 60      CON(2) #06
34 00A20 4        CON(1) #4
35 00A21 324496D6 NIBASC '#Dims'
36          37
37 00A2B C        CON(1) #C
38
39          * msg 0: "JPC "
40 00A2C E0      CON(2) #0E
41 00A2E 00      CON(2) #00
42 00A30 3        CON(1) #3
43 00A31 A4053402 NIBASC 'JPC '
44 00A39 C        CON(1) #C
45
46          * msg 1: "Driver Lex File"
47 00A3A 52      CON(2) #25
48 00A3C 10      CON(2) #01
49 00A3E B        CON(1) #B
50 00A3F E        CON(1) #E
51 00A40 44279667 NIBASC 'Driver L'
52          562702C4
53 00A50 56870264 NIBASC 'ex File'
54          96C656
55 00A5E C        CON(1) #C
56
57          * msg 2: "Not Found"
```

```
55 00A5F 81          CON(2) #18
56 00A61 20          CON(2) #02
57 00A63 8           CON(1) #8
58 00A64 E4F64702    NIBASC 'Not Foun'
    64F657E6
59 00A74 46          NIBASC 'd'
60 00A76 C           CON(1) #C
61
62                * msg 3: "Structure Mismatch"
63 00A77 C2          CON(2) #2C
64 00A79 30          CON(2) #03
65 00A7B B           CON(1) #B
66 00A7C F           CON(1) #F
67 00A7D 35472757    NIBASC 'Structur'
    36475727
68 00A8D 5602D496    NIBASC 'e Mismat'
    37D61647
69 00A9D 1           CON(1) #1
70 00A9E 3686        NIBASC 'ch'
71 00AA2 C           CON(1) #C
72
73                * msg 4: "Invalid Prompt"
74 00AA3 51          CON(2) #15
75 00AA5 40          CON(2) #04
76 00AA7 E           CON(1) #E
77 00AA8 CE          CON(2) #EC
78 00AAA 5           CON(1) #5
79 00AAB 0527F6D6    NIBASC 'Prompt'
    0747
80 00AB7 C           CON(1) #C
81
82                * msg 5: "Invalid Format"
83 00AB8 51          CON(2) #15
84 00ABA 50          CON(2) #05
85 00ABC E           CON(1) #E
86 00ABD CE          CON(2) #EC
87 00ABF 5           CON(1) #5
88 00AC0 64F627D6    NIBASC 'Format'
    1647
89 00ACC C           CON(1) #C
90
91                * msg 7: "Var Not Found"
92 00ACD F0          CON(2) #0F
93 00ACF 70          CON(2) #07
94 00AD1 2           CON(1) #2
95 00AD2 651627      NIBASC 'Var'
96 00AD8 E           CON(1) #E
97 00AD9 8E          CON(2) #E8
98 00ADB C           CON(1) #C
99
100               * msg 8 to 14: "Sunday" to "Saturday"
101 00ADC F0          CON(2) #0F
102 00ADE 80          CON(2) #08
103 00AE0 2           CON(1) #2
104 00AE1 3557E6      NIBASC 'Sun'
105 00AE7 D           CON(1) #D   uses "day" block, saves 3 nibbles each time
106 00AE8 21          CON(2) #12
```



```
107 00AEA C          CON(1) #C
108
109 00AEB F0         CON(2) #0F
110 00AED 90         CON(2) #09
111 00AEF 2          CON(1) #2
112 00AF0 D4F6E6     NIBASC 'Mon'
113 00AF6 D          CON(1) #D
114 00AF7 21         CON(2) #12
115 00AF9 C          CON(1) #C
116
117 00AFA 11         CON(2) #11
118 00AFC A0         CON(2) #0A
119 00AFE 3          CON(1) #3
120 00AFF 45575637   NIBASC 'Tues'
121 00B07 D          CON(1) #D
122 00B08 21         CON(2) #12
123 00B0A C          CON(1) #C
124
125 00B0B 51         CON(2) #15
126 00B0D B0         CON(2) #0B
127 00B0F 5          CON(1) #5
128 00B10 755646E6   NIBASC 'Wednes'
    5637
129 00B1C D          CON(1) #D
130 00B1D 21         CON(2) #12
131 00B1F C          CON(1) #C
132
133 00B20 31         CON(2) #13
134 00B22 C0         CON(2) #0C
135 00B24 4          CON(1) #4
136 00B25 45865727   NIBASC 'Thurs'
    37
137 00B2F D          CON(1) #D
138 00B30 21         CON(2) #12
139 00B32 C          CON(1) #C
140
141 00B33 F0         CON(2) #0F
142 00B35 D0         CON(2) #0D
143 00B37 2          CON(1) #2
144 00B38 642796     NIBASC 'Fri'
145 00B3E D          CON(1) #D
146 00B3F 21         CON(2) #12
147 00B41 C          CON(1) #C
148
149 00B42 31         CON(2) #13
150 00B44 E0         CON(2) #0E
151 00B46 4          CON(1) #4
152 00B47 35164757   NIBASC 'Satur'
    27
153 00B51 D          CON(1) #D
154 00B52 21         CON(2) #12
155 00B54 C          CON(1) #C
156
157                * msg 15: "Function Interrupted"
158 00B55 03         CON(2) #30
159 00B57 F0         CON(2) #0F
160 00B59 B          CON(1) #B
```

```
161 00B5A F          CON(1) #F
162 00B5B 6457E636   NIBASC 'Function'
      4796F6E6
163 00B6B 0294E647   NIBASC ' Interru'
      56272757
164 00B7B 3          CON(1) #3
165 00B7C 07475646   NIBASC 'pted'
166 00B84 C          CON(1) #C
167
168                * msg 16: "Removed Keyword"
169 00B85 52         CON(2) #25
170 00B87 01         CON(2) #10
171 00B89 B          CON(1) #B
172 00B8A E          CON(1) #E
173 00B8B 2556D6F6   NIBASC 'Removed '
      67564602
174 00B9B B4569777   NIBASC 'Keyword'
      F62746
175 00BA9 C          CON(1) #C
176
177                * msg 17: "(c) 1986, 1987, 1988 PPC-Paris"
178 00BAA 54         CON(2) #45
179 00BAC 11         CON(2) #11
180 00BAE B          CON(1) #B
181 00BAF F          CON(1) #F
182 00BB0 82369202   NIBASC '(c) 1986'
      13938363
183 00BC0 C2021393   NIBASC ', 1987, '
      8373C202
184 00BD0 B          CON(1) #B
185 00BD1 D          CON(1) #D
186 00BD2 13938383   NIBASC '1988 PPC'
      02050534
187 00BE2 D2051627   NIBASC '-Paris'
      9637
188 00BEE C          CON(1) #C
189
190                * "day" building block (uses 12 nibbles, to save 21 nib,
191                *                               gain is 9 nibbles...)
192 00BEF C0         CON(2) #0C
193 00BF1 21         CON(2) #12
194 00BF3 2          CON(1) #2
195 00BF4 461697     NIBASC 'day'
196 00BFA C          CON(1) #C
197
198 00BFB FF         NIBHEX FF
199
200 00BFD           END
```

=MSGTB1 Abs 2584 #00000A18 - 28

Input Parameters

Source file name is msg.a

Listing file name is msg.l

Object file name is msg.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE DECOMP
2
3          * *****
4          *  Decompile module
5          *
6          *  Provides the decompile routines for all keywords
7          *
8          *  Rebuilt from JPC rev D binary and from various sources,
9          *  J-F G. 2005
10         *
11         *  (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
12         *  *****
13
14 00BFD          ABS      #0BFD
15
16          * HPIL ROM calls:
17  PACKd EQU      #7B4A
18
19          *** decompile entry for MAP
20 00BFD 3132     =MAPd  LCASC  '#'
21 00C01 962          ?A=C  B          #<channel>?
22 00C04 90          GOYES  dropdc  yes, DROPDC can handle it
23 00C06 8F00000    GOSBVL =FILDC* decompile filespec
24 00C0D 8D00000    dropdc  GOVLNG =DROPDC decompile expr list
25
26
27          *** decompile entry for END
28 00C14 14B       =END2d  A=DAT1 B
29 00C17 7C41          GOSUB  eolxck
30 00C1B 474          GOC     ENDWd      trouve tEOL (ou equivalent)
31 00C1E D6          C=A     A          C(0) := quartet lu
32 00C20 170          D1=D1+ 1
33 00C23 8F00000    GOSBVL =TBLJMC
34 00C2A 510          REL(3) ENDLd
35 00C2D 220          REL(3) ENDSd
36 00C30 300          REL(3) ENDI d
37
38 00C33 339464     ENDI d  LCASC  'FI'
39 00C39 23          P=      2*2-1
40 00C3B 6530          GOTO   ENDD10
41 00C3F 37C4F4F4   ENLD d  LCASC  'POOL'
42          05
43 00C49 27          P=      2*4-1
44 00C4B 6520          GOTO   ENDD10
45 00C4F 3B3554C4   ENDS d  LCASC  'TCELES'
46          543445
47 00C5D 2B          P=      2*6-1
48 00C5F 6110          GOTO   ENDD10
49 00C63 39758494   ENDW d  LCASC  'ELIHW'
50          C454
51 00C6F 29          P=      2*5-1
52 00C71 7D61       ENDD10 GOSUB  outnbc
53 00C75 14B        A=DAT1 B
54
55         *
56         *  ATTENTION : le code continue !!!
57         *
```

```
55          *** decompile entry for
56          * SLEEP, BELL, PCR, PFF, LEAVE, REPEAT, LOOP, ELSE
57          * (no argument)
58 00C78      =NOARGd
59 00C78 8D00000 outela GOVLNG =OUTELA
60
61
62          *** decompile entry for UNDERLINE
63 00C7F      =UNDERd
64 00C7F 335402          LCASC ' E'
65 00C85 181          D0=D0- 2          Retour sur l'espace
66 00C88 8F00000          GOSBVL =OUT2TC  Affichage de "E"
67 00C8F 14B          A=DAT1 B          Token suivant (ON/OFF)
68          *
69          * le code se poursuit dans ONOFFd
70          *
71
72          *** decompile entry for ATTN, BOLD, PERF, WRAP, ROMAN
73          * arg on/off
74 00C92 8D00000 =ONOFFd GOVLNG =TRACDC  DÅcompilation du token
75
76
77          *** decompile entry for
78          * DMY, MDY, INVERSE, ENDUP, EXECUTE, FKEY, GPSET, GLINE,
79          * MARGIN, PLF, MODE, PAGELEN, STACK, SELECT, WHILE, UNTIL
80          * VARSWAP, SYSEdit
81 00C99 8D00000 =STDd GOVLNG =DROPDC
82
83          *** decompile entry for EXIT
84 00CA0 8F00000 =EXITd GOSBVL =VARDC
85 00CA7 8D00000          GOVLNG =OUTEL1
86
87          *** decompile for LEX ON
88 00CAE 8F00000 =LXOND GOSBVL =FILDC*
89 00CB5 3502F4E4          LCASC 'NO '
90 00CBD 25          P= 5
91 00CBF 61BF          GOTO ENDD10
92
93          *** decompile for LEX OFF
94 00CC3 8F00000 =LXOFFd GOSBVL =FILDC*
95 00CCA 3702F464          LCASC 'FFO '
96          64
96 00CD4 27          P= 7
97 00CD6 6A9F          GOTO ENDD10
98
99
100          *** decompile entry for RENUMREM, KA, SHRINK
101          * also used for LXON, LXOFF (rev X)
102 00CDA      =LXDRVd
103 00CDA      =RENUMd
104 00CDA      =SHRIND
105 00CDA 8D00000 =KAd GOVLNG =LISTDC
106
107
108          *** decompile entry for WREC
109 00CE1 8F00000 =WRECd GOSBVL =EXPRDC 1er paramÉtre
110 00CE8 31C2          LCASC ','
```

```
111 00CEC 8F00000          GOSBVL =OUTBY+
112 00CF3 8F00000          GOSBVL =EXPRDC  2Éme paramÉtre  GOSUB exprdc
113 00CFA 31C2             LCASC  ', '
114 00CFE 8F00000          GOSBVL =OUTBY+
115 00D05 8E0000          GOSUBL =JUMPER
116 00D0B A4B70           CON(5) PACKd    3Éme paramÉtre: HPIL
117 00D10 01              RTN
118
119
120 00D12 31C2            virgud  LCASC  ', '
121 00D16 8D00000          GOVLNG =OUTBY+
122
123                        *** decompile entry for FINPUT
124 00D1D 7920            =FINPud  GOSUB  exprdc
125 00D21 7DEF            GOSUB  virgud
126 00D25 7120            GOSUB  exprdc
127 00D29 75EF            GOSUB  virgud
128 00D2D 170             D1=D1+ 1
129 00D30 7610            GOSUB  exprdc
130 00D34 8F00000          GOSBVL =EOLXC*
131 00D3B 73DF            GOSUB  virgud
132 00D3F 7700            GOSUB  exprdc
133 00D43 8D00000  outell  GOVLNG =OUTEL1
134
135 00D4A 8D00000  exprdc  GOVLNG =EXPRDC
136
137
138                        *** decompile entry for IF
139 00D51 75FF            =IF2d   GOSUB  exprdc
140 00D55 39024584        LCASC  'NEHT '
141      54E4
141 00D61 29              P=      2*5-1
142 00D63 6D0F            GOTO   ENDD10
143
144 00D67 8D00000  eolxck  GOVLNG =EOLXCK
145
146                        *** decompile entry for CASE
147 00D6E 75FF            =CASEd  GOSUB  eolxck
148 00D72 521             GONC   CSd10
149 00D75 3754C435        LCASC  'ESLE '
150      54
150 00D7F 27              P=      4*2-1
151 00D81 6FEE            GOTO   ENDD10
152 00D85 1533            CSd10  A=DAT1 X
153 00D89 3100            LC(2)  =tRELOP
154 00D8D 966             ?A#C   B
155 00D90 31              GOYES  CSd20
156
157      *
157      * <relop> trouve. Il faut le sortir en ASCII
158      *
159 00D92 8F00000          GOSBVL =ARITH
160 00D99 7540            GOSUB  outnbc
161 00D9D 172             D1=D1+ 3
162 00DA0 522             GONC   CSd30    B.E.T.
163
164      *
164      * <relop> n'existe pas. Il faut decompiler une
165      * expression, puis voir ce qu'il y a apres...
```

```

166
167 00DA3 73AF      Csd20  GOSUB  exprdc
168 00DA7 3100      LC(2)  =tTO
169 00DAB 966       ?A#C   B
170 00DAE C1        GOYES  Csd40   c'est donc tCOMMA ou EOL
171
172                *
173                * TO : on s'attend donc maintenant a une exp.
174                *
174 00DB0 370245F4  LCASC  ' OT '
      02
175 00DBA 27        P=      4*2-1
176 00DBC 7220      GOSUB  outnbc
177
178 00DC0 171       D1=D1+ 2
179 00DC3 14B      Csd30  A=DAT1 B
180 00DC6 708F      GOSUB  exprdc
181 00DCA 8F00000  Csd40  GOSBVL =EOLXC*
182
183                *
184                * Si on est revenu, c'est qu'il y a une tCOMMA
185                *
185 00DD1 31C2      LCASC  ', '   on l'envoie...
186 00DD5 21        P=      2*1-1   ... dans le flot
187 00DD7 7700      GOSUB  outnbc ... de sortie
188 00DDB 171       D1=D1+ 2   on l'oublie...
189 00DDE 66AF      GOTO   Csd10 ... et on continue
190
191 00DE2 8D00000  outnbc  GOVLNG =OUTNBC
192 00DE9 8D00000  fildc*  GOVLNG =FILDC*
193 00DF0 8D00000  vardc   GOVLNG =VARDC
194
195                *** decompile entry for ADCREATE
196 00DF7 7EEF      =ADCREd GOSUB fildc*
197 00DFB 8F00000  ADcre   GOSBVL =EOLXC*
198 00E02 7C0F      GOSUB  virgud
199 00E06 704F      GOSUB  exprdc
200 00E0A 8D00000  GOVLNG =OUTEL1
201
202                *** decompile entry for ADDELETE
203 00E11 74DF      =ADDELd GOSUB fildc*
204 00E15 79FE      GOSUB  virgud
205 00E19 7D2F      GOSUB  exprdc
206 00E1D 6DDF      GOTO   ADcre
207
208                *** decompile entry for ADGET
209 00E21 74CF      =ADGETd GOSUB fildc*
210 00E25 79EE      GOSUB  virgud
211 00E29 73CF      GOSUB  vardc
212 00E2D 71EE      GOSUB  virgud
213 00E31 751F      GOSUB  exprdc
214 00E35 65CF      GOTO   ADcre
215
216                *** decompile entry for ADPUT
217 00E39 7CAF      =ADPUTd GOSUB fildc*
218 00E3D 71DE      GOSUB  virgud
219 00E41 7BAF      GOSUB  vardc
220 00E45 65BF      GOTO   ADcre
221

```



```
222          *** decompile entry for obsolete keywords
223 00E49 1C7      =OBSOLd D1=D1- 8
224 00E4C D2          C=0      A
225 00E4E 14F      C=DAT1 B
226 00E51 133      AD1EX
227 00E54 CA        A=A+C  A
228 00E56 131      D1=A
229 00E59 14B      A=DAT1 B
230 00E5C 8D00000   GOVLNG =OUTELA
231
232
233          *****
234          * DBLSTd, PBLSTd
235          *
236          * But : decompiler DBLIST, PBLIST, RENUMREM
237          * Historique :
238          * 86/05/.. : JPB      conception & codage
239          * 88/01/10 : PD & JT nouvelle syntaxe
240          *****
241
242          *** decompile entry for PBLIST, DBLIST
243 00E63 848      =PBLISd ST=0 8      Premiere fois : pas de ','
244          * Boucle de decompilation des elements
245          * A l'entree, D1 pointe sur le token a decompiler.
246          * Les alternatives sont :
247          * <file spec>
248          * [ , <line #> ]      (S8=1 si affichage ',')
249          * TO <file spec>
250          * INDENT <exp num>
251 00E66 8F00000   LSTD00 GOSBVL =EOLXC* No return if end of statem.
252 00E6D 8F00000   GOSBVL =FINDA
253 00E74 00          CON(2) =tCOMMA <line#>      (LSTDC+)
254 00E76 D10        REL(3) LSTD10
255 00E79 00          CON(2) =tTO TO <file>
256 00E7B 630        REL(3) LSTD20
257 00E7E 00          CON(2) =tXWORD INDENT <exp num>
258 00E80 450        REL(3) LSTD30
259 00E83 00          NIBHEX 00
260          * <file spec>
261 00E85 8F00000   GOSBVL =FILDC*
262 00E8C 858        LSTD05 ST=1 8
263 00E8F 66DF      GOTO LSTD00
264
265          * [ , ] <line#>
266 00E93 868        LSTD10 ?ST=0 8      Besoin de sortir ', ' ?
267 00E96 D0          GOYES LSTD12 Non
268 00E98 31C2        LCASC ', ' Affichage de ', '
269 00E9C 8F00000   GOSBVL =OUTBYT
270 00EA3 171        LSTD12 D1=D1+ 2      Passer tCOMMA
271 00EA6 8F00000   GOSBVL =LIN#DC Envoyer <line#>
272 00EAD 6EDF      GOTO LSTD05
273
274          * TO <file spec>
275 00EB1 7A40        LSTD20 GOSUB OUTSPC
276 00EB5 3545F402   LCASC ' OT'
277 00EBD 25          P= 3*2-1
278 00EBF 8F00000   GOSBVL =OUTNBC
```

```
279 00EC6 171          D1=D1+ 2
280 00EC9 8F00000      GOSBVL =FILDC*
281 00ED0 6BBF         GOTO   LSTD05
282
283                   * INDEnt <exp num>
284 00ED4 7720          LSTD30 GOSUB  OUTSPC
285 00ED8 3D94E444      LCASC   ' TNEDNI '
      54E44502
286 00EE8 2D           P=      7*2-1
287 00EEA 8F00000      GOSBVL =OUTNBC
288 00EF1 175           D1=D1+ 6
289 00EF4 8F00000      GOSBVL =EXPRDC
290 00EFB 609F         GOTO   LSTD05
291
292                   *****
293                   * OUTSPC
294                   *
295                   * But : afficher un espace si TO ou INDEnt ne
296                   * suivent pas directement D/PBLIST.
297                   * Entree :
298                   * - S8 = 0 si directemenbt derriere D/PBLIST
299                   * Sortie :
300                   * - un blanc dans le flot ASCII
301                   * Appelle : OUTBYT
302                   * Utilise : A(B), C(B), D0
303                   * Niveaux : 1
304                   * Historique :
305                   * 88/01/10: PD & JT conception & codage
306                   *****
307
308 00EFF 868           OUTSPC ?ST=0 8
309 00F02 00            RTNYES
310 00F04 3102          LCASC   ' '
311 00F08 8D00000      GOVLNG =OUTBYT
312
313
314                   *** decompile entry for DDIR, PDIR
315 00F0F 8F00000      =DDIRd GOSBVL =EOLXC*
316 00F16 313F          LCHEX   F3
317 00F1A 962           ?C=A   B
318 00F1D 02            GOYES  dird1
319 00F1F 318F          LCHEX   F8
320 00F23 962           ?C=A   B
321 00F26 F2            GOYES  dird2
322 00F28 8F00000      GOSBVL =FILDC*
323 00F2F 8F00000      GOSBVL =BLNKCK
324 00F36 8F00000      GOSBVL =EOLXC*
325 00F3D 859          dird1  ST=1   9
326 00F40 8F00000      GOSBVL =GTEXT1
327 00F47 8F00000      GOSBVL =FILDC*
328 00F4E 8D00000      GOVLNG =OUTEL1
329 00F55 859          dird2  ST=1   9
330 00F58 8F00000      GOSBVL =GTEXT1
331 00F5F 6FAF         GOTO   DDIRd
332
333 00F63              END
```


Input Parameters

Source file name is decomp.a
Listing file name is decomp.l
Object file name is decomp.o
Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE PARSE
2
3          * *****
4          * Parse module
5          *
6          * Provides the parse routines for all keywords
7          *
8          * Rebuilt from JPC rev D binary and from various sources,
9          *   J-F G. 2005
10         * Updated with changes from rev X (R. Rosenbaum)
11         *
12         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
13         * *****
14
15 00F63          ABS      #0F63
16
17         * HPIL ROM calls:
18 DVCSPP EQU     #7925
19
20         * *****
21         * parse entries
22         * note: due to the numerous identical
23         * labels to commun ROM entries, numeric
24         * incremental suffixes have been used:
25         * e.g. respt1, respt2, ... to RESPTR jumps
26         * *****
27
28         *** parse entry for MAP
29 00F63 8F00000 =MAPp  GOSBVL =#CK      #?
30 00F6A 401      GOC      pfsp      no, try filespec
31 00F6D 8F00000 GOSBVL =OUT1T+ output # token
32 00F74 7B50     GOSUB   numck     channel no. (num expr)
33 00F78 531      GONC     pstr1
34 00F7B 8F00000 pfsp    GOSBVL =FSPECp parse filespec
35 00F82 475      GOC      badf      invalid filespec
36 00F85 8F00000 GOSBVL =NTOKEN get next token
37 00F8C 8F00000 pstr1  GOSBVL =COMCK+ comma? -> output token
38 00F93 5D4      GONC     syntx     no, syntax error
39 00F96 8F00000 GOSBVL =STRNGP <str1>
40
41         * the following commas are not tokenized, so that
42         * one call of EXPEXC will evaluate all parameters left
43         * DROPDC will insert commas between them by default
44 00F9D 7B20     GOSUB   comck     comma?
45 00FA1 5F3      GONC     syntx     no, syntax error
46 00FA4 8F00000 GOSBVL =STRNGP <str2>
47 00FAB 7D10     GOSUB   comck     comma?
48 00FAF 551      GONC     resptr    no, done
49 00FB2 7D10     GOSUB   numck     <num1>
50 00FB6 7B00     GOSUB   resptr
51 00FBA 7E00     GOSUB   comck     yet another comma?
52 00FBE 560      GONC     resptr
53 00FC1 7E00     GOSUB   numck     <num2>
54 00FC5 8D00000 resptr GOVLNG =RESPTR parsing done
55
56         * local entry points for mainframe routines that are called
57         * often enough (at least 3 times) that this saves memory
58 00FCC 8D00000 comck  GOVLNG =COMCK
59 00FD3 8D00000 numck  GOVLNG =NUMCK
```

```
58          * parse errors
59 00FDA 8D00000 badf   GOVLNG =FSPECe "Invalid Filespec"
60 00FE1 8D00000 syntx  GOVLNG =SYNTXe "Syntax"
61
62
63          *** parse entry for VARSWAP
64 00FE8          =VARSWp
65 00FE8 7430          GOSUB  Swap+      Compile 1 var.
66 00FEC 848          ST=0      8
67 00FEF 550          GONC   vs1   o0FDF
68 00FF2 858          ST=1      8
69 00FF5 8F00000 vs1   GOSBVL =NTOKEN  Cherche le token suivant
70          *          la var. prÃcÃdente
71 00FFC 8F00000          GOSBVL =COMCK+  Est-ce 1 virgule?
72 01003 551          GONC   Syntxe  Non
73 01006 7610         GOSUB  Swap+      Oui: compile 1 2nd var.
74 0100A 590          GONC   vs2   o0FFE
75 0100D 868          ?ST=0    8
76 01010 90          GOYES  Syntxe
77 01012 03          Rtncc  RTNCC
78 01014 868          vs2    ?ST=0    8
79 01017 BF          GOYES  Rtncc
80 01019 8D00000 Syntxe GOVLNG =SYNTXe
81
82 01020 8F00000 Swap+ GOSBVL =EXPPAR
83 01027 41F          GOC    Syntxe
84 0102A 870          ?ST=1    0
85 0102D CE          GOYES  Syntxe
86 0102F 831          ?XM=0
87 01032 50          GOYES  respt1
88 01034 54E          GONC   Syntxe
89 01037 8F00000 respt1 GOSBVL =RESPTR
90 0103E 873          ?ST=1    3
91 01041 00          RTNYES
92 01043 01          RTN
93
94
95          STITLE  Misc module parse 1
```

```
96
97
98          *** parse entry for KA
99 01045 8F00000 =KAp  GOSBVL =EOLCK
100 0104C 4F5    GOC   respt2
101 0104F 7950   GOSUB respt2
102
103
104          *** parse entry for SHRINK
105          * also used for LXON and LXOFF (rev Ex)
106 01053        =LXDRVp
107 01053 8F00000 =SHRINp GOSBVL =FSPECp
108 0105A 500    RTNNC
109
110 0105D 8D00000 GOVLNG =FSPECe
111
112          *** parse entry for ENDUP, EXECUTE, FKEY, FIND, SYSEDT
113 01064 8D00000 =STRNGp GOVLNG =STRNGP
114
115
116          *** parse entry for INVERSE
117 0106B 8F00000 =INVERp GOSBVL =EOLCK
118 01072 493    GOC   respt2
119 01075 7330   GOSUB respt2
120 01079 302    LC(1) 2
121 0107C 6600   GOTO  LIp10
122
123
124          *** parse entry for GLINE
125 01080 305    =GLINEp LC(1) 5      5 paramÉtres È parser
126 01083 A0E    LIp10 C=C-1 P      sauvegarde du compteur dans
127 01086 10A    R2=C      R2
128 01089 8F00000 GOSBVL =NUMCK  parser une expression num.
129 01090 7810   GOSUB respt2
130 01094 11A    C=R2      restauration du compteur
131 01097 90A    ?C=0 P      sortie si = 0
132 0109A 21     GOYES respt2
133 0109C 8F00000 GOSBVL =COMCK  sinon, on cherche une ", "
134 010A3 5F0    GONC  mspare si non trouvÁe, erreur
135 010A6 11A    C=R2      restauration du compteur
136 010A9 49D    GOC   LIp10 B.E.T.
137 010AC 8D00000 respt2 GOVLNG =RESPTR
138 010B3 8D00000 mspare GOVLNG =MSPARe "missing parameter"
139
140
141          *** parse entry for SLEEP, DMY, MDY,
142          * BELL, PCR, PFF, LEAVE, REPEAT, LOOP
143 010BA 03     =NOARGp RTNCC
144
145
146          STITLE  Struc2 module parse
```



```
147
148      * *****
149      * parse routines for struc2 module
150      * *****
151
152      id      EQU    #E1
153
154      tEND2   EQU    66
155      tWHILE  EQU    67
156      tREPEAT EQU    68
157      tUNTIL  EQU    69
158      tLEAVE  EQU    70
159
160      tLOOP   EQU    96
161      tSELECT EQU    97
162      tCASE   EQU    98
163      tIF2    EQU    99
164      tELSE2  EQU    100
165
166      *
167      * quartets de reconnaissance de :
168      *
169      qENDL   EQU    0      END LOOP
170      qENDS   EQU    1      END SELECT
171      qENDI   EQU    2      END IF
172
173 010BC 8D00000 wrdscn GOVLNG =WRDSCN
174
175      *** parse entry for END
176 010C3 75FF   =END2p  GOSUB  wrdscn
177 010C7 00     CON(2) =tXWORD
178 010C9 1E34   CON(4) (tWHILE)~(id)
179 010CD 420    REL(3) ENDWp
180 010D0 00     CON(2) =tXWORD
181 010D2 1E06   CON(4) (tLOOP)~(id)
182 010D6 820    REL(3) ENDLp
183 010D9 00     CON(2) =tXWORD
184 010DB 1E16   CON(4) (tSELECT)~(id)
185 010DF 710    REL(3) ENDSp
186 010E2 00     CON(2) =tXWORD
187 010E4 1E36   CON(4) (tIF2)~(id)
188 010E8 410    REL(3) ENDIp
189 010EB 00     CON(2) 0
190 010ED 6351   GOTO   rest*
191
192      *
193      * Tokenisation du END WHILE / LOOP / SELECT / IF :
194      *
195      * Dans la chaine tokenisee (pointee par D0), nous
196      * avons les 6 quartets pour le codage du END (le
197      * notre, pas celui du systeme), puis un quartet
198      * de reconnaissance, servant a identifier laquelle
199      * des 4 structures nous terminons :
200      *   qENDL = 0      (necessaire)
201      *   qENDS = 1
202      *   qENDI = 2
203      * Si il n'y a pas de quartet de reconnaissance,
204      * c'est un END WHILE
```

```

204      *
205
206 010F1      ENDWp
207 010F1 185      D0=D0- 6      On oublie tWHILE
208 010F4 01      RTN      Cy := 0
209
210 010F6 21      ENDSp P=      =qENDS
211 010F8 6500      GOTO      ENDLp
212 010FC 22      ENDIp P=      =qENDI
213 010FE      ENDLp
214      *
215      * tXWORD id tEND2 tXWORD id tLOOP/SELECT/IF
216      *      ^
217      *      D0
218      *
219 010FE 185      D0=D0- 6      On retire le token reconnu
220      *
221      * tXWORD id tEND2 tXWORD id tLOOP/SELECT/IF
222      *      ^
223      *      D0
224      *
225 01101 80C0      C=P      0      C(0) := qEND L/S/I
226 01105 20      P=      0      c'est plus propre...
227 01107 8D00000  outnib GOVLNG =OUTNIB
228      *
229      * CASE ELSE
230      *
231 0110E 185      CS00p D0=D0- 6      tELSE nous interesse pas
232 01111 01      RTN      Cy := 0 par le D0=D0- 6
233
234      *****
235      * CASEp
236      *
237      * Syntaxes :
238      * CASE ELSE :
239      * tXWORD id tCASE
240      * <relop> <exp>
241      * tRELOP <1 quartet> texp
242      * <exp> TO <exp>
243      * texp tTO texp
244      * <exp>
245      * texp
246      * CASE <clause> , <clause>
247      * tXWORD id tCASE tclause tCOMMA tclause
248      *****
249
250      * parse entry for CASE
251 01113      =CASEp
252      *
253      * EXPTYP := 0 ;
254      *
255 01113 137      CD1EX
256 01116 1F00000      D1=(5) =F-R0-3
257 0111D AC2      C=0 S
258 01120 1554      DAT1=C S
259 01124 135      D1=C
260      *

```

```
261          * ELSE ?
262          *
263 01127 719F          GOSUB wrdscn
264 0112B 00          CON(2) =tXWORD
265 0112D 1E46          CON(4) (tELSE2)~(id)
266 01131 DDF          REL(3) CS00p
267 01134 00          CON(2) 0
268          *
269          * analyse des clauses
270          *
271 01136 7E40          GOSUB respt3
272 0113A          CSp10
273          *
274          * <relop> ?
275          *
276 0113A 8F00000      GOSBVL =NTOKEN
277 01141 3100          LC(2) =tRELOP
278 01145 966          ?A#C B
279 01148 A1          GOYES CSp20 <exp> TO <exp> ou <exp>
280          *
281          * CASE <relop> <exp>
282          *
283 0114A 8F00000      GOSBVL =OUTBYT stocker tRELOP
284 01151 AA6          C=A XS C(XS) := specificateur
285 01154 BB6          CSR X
286 01157 BB6          CSR X C(0) := specifier
287 0115A 79AF          GOSUB outnib stocker le "specifier"
288          *
289          * "CASE <25 @"
290          * ^
291          * D1
292          * tXWORD id tCASE tRELOP q
293          * ^
294          * D0
295          *
296          *
297 0115E 6B10          GOTO CSp30 parse expression & fin
298          *
299          * CASE <exp1> [ TO <exp2 ]
300          *
301 01162          CSp20
302 01162 7220          GOSUB respt3
303 01166 7430          GOSUB expck parse et verifie exp1
304          *
305          * "CASE 32 TO 127 @"
306          * ^
307          * D1
308          * tXWORD id tCASE t32
309          * ^
310          * D0
311          *
312 0116A 3100          LC(2) =tTO
313 0116E 966          ?A#C B
314 01171 D0          GOYES CSp40
315 01173 8F00000      GOSBVL =OUTBYT On n'incremente pas D1
316          *
317          * "CASE 32 TO 127 @"
```

```

318          *           ^
319          *           D1
320          * tXWORD id tCASE t32 tTO
321          *           ^
322          *           D0
323          *
324 0117A 7020   CSp30  GOSUB  expck
325 0117E       CSp40
326          *
327          * "... , ..."
328          *           ^
329          *           D1
330          * ... t<dernier token reconnu>
331          *           ^
332          *           D0
333          *
334          * A(B) = token a analyser (tCOMMA ou tEOL)
335          * D0 = ^ flot de sortie
336          * D1 = ^ passe le token A(B)
337          *
338 0117E 8F00000   GOSBVL =COMCK+
339 01185 44B       GOC   CSp10   ", " trouvee, on recommence
340
341 01188 8D00000   respt3 GOVLNG =RESPTR
342
343
344          *** parse entry for SELECT
345 0118F         =SELECP
346 0118F 8F00000   GOSBVL =EXPPAR
347 01196 870       ?ST=1  0
348 01199 93        GOYES  ivexp
349 0119B 5CE       GONC   respt3   B.E.T.
350
351          *****
352          * expck
353          *
354          * But: analyser l'expression pointee par D1, et
355          *   verifier que le type correspond a celui des
356          *   expressions precedentes.
357          * Entree:
358          *   - D0 = ^ flot de sortie
359          *   - D1 = ^ flot d'entree
360          * Sortie:
361          *   - A(B) = token suivant
362          *   - P = 0
363          *   - A(S) = C(S) = type de l'expression
364          *   - D0 = ^ flot de sortie (passee l'exp)
365          *   - D1 = ^ passe le 1er token non reconnu
366          * Appelle: EXPPAR
367          * Niveaux: 4 (EXPPAR)
368          * Abime: A-D, R0-R1, ST(0-3,7,11), FUNCD0, F-R0-3
369          * Detail: verification du type :
370          *   type := type de l'expression
371          *   si EXPTYP = 0
372          *   alors EXPTYP := type
373          *   sinon si EXPTYP # type alors erreur
374          *   fin si

```

```
375          * Historique:
376          *   87/02/28: conception & codage
377          * *****
378
379 0119E 8F00000   expck  GOSBVL =EXPPAR
380 011A5 870          ?ST=1  0          not valid expression ?
381 011A8 A2          GOYES  ivexp
382 011AA 301          LC(1)  1          num exp. <==> 1
383 011AD 873          ?ST=1  3          not string expression ?
384 011B0 40          GOYES  expck1
385 011B2 E6          C=C+1  A          string exp. <==> 2
386          *
387          * C(0) := type de l'expression
388          *
389 011B4 816          expck1 CSRC   W          C(S) := type
390 011B7 137          CD1EX
391 011BA 1F00000     D1=(5) =F-R0-3
392 011C1 1534         A=DAT1 S          A(S) := (EXPTYP)
393 011C5 948         ?A=0   S
394 011C8 11          GOYES  expck2
395 011CA 135         D1=C          restaurer D1
396 011CD 942         ?A=C   S
397 011D0 00          RTNYES          expressions meme type
398 011D2 8D00000     ivexp  GOVLNG =IVEXPe "Invalid Expr"
399
400 011D9 1554         expck2 DAT1=C S          EXPTYP := type
401 011DD 135         D1=C
402 011E0 01          RTN
403
404          * parse entry for IF
405 011E2           =IF2p
406          *
407          * "IF A+B THEN <eol>"
408          *   ^
409          *   D1
410          *
411          * tXWORD id tIF2
412          *   ^
413          *   D0
414          *
415 011E2 8F00000     GOSBVL =NUMCK
416          *
417          * "IF A+B THEN <eol>"
418          *   ^
419          *   D1
420          * tXWORD id tIFSTR tA tB t+
421          *   ^
422          *   D0
423          *
424 011E9 7B9F          GOSUB  respt3
425          *
426          * "IF A+B THEN <eol>"
427          *   ^
428          *   D1
429          *
430 011ED 7BCE          GOSUB  wrdscn
431 011F1 00          CON(2) =tTHEN
```

```

432 011F3 900          REL(3) THENp
433 011F6 00          CON(2) 0
434 011F8 6840        GOTO  rest*
435                    *
436                    * "IF A+B THEN <eol>"
437                    *           ^
438                    *           D1
439                    * tXWORD id tIFSTR tA tB t+ tTHEN
440                    *           ^
441                    *           D0
442                    *
443 011FC 181          THENp  D0=D0- 2          On oublie le token de THEN
444                    *
445                    * On fait maintenant le test decisif :
446                    * si on trouve tEOL, t@, t!, c'est a nous !
447                    * sinon, on repasse la main, via REST*, au IF
448                    * interne.
449                    * On ne peut pas utiliser EOLCK, car il accepte la
450                    * presence de tELSE. Hum...
451                    *
452 011FF 8F00000      GOSBVL =NTOKEN
453
454 01206 3100          LC(2)  =tEOL
455 0120A 962           ?A=C  B
456 0120D 81           GOYES  respt4
457 0120F 3104          LCASC  '@'
458 01213 962           ?A=C  B
459 01216 F0           GOYES  respt4
460 01218 3112          LC(2)  '!'
461 0121C 962           ?A=C  B
462 0121F 60           GOYES  respt4
463 01221 6F10          GOTO  rest*
464
465 01225 626F          respt4 GOTO respt3
466                    *
467                    * Tokenisation finale :
468                    * "IF A+B THEN <eol>"
469                    *           ^
470                    *           D1
471                    * tXWORD id tIFSTR tA tB t+ tEOL
472                    *           ^
473                    *           D0
474                    *
475
476                    *** parse entry for ELSE
477 01229              =ELSE2p
478 01229 137          CD1EX          Sauve D1
479 0122C 1F00000      D1=(5) =S-R0-3  "IF statement in progress"
480 01233 1574          C=DAT1 S
481 01237 135          D1=C          restaure D1
482 0123A 94E          ?C#0  S          "IF statement in progress"
483 0123D 40           GOYES  rest*
484 0123F 01           RTN          Ok, on prend (RTNCC)
485 01241 6620          rest*  GOTO Rest*  C'est pas a nous
486
487
488                    *****

```

```
489          * UNDERp
490          * But: analyser UNDERLINE ON|OFF
491          ****
492
493          *** parse entry for UNDERLINE
494 01245      =UNDERp
495 01245 8F00000      GOSBVL =CNVWUC  Conversion en majuscule
496 0124C 3154      LCASC 'E'
497 01250 966      ?A#C B      Est-ce "E" ?
498 01253 C1      GOYES syntax Non: "Erreur de syntaxe"
499 01255 171      D1=D1+ 2      Oui: caractÉre suivant
500          *
501          * Le code de UNDERp se poursuit dans ONOFFp
502          *
503
504          ****
505          * ONOFFp
506          * But: analyser ON|OFF
507          ****
508
509          *** parse entry for ATTN, BOLD, PERF, WRAP, ROMAN
510 01258      =ONOFFp
511 01258 706E      GOSUB wrdscn  Token suivant
512 0125C 00      CON(2) =tON      Est-ce ON ?
513 0125E C5E      REL(3) NOARGp  Oui: retour
514 01261 00      CON(2) 1+=tON  (<==> =tOFF)
515 01263 75E      REL(3) NOARGp
516 01266 00      NIBHEX 00      Fin de la table
517          *
518          * Si nous sommes arrivÀs lÈ, c'est que le token
519          * n'Àtait ni tON, ni tOFF. Alors, on ne reconnaît
520          * pas l'ordre, et on repasse la main au systÉme
521          * d'exploitation :
522          *
523 01268 8D00000  Rest* GOVLNG =REST*
524 0126F 8D00000  syntax GOVLNG =SYNTXe
525
526          STITLE Misc parse
```

```
527 *****
528 * PLFp
529 * But: analyser PLF [ <num> ]
530 *****
531 *** parse entry for MARGIN, PLF
532 01276 =MARGIp
533 01276 =PLFp
534 01276 8F00000 GOSBVL =EOLCK Fin de ligne ?
535 0127D 4A0 GOC RESp
536 01280 740F GOSUB respt3 Non : il y a donc un nb
537 01284 6C10 GOTO NUMp alors parsons-le.
538 01288 6FFE RESp GOTO respt3
539
540 *****
541 * PAGELp
542 * But: analyser PAGELEN [num [, num] ]
543 * Principe: on teste le premier paramêtre, et s'il
544 * existe, on continue sur DELAYp
545 *****
546 *** parse entry for PAGELEN
547 0128C =PAGELp
548 0128C 8F00000 GOSBVL =EOLCK Premier param. existe ?
549 01293 44F GOC RESp Non: on revient a Basic
550 01296 7EEE GOSUB respt3 Oui: on branche sur
551 0129A 8D00000 GOVLNG =DELAYp DELAY num [, num]
552
553
554 *** parse entry for GPSET, MODE, STACK, WHILE, UNTIL
555 012A1 NUMp
556 012A1 8D00000 =FIXp GOVLNG =FIXP
557
558
559 *** parse entry for EXIT
560 012A8 8D00000 =EXITp GOVLNG =NXTp
561
562
563 *** parse entry for EDIT
564 012AF =EDITp
565 **** changes
566
567 * rev D:
568 * GOSUB eolck+
569 * GOC RESp
570 * GONC EDP10 B.E.T.
571 *RNMP05 ST=1 8
572 *EDP10 GOSBVL =FSPECp
573 * GONC RNMP25
574 * GOVLNG =FSPECe
575 *RNMP25
576 * ?ST=0 8
577 * GOYES RNMP30
578 * RTNCC
579 *RNMP30
580 * GOSBVL =WRDSCN
581 * CON(2) =tTO
582 * REL(3) RNMP05
583 * CON(2) 0
```



```
584          *respt5 GOTO  respt9  (GOVLNG RESPTR)
585          * rev X
586 012AF 8DC0B30          GOVLNG #03B0C not supported
587          **** end of changes
588
589          **** changes
590          * removed in rev Ex (no more needed):
591          *** parse entry for LEX ON/OFF
592          *=LEXp  CD0EX
593          *          D0=(5) =STSAVE
594          *          DAT0=C A
595          *          D0=C
596          *          GOSBVL =FSPECp
597          *          GONC  lex
598          *          GOVLNG =FSPECe
599          *lex      GOSBVL =WRDSCN  Token suivant
600          *          CON(2) =tON      Est-ce ON ?
601          *          REL(3) LON
602          *          CON(2) 1+=tON    (<==> =tOFF)
603          *          REL(3) LOFF
604          *          CON(2) 0        Fin de la table
605          *          GOVLNG =SYNTXe
606          *
607          *LON      AD0EX
608          *          D0=(5) =STSAVE
609          *          C=DAT0 A
610          *          D0=C
611          *          D0=D0- 2
612          *          LCHEX  #0A
613          *          DAT0=C B
614          *          D0=A
615          *
616          *LOFF    D0=D0- 2
617          *          RTNCC
618          **** end of changes
619
620
621 012B6 8F00000 STRck+  GOSBVL =STRGCK  1er paramÉtre : alpha
622 012BD 8F00000          GOSBVL =COMCK+
623 012C4 400          RTNC
624 012C7 8D00000          GOVLNG =SYNTXe
625
626          *** parse entry for POKE$
627 012CE          =POKEp
628 012CE 74EF          GOSUB  STRck+
629 012D2 8D00000          GOVLNG =STRNGP
630
631          *** parse entry for WREC
632 012D9 79DF          =WRECP  GOSUB  STRck+
633 012DD 8F00000          GOSBVL =GNXTCR
634 012E4 8F00000          GOSBVL =R3=D10
635 012EB 8F00000          GOSBVL =EXPPAR  2Éme paramÉtre: alpha
636          *          *          ou numÁrique.
637 012F2 870          ?ST=1 0
638 012F5 91          GOYES  ivexpe
639 012F7 8F00000          GOSBVL =COMCK+
640 012FE 591          GONC  syntxe
```

```
641 01301 8E0000          GOSUBL =JUMPER
642 01307 52970          CON(5) DVCSPp  3Éme paramÉtre: spÅcifi-
643                      *          *          cateur d'appareil HPIL.
644 0130C 03            RTNCC
645
646 0130E 854          ivexpe  ST=1  4
647 01311 8D00000      GOVLNG =IVEXPe
648
649 01318 8D00000      syntxe  GOVLNG =SYNTXe
650
651
652                      STITLE FINPUT parse
```

```

653          *
654          * Syntaxe: FINPUT I$,M$, [P$], A
655          *
656
657 0131F 8D00000 ivpare GOVLNG =IVPARE
658
659 01326 7D22      comma  GOSUB  respt9
660 0132A 8F00000      GOSBVL =NTOKEN
661 01331 8F00000      GOSBVL =COMCK+      check & output tCOMMA
662 01338 400        RTNC                      Ok
663 0133B 8D00000      GOVLNG =MSPARE
664
665          *** parse entry for FINPUT
666 01342          =FINPUp
667          *          ST=0      8          Dummy arrays are valid
668 01342 859          ST=1      9          Single string variable parse
669 01345 8F00000      GOSBVL =READP5      supporte (mais oui !)
670 0134C 76DF        GOSUB  comma
671          * M$
672 01350 7C50        GOSUB  exppar          M$
673 01354 873          ?ST=1  3
674 01357 8C          GOYES  ivpare
675 01359 79CF        GOSUB  comma
676          * P$ ou A
677 0135D 136          CD0EX          !
678 01360 10A          R2=C          ! Quartet de reconnaissance := 0
679 01363 134          D0=C          ! si il n'y a pas P$
680 01366 D2          C=0      A          !
681 01368 8F00000      GOSBVL =OUTNIB      !
682          *** changes
683          *          GOSUB  SINp10
684 0136F 7D30        GOSUB  exppar
685 01373 873          ?ST=1  3
686 01376 D1          GOYES  SINp10
687          *** end of changes
688 01378 11A          C=R2          !
689 0137B 136          CD0EX          ! Quartet de reconnaissance := F
690 0137E AC0          A=0      S          ! si il y a P$
691 01381 A4C          A=A-1  S          !
692 01384 1504        DAT0=A  S          !
693 01388 134          D0=C
694 0138B 779F        GOSUB  comma
695          * A
696 0138F 7D10        GOSUB  exppar
697 01393 870        SINp10 ?ST=1  0          No valid expression ?
698 01396 C0          GOYES  ivvare
699 01398 863          ?ST=0  3          Valid string expression ?
700 0139B 70          GOYES  ivvare          invalid var
701 0139D 831          ?XM=0          expression = passable par reference
702 013A0 90          GOYES  respt6
703 013A2 8D00000      ivvare GOVLNG =IVVARE
704 013A9 8D00000      respt6 GOVLNG =RESPTR
705
706 013B0 8D00000      exppar GOVLNG =EXPPAR
707
708
709          STITLE ADxxx parse

```

```
710
711 013B7 8F00000 fspec+ GOSBVL =FSPECp
712 013BE 500 RTNNC
713 013C1 8D00000 GOVLNG =FSPECe
714 013C8 8F00000 numck+ GOSBVL =NUMCK
715 013CF 8D00000 respt7 GOVLNG =RESPTR
716 013D6 8F00000 varp+ GOSBVL =VARP
717 013DD 500 RTNNC
718 013E0 8D00000 GOVLNG =IVVARE
719 013E7 8F00000 ntok+ GOSBVL =NTOKEN
720 013EE 8F00000 GOSBVL =COMCK+
721 013F5 400 RTNC
722 013F8 8D00000 GOVLNG =MSPARE
723
724 *** parse entry for ADCREATE
725 013FF 74BF =ADCREp GOSUB fspec+
726 01403 8F00000 ADp+ GOSBVL =EOLCK
727 0140A 44C GOC respt7
728 0140D 7EBF GOSUB respt7
729 01411 72DF GOSUB ntok+
730 01415 8F00000 GOSBVL =STRGCK
731 0141C 62BF GOTO respt7
732
733 *** parse entry for ADDELETE
734 01420 739F =ADDELp GOSUB fspec+
735 01424 7FBF GOSUB ntok+
736 01428 7C9F GOSUB numck+
737 0142C 66DF GOTO ADp+
738
739 *** parse entry for ADGET
740 01430 738F =ADGETp GOSUB fspec+
741 01434 7FAF GOSUB ntok+
742 01438 7A9F GOSUB varp+
743 0143C 77AF GOSUB ntok+
744 01440 748F GOSUB numck+
745 01444 6EBF GOTO ADp+
746
747 *** parse entry for ADPUT
748 01448 7B6F =ADPUTp GOSUB fspec+
749 0144C 779F GOSUB ntok+
750 01450 728F GOSUB varp+
751 01454 6EAF GOTO ADp+
752
753 STITLE BLIST module parse
```

```
754
755 *****
756 * DBLSTp, PBLSTp
757 *
758 * But : parse quelque chose de tres complique...
759 * Historique :
760 * 86/05/...: JPB reecriture d'apres I.D.S.
761 * 88/01/10: PD & JT documentation
762 * 88/01/31: PD & JT correction de PBLIST TOTO
763 *****
764
765 *** parse entry for PBLIST, DBLIST
766 01458 =PBLISp
767 * Mettre a 0 les flags PRSETO et PRSEIN
768 * qui interdisent de mettre deux fois la meme
769 * option.
770 01458 AEO A=0 B Le 88/01/31 : gagne quelques
771 0145B 136 CD0EX quartets
772 0145E 1B00000 D0=(5) =FUNCR0
773 01465 148 DAT0=A B Pas encore vu de TO/INDENT
774 01468 134 D0=C
775 * [<file>] [, <li#> [, <li#>]] [TO <file>][INDENT <n>]
776 * Autrefois inspire de LISTP (#03B92 / HP71B)
777 0146B 7690 GOSUB eolckx
778 0146F 425 GOC respt8 tEOL trouve
779 01472 7211 GOSUB CKINDT
780 01476 417 GOC LSTP60 tINDENT trouve
781 01479 7540 GOSUB respt8
782 * <file> ou bien encore TO
783 * ^ ^
784 0147D 8F00000 GOSBVL =FSPECp specificateur valide ?
785 01484 5A1 GONC LSTP12 oui
786 * si S7=1
787 * alors reserved word in A
788 * (TO, ALL, KEYS, INTO, CARD)
789 * sinon bad file parse
790 * fin si
791 01487 867 ?ST=0 7 Bad file parse ?
792 0148A F1 GOYES LSTP20 Oui : peut-etre <line#>
793 * reserved word in A. Est-ce tTO ?
794 0148C 181 D0=D0- 2 Revient sur tTO
795 0148F 3100 LC(2) =tTO
796 01493 962 ?A=C B
797 01496 33 GOYES LSTP40
798
799 01498 8D00000 fspece GOVLNG =FSPECe Illegal file spec
800
801 * <file> , ...
802 * ^
803 0149F 8F00000 LSTP12 GOSBVL =COMCK Comma ?
804 014A6 5D0 GONC LSTP30 Non : TO ou INDENT ou vraie fin
805
806 * <file> , <line> ...
807 * ^
808
809 014A9 7EE0 LSTP20 GOSUB ck2li# <line1> [ , <line2> ]
810 * ... TO <file>
```

```
811 * ^
812 014AD 8F00000 LSTP29 GOSBVL =NTOKEN Pour avancer
813 014B4 75C0 LSTP30 GOSUB CKTO
814 014B8 401 GOC LSTP40
815 014BB 79C0 GOSUB CKINDT
816 014BF 482 GOC LSTP60
817 014C2 8D00000 respt8 GOVLNG =RESPTR
818
819 * Traitement de TO <file>
820 014C9 3400000 LSTP40 LC(5) (=FUNCR0)+0
821 014D0 7A80 GOSUB CKSEEN
822 014D4 8F00000 GOSBVL =OUT1TK
823 014DB 8F00000 GOSBVL =FSPECp specificateur valide ?
824 014E2 5AC GONC LSTP29 Oui : on continue
825 014E5 42B GOC fspece Non : erreur (B.E.T.)
826
827 * Traitement de INDENT <expression>
828 014E8 3400000 LSTP60 LC(5) (=FUNCR0)+1
829 014EF 7B60 GOSUB CKSEEN
830 014F3 8F00000 GOSBVL =OUT3TK
831 014FA 8F00000 GOSBVL =NUMCK
832 01501 62BF GOTO LSTP30
833
834 *****
835 * eolckx
836 *
837 * But : idem EOLCK, mais avec NTOKEN au lieu de
838 * WRDSCN.
839 * Entree :
840 * - D1 = ^ flot ascii
841 * Sortie :
842 * - Cy = 1 si tEOL trouve
843 * Appelle : NTOKEN, FINDA
844 * Niveaux : 3
845 * Abime : A, B, C, P, D0, D1, R0, S0-3, S11
846 * Historique :
847 * 88/01/31: PD & JT recodage apres suppression
848 *****
849
850 01505 8F00000 eolckx GOSBVL =NTOKEN
851 0150C 8F00000 GOSBVL =FINDA
852 01513 00 CON(2) =tEOL
853 01515 610 REL(3) rtnsc
854 01518 00 CON(2) =t@
855 0151A 110 REL(3) rtnsc
856 0151D 00 CON(2) =t!
857 0151F C00 REL(3) rtnsc
858 01522 00 CON(2) =tELSE
859 01524 700 REL(3) rtnsc
860 01527 00 NIBHEX 00
861 01529 03 RTNCC Cy = 0 : non trouve
862 0152B 02 rtnsc RTNSC Cy = 1 : trouve
863
864 *****
865 * RENUMp
866 *
867 * But : parse RENUMREM [<11>[,<12>[,<13>[,<14>]]]]
```

```
868          * Historique :
869          *   86/05/..: JPB      reecriture d'apres I.D.S.
870          *   88/01/10: PD & JT documentation
871          * *****
872
873          *** parse entry for RENUMREM
874 0152D 7C10 =RENUMp GOSUB eolck+
875 01531 452          GOC  respt9  EOL : sortir
876 01534 7360          GOSUB ck2li#
877 01538 869          ?ST=0  9      2 <line#> trouves ?
878 0153B 01          GOYES  rtncc  Non : sortir
879 0153D 8F00000      GOSBVL =COMCK  Comma ?
880 01544 521          GONC  respt9  Non : sortir
881 01547 7050          GOSUB  ck2li#
882 0154B 03          rtncc  RTNCC      Ok
883
884 0154D 8F00000      eolck+ GOSBVL =EOLCK  fin de commande ?
885 01554 400          RTNC      oui : Cy=1
886 01557 8D00000      respt9 GOVLNG =RESPTR non : on revient
887
888          * *****
889          * CKSEEN
890          *
891          * But : verifier que le token TO ou INDENT n'a pas
892          *   deja ete vu durant la parse.
893          * Entree :
894          *   - C(A) = adresse du flag (PRSETO ou PRSEIN)
895          * Sortie :
896          *   - si deja vu, alors erreur
897          *   sinon flag mis a jour, Cy = 0
898          * Niveaux : 0
899          * Utilise : C(S)
900          * Historique :
901          *   88/01/10: PD & JT conception & codage
902          * *****
903
904 0155E 136          CKSEEN CD0EX
905 01561 1564          C=DAT0 S
906 01565 94E          ?C#0  S      Deja vu ?
907 01568 E0          GOYES  CKS10  oui : erreur
908 0156A B46          C=C+1  S
909 0156D 1544          DAT0=C  S
910 01571 136          CD0EX
911 01574 01          RTN
912 01576 8D00000      CKS10  GOVLNG =SYNTXe "Syntax Error"
913
914          * *****
915          * CKTO, CKINDT
916          *
917          * But : tester tTO ou tINDENT
918          * Entree :
919          *   - A = token a tester
920          * Sortie :
921          *   Cy = 1 : token trouve
922          *   Cy = 0 : token cherche non trouve
923          * Niveaux : 0
924          * Utilise : C
```

```
925          * Historique :
926          *   88/01/10: PD & JT conception & codage
927          *****
928
929 0157D 3100      CKTO   LC(2)  =tTO
930 01581 962          ?A=C   B
931 01584 00          RTNYES
932 01586 01          RTN
933
934 01588 35FE1ED6  CKINDT  LCHEX  6DE1EF      LC(6)  (=tINDENT)~(=id)~(=tXWORD)
935 01590 25          P=      5
936 01592 912          ?A=C   WP
937 01595 20          GOYES  CKIN10
938 01597 20          CKIN10 P=      0
939 01599 01          RTN
940
941          *****
942          * ck2li#
943          *
944          * But : parse " <line#1> [ , <line#2> ] "
945          * Entree :
946          *   - D1 = ^ blancs optionnels avant
947          * Sortie :
948          *   - S9 = 1 si les deux <line#> ont ete reconnus
949          * Appelle : ck1li#, COMCK, NTOKNL, OUT3TK
950          * Niveaux : 4 (NTOKNL)
951          * Utilise : A-C, P, D0, D1, R0, S0-S3, S11
952          * Historique :
953          *   86/05/..: JPB      reecriture d'apres I.D.S.
954          *   88/01/10: PD & JT separation & documentation
955          *****
956
957 0159B 849      ck2li# ST=0   9
958 0159E 7D00      GOSUB  ck1li# <line1>
959 015A2 8F00000  GOSBVL =COMCK  Comma ?
960 015A9 5A2      GONC   respt0  Non : sortir
961 015AC 859      ST=1   9      On reconnait les 2 <line#>
962
963 015AF 8F00000  ck1li# GOSBVL =NTOKNL  Idem NTOKEN, autorise line#
964 015B6 3100      LC(2)  =tLINE#
965 015BA 966          ?A#C   B
966 015BD 01          GOYES  LSTPE   Pas <line#> : erreur
967 015BF 3100      LC(2)  =tCOMMA
968 015C3 AEA          A=C    B
969 015C6 8D00000  GOVLNG =OUT3TK
970
971 015CD 8D00000  LSTPE  GOVLNG =IVPARE  INVALID (MISSING PARM)
972 015D4 8D00000  respt0 GOVLNG =RESPTR
973
974
975          *** parse entry for DDIR, PDIR
976 015DB 8F00000  =DDIRp GOSBVL =EOLCK
977 015E2 41F      GOC    respt0
978 015E5 7BEF      GOSUB  respt0
979 015E9 8F00000  GOSBVL =FSPECp
980 015F0 502      GONC   dirpl
981 015F3 867      ?ST=0  7
```



```
982 015F6 41          GOYES  Fspece
983 015F8 3100       LC(2)  =tALL
984 015FC 962        ?C=A   B
985 015FF 21         GOYES  dirp1
986 01601 3100       LC(2)  =tTO
987 01605 962        ?C=A   B
988 01608 42         GOYES  dirp3
989 0160A 8D00000    Fspece  GOVLNG =FSPECe
990 01611 8F00000    dirp1  GOSBVL =NTOKEN
991 01618 3100       LC(2)  =tTO
992 0161C 962        ?C=A   B
993 0161F 60         GOYES  dirp2
994 01621 62BF       GOTO   respt0
995 01625 8F00000    dirp2  GOSBVL =OUT1TK
996 0162C 8F00000    dirp3  GOSBVL =FSPECp
997 01633 46D        GOC    Fspece
998 01636 03         RTNCC
999
1000 01638          END
```


eolck+	Abs	5453	#0000154D	-	884	874				
eolckx	Abs	5381	#00001505	-	850	777				
expck	Abs	4510	#0000119E	-	379	303	324			
expck1	Abs	4532	#000011B4	-	389	384				
expck2	Abs	4569	#000011D9	-	400	394				
exppar	Abs	5040	#000013B0	-	706	672	684	696		
fspec+	Abs	5047	#000013B7	-	711	725	734	740	748	
fspece	Abs	5272	#00001498	-	799	825				
id	Abs	225	#000000E1	-	152	178	181	184	187	265
ivexp	Abs	4562	#000011D2	-	398	348	381			
ivexpe	Abs	4878	#0000130E	-	646	638				
ivpare	Abs	4895	#0000131F	-	657	674				
ivvare	Abs	5026	#000013A2	-	703	698	700			
mspare	Abs	4275	#000010B3	-	138	134				
ntok+	Abs	5095	#000013E7	-	719	729	735	741	743	749
numck	Abs	4051	#00000FD3	-	57	32	48	52		
numck+	Abs	5064	#000013C8	-	714	736	744			
outnib	Abs	4359	#00001107	-	227	287				
pfsp	Abs	3963	#00000F7B	-	34	30				
pstr1	Abs	3980	#00000F8C	-	37	33				
qENDI	Abs	2	#00000002	-	171	212				
qENDL	Abs	0	#00000000	-	169					
qENDS	Abs	1	#00000001	-	170	210				
respt0	Abs	5588	#000015D4	-	972	960	977	978	994	
respt1	Abs	4151	#00001037	-	89	87				
respt2	Abs	4268	#000010AC	-	137	100	101	118	119	129 132
respt3	Abs	4488	#00001188	-	341	271	302	349	424	465 536
					538	550				
respt4	Abs	4645	#00001225	-	465	456	459	462		
respt6	Abs	5033	#000013A9	-	704	702				
respt7	Abs	5071	#000013CF	-	715	727	728	731		
respt8	Abs	5314	#000014C2	-	817	778	781			
respt9	Abs	5463	#00001557	-	886	659	875	880		
resptr	Abs	4037	#00000FC5	-	53	47	49	51		
rest*	Abs	4673	#00001241	-	485	190	434	463	483	
rtnc	Abs	5451	#0000154B	-	882	878				
rtnsc	Abs	5419	#0000152B	-	862	853	855	857	859	
syntax	Abs	4719	#0000126F	-	524	498				
syntx	Abs	4065	#00000FE1	-	60	38	44			
syntxe	Abs	4888	#00001318	-	649	640				
t!	Ext			-	856					
t@	Ext			-	854					
tALL	Ext			-	983					
tCASE	Abs	98	#00000062	-	162					
tCOMMA	Ext			-	967					
tELSE	Ext			-	858					
tELSE2	Abs	100	#00000064	-	164	265				
tEND2	Abs	66	#00000042	-	154					
tEOL	Ext			-	454	852				
tIF2	Abs	99	#00000063	-	163	187				
tLEAVE	Abs	70	#00000046	-	158					
tLINE#	Ext			-	964					
tLOOP	Abs	96	#00000060	-	160	181				
tON	Ext			-	512	514				
tRELOP	Ext			-	277					
tREPEAT	Abs	68	#00000044	-	156					
tSELECT	Abs	97	#00000061	-	161	184				

tTHEN	Ext		-	431					
tTO	Ext		-	312	795	929	986	991	
tUNTIL	Abs	69	#00000045	-	157				
tWHILE	Abs	67	#00000043	-	155	178			
tXWORD	Ext		-	177	180	183	186	264	
varp+	Abs	5078	#000013D6	-	716	742	750		
vs1	Abs	4085	#00000FF5	-	69	67			
vs2	Abs	4116	#00001014	-	78	74			
wrdscn	Abs	4284	#000010BC	-	173	176	263	430	511

Input Parameters

Source file name is parse.a

Listing file name is parse.l

Object file name is parse.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE JUMPER
2
3 01638          ABS      #1638
4
5          *****
6          **
7          ** Name:          JUMPER - Jump to a routine in HPIL ROM
8          **
9          ** Category:      ADDCAL
10         **
11         ** Purpose:
12         **     By giving the offset of a routine entry in the HPIL
13         **     ROM, this routine will find the absolute start
14         **     address of the HPIL ROM and do an indirect jump to
15         **     specified routine.
16         **
17         ** Entry:
18         **     RSTK points to the 5-nibble offset from the start of
19         **     the LEX file to the desired entry point.
20         **
21         ** Exit:
22         **     LEX file found:
23         **     Jumps to desired routine with all CPU registers pre-
24         **     served, including carry and mode (DEC/HEX), with the
25         **     exception of SB ("Sticky Bit")
26         **     Execution will return to after the 5 nibbles offset.
27         **
28         **     LEX file not found:
29         **     Jumps directly to MFERR with error "XWORD Not Found"
30         **
31         ** Calls:          I/OFND
32         **
33         ** Uses.....
34         **     Inclusive: SNAPBF[44:0]
35         **
36         ** Stk lvls:      2 (I/OFND)
37         **
38         ** NOTE: 1) Stk lvls are used only within this routine and do
39         **     not apply to the destination routine (ie the use
40         **     is only a transient usage within this routine, and
41         **     nothing remains on RSTK when this routine jumps to
42         **     the target routine except whatever was on the RSTK
43         **     on entry to this routine)
44         **
45         **     2) The proper way to set up the RSTK as needed for
46         **     the entry conditions to this routine:
47         **
48         **         .
49         **         . (Assembly code preceding the call)
50         **         .
51         **         GOSUBL =JUMPER
52         **         CON(5) ({targetaddr})-({target LEX table addr})
53         **         .
54         **         . (Continue with assembly code here)
55         **         .
56         **
57         *****
```

```

58          **
59
60          SNAPBF EQU #2F7F0      the snapshot buffer
61          bLEX   EQU #00BFC      the LEX buffer ID #
62          I/OFND EQU #118BA      finds an I/O buffer given the ID #
63          LEXPIL EQU #000FF      the HPIL ROM ID #
64          eXWORD EQU #00023      mainframe error # for XWORD
65          BSERR  EQU #0939A      the BASIC error reporting routine
66          *
67          *
68 01638     =JUMPER
69          *
70          * Save D1, C[W], A[W], B[A], P, carry, and mode in SNAPBF
71          * (Total size of SNAPBF is 16+16+5+5+5, or 47 nibbles. This
72          * routine uses 45 of those nibbles)
73
74 01638 06          RSTK=C
75 0163A 137        CD1EX
76 0163D 1F0F7F2   D1=(5) =SNAPBF
77 01644 145        DAT1=C A          Write D1 @ SNAPBF
78 01647 1D5F      D1=(2) (=SNAPBF)+5
79 0164B 07        C=RSTK
80 0164D 1557      DAT1=C W          Write C[W] @ SNAPBF + 5
81 01651 1E508F   D1=(4) (=SNAPBF)+21
82 01657 1517      DAT1=A W          write A[W] @ SNAPBF + 21
83 0165B 1D51     D1=(2) (=SNAPBF)+37
84 0165F D9        C=B A
85 01661 80F5     CPEX 5          Save P @ SNAPBF + 42
86 01665 26       P= 6
87 01667 A82      C=0 P
88 0166A 550      GONC JUMP05      C[6]='0' means carry clear
89 0166D A0E      C=C-1 P          C[6]#'0' means carry set
90 01670 27       JUMP05 P= 7
91 01672 A82      C=0 P
92 01675 A0E      C=C-1 P          C[7]="9" means decimal mode
93 01678 15D7     DAT1=C 8          Write B[A],P,Carry,mode@SNAPBF+37
94 0167C 04       SETHEX           Force HEX mode for I/OFND
95          *
96          * Now A[W],B[A],C[W],P and D1 are available for use
97          *
98 0167E 20       P= 0
99 01680 32CFB    LC(3) =bLEX       Find the LEX buffer
100 01685 8FAB811 GOSBVL =I/OFND
101 0168C 543     GONC JUMP90       Not there!! (Error)
102          *
103          * Found the LEX buffer...D1 points to it
104          *
105          * Search the LEX buffer for the HPIL ROM LEX ID
106          *
107 0168F 31FF     LC(2) =LEXPIL     C[B] = HPIL ROM LEX ID
108 01693 D5       B=C A
109 01695 D0       A=0 A
110 01697 E4       A=A+1 A          A[B] = 1
111          *
112 01699 15F5     JUMP10 C=DAT1 6
113 0169D 96A     ?C=0 B           End of LEX buffer?
114 016A0 12      GOYES JUMP90     Yes...exit

```



```

115 016A2 965          ?B#C  B      Right ID?
116 016A5 61          GOYES  JUMP20   No...try next one
117                    *
118                    * LEX ID number matches...check if the token # is in the range
119                    *
120 016A7 BF6          CSR    W
121 016AA F6           CSR    A      C[3:0] is now the token range
122 016AC 9E2          ?A<C  B      Too small ?
123 016AF C0           GOYES  JUMP20   Yes...keep looking
124 016B1 F6           CSR    A
125 016B3 F6           CSR    A
126 016B5 B62         C=C-A  B      If no carry, token # is in range
127 016B8 551         GONC   JUMP30   In range...process offset
128 016BB 17A         JUMP20 D1=D1+ 11  Not in range...goto next LEX entry
129 016BE 5AD         GONC   JUMP10   Go always
130                    * _
131                    * _
132                    *
133 016C1 333200       JUMP90 LC(4)  =eXWORD  "XWORD Not Found"
134 016C7 8DA9390       GOVLNG =BSERR  Do NOT return to caller if error
135                    * _
136                    * _
137                    *
138                    * Found the requested LEX table
139                    *
140 016CE 175          JUMP30 D1=D1+ 6      Point to address of main table
141 016D1 147          C=DAT1 A      Read the address of table into C
142 016D4 D5          B=C    A      Put address of table into B[A]
143                    *
144                    * Now get offset from main table start from the RSTK pointer
145                    *
146 016D6 07          C=RSTK      Get address of offset...
147 016D8 135         D1=C        ...into D1
148 016DB 174         D1=D1+ 5    Skip the offset field
149 016DE 137         CD1EX
150 016E1 06          RSTK=C      Put return address back on RSTK
151 016E3 147         C=DAT1 A    Read offset from main table
152 016E6 C9          C=C+B  A    Add address of main table
153 016E8 06          RSTK=C      Push desired address onto RSTK
154                    *
155                    * Now restore the registers and jump to the routine
156                    *
157 016EA 1F508F2       D1=(5) (=SNAPBF)+21
158 016F1 1537        A=DAT1 W    Restore A[W]
159 016F5 17F         D1=D1+ 16   Position to carry/mode/B[A] save area
160 016F8 15F7        C=DAT1 8
161 016FC D5          B=C    A    Restore B[A]
162 016FE 27          P=     7    Check mode
163 01700 B06         C=C+1  P    If carry, hex mode
164 01703 440         GOC    JUMP50
165 01706 05         SETDEC      No carry = DEC mode
166 01708 26         JUMP50 P=    6
167 0170A 90E        ?C#0  P
168 0170D 20         GOYES  JUMP60   Set carry if C[7]#0
169 0170F 80D5       JUMP60 P=C    5    Eestore P from C[5]
170 01713 1E5F7F     D1=(4) (SNAPBF)+5
171 01719 1577        C=DAT1 W    Restore C[W]

```

```
172 0171D 1D0F          D1=(2) SNAPBF
173 01721 06           RSTK=C          (Temporarily save C[A] on RSTK)
174 01723 147         C=DAT1 A
175 01726 135         D1=C          Restore D1
176 01729 07         C=RSTK        (Restore C[A] from RSTK)
177 0172B 01         RTN          Jump to the routine
178
179 0172D             END
```

BSERR	Abs	37786	#0000939A	-	65	134						
I/OFND	Abs	71866	#000118BA	-	62	100						
JUMP05	Abs	5744	#00001670	-	90	88						
JUMP10	Abs	5785	#00001699	-	112	129						
JUMP20	Abs	5819	#000016BB	-	128	116	123					
JUMP30	Abs	5838	#000016CE	-	140	127						
JUMP50	Abs	5896	#00001708	-	166	164						
JUMP60	Abs	5903	#0000170F	-	169	168						
JUMP90	Abs	5825	#000016C1	-	133	101	114					
=JUMPER	Abs	5688	#00001638	-	68							
LEXPIL	Abs	255	#000000FF	-	63	107						
SNAPBF	Abs	194544	#0002F7F0	-	60	76	78	81	83	157	170	
					172							
bLEX	Abs	3068	#00000BFC	-	61	99						
eXWORD	Abs	35	#00000023	-	64	133						

Input Parameters

Source file name is jumper.a
Listing file name is jumper.l
Object file name is jumper.o
Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE PPOLL
2
3          * *****
4          * PPOLL module
5          *
6          * Keyword(s):
7          *   NLOOP, PPOLL, SRQ, SLEEP
8          * Author(s):
9          *   Jean-Francois Garnier
10         * History:
11         *   Published in JPC #37
12
13         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14         * *****
15
16 0172D          ABS      #172D
17
18         * HPIL ROM calls:
19 GETLPs EQU      #1D15
20 PUTC   EQU      #6B1C
21 GET    EQU      #6751
22
23         * *****
24         * NLOOP entry
25         * *****
26 0172D 801          NIBHEX 801
27 01730 8E0000      =NLOOpe GOSUBL =JUMPER  cherche la boite
28 01736 51D10      CON(5) GETLPs  au lettre hpil
29 0173B 476        GOC      err      erreur?
30 0173E 330010     LC(4)  #0100   demande le
31 01744 7370      GOSUB  SEND    nbre de periph
32 01748 D0        A=0    A
33 0174A AF5       B=C    W
34 0174D AEA       A=C    B      AAD
35 01750 7780     GOSUB  MPY
36 01754 CA       A=A+C  A      AEP
37 01756 7180     GOSUB  MPY
38 0175A 7880     GOSUB  MPY0
39 0175E CA       A=A+C  A      AES
40 01760 542      GONC   SRQ1   B.E.T.
41
42         * *****
43         * PPOLL entry
44         * *****
45 01763 801          NIBHEX 801
46 01766 7D30      =PPOLLe GOSUB  SIDY    envoie IDY 0
47 0176A AEA       A=C    B      resultat du ppoll
48 0176D 571      GONC   SRQ1   B.E.T.
49
50         * *****
51         * SRQ entry
52         * *****
53 01770 801          NIBHEX 801
54 01773 7030      =SRQe  GOSUB  SIDY    envoie IDY 0
55 01777 F6        CSR    A
56 01779 F6        CSR    A
57 0177B A8A       A=C    P      (P=0)
```

```

58 0177E 301          LC(1)  1
59 01781 0E06        A=A&C  P      isole bit SRQ
60 01785 8F00000    SRQ1   GOSBVL  =HDFLT  conv flottant
61 0178C 1B00000    D0=(5) =FUNCD0
62 01793 146        C=DAT0  A
63 01796 136        CD0EX           restaure D0
64 01799 AF6        C=A     W      resultat
65 0179C 8D00000    GOVLNG  =FNRTN1  fin
66
67 017A3 6000      err    GOTO    =PILERR  HPIL error ??
68
69 017A7 8E0000    SIDY   GOSUBL  =JUMPER  cherche la boite
70 017AD 51D10     CON(5) GETLPs  au lettre hpil
71 017B2 40F       GOC     err      erreur?
72 017B5 3300E1    LC(4)   #1E00
73 017BB 8E0000    SEND   GOSUBL  =JUMPER  envoie la commande
74 017C1 C1B60     CON(5) PUTC   au processeur d'e/s
75 017C6 4CD       GOC     err      erreur?
76 017C9 8E0000    GOSUBL  =JUMPER  recoit la
77 017CF 15760     CON(5) GET    reponse
78 017D4 4EC       GOC     err      erreur?
79 017D7 D0        A=0     A      initialise A pour
80 017D9 03        RTNCC           la suite
81
82 017DB BF5       MPY    BSR     W
83 017DE BF5       BSR     W
84 017E1 D2        C=0     A
85 017E3 AE9       C=B     B
86 017E6 C6       MPY0   C=C+C   A      2
87 017E8 D7        D=C     A
88 017EA C7        D=D+D   A      4
89 017EC CB        C=C+D   A      6
90 017EE F2        CSL     A      96
91 017F0 CB        C=C+D   A     100
92 017F2 01        RTN
93
94
95
96
97
98 017F4 00000     REL(5) =NOARGd
99 017F9 00000     REL(5) =NOARGp
100 017FE 136      =SLEEPe CD0EX   sauve D0
101 01801 108      R0=C      et D1 en
102 01804 137      CD1EX     R0 et R1
103 01807 109      R1=C
104 0180A 8F00000    GOSBVL  =SLEEP  en sommeil lâger
105 01811 8F00000    GOSBVL  =CKSREQ  traite les SRQ
106 01818 119      C=R1
107 0181B 135      D1=C      restaure D1
108 0181E 118      C=R0      et
109 01821 134      D0=C      D0
110 01824 8D00000    GOVLNG  =NXTSTM
111
112 0182B          END

```

CKSREQ	Ext		-	105				
FNRTN1	Ext		-	65				
FUNCD0	Ext		-	61				
GET	Abs	26449	#00006751	-	21	77		
GETLPs	Abs	7445	#00001D15	-	19	28	70	
HDFLT	Ext		-	60				
JUMPER	Ext		-	27	69	73	76	
MPY	Abs	6107	#000017DB	-	82	35	37	
MPY0	Abs	6118	#000017E6	-	86	38		
=NLOOPe	Abs	5936	#00001730	-	27			
NOARGd	Ext		-	98				
NOARGp	Ext		-	99				
NXTSTM	Ext		-	110				
PILERR	Ext		-	67				
=PPOLLe	Abs	5990	#00001766	-	46			
PUTC	Abs	27420	#00006B1C	-	20	74		
SEND	Abs	6075	#000017BB	-	73	31		
SIDY	Abs	6055	#000017A7	-	69	46	54	
SLEEP	Ext		-	104				
=SLEEPe	Abs	6142	#000017FE	-	100			
SRQ1	Abs	6021	#00001785	-	60	40	48	
=SRQe	Abs	6003	#00001773	-	54			
err	Abs	6051	#000017A3	-	67	29	71	75 78

Input Parameters

Source file name is ppoll.a

Listing file name is ppoll.l

Object file name is ppoll.o

Flags set on command line
None

Errors

None


```
1          TITLE JPC LEX: MODULE MMLEX
2
3          * *****
4          * Mass memory functions
5          *
6          * Keyword(s):
7          *   MAXD, MEMM, MEMD, MAXM
8          * Author(s):
9          *   Michel Martinet
10         * History:
11         *   Published in JPC #30
12         *
13         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14         * *****
15
16 0182B          ABS      #182B
17
18         * HPIL ROM calls:
19         DEVPAR EQU      #1BF0
20         CHKMAS EQU      #425C
21         GDIRST EQU      #4843
22         ENDTAP EQU      #44D9
23         READ# EQU       #44FF
24
25         * local symbols:
26         ePIL EQU        2
27         Flg EQU         11
28         memflg EQU      10
29
30
31         * *****
32         * La routine VERTAP verifie si le parametre
33         * pointe par D1 correspond bien a une memoire
34         * de masse d'identificateur appareil 16.
35         * *****
36
37 0182B 22       Dvnmfd P=   ePIL      Classe d'erreur 32 a 47
38 0182D D2       C=0      A          Erreur 32.
39 0182F 492     GOC       erreur    B.E.T.
40
41 01832 8E0000   VERTAP  GOSUBL =JUMPER
42 01838 0FB10   CON(5)  DEVPAR      Routine d'analyse du
43         *                ) specificateur d' appareil
44         *                ) de la fonction. En sortie
45         *                ) D1 ^ Math Stack - 16 pour
46         *                ) un parametre numerique.
47 0183D 4B1     GOC       erreur    Si carry: ERREUR
48 01840 93B     ?D=0     X          D[X] contient l'adresse du
49         *                ) peripherique. Si D[X] = 0,
50 01843 8E       GOYES   Dvnmfd     le pAriphArique n'est pas
51         *                ) prAent. On renvoie
52         *                ) "Device Not Found"
53 01845 137     CD1EX      ) Sauvegarde de D1 pour
54 01848 108     R0=C      ) le retour A BASIC (fin).
55 0184B 8E0000   GOSUBL  =JUMPER  VArification de l'AID du
56 01851 C5240   CON(5)  CHKMAS    pAriphArique (16).
57 01856 500     RTNNC      Cy=0 : OK, Cy=1 : ERREUR
```

```
58
59 01859 6000      erreur GOTO      =PILEERR  HPIL erreur
60
61                * *****
62                * MAXD entry
63                * *****
64
65 0185D C11              NIBHEX C11      Alpha ou un numÃrique.
66 01860 136            =MAXDe CD0EX
67 01863 06             RSTK=C
68 01865 7DD1           GOSUB  r<rstk
69 01869 75CF           GOSUB  VERTAP
70 0186D 8E0000         GOSUBL =JUMPER
71 01873 34840         CON(5) GDIRST  Get DIRectory SStart
72 01878 40E           GOC      erreur  Erreur si carry
73 0187B AFB           C=D      W      D[5:8] = 1. du dir. (enr.)
74 0187E 8F00000       GOSBVL =CSRW5  C[A] = longueur du dir.
75 01885 DA            A=C      A      A[A] = longueur du dir.
76 01887 F0            ASL      A      A[A] = 16 fois 1. du dir.
77 01889 81C           ASRB      A[A] = 1. du dir. * 8
78
79                *****
80                * SORTIE: commune, RETOUR A BASIC : Replace
81                * 4 niveaux de pile des retours, le compteur
82                * programme dans D0, transforme A[A] en nombre
83                * flottant dans A[W], replace D1 È sa place
84                * sur la Math-Stack puis retour È Basic via
85                * FNRTN4. (Rembobine le support)
86                *****
87
88 0188C 7000          SORTIE GOSUB =JUMPER  Rembobine le support
89 01890 9D440         CON(5) ENDTAP  de mÃmoire de masse
90 01895 472           GOC      Erreur
91 01898 24            P=      4      ) Restitution de 5 niveaux
92 0189A 8F00000       GOSBVL =RSTK<R  ) de pile
93 018A1 07            C=RSTK  ) puis de
94 018A3 134           D0=C      ) D0 (4 niveaux & PC)
95 018A6 8F00000       GOSBVL =HDFLT  A[W] = longueur
96 018AD 118           C=R0      R0 = D1 (MS-16)
97 018B0 135           D1=C      D1 = MS-16
98 018B3 AF6           C=A      W      C[W] = longueur
99 018B6 8D00000       GOVLNG =FNRTN4  retour È BASIC
100
101 018BD 6B9F          Erreur GOTO  erreur  Ralonge
102
103 018C1 8D00000      memerr GOVLNG =MEMERR
104
105
106                * *****
107                * MEMM entry
108                * *****
109 018C8 C11              NIBHEX C11
110 018CB 84A            =MEMMe ST=0  memflg
111 018CE 85B            ST=1  Flg
112 018D1 531            GONC   MXMC
113
114
```

```

115          * *****
116          * MEMD entry
117          * *****
118 018D4 C11          NIBHEX C11
119 018D7 85A        =MEMDe ST=1 memflg
120 018DA 84B          ST=0 Flg
121 018DD 570          GONC MXMC
122
123
124          * *****
125          * MAXM entry
126          * *****
127 018E0 C11          NIBHEX C11      Alpha ou un numÃrique
128 018E3 08          =MAXMe CLRST      Ni memflg ni Flg
129 018E5 136          MXMC CD0EX
130 018E8 06          RSTK=C
131 018EA 7851         GOSUB r<rstk
132 018EE 704F         GOSUB VERTAP
133 018F2 110          A=R0              R0 = MS-16
134 018F5 DB          C=D A              ) D[X] = device address
135 018F7 06          RSTK=C              ) RSTK = D[X]
136 018F9 D2          C=0 A
137 018FB 320F1       LC(3) #1F0        1F0 = 512-16 = 496
138 01900 EA          A=A-C A            A[A] = MS-512
139 01902 8F00000     GOSBVL =D=AVMS
140 01909 D6          C=A A              C[A] = AVMEMS
141 0190B 8B3         ?C<D A            Y a-t-il assez de mem. ?
142 0190E 3B          GOYES memerr      Non: Erreur
143 01910 109         R1=C              Sauvegarde de MS-512
144 01913 135         D1=C              D1 = Math Stack - 512
145 01916 D0          A=0 A              A[A] = # enreg. È lire
146 01918 07          C=RSTK             ) Restitution de
147 0191A D7          D=C A              ) dev. add. dans D[X]
148 0191C 8E0000     GOSUBL =JUMPER ) Lecture de l'enreg. A[A]
149 01922 FF440     CON(5) READ# ) en commençant È D1 ^
150 01927 459       GOC Erreur
151 0192A 111         A=R1              A[A] = MS-512
152 0192D 87A         ?ST=1 memflg      Est-on entrÃ via MEMDIR ?
153 01930 13          GOYES MDIRXQ
154 01932 D2          C=0 A
155 01934 3103       LC(2) 48
156 01938 C2          C=C+A A
157 0193A 135         D1=C              ^ pointe ^ l. max support
158 0193D 24          P= 4              ) Calcul de
159 0193F 7121         GOSUB GETALR ) la longueur maximale
160 01943 7801         GOSUB getA* ) du support
161 01947 7401         GOSUB getA* ) de mÃmoire de masse
162 0194B 86B         ?ST=0 Flg      Est-on entrÃ via MAXMEM
163 0194E 60          GOYES MXM10
164 01950 7800         GOSUB MEM00      Autrement exÃcution de MEMM
165 01954 F0          MXM10 ASL A          ) multiplication de A[A]
166 01956 F0          ASL A              ) par 256 (dclg. d'1 octet)
167 01958 633F         GOTO SORTIE      prÃ retour È BASIC
168
169 0195C D8          MEM00 B=A A          B[A] = # enreg. maxi
170 0195E 111         A=R1
171 01961 131         MDIRXQ D1=A          D1^ MS-512

```

```

172 01964 17F          D1=D1+ 16      D1^ codage # Rec. Start DIR
173 01967 24          P=           4
174 01969 77F0        GOSUB  GETALR  => A[A]= 00002 dir. start
175 0196D E8          B=B-A  A       B[A] = # maxrec - vol.label
176 0196F 102         R2=A          R2= dir. start
177 01972 177         D1=D1+ 8      D1^ code long. directory
178 01975 24          P=           4
179 01977 79E0        GOSUB  GETALR  => A[A]= 0xxxx Dir.Len
180 0197B E8          B=B-A  A       B[A]=maxmem aprEs format.
181 0197D 87B         ?ST=1  Flg     Est-on entrÅ via MEMM
182 01980 40          GOYES  MEM10
183 01982 D6          C=A  A        Sauvegarde de DirLen dans C
184 01984 CC          MEM10 A=A-1  A      ) A => R2 = compteur
185 01986 122         AR2EX          ) A[A] = DIR START
186 01989 DC          ABEX  A        A[A]=Maxmem aprEs format.
187 0198B 103         R3=A          R3=compteur mem :tape
188 0198E 87B         ?ST=1  Flg     Est-on entrÅ via MEMM
189 01991 A0          GOYES  MEM20
190 01993 F2          CSL  A         ) Non: division de C[A] par
191 01995 81E         CSRB          ) 8 pour obtenir le max dir
192 01998 10B         R3=C          ) R3 compteur mem dir
193 0199B 7C00        MEM20 GOSUB  GET#FL Routine commune de comptage
194 0199F 113         A=R3          A= rÅsulat GET#FL
195 019A2 87B         ?ST=1  Flg     Est-on entrÅ via MEMM ?
196 019A5 00          RTNYES
197 019A7 64EE        GOTO  SORTIE  Retour de MEMM
198
199 019AB 111         GET#FL A=R1
200 019AE 131         D1=A          D1^ MS-512
201 019B1 D4          A=B  A        A[A] = # enregis. È lire
202 019B3 8E0000      GOSUBL =JUMPER
203 019B9 FF440       CON(5) READ#
204 019BE 4C1         GOC  ERReur
205 019C1 E5          B=B+1  A       B[A] = prochain # enr.
206 019C3 111         A=R1
207 019C6 131         D1=A          D1 ^ MS - 512
208 019C9 7210        GOSUB  GET#E   Routine de comptage des 8
209 *                  *                  entrÅes directory d'un enr.
210 019CD 112         A=R2          ) DÅcrÅmentation de R2
211 019D0 CC          A=A-1  A      ) # d'enregistrements du
212 019D2 400         RTNC          ) dir & retour si fin
213 019D5 102         R2=A          Autrement sauvegarde
214 019D8 52D         GONC  GET#FL  B.E.T
215
216 019DB 6D7E        ERReur GOTO  erreur
217
218 019DF 34FFFF0     GET#E  LC(5)  #FFFF  C[3:0]=FFFF, masque de com-
219 *                  *                  paraison du rideau de fin d
220 *                  *                  de directory
221 019E6 28          P=           8      8 = 8 entrÅes / enregist.
222 019E8 D0          A=0  A
223 019EA 0D          get#e P=P-1      Compteur
224 019EC 400         RTNC          Sortie si fin
225 019EF 17F         D1=D1+ 16
226 019F2 173         D1=D1+ 4      D1^ type de fichier
227 019F5 15B3        A=DAT1 4      A[3:0] =type de fichier
228 019F9 8A2         ?A=C  A       Est-ce le rideau (FFFF)

```

```
229 019FC 64          GOYES R3=OK   Oui: fin du s.prog.
230 019FE 8AC          ?A#0  A       A=0 => fichier purg  !
231 01A01 E0          GOYES R3-1
232 01A03 17B        incr  D1=D1+ 12   ) On passe   l'entr e
233 01A06 17F          D1=D1+ 16     ) suivante
234 01A09 17F        MEM30 D1=D1+ 16     )
235 01A0C 5DD          GONC  get#e   B.E.T
236
237 01A0F 87B        R3-1  ?ST=1 Flg   MEMM ??
238 01A12 D0          GOYES R3-M    Oui => R3- long fichier
239 01A14 113          A=R3         ) autrement R3
240 01A17 CC          A=A-1  A      ) -1 seulement pour
241 01A19 103          R3=A        ) MEMDIR
242 01A1C 56E          GONC  incr    B.E.T.
243
244 01A1F 17B        R3-M  D1=D1+ 12   D1^ long. du fich. en #rec.
245 01A22 80C4        C=P        4     Sauveg. pointeur # dir
246 01A26 06          RSTK=C
247 01A28 24          P=        4
248 01A2A 7630        GOSUB GETALR A[A] long. # rec. du fich.
249 01A2E 07          C=RSTK
250 01A30 80F4        CPEX      4     Restitution
251 01A34 12B          CR3EX     )
252 01A37 E2          C=C-A  A     ) R3 = R3 - # rec
253 01A39 12B          CR3EX     )
254 01A3C 177          D1=D1+  8     Ajustement
255 01A3F 59C          GONC  MEM30   B.E.T
256
257 01A42 07          R3=OK  C=RSTK   Fin pop et retour
258 01A44 01          RTN
259
260 *****
261 * Cette routine sauvegarde 5 niveaux de pile
262 *****
263
264 01A46 24          r<rstk P=      4
265 01A48 8D00000      GOVLNG =R<RSTK I N D I P E N S A B L E !!
266
267
268 *****
269 * Routine de calcul de la longueur maxi
270 * du support (multiplication)
271 *****
272
273 01A4F 102          getA*  R2=A
274 01A52 24          P=      4
275 01A54 7C00        GOSUB  GETALR
276 01A58 11A          C=R2
277 01A5B 8F00000      GOSBVL =A-MULT
278 01A62 01          RTN
279
280 *****
281 * GETALR: GET A Left-Right.
282 * Cette routine permet de charger dans le registre
283 * A n octets. Les octets sont d cal s vers la gau-
284 * che   chaque nouvel arrivant.
285 *
```

```
286      * Entrée: D1^ premier octet à charger dans A
287      *          P= Nombre d'octets à charger
288      *
289      * Sortie: A[2*(Pentrée):0] = octets, le dernier
290      *          étant toujours dans A[B].
291      *          P=0
292      *          D1 ^ après dernier octet
293      *
294      * Utilise : C[B], A[W], P , D1
295      *
296      *****
297
298 01A64 14F      GETALR C=DAT1 B
299 01A67 171          D1=D1+ 2
300 01A6A BF0          ASL      W
301 01A6D BF0          ASL      W
302 01A70 AEA          A=C      B
303 01A73 0D          P=P-1
304 01A75 880          ?P#      0
305 01A78 CE          GOYES    GETALR
306 01A7A 01          RTN
307
308 01A7C          END
```

A-MULT	Ext		-	277						
CHKMAS	Abs	16988	#0000425C	-	20	56				
CSRW5	Ext			-	74					
D=AVMS	Ext			-	139					
DEVPAR	Abs	7152	#00001BF0	-	19	42				
Dvnfnd	Abs	6187	#0000182B	-	37	50				
ENDTAP	Abs	17625	#000044D9	-	22	89				
ERReur	Abs	6619	#000019DB	-	216	204				
Erreur	Abs	6333	#000018BD	-	101	90	150			
FNRTN4	Ext			-	99					
Flg	Abs	11	#0000000B	-	27	111	120	162	181	188
					237					195
GDIRST	Abs	18499	#00004843	-	21	71				
GET#E	Abs	6623	#000019DF	-	218	208				
GET#FL	Abs	6571	#000019AB	-	199	193	214			
GETALR	Abs	6756	#00001A64	-	298	159	174	179	248	275
HDFLT	Ext			-	95					305
JUMPER	Ext			-	41	55	70	88	148	202
=MAXDe	Abs	6240	#00001860	-	66					
=MAXMe	Abs	6371	#000018E3	-	128					
MDIRXQ	Abs	6497	#00001961	-	171	153				
MEM00	Abs	6492	#0000195C	-	169	164				
MEM10	Abs	6532	#00001984	-	184	182				
MEM20	Abs	6555	#0000199B	-	193	189				
MEM30	Abs	6665	#00001A09	-	234	255				
=MEMDe	Abs	6359	#000018D7	-	119					
MEMERR	Ext			-	103					
=MEMMe	Abs	6347	#000018CB	-	110					
MXM10	Abs	6484	#00001954	-	165	163				
MXMC	Abs	6373	#000018E5	-	129	112	121			
PILERR	Ext			-	59					
R3-1	Abs	6671	#00001A0F	-	237	231				
R3-M	Abs	6687	#00001A1F	-	244	238				
R3=OK	Abs	6722	#00001A42	-	257	229				
R<RSTK	Ext			-	265					
READ#	Abs	17663	#000044FF	-	23	149	203			
RSTK<R	Ext			-	92					
SORTIE	Abs	6284	#0000188C	-	88	167	197			
VERTAP	Abs	6194	#00001832	-	41	69	132			
ePIL	Abs	2	#00000002	-	26	37				
erreur	Abs	6233	#00001859	-	59	39	47	72	101	216
get#e	Abs	6634	#000019EA	-	223	235				
getA*	Abs	6735	#00001A4F	-	273	160	161			
incrÅ	Abs	6659	#00001A03	-	232	242				
memerr	Abs	6337	#000018C1	-	103	142				
memflg	Abs	10	#0000000A	-	28	110	119	152		
r<rstk	Abs	6726	#00001A46	-	264	68	131			

Input Parameters

Source file name is mmlex.a

Listing file name is mmlex.l

Object file name is mmlex.o

Flags set on command line
None

Errors

None


```
1          TITLE JPC LEX: MODULE RW
2
3          * *****
4          * Direct mass storage record Read/Write module
5          *
6          * Keyword(s):
7          *   WREC, RREC
8          *   Note: this module also includes the =PILERR entry point
9          *         used by other modules dealing with HPIL.
10         * Author(s):
11         *   Michel Martinet
12         * History:
13         *   Published in JPC #45
14         *
15         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
16         * *****
17
18 01A7C          ABS      #1A7C
19
20         * HPIL ROM calls:
21         READ# EQU      #44FF
22         WRITE# EQU     #453F
23         GETPIL EQU     #6E0B
24         GETMBX EQU     #3B62
25         DEVPAR EQU     #1BF0
26         CHKMAS EQU     #425C
27         GETERR EQU     #6791
28
29         * local symbols:
30         Attn EQU      12
31         eABORT EQU    4
32         eNODEF EQU    3
33         ePIL EQU      2
34         eTAPE EQU     1
35         reclen EQU    512
36         sSTK EQU      7
37
38
39 01A7C 6AE0     ERReur GOTO  erreur
40
41         * *****
42         * WREC entry
43         * *****
44 01A80 00000          REL(5) =WRECD
45 01A85 00000          REL(5) =WRECP
46 01A8A 8F00000 =WRECe GOSBVL =EXPEX- ) DÅpose les 2 premiers
47 01A91 161          D0=D0+ 2 ) paramÉtres
48 01A94 8F00000          GOSBVL =EXPEXC ) de l'ordre
49 01A9B 161          D0=D0+ 2 ) WREC sur la M.S.
50 01A9E 137          CD1EX
51 01AA1 06           RSTK=C           RSTK = Math Stack
52 01AA3 847          ST=0   sSTK
53 01AA6 8E0000          GOSUBL =JUMPER
54 01AAC B0E60          CON(5) GETPIL   Cette routine permet
55         *           d'Åvaluer le spÅcificateur
56         *           d'appareil.
57 01AB1 4AC          GOC      ERReur
```

```

58 01AB4 07          C=RSTK
59 01AB6 135        D1=C          D1 = Math Stack
60 01AB9 8E0000    GOSUBL =JUMPER
61 01ABF 26B30    CON(5) GETMBX  Routine de recherche de
62                *          l'adresse de la MAILBOX de
63                *          la boucle d'acrite dans le
64                *          spAcificateur d'appareil.
65                *          Au retour: D0 ^ M.B.
66 01AC4 7680      GOSUB  VERT10  VArification de la prAsence
67                *          de la mAmoire de masse sur
68                *          la boucle.
69 01AC8 110        A=R0          ) Mise en place de D1 sur
70 01ACB 131        D1=A          ) le deuxiEme paramEtre de
71                *          ) l'ordre WREC
72 01ACE DB         C=D      A      Sauvegarde de l'adresse de
73 01AD0 06         RSTK=C      la mAmoire de masse dans
74                *          la pile de retour
75 01AD2 14B        A=DAT1 B      ) l'adresse d'Acriture est-
76                *          ) elle en HEXA ?
77 01AD5 31F0      LC(2)  15      )
78 01AD9 966        ?A#C  B      )
79 01ADC E0         GOYES  rndahx  Non => RNDAHX
80 01ADE 8F00000    GOSBVL =ADDRCK  VArification de la validitA
81                *          de l'adresse (2e param.)
82 01AE5 D9         C=B      A
83 01AE7 411        GOC      EXEC00
84 01AEA 8F00000    rndahx GOSBVL =RNDAHX
85 01AF1 504        GONC   ivarg
86 01AF4 17F        D1=D1+ 16
87 01AF7 D6         C=A      A
88
89 01AF9 06        EXEC00 RSTK=C      ) adresse dans la pile de
90                *          ) retour
91 01AFB 8F00000    GOSBVL =REVPOP  Renversement de la chaÑne
92                *          sur la Mathstack.
93 01B02 D2         C=0      A      ) La longueur de la chaÑne
94 01B04 32002      LC(3)  reclen  ) est-elle de 256 octets ?
95 01B09 8A6        ?A#C  A
96 01B0C 62         GOYES  ivarg      Si non => ERREUR
97 01B0E 07         C=RSTK      ) Restitution de l'adresse
98 01B10 DA         A=C      A      ) de l'enregist. dans A(A)
99 01B12 07         C=RSTK      ) Mise en place de
100                *          ) l'adresse de la mAmoire
101 01B14 D7        D=C      A      ) de masse dans D(X).
102                *
103 01B16 8E0000    GOSUBL =JUMPER      *****
104 01B1C F3540     CON(5) WRITE#      * E C R I T U R E *
105                *          *****
106 01B21 454       GOC      erreur  Au retour de la
107                *          routine, si Cy=1,
108                *          nous sommes en prAsence
109                *          d'une erreur.
110 01B24 8F00000    GOSBVL =COLLAP  Remise en place de MTHSTK
111 01B2B 8D00000    GOVLNG =NXTSTM  BASIC continu
112 01B32 8D00000    ivarg  GOVLNG =ARGERR
113
114                *****

```

```

115      * La routine VERTAP verifie si le paramÉtre
116      * pointÀ par D1 correspond bien È une mÅmoire
117      * de masse d'identificateur appareil 16.
118      *****
119 01B39 22      Dvnfnd P=      ePIL      Classe d'erreur 32 È 47
120 01B3B D2      C=0      A      Erreur 32.
121 01B3D 492     GOC      erreur   B.E.T.
122 01B40 8E0000  VERTAP  GOSUBL =JUMPER
123 01B46 0FB10   CON(5)  DEVPAR   Routine d'analyse du
124      *      spÅcificateur d' appareil
125      *      de la fonction. En sortie
126      *      D1 ^ Math Stack - 16 pour
127      *      un paramÉtre numÅrique.
128 01B4B 4B1     GOC      erreur   Si carry: ERREUR
129 01B4E 93B     VERT10  ?D=0    X      D[X] contient l'adresse du
130      *      pÅriphÅrique. Si D[X] = 0,
131 01B51 8E      GOYES   Dvnfnd   le pÅriphÅrique n'est pas
132      *      prÀsent. On renvoie
133      *      "Device Not Found"
134 01B53 137     CD1EX   ) Sauvegarde de D1 dans R0
135      *      ) pour le retour
136 01B56 108     R0=C    ) È BASIC (fin).
137 01B59 8E0000  GOSUBL =JUMPER  VÅrification de l'AID du
138 01B5F C5240   CON(5)  CHKMAS  pÅriphÅrique (16).
139 01B64 500     RTNNC   Cy=0 : OK, Cy=1 : ERREUR
140
141
142      *****
143      * Traitement des erreurs HPIL. P indique la
144      * classe d'erreur (mÅmoire de masse... etc.) et
145      * C[0] le numÅro de l'erreur. Le retour È la
146      * main loop se fait via BSERR.
147      *****
148 01B67      =PILERR
149 01B67 891     erreur ?P=      eTAPE
150 01B6A 93      GOYES  ERROR1
151 01B6C 892     ?P=      ePIL
152 01B6F 43      GOYES  ERROR1
153 01B71 884     ?P#      eABORT
154 01B74 92      GOYES  ERROR0
155 01B76 8E0000  GOSUBL =JUMPER
156 01B7C 26B30   CON(5)  GETMBX
157 01B81 7001   GOSUB  atnchk
158 01B85 471     GOC      ERROR0
159 01B88 8E0000  GOSUBL =JUMPER
160 01B8E 19760   CON(5)  GETERR
161 01B93 570     GONC    ERROR-
162 01B96 884     ?P#      eABORT
163 01B99 A0      GOYES  ERROR1
164 01B9B 24      ERROR-  P=      eABORT
165 01B9D 80C0   ERROR0  C=P     0
166 01BA1 23      P=      eNODEF
167 01BA3 80C1   ERROR1  C=P     1
168 01BA7 22      P=      2
169 01BA9 31FF    LCHEX   FF
170 01BAD DA      A=C     A
171 01BAF 20      P=      0

```

```

172 01BB1 8D00000 bserr GOVLNG =BSERR Envoyez l'erreur !
173 01BB8 4EA Erreur GOC erreur Ralonge
174
175
176
177 * *****
178 * RREC entry
179 * *****
180 01BBB CC22 NIBHEX CC22 Alpha ou un numÃrique pour
181 * les deux paramÃtres.
182 01BBF 136 =RRECe CD0EX ) Sauvegarde de D0 dans
183 01BC2 06 RSTK=C ) la pile de retour
184 01BC4 74B0 GOSUB r<rstk Sauvegarde de 5 niveaux
185 01BC8 747F GOSUB VERTAP VÃrification mÃm. de masse
186 01BCC 110 A=R0 R0 = MS-16
187 01BCF 131 D1=A
188 01BD2 17F D1=D1+ 16 D1 = M.S.
189 01BD5 DB C=D A ) D[X] = device address
190 01BD7 06 RSTK=C ) RSTK = D[X]
191 01BD9 14B A=DAT1 B L'adresse est-elle
192 01BDC 31F0 LC(2) 15 donnÃe sous forme de
193 01BE0 966 ?A#C B chaÃne ?
194 01BE3 C0 GOYES rndah2 Non => RNDAHX
195 01BE5 8F00000 GOSBVL =ADDRCK VÃrification validitÃ
196 01BEC 411 GOC EXEC10
197 01BEF 8F00000 rndah2 GOSBVL =RNDAHX
198 01BF6 5A7 GONC Ivarg
199 01BF9 17F D1=D1+ 16
200 01BFC D8 B=A A
201
202 01BFE 133 EXEC10 AD1EX
203 01C01 D2 C=0 A
204 01C03 32012 LC(3) (reclen)+16 ( 210H = 528D
205 * (Soit la longueur de
206 * ( l'enregistrement plus
207 * ( 16 quartets d'en-tÃte)
208 01C08 EA A=A-C A A[A] = MS-528
209 01C0A 8F00000 GOSBVL =D=AVMS
210 01C11 D6 C=A A C[A] = AVMEMS
211 01C13 8B3 ?C<D A Y a-t-il assez de mem. ?
212 01C16 F5 GOYES memerr Non: Erreur
213 01C18 109 R1=C Sauvegarde de MS-528
214 01C1B 135 D1=C D1 = Math Stack - 512
215 01C1E 17F D1=D1+ 16
216 01C21 D4 A=B A A[A] = # enreg. Ã lire
217 01C23 07 C=RSTK ) Restitution de
218 01C25 D7 D=C A ) dev. add. dans D[X]
219 *
220 01C27 8E0000 GOSUBL =JUMPER *****
221 01C2D FF440 CON(5) READ# * L E C T U R E *
222 * *****
223 01C32 458 GOC Erreur
224 01C35 111 A=R1 A[A] = MS-528
225 01C38 131 D1=A
226 01C3B AF2 C=0 W ) construction e l'en-tÃte
227 * ) de la chaÃne avant le
228 * ) retour Ã BASIC

```

```
229 01C3E 34F0002      LCHEX  2000F      ) C(W)= 000000000002000F
230 01C45 1557         DAT1=C W         O.K.
231 01C49 133         AD1EX
232 01C4C 101         R1=A
233 01C4F 24          P=      4        ) Restitution de 5 niveaux
234 01C51 8F00000     GOSBVL =RSTK<R  ) de pile
235 01C58 07          C=RSTK          ) puis de
236 01C5A 134         D0=C            ) D0 (4 niveaux & PC)
237 01C5D 119         C=R1            Restitution pointeur
238 01C60 135         D1=C            D1 = MS+528
239 01C63 8F00000     GOSBVL =REV$    On retourne l'ordre des
240                                *          caractÉres sur la M.Stack.
241 01C6A 8D00000     GOVLNG =EXPR    Retour È BASIC.
242 01C71 60CE        Ivarg  GOTO  ivarg
243 01C75 8D00000     memerr GOVLNG =MEMERR
244
245                                *****
246                                * ROUTINE R<RSTK
247                                * Cette routine sauvegarde 5 niveaux de pile
248                                *****
249 01C7C 24          r<rstk P=      4
250 01C7E 8D00000     GOVLNG =R<RSTK I N D I S P E N S A B L E !
251                                *          Pour l'exÃcution d'une
252                                *          fonction faisant appel È
253                                *          des routines HPIL.
254
255                                *****
256                                * ROUTINE ATNCHK
257                                * Cette routine a ÃtÃ pompÃe dans les IDS V, elle
258                                * n'est pas supportÃe. Elle permet la sortie
259                                * de l'erreur Aborted lorsque l'on arrÃte l'exÃcu-
260                                * tion des routines HPIL en appuyant deux fois sur
261                                * la touche ATTN.
262                                *****
263 01C85 86C          atnchk ?ST=0  Attn
264 01C88 62          GOYES  ATNCHc
265 01C8A 06          RSTK=C
266 01C8C 136         CD0EX
267 01C8F 1B00000     D0=(5) =ATNFLG
268 01C96 1564         C=DAT0 S
269 01C9A 134         D0=C
270 01C9D 07          C=RSTK
271 01C9F 94A         ?C=0  S
272 01CA2 C0          GOYES  ATNCHc
273 01CA4 B46         C=C+1  S
274 01CA7 460         GOC    ATNCHc
275 01CAA 24          P=      eABORT
276 01CAC 02          RTNSC
277 01CAE 03          ATNCHc RTNCC
278
279
280 01CB0              END
```


Input Parameters

Source file name is rw.a

Listing file name is rw.l

Object file name is rw.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE DDIR
2
3          * *****
4          * Directory listing commands
5          *
6          * Keyword(s):
7          *   DDIR, PDIR
8          * Author(s):
9          *   Jean-Jacques Dhenin
10         * History:
11         *   No source available (never published)
12         *   Source file rebuilt from JPC LEX disassembly
13         *   Bug fixes by Rodger Rosenbaum in rev X
14         *
15         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
16         * *****
17
18 01CB0          ABS #1CB0
19
20         * HPIL ROM calls:
21         FINDf+ EQU #46A6  FINDf+ renamed due to same mainframe entry
22         START EQU #07E8
23         GETDI! EQU #47D7
24         GETMBX EQU #3B62
25         UTLEND EQU #07CC
26         BLDCAT EQU #6300
27         NXTENT EQU #4A1E
28         SEEKRD EQU #62D8
29         ENDTAP EQU #44D9
30         GETDR+ EQU #47F9
31         GETERR EQU #6791
32
33
34         * *****
35         * DDIR entry
36         * *****
37 01CB0 00000          REL(5) =DDIRd
38 01CB5 00000          REL(5) =DDIRp
39 01CBA 31F0          =DDIRe LCHEX 0F
40 01CBE 501          GONC 01D46
41
42         * *****
43         * PDIR entry
44         * *****
45 01CC1 00000          REL(5) =DDIRd
46 01CC6 00000          REL(5) =DDIRp
47 01CCB 31F1          =PDIRe LCHEX 1F
48 01CCF 1F00000      01D46 D1=(5) =MLFFLG
49 01CD6 14D          DAT1=C B
50 01CD9 AF2          C=0 W
51 01CDC 1EBC8F        D1=(4) #F8CB
52 01CE2 1554          DAT1=C S
53 01CE6 1E0000        D1=(4) =MAINST
54 01CEC 147          C=DAT1 A
55 01CEF 1E5C8F        D1=(4) #F8C5
56 01CF5 145          DAT1=C A
57 01CF8 1DAC          D1=(2) #CA
```



```
58 01CFC 1554      DAT1=C S
59 01D00 14A      A=DAT0 B
60 01D03 8F00000  GOSBVL =EOLXCK
61 01D0A 560      GONC  o1D88
62 01D0D 6422     GOTO  o1F93
63 01D11 313F     o1D88 LCHEX F3
64 01D15 962      ?C=A B
65 01D18 67       GOYES o1DED
66 01D1A 318F     LCHEX F8
67 01D1E 966      ?C#A B
68 01D21 61       GOYES o1DAE
69 01D23 30E     LCHEX E
70 01D26 15D0     DAT1=C 1
71 01D2A 161     D0=D0+ 2
72 01D2D 505     GONC  o1DDD
73 01D30 8D00000  o1DA7 GOVLNG =BSERR
74 01D37 301     o1DAE LCHEX 1
75 01D3A 15D0     DAT1=C 1
76 01D3E 8F00000  GOSBVL =FSPECx
77 01D45 4AE     GOC  o1DA7
78 01D48 1F00000  D1=(5) =STMTD1
79 01D4F 1517     DAT1=A W
80 01D53 17F     D1=D1+ 16
81 01D56 AFB     C=D W
82 01D59 1557     DAT1=C W
83                *** changes ver. X
84 01D5D A46     C=C+C S
85 01D60 541     GONC  MASSOK
86 01D63 F2      CSL   A
87 01D65 C6      C=C+C A
88 01D67 C6      C=C+C A
89 01D69 4B0     GOC  MASSOK
90 01D6C 33F2FF  LCHEX FF2F      BAD DEVICE
91 01D72 5DB     GONC  o1DA7      BET
92 01D75         MASSOK
93                *** end of changes
94 01D75 174     D1=D1+ 5
95 01D78 118     C=R0
96 01D7B 145     DAT1=C A
97 01D7E 14A     o1DDD A=DAT0 B
98 01D81 313F     LCHEX F3
99 01D85 962      ?C=A B
100 01D88 60      GOYES o1DED
101 01D8A 68A0    GOTO  o1E92
102 01D8E 161     o1DED D0=D0+ 2
103 01D91 8F00000  GOSBVL =FSPECx
104 01D98 472     GOC  o1E1F
105 01D9B 1F0D8F2  D1=(5) #2F8D0
106 01DA2 AFB     C=D W
107 01DA5 1557     DAT1=C W
108 01DA9 17F     D1=D1+ 16
109 01DAC 1517     DAT1=A W
110 01DB0 8F00000  GOSBVL =FINDF+
111 01DB7 4C0     GOC  o1E23
112 01DBA 33B300  LCHEX 003B
113 01DC0 6F6F    o1E1F GOTO  o1DA7
114 01DC4 1FBC8F2  o1E23 D1=(5) #2F8CB
```

```
115 01DCB 301      LCHEX 1
116 01DCE 1550     DAT1=C P
117 01DD2 1D0D     D1=(2) #D0
118 01DD6 1577     C=DAT1 W
119 01DDA AF7      D=C W
120 01DDD D2       C=0 A
121 01DDF 3192     LCHEX 29
122 01DE3 8F00000  GOSBVL =CRETf+
123 01DEA 45D      GOC 01E1F
124 01DED 119      C=R1
125 01DF0 1F2D8F2  D1=(5) #2F8D2
126 01DF7 145      DAT1=C A
127 01DFA 1D0E     D1=(2) #E0
128 01DFE 1537     A=DAT1 W
129 01E02 135      D1=C
130 01E05 1517     DAT1=A W
131 01E09 17F      D1=D1+ 16
132 01E0C 35100004 LCHEX 400001
133 01E14 15D5     DAT1=C 6
134 01E18 17F      D1=D1+ 16
135 01E1B 174      D1=D1+ 5
136 01E1E D2       C=0 A
137 01E20 CE       C=C-1 A
138 01E22 15D3     DAT1=C 4
139 01E26 137      CD1EX
140 01E29 1FCC8F2  D1=(5) #2F8CC
141 01E30 145      DAT1=C A
142 01E33 1FAC8F2  01E92 D1=(5) #2F8CA
143 01E3A 15B0     A=DAT1 1
144 01E3E 908      ?A=0 P
145 01E41 56       GOYES 01F05
146 01E43 30E      LCHEX E
147 01E46 902      ?C=A P
148 01E49 D5       GOYES 01F05
149 01E4B AE0      A=0 B
150 01E4E 1510     DAT1=A P
151 01E52 1B00000  D0=(5) =F-R1-0
152 01E59 146      C=DAT0 A
153 01E5C 108      R0=C
154 01E5F 184      D0=D0- 5
155 01E62 1567     C=DAT0 W
156 01E66 AF7      D=C W
157 01E69 18F      D0=D0- 16
158 01E6C 1527     A=DAT0 W
159 01E70 308      LCHEX 8
160 01E73 816      CSRC
161 01E76 943      ?D=C S
162 01E79 92       GOYES 01F01
163 01E7B 8F00000  GOSBVL =FINDf+
164 01E82 5A4      GONC 01F2C
165 01E85 876      ?ST=1 6
166 01E88 85       GOYES 01F41
167 01E8A 94A      ?C=0 S
168 01E8D 35       GOYES 01F41
169 01E8F ACB      C=D S
170 01E92 A4E      C=C-1 S
171 01E95 4F3      GOC 01F34
```

```
172 01E98 A4E      C=C-1 S
173 01E9B 4E0      GOC o1F09
174 01E9E 6140     GOTO o1F41
175 01EA2 64D1     o1F01 GOTO o20D8
176 01EA6 6B80     o1F05 GOTO o1F93
177 01EAA B67      o1F09 D=D+1 B
178 01EAD 581      GONC o1F25
179 01EB0 1BAC8F2  D0=(5) #2F8CA
180 01EB7 1544     DAT0=C S
181 01EBB 7B21     GOSUB o204B
182 01EBF 5D5      GONC o1F7E
183 01EC2 65A1     o1F21 GOTO o20C9
184 01EC6 7471     o1F25 GOSUB o209F
185 01ECA 47F      GOC o1F21
186 01ECD 6750     o1F2C GOTO o1F86
187 01ED1 6730     o1F30 GOTO o1F6A
188 01ED5 6C50     o1F34 GOTO o1F93
189                * RTNCC * change: removed in ver. X
190 01ED9 8D00000  o1F3A GOVLNG =NXTSTM
191 01EE0 8D00000  o1F41 GOVLNG =BSERR
192 01EE7 1FAC8F2  o1F48 D1=(5) #2F8CA
193 01EEE 1574     C=DAT1 S
194 01EF2 94A      ?C=0 S
195 01EF5 41       GOYES o1F6A
196 01EF7 B46      C=C+1 S
197 01EFA 4B1      GOC o1F77
198 01EFD 1554     DAT1=C S
199 01F01 75E0     GOSUB o204B
200 01F05 6410     GOTO o1F7B
201 01F09 8F00000  o1F6A GOSBVL =NOSCRL
202 01F10 8C7CFF   GOLONG o1F3A
203 01F16 7741     o1F77 GOSUB o20C2
204 01F1A 46B      o1F7B GOC o1F30
205 01F1D 14B      o1F7E A=DAT1 B
206 01F20 968      ?A=0 B
207 01F23 3F       GOYES o1F77
208 01F25 137      o1F86 CD1EX
209 01F28 1F00000  D1=(5) =TRFMBF
210 01F2F 145      DAT1=C A
211 01F32 1F00000  o1F93 D1=(5) =TRFMBF
212 01F39 143      A=DAT1 A
213 01F3C 130      D0=A
214 01F3F 14E      C=DAT0 B
215 01F42 96A      ?C=0 B
216 01F45 2A       GOYES o1F48
217 01F47 1D5C     D1=(2) #C5
218 01F4B 141      o1FAC DAT1=A A
219 01F4E 131      D1=A
220 01F51 840      ST=0 0
221 01F54 8F00000  GOSEVL =CAT$20
222 01F5B 8F00000  GOSEVL =OBCOLL
223 01F62 1F00000  D1=(5) =AVMEMS
224 01F69 147      C=DAT1 A
225 01F6C 181      D0=D0- 2
226 01F6F 132      AD0EX
227 01F72 141      DAT1=A A
228 01F75 1C4      D1=D1- 5
```

```
229 01F78 145      DAT1=C A
230 01F7B 1F7D8F2  D1=(5) #2F8D7
231 01F82 119      C=R1
232 01F85 1557     DAT1=C W
233 01F89 8EF140   GOSUBL o243D
234 01F8F 1F7D8F2  D1=(5) #2F8D7
235 01F96 1577     C=DAT1 W
236 01F9A 109      R1=C
237 01F9D 7330     GOSUB o2035
238 01FA1 5E2      GONC o2031
239 01FA4 1F00000  D1=(5) =TRFMBF
240 01FAB 143      A=DAT1 A
241 01FAE D2       C=0 A
242 01FB0 3102     LCHEX 20
243 01FB4 CA       A=A+C A
244 01FB6 130      D0=A
245 01FB9 142      A=DAT0 A
246 01FBC 136      CD0EX
247 01FBF CA       A=A+C A
248 01FC1 130      D0=A
249 01FC4 14E      C=DAT0 B
250 01FC7 96E      ?C#0 B
251 01FCA 18       GOYES o1FAC
252 01FCC 6A1F     GOTO o1F48
253 01FD0 683F     o2031 GOTO o1F6A
254 01FD4 136      o2035 CD0EX
255 01FD7 1B00000  D0=(5) =ATNFLG
256 01FDE 1564     C=DAT0 S
257 01FE2 134      D0=C
258 01FE5 A4E      C=C-1 S
259 01FE8 01       RTN
260 01FEA 20       o204B P= 0
261 01FEC 31EF     LCHEX FE
262 01FF0 8F00000  GOSEVL =CNFFND
263 01FF7 938      ?A=0 X
264 01FFA 00       RTNYES
265 01FFC 170      D1=D1+ 1
266 01FFF AB8      B=A X
267 02002 15F7     C=DAT1 8
268 02006 AF7      D=C W
269 02009 172      D1=D1+ 3
270 0200C 15F2     C=DAT1 3
271 02010 172      D1=D1+ 3
272 02013 1574     C=DAT1 S
273 02017 B46      C=C+1 S
274 0201A AC7      D=C S
275 0201D 173      D1=D1+ 4
276 02020 F2       CSL A
277 02022 F2       CSL A
278 02024 308     LCHEX 8
279 02027 133     AD1EX
280 0202A 135     D1=C
281 0202D BF0     ASL W
282 02030 BF0     ASL W
283 02033 BF0     ASL W
284 02036 AB4     A=B X
285 02039 101     R1=A
```

```
286 0203C 03          RTNCC
287 0203E A6F        o209F D=D-1 B
288 02041 DB          C=D A
289 02043 10B         R3=C
290 02046 70AF        GOSUB o204B
291 0204A 400         RTNC
292 0204D 590         GONC o20B8
293 02050 7D00        o20B1 GOSUB o20C2
294 02054 400         RTNC
295 02057 11B        o20B8 C=R3
296 0205A 967         ?D#C B
297 0205D 3F          GOYES o20B1
298 0205F 03          RTNCC
299 02061 8D00000    o20C2 GOVLNG =ROMFND
300 02068 3104        o20C9 LCHEX 40
301 0206C 8D00000    GOVLNG =MFERR
302 02073 69C2        o20D4 GOTO o23CC
303 02077 DB          o20D8 C=D A
304 02079 1F00000    D1=(5) =STMTD1
305 02080 145         DAT1=C A
306 02083 8A8         ?A=0 A
307 02086 41          GOYES o20FB
308 02088 8E0000     GOSUBL =JUMPER
309 0208E 6A640       CON(5) FINDf+
310 02093 4FD         GOC o20D4
311 02096 6320        GOTO o211B
312 0209A 8E0000     o20FB GOSUBL =JUMPER
313 020A0 8E700       CON(5) START
314 020A5 560         GONC o210D
315 020A8 6492        o2109 GOTO o23CC
316 020AC 8E0000     o210D GOSUBL =JUMPER
317 020B2 7D740       CON(5) GETDI!
318 020B7 40F         GOC o2109
319 020BA AFB         o211B C=D W
320 020BD 7DE1        GOSUB o230F
321 020C1 F2          CSL A
322 020C3 81E         CSRB
323 020C6 CE          C=C-1 A
324 020C8 74F1        GOSUB o2321
325 020CC D9          C=B A
326 020CE 75F1        GOSUB o2328
327 020D2 10A         R2=C
328 020D5 10B         R3=C
329 020D8 8F00000    GOSBVL =R<RST2
330 020DF 8E0000     GOSUBL =JUMPER
331 020E5 26B30       CON(5) GETMBX
332 020EA 7FA1        GOSUB o22FE
333 020EE 7D21        GOSUB o2280
334 020F2 80CE        C=P 14
335 020F6 AC2         C=0 S
336 020F9 550         GONC o2160
337 020FC B46         C=C+1 S
338 020FF 7BC1        o2160 GOSUB o235D
339 02103 8F00000    GOSBVL =RST2<R
340 0210A 8E0000     GOSUBL =JUMPER
341 02110 26B30       CON(5) GETMBX
342 02115 72D1        GOSUB o237A
```

```
343 02119 80DE      P=C 14
344 0211D 94E      ?C#0 S
345 02120 D3       GOYES o21BE
346 02122 8AA      ?C=0 A
347 02125 43       GOYES o21BA
348 02127 11B      C=R3
349 0212A 7081     GOSUB o230F
350 0212E 7381     GOSUB o2316
351 02132 10B      R3=C
352 02135 76C1     GOSUB o238E
353 02139 D5       B=C A
354 0213B 1F00000  D1=(5) =AVMEMS
355 02142 143      A=DAT1 A
356 02145 174      D1=D1+ 5
357 02148 147      C=DAT1 A
358 0214B E9       C=C-B A
359 0214D 8BA      ?C>=A A
360 02150 51       GOYES o21C6
361 02152 8D00000  GOVLNG =MEMERR
362 02159 6CE0     o21BA GOTO o22A7
363 0215D 6FD1     o21BE GOTO o23CC
364 02161 66F0     o21C2 GOTO o22B9
365 02165 145     o21C6 DAT1=C A
366 02168 786E     o21C9 GOSUB o2035
367 0216C 54F      GONC o21C2
368 0216F 8E0000  GOSUBL =JUMPER
369 02175 CC700    CON(5) UTLEND
370 0217A 8E0000  GOSUBL =JUMPER
371 02180 00360    CON(5) BLDCAT
372 02185 11B      C=R3
373 02188 1F7D8F2  D1=(5) #2F8D7
374 0218F 1557     DAT1=C W
375 02193 7271     GOSUB o2398
376 02197 1F7D8F2  D1=(5) #2F8D7
377 0219E 1577     C=DAT1 W
378 021A2 10B      R3=C
379 021A5 1F00000  D1=(5) =STMTD1
380 021AC 147      C=DAT1 A
381 021AF D7       D=C A
382 021B1 8E0000  GOSUBL =JUMPER
383 021B7 8E700    CON(5) START
384 021BC 40A      GOC o21BE
385 021BF 11B      C=R3
386 021C2 10A      R2=C
387 021C5 7B00     GOSUB o2235
388 021C9 439      GOC o21BE
389 021CC 8AE      ?C#0 A
390 021CF 99       GOYES o21C9
391 021D1 547      GONC o22A7
392 021D4 8AA      o2235 ?C=0 A
393 021D7 36       GOYES o229B
394 021D9 94A      ?C=0 S
395 021DC 22       GOYES o225F
396 021DE 7ED0     GOSUB o2321
397 021E2 8E0000  GOSUBL =JUMPER
398 021E8 E1A40    CON(5) NXTENT
399 021ED 8E0000  GOSUBL =JUMPER
```

```
400 021F3 8D260      CON(5) SEEKRD
401 021F8 400        RTNC
402 021FB 11A        o225C C=R2
403 021FE 8AA        o225F ?C=0 A
404 02201 92         GOYES o228B
405 02203 CE         C=C-1 A
406 02205 75A0       GOSUB o230F
407 02209 7F90       GOSUB o230D
408 0220D 8E0000     GOSUBL =JUMPER
409 02213 E1A40      CON(5) NXTENT
410 02218 7460       GOSUB o22E1
411 0221C 400        RTNC
412 0221F 91A        o2280 ?C=0 WP
413 02222 9D         GOYES o225C
414 02224 B16        C=C+1 WP
415 02227 541        GONC o229D
416 0222A 11B        o228B C=R3
417 0222D D2         C=0 A
418 0222F AC2        C=0 S
419 02232 A4E        C=C-1 S
420 02235 10B        R3=C
421 02238 D2         C=0 A
422 0223A 03         o229B RTNCC
423 0223C 11A        o229D C=R2
424 0223F 10B        R3=C
425 02242 E6         C=C+1 A
426 02244 03         RTNCC
427 02246 D0         o22A7 A=0 A
428 02248 1F00000    D1=(5) =NEEDSC
429 0224F 1590       DAT1=A 1
430 02253 8AA        ?C=0 A
431 02256 51         GOYES o22CC
432 02258 1F00000    o22B9 D1=(5) =AVMEME
433 0225F 143        A=DAT1 A
434 02262 7990       GOSUB o238E
435 02266 CA         A=A+C A
436 02268 141        DAT1=A A
437 0226B 8E0000     o22CC GOSUBL =JUMPER
438 02271 9D440      CON(5) ENDTAP
439 02276 8C16CF     GOLONG o1F3A
440 0227C 60EE       GOTO o21BE
441 02280 DA         o22E1 A=C A
442 02282 7140       GOSUB o2328
443 02286 AC2        C=0 S
444 02289 10A        R2=C
445 0228C 814        ASRC
446 0228F 8E0000     GOSUBL =JUMPER
447 02295 9F740      CON(5) GETDR+
448 0229A 400        RTNC
449 0229D 1F519F2    o22FE D1=(5) #2F915
450 022A4 15F3       C=DAT1 4
451 022A8 23         P= 3
452 022AA 03         RTNCC
453 022AC E6         o230D C=C+1 A
454 022AE 8D00000    o230F GOVLNG =CSRC5
455 022B5 D2         o2316 C=0 A
456 022B7 E6         C=C+1 A
```

```
457 022B9 8D00000 GOVLNG =CSLC5
458 022C0 8D00000 o2321 GOVLNG =CSLC6
459 022C7 8D00000 o2328 GOVLNG =CSRC6
460 *** changes: removed in ver. X:
461 * GOVLNG =OUTELA
462 * GOVLNG =CRLFSD
463 * GOVLNG =CRLFND
464 * GOVLNG =FINDA
465 * P= 4
466 * GOVLNG =R<RSTK
467 * P= 4
468 * GOVLNG =RSTK<R
469 *** end of changes
470 022CE 136 o235D CD0EX
471 022D1 06 RSTK=C
472 022D3 136 CD0EX
473 022D6 1B00000 D0=(5) =F-R1-0
474 022DD 1547 DAT0=C W
475 022E1 136 o2370 CD0EX
476 022E4 07 C=RSTK
477 022E6 136 CD0EX
478 022E9 01 RTN
479 022EB 136 o237A CD0EX
480 022EE 06 RSTK=C
481 022F0 1B00000 D0=(5) =F-R1-0
482 022F7 1567 C=DAT0 W
483 022FB 65EF GOTO o2370
484 022FF 20 o238E P= 0
485 02301 D2 C=0 A
486 02303 3105 LCHEX 50
487 02307 03 RTNCC
488 02309 8F00000 o2398 GOSBVL =OBCOLL
489 02310 1F00000 D1=(5) =AVMEME
490 02317 143 A=DAT1 A
491 0231A D2 C=0 A
492 0231C 3105 LCHEX 50
493 02320 D5 B=C A
494 02322 1C9 D1=D1- 10
495 02325 147 C=DAT1 A
496 02328 C9 C=C+B A
497 0232A 174 D1=D1+ 5
498 0232D 145 DAT1=C A
499 02330 E9 C=C-B A
500 02332 8F00000 GOSBVL =MOVE*M
501 02339 6470 GOTO o243D
502 0233D 890 o23CC ?P= 0
503 02340 C5 GOYES o242B
504 02342 891 ?P= 1
505 02345 75 GOYES o242B
506 02347 892 ?P= 2
507 0234A 25 GOYES o242B
508 0234C 884 ?P# 4
509 0234F 74 GOYES o2425
510 02351 8E0000 GOSUBL =JUMPER
511 02357 26B30 CON(5) GETMBX
512 0235C 86C ?ST=0 12
513 0235F 22 GOYES o2410
```



```
514 02361 06          RSTK=C
515 02363 136         CD0EX
516 02366 1B00000     D0=(5) =ATNFLG
517 0236D 1564        C=DAT0 S
518 02371 134         D0=C
519 02374 07          C=RSTK
520 02376 94A         ?C=0 S
521 02379 80          GOYES o2410
522 0237B B46         C=C+1 S
523 0237E 551         GONC o2423
524 02381 8E0000     o2410 GOSUBL =JUMPER
525 02387 19760       CON(5) GETERR
526 0238C 570         GONC o2423
527 0238F 884         ?P# 4
528 02392 A0          GOYES o242B
529 02394 24          o2423 P= 4
530 02396 80C0       o2425 C=P 0
531 0239A 23          P= 3
532 0239C 80C1       o242B C=P 1
533 023A0 22          P= 2
534 023A2 31FF        LCHEX FF
535 023A6 20          P= 0
536 023A8 8C63BF     GOLONG o1F41
537 023AE 1BBC8F2    o243D D0=(5) #2F8CB
538 023B5 1524        A=DAT0 S
539 023B9 94C        ?A#0 S
540 023BC 05          GOYES o248D
541 023BE 1B00000     D0=(5) =MLFFLG
542 023C5 30F         LCHEX F
543 023C8 1540        DAT0=C P
544 023CC 8F00000     GOSBVL =CKINFO
545 023D3 1F00000     D1=(5) =AVMEMS
546 023DA 143         A=DAT1 A
547 023DD 1C4         D1=D1- 5
548 023E0 147         C=DAT1 A
549 023E3 EA          A=A-C A
550 023E5 AF1         B=0 W
551 023E8 D8          B=A A
552 023EA 81D         BSRB
553 023ED 854         ST=1 4
554 023F0 8F00000     GOSEVL =SNDWD+
555                    *** change: added in ver. X
556 023F7 1F00000     D1=(5) =CURSOR
557 023FE 3110        LCHEX 01
558 02402 14D         DAT1=C B
559                    *** end of change
560 02405 8D00000     GOVLNG =SENDEL
561 0240C 1B00000     o248D D0=(5) =AVMEMS
562 02413 142         A=DAT0 A
563 02416 131         D1=A
564 02419 173         D1=D1+ 4
565 0241C 184         D0=D0- 5
566 0241F 146         C=DAT0 A
567 02422 8F00000     GOSBVL =MOVED2
568 02429 1B00000     D0=(5) =AVMEMS
569 02430 AF0         A=0 W
570 02433 142         A=DAT0 A
```

```
571 02436 184      D0=D0- 5
572 02439 146      C=DAT0 A
573 0243C EA       A=A-C A
574 0243E 81C      ASRB
575 02441 D8       B=A A
576 02443 8F00000  GOSBVL =SWPBYT
577 0244A 146      C=DAT0 A
578 0244D 135      D1=C
579 02450 1593     DAT1=A 4
580 02454 822      SB=0
581 02457 81D      BSRB
582 0245A D0       A=0 A
583 0245C 832      ?SB=0
584 0245F 40       GOYES 024E4
585 02461 E4       A=A+1 A
586 02463 D2       024E4 C=0 A
587 02465 304      LCHEX 4
588 02468 C4       A=A+A A
589 0246A CA       A=A+C A
590 0246C 164      D0=D0+ 5
591 0246F 146      C=DAT0 A
592 02472 C2       C=C+A A
593 02474 144      DAT0=C A
594 02477 D2       C=0 A
595 02479 10B      R3=C
596 0247C 1B2D8F2  D0=(5) #2F8D2
597 02483 146      C=DAT0 A
598 02486 19CC     D0=(2) #CC
599 0248A 142      A=DAT0 A
600 0248D 8F00000  GOSBVL =RPLLIN
601 02494 4E0      GOC 02524
602 02497 1BCC8F2  D0=(5) #2F8CC
603 0249E 140      DAT0=A A
604 024A1 01       RTN
605 024A3 8CB3AF   02524 GOLONG 01F41
606
607
608 024A9          END
```

ATNFLG	Ext		-	255	516					
AVMEME	Ext		-	432	489					
AVMEMS	Ext		-	223	354	545	561	568		
BLDCAT	Abs	25344	#00006300	-	26	371				
BSERR	Ext		-	73	191					
CAT\$20	Ext		-	221						
CKINFO	Ext		-	544						
CNFFND	Ext		-	262						
CRETf+	Ext		-	122						
CSLC5	Ext		-	457						
CSLC6	Ext		-	458						
CSRC5	Ext		-	454						
CSRC6	Ext		-	459						
CURS0R	Ext		-	556						
DDIRd	Ext		-	37	45					
=DDIRe	Abs	7354	#00001CBA	-	39					
DDIRp	Ext		-	38	46					
ENDTAP	Abs	17625	#000044D9	-	29	438				
EOLXCK	Ext		-	60						
F-R1-0	Ext		-	151	473	481				
FINDF+	Ext		-	110	163					
FINDF+	Abs	18086	#000046A6	-	21	309				
FSPECx	Ext		-	76	103					
GETDI!	Abs	18391	#000047D7	-	23	317				
GETDR+	Abs	18425	#000047F9	-	30	447				
GETERR	Abs	26513	#00006791	-	31	525				
GETMBX	Abs	15202	#00003B62	-	24	331	341	511		
JUMPER	Ext		-	308	312	316	330	340	368	370
				382	397	399	408	437	446	510
				524						
MAINST	Ext		-	53						
MASSOK	Abs	7541	#00001D75	-	92	85	89			
MEMERR	Ext		-	361						
MFERR	Ext		-	301						
MLFFLG	Ext		-	48	541					
MOVE*M	Ext		-	500						
MOVED2	Ext		-	567						
NEEDSC	Ext		-	428						
NOSCRL	Ext		-	201						
NXTENT	Abs	18974	#00004A1E	-	27	398	409			
NXTSTM	Ext		-	190						
OBCOLL	Ext		-	222	488					
=PDIRe	Abs	7371	#00001CCB	-	47					
R<RST2	Ext		-	329						
ROMFND	Ext		-	299						
RPLLIN	Ext		-	600						
RST2<R	Ext		-	339						
SEEKRD	Abs	25304	#000062D8	-	28	400				
SENDEL	Ext		-	560						
SNDWD+	Ext		-	554						
START	Abs	2024	#000007E8	-	22	313	383			
STMTD1	Ext		-	78	304	379				
SWPBYT	Ext		-	576						
TRFMBF	Ext		-	209	211	239				
UTLEND	Abs	1996	#000007CC	-	25	369				
o1D46	Abs	7375	#00001CCF	-	48	40				
o1D88	Abs	7441	#00001D11	-	63	61				

o1DA7	Abs	7472	#00001D30	-	73	77	91	113				
o1DAE	Abs	7479	#00001D37	-	74	68						
o1DDD	Abs	7550	#00001D7E	-	97	72						
o1DED	Abs	7566	#00001D8E	-	102	65	100					
o1E1F	Abs	7616	#00001DC0	-	113	104	123					
o1E23	Abs	7620	#00001DC4	-	114	111						
o1E92	Abs	7731	#00001E33	-	142	101						
o1F01	Abs	7842	#00001EA2	-	175	162						
o1F05	Abs	7846	#00001EA6	-	176	145	148					
o1F09	Abs	7850	#00001EAA	-	177	173						
o1F21	Abs	7874	#00001EC2	-	183	185						
o1F25	Abs	7878	#00001EC6	-	184	178						
o1F2C	Abs	7885	#00001ECD	-	186	164						
o1F30	Abs	7889	#00001ED1	-	187	204						
o1F34	Abs	7893	#00001ED5	-	188	171						
o1F3A	Abs	7897	#00001ED9	-	190	202	439					
o1F41	Abs	7904	#00001EE0	-	191	166	168	174	536	605		
o1F48	Abs	7911	#00001EE7	-	192	216	252					
o1F6A	Abs	7945	#00001F09	-	201	187	195	253				
o1F77	Abs	7958	#00001F16	-	203	197	207					
o1F7B	Abs	7962	#00001F1A	-	204	200						
o1F7E	Abs	7965	#00001F1D	-	205	182						
o1F86	Abs	7973	#00001F25	-	208	186						
o1F93	Abs	7986	#00001F32	-	211	62	176	188				
o1FAC	Abs	8011	#00001F4B	-	218	251						
o2031	Abs	8144	#00001FD0	-	253	238						
o2035	Abs	8148	#00001FD4	-	254	237	366					
o204B	Abs	8170	#00001FEA	-	260	181	199	290				
o209F	Abs	8254	#0000203E	-	287	184						
o20B1	Abs	8272	#00002050	-	293	297						
o20B8	Abs	8279	#00002057	-	295	292						
o20C2	Abs	8289	#00002061	-	299	203	293					
o20C9	Abs	8296	#00002068	-	300	183						
o20D4	Abs	8307	#00002073	-	302	310						
o20D8	Abs	8311	#00002077	-	303	175						
o20FB	Abs	8346	#0000209A	-	312	307						
o2109	Abs	8360	#000020A8	-	315	318						
o210D	Abs	8364	#000020AC	-	316	314						
o211B	Abs	8378	#000020BA	-	319	311						
o2160	Abs	8447	#000020FF	-	338	336						
o21BA	Abs	8537	#00002159	-	362	347						
o21BE	Abs	8541	#0000215D	-	363	345	384	388	440			
o21C2	Abs	8545	#00002161	-	364	367						
o21C6	Abs	8549	#00002165	-	365	360						
o21C9	Abs	8552	#00002168	-	366	390						
o2235	Abs	8660	#000021D4	-	392	387						
o225C	Abs	8699	#000021FB	-	402	413						
o225F	Abs	8702	#000021FE	-	403	395						
o2280	Abs	8735	#0000221F	-	412	333						
o228B	Abs	8746	#0000222A	-	416	404						
o229B	Abs	8762	#0000223A	-	422	393						
o229D	Abs	8764	#0000223C	-	423	415						
o22A7	Abs	8774	#00002246	-	427	362	391					
o22B9	Abs	8792	#00002258	-	432	364						
o22CC	Abs	8811	#0000226B	-	437	431						
o22E1	Abs	8832	#00002280	-	441	410						
o22FE	Abs	8861	#0000229D	-	449	332						

o230D	Abs	8876	#000022AC -	453	407			
o230F	Abs	8878	#000022AE -	454	320	349	406	
o2316	Abs	8885	#000022B5 -	455	350			
o2321	Abs	8896	#000022C0 -	458	324	396		
o2328	Abs	8903	#000022C7 -	459	326	442		
o235D	Abs	8910	#000022CE -	470	338			
o2370	Abs	8929	#000022E1 -	475	483			
o237A	Abs	8939	#000022EB -	479	342			
o238E	Abs	8959	#000022FF -	484	352	434		
o2398	Abs	8969	#00002309 -	488	375			
o23CC	Abs	9021	#0000233D -	502	302	315	363	
o2410	Abs	9089	#00002381 -	524	513	521		
o2423	Abs	9108	#00002394 -	529	523	526		
o2425	Abs	9110	#00002396 -	530	509			
o242B	Abs	9116	#0000239C -	532	503	505	507	528
o243D	Abs	9134	#000023AE -	537	233	501		
o248D	Abs	9228	#0000240C -	561	540			
o24E4	Abs	9315	#00002463 -	586	584			
o2524	Abs	9379	#000024A3 -	605	601			

Input Parameters

Source file name is ddir.a

Listing file name is ddir.l

Object file name is ddir.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE FILES
2
3          * *****
4          * FILESIZE module
5          *
6          * Keyword(s):
7          *   FILESIZE
8          * Author(s):
9          * History:
10         *   Initial version by Henri Kudelski published in JPC #23
11         *   Code rewritten at some stage, unknown author
12         *   Source file rebuilt from JPC LEX desassembly
13         *
14         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
15         * *****
16
17 024A9          ABS      #24A9
18
19         * HPIL ROM calls:
20         FINDf+ EQU #46A6  FINDf+ renamed due to same mainframe entry
21
22 024A9 8D00000 ave=d1 GOVLNG =AVE=D1
23
24
25         * *****
26         * FILESIZE entry
27         * *****
28 024B0 411          NIBHEX 411
29         * o2534
30 024B3 136         =FILESe CD0EX
31 024B6 06          RSTK=C
32 024B8 24          P=      4
33 024BA 8F00000     GOSBVL =R<RSTK
34 024C1 74EF        GOSUB  ave=d1
35 024C5 8F00000     GOSBVL =FILXQ$
36 024CC 562         GONC   o2574
37 024CF 76DF        GOSUB  ave=d1
38 024D3 8F00000     GOSBVL =FINDF
39 024DA 455         GOC   o25B1
40 024DD 17F         D1=D1+ 16
41 024E0 17F         D1=D1+ 16
42 024E3 AF0         A=0 W
43 024E6 143         A=DAT1 A
44 024E9 AF2         C=0 W
45 024EC 3112        LCHEX 21
46 024F0 5A0         GONC   o257C
47
48 024F3 6340        o2574 GOTO o25B8
49 024F7 3162        o2578 LCHEX 26
50 024FB A72         o257C C=C+A W
51 024FE 81E         CSRB
52 02501 8F00000     o2582 GOSBVL =HXDCW
53 02508 8F00000     GOSBVL =FLOAT
54 0250F 04          o2590 SETHEX
55 02511 24          P= 4
56 02513 8F00000     GOSBVL =RSTK<R
57 0251A 07          C=RSTK
```

```
58 0251C 136      CDOEX
59 0251F 8F00000  GOSBVL =D1MSTK
60 02526 AF6      C=A W
61 02529 8D00000  GOVLNG =FNRTN1
62
63 02530 AF0      o25B1 A=0 W
64 02533 6BDF     GOTO o2590
65
66 02537 330202  o25B8 LCHEX 2020
67 0253D 108     R0=C
68 02540 8F00000  GOSBVL =POLL
69 02547 50      CON(2) #05
70 02549 4D0     GOC bserr
71 0254C 831     ?XM=0
72 0254F 82      GOYES o25F8
73 02551 33A300  LCHEX 003A
74 02557 8D00000  bserr GOVLNG =BSERR
75
76
77 0255E 881     error ?P# 1
78 02561 01      GOYES PILerr
79 02563 AEA     A=C B
80 02566 BE0     ASL B
81 02569 306     LCHEX 6
82 0256C 902     ?C=A P
83 0256F 1C      GOYES o25B1
84 02571 8C0000  PILerr GOLONG =PILERR  HPIL error
85
86 02577 101     o25F8 R1=A
87 0257A 8F00000  GOSBVL =D1MSTK
88 02581 8F00000  GOSBVL =POP1S
89 02588 137     CD1EX
90 0258B C2      C=C+A A
91 0258D 135     D1=C
92 02590 751F    GOSUB ave=d1
93 02594 111     A=R1
94 02597 8E0000  GOSUBL =JUMPER
95 0259D 6A640   CON(5) FINDf+
96 025A2 4BB     GOC error
97 025A5 D7      D=C A
98 025A7 D0      A=0 A
99 025A9 14B     A=DAT1 B
100 025AC F0     ASL A
101 025AE F0     ASL A
102 025B0 171    D1=D1+ 2
103 025B3 14B     A=DAT1 B
104 025B6 8F00000  GOSBVL =FTYPF#
105 025BD 20     P= 0
106 025BF 136    CDOEX
107 025C2 1D93   D1=(2) #39
108 025C6 AF0    A=0 W
109 025C9 15B7   A=DAT1 8
110 025CD AF2    C=0 W
111 025D0 1560   C=DAT0 P
112 025D4 A0E    C=C-1 P
113 025D7 483    GOC o2691
114 025DA A0E    C=C-1 P
```



```
115 025DD 463      GOC o2695
116 025E0 A0E      C=C-1 P
117 025E3 4C1      GOC o2681
118 025E6 DB       C=D A
119 025E8 AF0      A=0 W
120 025EB DA       A=C A
121 025ED F0       ASL A
122 025EF BF0      ASL W
123 025F2 3431000  LCHEX 00013
124 025F9 A72      C=C+A W
125 025FC 640F     o267D GOTO o2582
126 02600 F0       o2681 ASL A
127 02602 AD0      A=0 M
128 02605 F0       ASL A
129 02607 171      D1=D1+ 2
130 0260A 14B      A=DAT1 B
131 0260D BF0      ASL W
132 02610 66EE     o2691 GOTO o2578
133 02614 23       o2695 P= 3
134 02616 A96      C=A WP
135 02619 BF4      ASR W
136 0261C BF4      ASR W
137 0261F BF4      ASR W
138 02622 F4       ASR A
139 02624 20       P= 0
140 02626 8F00000  GOSBVL =MPY
141 0262D AF2      C=0 W
142 02630 3171     LCHEX 17
143 02634 A72      C=C+A W
144 02637 54C      GONC o267D
145
146 0263A          END
```

AVE=D1	Ext		-	22			
BSERR	Ext		-	74			
D1MSTK	Ext		-	59	87		
=FILESe	Abs	9395 #000024B3	-	30			
FILXQ\$	Ext		-	35			
FINDF	Ext		-	38			
FINDf+	Abs	18086 #000046A6	-	20	95		
FLOAT	Ext		-	53			
FNRTN1	Ext		-	61			
FTYPF#	Ext		-	104			
HXDCW	Ext		-	52			
JUMPER	Ext		-	94			
MPY	Ext		-	140			
PILERR	Ext		-	84			
PILerr	Abs	9585 #00002571	-	84	78		
POLL	Ext		-	68			
POP1S	Ext		-	88			
R<RSTK	Ext		-	33			
RSTK<R	Ext		-	56			
ave=d1	Abs	9385 #000024A9	-	22	34	37	92
bserr	Abs	9559 #00002557	-	74	70		
error	Abs	9566 #0000255E	-	77	96		
o2574	Abs	9459 #000024F3	-	48	36		
o2578	Abs	9463 #000024F7	-	49	132		
o257C	Abs	9467 #000024FB	-	50	46		
o2582	Abs	9473 #00002501	-	52	125		
o2590	Abs	9487 #0000250F	-	54	64		
o25B1	Abs	9520 #00002530	-	63	39	83	
o25B8	Abs	9527 #00002537	-	66	48		
o25F8	Abs	9591 #00002577	-	86	72		
o267D	Abs	9724 #000025FC	-	125	144		
o2681	Abs	9728 #00002600	-	126	117		
o2691	Abs	9744 #00002610	-	132	113		
o2695	Abs	9748 #00002614	-	133	115		

Input Parameters

Source file name is files.a

Listing file name is files.l

Object file name is files.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE POLHND
2
3          * *****
4          * Poll handler module
5          *
6          * JPC LEX poll handler
7          *
8          * Rebuilt from JPC rev D binary and from various sources,
9          *   J-F Garnier 2005
10         * Updated with rev X changes (R. Rosenbaum)
11         *
12         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
13         * *****
14
15 0263A          ABS      #263A
16
17 RESJPC EQU     #2F991      JPC reserved location
18
19
20         *** changes: moved here in rev X (see below):
21 0263A 6D26     wtky     GOTO   Wtky
22 0263E 8C0070   cfg     GOLONG  Cfg
23 02644 8CC570   pwr     GOLONG  Pwr
24 0264A 8C7990   cldst   GOLONG  Cldst
25 02650 8C5970   ftype   GOLONG  Ftype
26         *** end of changes
27
28         * *****
29         * POLL Handler entry
30         * *****
31 02656 3100     =POLHND LC(2)  =pKYDF
32 0265A 961      ?B=C   B
33 0265D 86       GOYES   Kydf
34 0265F B66      C=C+1  B      (pWTKY)
35 02662 961      ?B=C   B
36 02665 5D       GOYES   wtky
37 02667 3100     LC(2)   =pCONFG
38 0266B 961      ?B=C   B
39 0266E 0D       GOYES   cfg
40 02670 B66      C=C+1  B      (pPWROF)
41 02673 961      ?B=C   B
42 02676 EC       GOYES   pwr
43 02678 3100     LC(2)   =pCLDST
44 0267C 961      ?B=C   B
45 0267F BC       GOYES   cldst
46 02681 3100     LC(2)   =pFTYPE
47 02685 961      ?B=C   B
48 02688 8C       GOYES   ftype
49         *** changes: added in rev X:
50 0268A B66      C=C+1  B      (pLIST2)
51 0268D 961      ?B=C   B
52 02690 D2       GOYES   LStxt
53         *** end of changes
54 02692 3100     LC(2)   =pEDIT
55 02696 961      ?B=C   B
56 02699 21       GOYES   edit
57 0269B 3100     LC(2)   =pMRGE2
```

```
58 0269F 961          ?B=C   B
59 026A2 F0          GOYES  mrge2
60 026A4 969          ?B=0   B      (pVER$)
61 026A7 01          GOYES  ver$
62 026A9 00          RTNSXM
63                    *** changes: moved from here in rev X (see above):
64                    *wtky   GOTO   Wtky
65                    *cfg    GOTO   Cfg
66                    *pwrof  GOLONG Pwrof
67                    *cldst  GOLONG Cldst
68                    *ftype  GOLONG Ftype
69                    *** end of changes
70 026AB 8C0D90      edit   GOLONG Edit
71 026B1 8C5D90      mrge2  GOLONG Mrge2
72 026B7 8C6890      ver$   GOLONG VER$
73 026BD 6C44        LSTtxt GOTO   LSTTXT      *** added in rev X
74
75
76                    STITLE  pKYDF handler
```

```
77
78 026C1 6290      asfld GOTO  Asfld
79
80      *** pKYDF entry
81 026C5 110      Kydf  A=R0
82 026C8 3102      LCASC  ' '      space?
83 026CC 962      ?C=A  B
84 026CF 2F       GOYES  asfld
85 026D1 3100      LC(2) =kcVIEW  view?
86 026D5 962      ?C=A  B
87 026D8 47       GOYES  view
88 026DA 3100      LC(2) =kcLFT   left?
89 026DE 962      ?C=A  B
90 026E1 F6       GOYES  left
91 026E3 3100      LC(2) =kcCALC  calc?
92 026E7 962      ?C=A  B
93 026EA 7D       GOYES  asfld
94
95      * finput management:
96 026EC 1B199F2   D0=(5) RESJPC  * in resv loc
97 026F3 15A0      A=DAT0 1
98 026F7 301      LCHEX  1
99 026FA 0E06      A=A&C  P
100 026FE 908      ?A=0   P
101 02701 93       GOYES  Kydf9
102 02703 110      A=R0
103 02706 D2       C=0    A
104      *** change: code added (undated and undocumented)
105 02708 3100      LC(2) =kcEOL
106 0270C 1BFB6F2   D0=(5) #2F6BF
107 02713 8A2      ?C=A  A
108 02716 23       GOYES  hKD003
109 02718 AC0      A=0    S
110 0271B 1504      DAT0=A S
111      *** end of change
112 0271F 3100      LC(2) =kcRT
113 02723 8A2      ?C=A  A
114 02726 61       GOYES  hKD000
115 02728 3100      LC(2) =kcFRT
116 0272C 8A2      ?C=A  A
117 0272F 11       GOYES  hKD001
118 02731 3100      LC(2) =kcRUN
119 02735 8A2      ?C=A  A
120 02738 C0       GOYES  hKD002
121 0273A 00       Kydf9  RTNSXM
122
123 0273C 6EE1      hKD000 GOTO  hKD010
124 02740 6A72      hKD001 GOTO  hKD050
125 02744 6473      hKD002 GOTO  hKD100
126 02748 6493      hKD003 GOTO  hKD150
127 0274C 6501      view   GOTO  View
128 02750 6281      left   GOTO  Left
129
130
131      STITLE Assembler fields
```

```

132
133      *** space & calc keys: manage assembler fields
134      MNEMF EQU      8
135      MODF  EQU     15
136      COMF  EQU     24
137
138      *****
139      * Auteur: Stephane Barizien 03/85 - Interception
140      *       de pKYDF et champs assembleur.
141      * 1ere modification: M.MARTINET et P.DAVID 10/85 -
142      *   transformation en FIELD ON | OFF.
143      * 2eme modification: M.MARTINET 12/85 - mise en
144      *   place de la bascule touche CALC et suppres-
145      *   sion token FIELD ON | OFF.
146      * *****
147 02754 1B00000 Asfld D0=(5) 14+(=FLGREG) Est-on sous Åditeur
148 0275B 1564      C=DAT0 S      de
149 0275F A46      C=C+C S      textes ?
150 02762 514      GONC  rtn      Non: retour.
151 02765 D2      C=0    A
152 02767 3100     LC(2) =kcCALC Est-ce
153 0276B 966      ?A#C B      la touche CALC ?
154 0276E 83      GOYES FXQ      Non: exÅcution suite prgm.
155 02770 1A199F   D0=(4) =RESJPC Autrement: changement
156 02776 1520     A=DAT0 P      d'Åtat
157 0277A 302     LC(1) 2      du flag
158 0277D D8      B=A    A      FIELD
159 0277F 0EF2     C=C&A A      "
160 02783 90A     ?C=0  P
161 02786 C0      GOYES Fon
162 02788 30D     LCHEX D
163 0278B 0E06    A=A&C P
164 0278F 590     GONC  Fok
165 02792 302     Fon  LC(1) 2
166 02795 0E0E    A=A!C P
167 02799 1500    Fok  DAT0=A P
168 0279D 840     ST=0  0
169 027A0 6AA0    GOTO  rtnxm0 et fin d'exÅcution
170 027A4 00      rtn  RTNSXM
171 027A6 1A199F   FXQ  D0=(4) RESJPC ExÅcution du programme
172 027AC 15A0     A=DAT0 1      Est-on en mode field on ?
173 027B0 302     LC(1) 2
174 027B3 0E06    A=A&C P
175 027B7 908     ?A=0  P
176 027BA AE      GOYES  rtn      Non: fin de programme
177 027BC 1A0000   D0=(4) =DSPBFS Oui: continu
178 027C2 32FB0   LC(3) #BF      (=DSPBFE) - (=DSPBFS) -1
179 027C7 AB5     B=C    X
180 027CA 14A     SP05  A=DAT0 B
181 027CD 968     ?A=0  B
182 027D0 F1      GOYES  SP08
183 027D2 3102    LCASC  ' '
184 027D6 966     ?A#C  B
185 027D9 D0      GOYES  SP06
186 027DB 161     D0=D0+ 2
187 027DE A3D     B=B-1 X
188 027E1 58E     GONC  SP05

```

```

189 027E4 00      rtn1   RTNSXM
190 027E6 31A2    SP06   LCASC  '*'      Y a-t-il des remarques ?
191 027EA 962     ?A=C   B
192 027ED 7B      GOYES  rtn      Oui: fin de programme
193 027EF 1A0000  SP08   D0=(4) =CURSOR VÃrification
194 027F5 14A     A=DAT0 B      position du curseur
195 027F8 3170    LC(2)  (MNEMF)-1 en fonction
196 027FC 9EA     ?A<=C B      des
197 027FF 41      GOYES  SP10     diffÃrents
198 02801 31E0    LC(2)  (MODF)-1 champs:
199 02805 9EA     ?A<=C B      MNEMF: mnemo-field
200 02808 B0      GOYES  SP10     MODF : modification-field
201 0280A 3171    LC(2)  (COMF)-1 COMF : comment-field
202 0280E 9E6     ?A>C   B
203 02811 3D      GOYES  rtn1
204 02813 E2      SP10   C=C-A   A      RedÃfinition
205 02815 E6      C=C+1  A      de la touche
206 02817 1A0000 D0=(4) =DEFADR espace
207 0281D 14C    DAT0=C B      avec
208 02820 161    D0=D0+ 2      le
209 02823 304    LCHEX  4      bon
210 02826 15C0   DAT0=C 1      nombre
211 0282A 7210    GOSUB  SKPTBL de
212 0282E 02020202 NIBASC ' ' CARCTERES BLANCS
      02020202
213 0283E 02      NIBASC ' '
214 02840 07      SKPTBL C=RSTK      Sortie de table
215 02842 160     D0=D0+ 1
216 02845 144     DAT0=C A
217 02848 850     ST=1   0      ST[0]=1 => dÃfinit. touche
218 0284B 821     rtnxm0 XM=0      ST[0]=0 => touche inhibee
219 0284E 03      RTNCC      Puis interception du Poll.
220
221              STITLE View cursor position
  
```



```

222
223      *** VIEW key management
224      * if in USER mode, view cursor position
225 02850 00      rtn2      RTNSXM
226 02852 DB      View      C=D      A      SFLAG? abime D(A), qu'il faut impérativement
227 02854 D5      B=C      A      sauver si le poll n'est pas intercepté
228 02856 3100    LC(2)    =flUSER
229 0285A 8F00000 GOSBVL  =SFLAG?  Tester le flag "User".
230 02861 D9      C=B      A
231 02863 D7      D=C      A
232 02865 5AE     GONC     rtn2     Si d'alarme, retour
233
234 02868 1F00000 D1=(5)  (=FUNCR0)-2  Zone de 22 caractères à afficher
235 0286F D2      C=0      A      C(A)=Longueur de la zone
236 02871 31C2    LC(2)    44      (22 octets)
237 02875 8F00000 GOSBVL  =WIPOUT  À remplir avec le code 0
238 0287C 1F00000 D1=(5)  =CURSOR
239 02883 AF2     C=0      W      On ne gardera que C(B) pour l'appel à HXDCW.
240 02886 14F     C=DAT1   B      C(B)=position du curseur.
241 02889 8F00000 GOSBVL  =HXDCW  Après l'appel: Mode DEC et résultat dans A:
242      *      *      A(W)=00...095 si le curseur est en 96ème
position
243 02890 B64     A=A+1   B      Curseur devient [1..96]
244 02893 04      SETHEX   Mode HEX pour conversion décimal-ASCII.
245 02895 1F00000 D1=(5)  (=FUNCR0)-2
246 0289C 3103    LCASC    '0'
247 028A0 814     ASRC     A: 60...09 (quartet de poids fort)
248 028A3 908     ?A=0    P      S'il est nul, on ne l'affiche pas (P=0 après
WIPOUT)
249 028A6 E0      GOYES    POL40
250
251 028A8 A6A     A=A+C   B      Conversion DEC-ASCII.
252 028AB 149     DAT1=A  B
253 028AE 171     D1=D1+  2      Poids faible en deuxième dans BUF.
254 028B1 AE0     A=0     B
255
256 028B4 810     POL40   ASLC     A(W)= 00.....06
257 028B7 A6A     A=A+C   B      Conversion
258 028BA 149     DAT1=A  B
259
260 028BD 1F00000 D1=(5)  (=FUNCR0)-2
261 028C4 8F00000 GOSBVL  =VIEWD1  Affichage du buffer pointé par D1.
262
263 028CB 821     XM=0    Poll intercepté
264 028CE 840     ST=0    0      Action (et non redéfinition).
265 028D1 01      RTN
266
267      STITLE CALC backspace key

```

```
268
269          *** left cursor management
270          * if in CALC mode, replace with backspace
271 028D3 1B00000 Left   D0=(5) (=SYSFLG)+11
272 028DA 142      A=DAT0 A
273 028DD C4      A=A+A A      Carry=1 si mode CALC
274 028DF 594      GONC   rtn3   Retour sinon
275          *** change: code added starting from JPC rev D.
276 028E2 07      C=RSTK
277 028E4 8F00000 GOSBVL =CSLC5
278 028EB 07      C=RSTK
279 028ED 8F00000 GOSBVL =CSLC5
280 028F4 07      C=RSTK
281 028F6 D5      B=C A
282 028F8 06      RSTK=C
283 028FA 8F00000 GOSBVL =CSRC5
284 02901 06      RSTK=C
285 02903 8F00000 GOSBVL =CSRC5
286 0290A 06      RSTK=C
287          * check that RDKEY was not called from CHEDIT
288 0290C 34FAC41 LCHEX 14CAF
289 02913 8A1      ?B=C A      called from CHEDIT, ignore it!
290 02916 31      GOYES  rtn3
291          *** end of change
292 02918 118      C=R0
293 0291B A6E      C=C-1 B
294 0291E 25      P=      5
295 02920 3100    LC(2) =k#BKSP C[9-5]=Code physique
296 02924 108      R0=C      Ni vu, ni connu, on change
297 02927 20      P=      0
298 02929 00      rtn3   RTNSXM
299
300
301          STITLE FINPUT poll handler
```

```
302
303 *
304 * Bits d'etat du display driver du HP71.
305 *
306 BitsOK EQU 1
307 NoChFC EQU 11
308 EscSt0 EQU 0
309 EscSt1 EQU 1
310
311 *****
312 * hKD010
313 *
314 * But: traiter le deplacement a droite (touche [->] dans
315 * FINPUT
316 * Entree: -
317 * Sortie:
318 * - interception de la touche signifie que nous
319 * n'autorisons pas le deplacment a droite
320 * Abime: A-C, D0, D1, ST
321 * Niveaux: 0
322 * Historique:
323 * 86/11/16: P.D. & J.T. conception & codage
324 * 86/11/24: P.D. ajout de documentation
325 *****
326
327 0292B hKD010
328 *
329 * Recherche s'il existe un caractere non protege dans le
330 * reste du display buffer
331 *
332 * for i := CURSOR + 1 to 96 do
333 * if dspmsk[i] = unprotected then exit ;
334 *
335 0292B 1B00000 D0=(5) =CURSOR
336 02932 AF0 A=0 W
337 02935 14A A=DAT0 B A(A) := CURSOR
338 02938 E4 A=A+1 A A(A) := next char.
339 0293A D8 B=A A B(A) := next char.
340 0293C 81C ASRB
341 0293F 81C ASRB A(B) := (next char) / 4
342 02942 3400000 LC(5) (=DSPMSK)+23
343 02949 E2 C=C-A A
344 0294B 134 D0=C D0 := dspmsk [next char]
345 *
346 * Il est bien evident que la boucle reellement ne sera pas une
347 * boucle caractere par caractere. La boucle va traiter un
348 * quartet de DSPMSK a la fois, c'est a dire quatre
349 * caracteres. Il y a donc un cas particulier a tester au
350 * préalable: le cas ou il n'y a pas exactement un multiple
351 * de 4.
352 *
353 0294E D4 A=B A A(A) := next char.
354 02950 31F5 LC(2) 95
355 02954 B62 C=C-A B C(B) := nb de car. restant
356 02957 416 GOC RTnsxm curseur a la fin
357 0295A AEA A=C B
358 0295D AE8 B=A B B(B) := sauvegarde du nb de car
```

```
359 02960 3130          LC(2)  3
360 02964 0E66          A=A&C  B          A(B) := X = nb de car mod 4
361 02968 962           ?C=A   B
362 0296B A2            GOYES  reste
363 0296D 21            P=     1
364 0296F A6C           A=A-1  B
365 02972 4C0           GOC    msk
366 02975 23            P=     3
367 02977 A6C           A=A-1  B
368 0297A 440           GOC    msk
369 0297D 27            P=     7
370 0297F 80CF         msk      C=P    15
371 02983 20            P=     0
372 02985 1524          A=DAT0 S
373 02989 0E46          A=A&C  S
374 0298D 946           ?C#A   S
375 02990 92            GOYES  RTnsxm
376
377 *
378 * Maintenant, le cas particulier a ete teste, il ne reste
379 * qu'a tester que tous les autres quartets de DSPMSK ont
380 * comme valeur #F
381
381 02992 180            D0=D0- 1
382 02995 81D           reste   BSRB
383 02998 81D           BSRB          B(B) := (nb de car.) / 4
384 0299B 5F0           GONC    rst20   B.E.T.
385 0299E 1564         rst10   C=DAT0 S
386 029A2 B46           C=C+1  S
387 029A5 531           GONC    RTnsxm
388 029A8 180           D0=D0- 1
389 029AB A6D           rst20   B=B-1  B
390 029AE 5FE           GONC    rst10
391
392 *****
393 * RIEN
394 *
395 * But: ne rien faire (ahhhhhhhhhh)
396 *   FINPUT
397 * Entree: -
398 * Sortie: Par RTNSXM, a CHEDIT, en declarant qu'on a agit
399 *   et qu'il n'y a pas besoin que le systeme continue
400 * Historique:
401 *   86/12/02: P.D. & J.T. conception & codage
402 *****
403 029B1 840           RIEN    ST=0   0          inhiber la touche
404 029B4 821           XM=0
405 029B7 01            RTN
406
407 029B9 00           RTnsxm  RTNSXM
408
409 *****
410 * hKD050
411 *
412 * But: traiter la touche [g][->] quand on est dans FINPUT
413 * Entree: -
414 * Sortie:
415 *   La touche est interceptee, c'est nous qu'on a traite
```

```
416      *   l'appui sur la touche, et c'est pas HP. Na !
417      * Detail:
418      *   2 cas:
419      *     - Le dernier caractere du display buffer est protege.
420      *       Dans ce cas, [g][->] nous amene sur le dernier
421      *       caractere du dernier champ non protege.
422      *     - Le dernier caractere du display n'est pas protege.
423      *       Alors; [g][->] nous amene apres le dernier
424      *       caractere introduit si c'est possible (c'est a dire
425      *       que le display buffer n'est pas plein).
426      * Historique:
427      *   86/11/21: P.D. & J.T. conception & codage
428      *   86/11/24: P.D.      ajout de documentation
429      *   86/11/29: P.D. & J.T. gestion de la video HPIL
430      * *****
431
432 029BB      hKD050
433      *
434      * Modification du 86/11/29 :
435      *
436      * Ce poll est destine a court-circuiter le display driver
437      * (de m...) du HP71. Quand la touche [g][->] est activee,
438      * c'est nous qui choisissons la position d'arrivee du
439      * curseur. Le probleme est de faire comprendre ca au module
440      * HPIL
441      *
442      * Ce driver se comporte d'une maniere plus propre que celui
443      * interne au HP71. Il sait jusqu'ou ne pas aller quand la
444      * sequence ESC CTRL C lui arrive. Il n'y a donc pas besoin
445      * de l'empecher d'agir. Mais ceci suppose de l'activer
446      * explicitement. Il n'est plus appele par le driver du HP71
447      * mais par nous.
448      *
449      *
450      *
451      * Comme dans le display-driver du HP71, nous appelons
452      * d'abord l'HPIL :
453      *
454 029BB 7690      GOSUB  ILFART
455      *
456      *
457      * Puis, nous gerons le LCD.
458      *
459      * j := 0 ;
460      * i := 96 ;
461      * while (i >= 0) and
462      *     ((dspmsk[i] = protected) or (display[i] = 0))
463      * do
464      *   begin
465      *     if dspmsk[i] # protected then j := 1 ;
466      *     i := i-1 ;
467      *   end ;
468      * if i<0 then exit poll ;
469      * if j#0 then cursor := j
470      *     else cursor := i ;
471      *
472 029BF 31F5      LC(2)  95
```

```
473 029C3 AE5          B=C   B          B(B) := i
474 029C6 1F00000      D1=(5) (=DSPBFE)-2 caractere courant
475 029CD AF0          A=0   W          A(S) := 0 : masque courant := 0
476 029D0 1B00000      D0=(5) =DSPMSK   A(B) := 0 : j := 0
477 029D7 6F20         GOTO  hKD070
478
479
480 * Dans la boucle, les assertions suivantes sont vraies:
481 *
482 *   A(S) = masque courant (mis a jour a la fin)
483 *   A(B) = j
484 *   B(B) = i
485 *   C(S) = DSPMSK [IP(i/4)]
486 *   D0 = ^ DSPMSK [IP(i/4)]
487 *   D1 = ^ DSPBFS [i]
488 *
489 029DB               hKD060
490 *
491 * Obtention d'un nouveau DSPMSK si necessaire, ainsi qu'une
492 * regeneration du masque courant.
493 *
494 029DB 94C           ?A#0   S
495 029DE C0           GOYES  hKD062
496 029E0 B44         A=A+1  S
497 029E3 1564        C=DAT0  S
498 029E7 160         D0=D0+ 1
499 029EA             hKD062
500 *
501 * Test de la protection
502 *
503 029EA AC5          B=C     S          On abime B(S)
504 029ED 0E44        B=B&A  S          B(S) := si dspmsk[i] = protected
505 *                                     alors #0
506 *                                     sinon 0
507 029F1 94D         ?B#0   S
508 029F4 D0           GOYES  hKD065          Pas la peine de continuer pour la
509 *                                     position : si le caractere est
510 *                                     protege.
511
512 *
513 * La position n'est pas protegee.
514 *
515 029F6 14F         C=DAT1  B
516 029F9 96E         ?C#0   B          il y a un caractere
517 029FC 31          GOYES  hKD080
518
519 029FE AE4         A=B     B          j := i ;
520
521 02A01 1C1         hKD065 D1=D1- 2
522 02A04 A44         A=A+A  S          masque mis a jour
523 02A07             hKD070
524 02A07 A6D         B=B-1  B
525 02A0A 50D        GONC   hKD060
526 *
527 * On est sorti de la boucle sans avoir rien trouve. Tout le
528 * display est donc protege, ou aucun caractere n'a encore
529 * ete introduit.
```

```
530 *
531 02A0D 00          RTNSXM
532
533 *
534 * On est sorti du WHILE.
535 *
536 02A0F            hKD080
537 * if j#0 then j := i ;
538 *
539 02A0F 96C        ?A#0   B
540 02A12 50          GOYES  hKD090
541
542 02A14 AE4        A=B     B           j := i
543 *
544 * curseur := j
545 *
546 02A17            hKD090
547 02A17 1B00000    D0=(5) =CURSOR
548 02A1E B64        A=A+1   B           Programmation heuristique !
549 02A21 148        DAT0=A  B
550 *
551 * Calcul de FIRSTC (premier caractere a afficher sur la
552 * fenetre de 22 caracteres du HP71) :
553 *
554 * FIRSTC := CURSOR - WINDLN si c'est possible
555 *
556 02A24 1900        D0=(2) =WINDLN
557 02A28 14E        C=DAT0  B
558 02A2B B6E        C=A-C   B           C(B) := CURSOR - WINDLN
559 02A2E 550        GONC    FST#0
560 02A31 AE2        C=0     B
561 02A34 1900      FST#0    D0=(2) =FIRSTC
562 02A38 14C        DAT0=C  B
563
564 *
565 * Les instructions suivantes (modifications sur ST) sont
566 * des ordres donnees au display driver:
567 *
568 02A3B 1900        D0=(2) (=DSPSTA)+3
569 02A3F 1563        C=DAT0  X
570 02A43 0A         ST=C
571
572 02A45 84B        ST=0    NoChFC      On a change FIRSTC
573 02A48 841        ST=0    BitsOK     Il faut reconstruire le display
574 02A4B 09         C=ST
575 02A4D 1543       DAT0=C  X           Dans les 12 bits du display driver
576
577 02A51 6F5F       GOTO    RIEN       interception
578
579 *****
580 * ILPART
581 *
582 * But: Envoyer le curseur a l'extreme droite sur la video
583 * HPIL courante.
584 * Entree: -
585 * Sortie:
586 * Abime: A-D, D0, D1, R3, ST
```

```
587 * Appelle: BDISPJ (HPIL display driver), R<RSTK, RSTK<R
588 * Niveaux: 0 (5 sauves par R<RSTK)
589 * Detail: D est sauvegarde dans R3
590 * Historique:
591 * 86/11/29: P.D. & J.T. conception & codage
592 *****
593
594 *** entry point, called from finput module:
595 02A55 =ILFART
596 02A55 1B00000 D0=(5) =DSPCHX
597 02A5C 142 A=DAT0 A
598 02A5F 8A8 ?A=0 A
599 02A62 00 RTNYES
600
601 02A64 DB C=D A
602 02A66 10B R3=C
603
604 02A69 24 P= 4 Sauvegarde 5 niveaux de pile
605 02A6B 8F00000 GOSBVL =R<RSTK
606
607 02A72 1B00000 D0=(5) =ESCSTA
608 02A79 301 LC(1) EscSt1
609 02A7C 15C0 DAT0=C 1
610
611 02A80 3130 LC(2) 3 CTRL C
612 02A84 7020 GOSUB ILSSEND
613 02A88 20 P= 0
614
615 02A8A 1B00000 D0=(5) =ESCSTA
616 02A91 300 LC(1) EscSt0
617 02A94 15C0 DAT0=C 1
618
619 02A98 24 P= 4 Restaure les niveaux de pile
620 02A9A 8F00000 GOSBVL =RSTK<R
621
622 02AA1 11B C=R3
623 02AA4 D7 D=C A
624 02AA6 01 RTN
625
626 *****
627 * ILSSEND
628 *
629 * But: appeler directement le display-driver du module HPIL
630 * Entree:
631 * - C(B) = caractere a afficher
632 * Sortie: -
633 * Abime: A-D, D0, D1, ST
634 * Appelle: BDISPJ (#F3637 dans l'HPIL:1B)
635 * Niveaux: 5
636 * Detail:
637 * le controle est rendu a la routine appelante directement
638 * Historique:
639 * 86/11/29: P.D. & J.T. conception & codage
640 *****
641
642 02AA8 AEA ILSSEND A=C B DATA BYTE
643 02AAB 1B00000 D0=(5) =DSPCHX
```



```
644 02AB2 146          C=DAT0 A
645 02AB5 06          RSTK=C
646 02AB7 01          RTN          GOVLNG =BDISPJ dans l'HPIL
647
648          *****
649          * hKD100
650          *
651          * But: traiter le touche [RUN] quand on est dans FINPUT
652          * Entree: -
653          * Sortie: -
654          * La touche est interceptee, pour inhiber la definition
655          * eventuelle de la touche [RUN]. Chez moi, j'ai toujours
656          * un DEF KEY '#46','RUN' ;
657          * Pas vous?
658          * Appelle: kRUN (tombe dedans)
659          * Historique:
660          * 86/12/02: P.D. & J.T. conception & codage
661          *****
662
663 02AB9 1F00000 hKD100 D1=(5) (=DEFADR)+2
664 02AC0 AC2          C=0 S
665 02AC3 1554          DAT1=C S
666 02AC7 170          D1=D1+ 1
667 02ACA 7200          GOSUB hKD110
668 02ACE F0           CON(2) 15 RUN KEY
669 02AD0 07           hKD110 C=RSTK
670 02AD2 145          DAT1=C A
671 02AD5 850          ST=1 0
672 02AD8 821          XM=0
673 02ADB 01          RTN
674
675
676          *** change: code added. (undated and undocumented)
677 02ADD 1524          hKD150 A=DAT0 S
678 02AE1 94C          ?A#0 S
679 02AE4 5D           GOYES hKD100
680 02AE6 1F00000          D1=(5) =LDCSPC
681 02AED 143          A=DAT1 A
682 02AF0 1EAF8F          D1=(4) #F8FA
683 02AF6 147          C=DAT1 A
684 02AF9 AC0          A=0 S
685 02AFC 8A6          ?C#A A
686 02AFF 50           GOYES hKD160
687 02B01 B44          A=A+1 S
688 02B04 1504          hKD160 DAT0=A S
689 02B08 00           RTNSXM
690          *** end of change
691
692          STITLE pLIST2 handler
```

```
693
694          ***** changes:
695          *          RTNSXM          * present in rev D, removed in X
696
697          *** added in rev X:
698          *** pLIST2 handler
699 02B0A 3400000 LSTTXT LC(5) =fTEXT
700 02B11 8A2          ?A=C A
701 02B14 40          GOYES list01
702 02B16 00          RTNSXM
703 02B18 AF2          list01 C=0 W
704 02B1B AF0          A=0 W
705 02B1E E6          C=C+1 A
706 02B20 109          R1=C
707 02B23 36575840    LCHEX 1048575
708          1
709 02B2C 10B          R3=C
710 02B2F 14A          A=DAT0 B
711 02B32 3100          LC(2) =tCOMMA
712 02B36 966          ?A#C B
713 02B39 C3          GOYES LIST07
714 02B3B 160          D0=D0+ 1
715 02B3E 142          A=DAT0 A
716 02B41 F4          ASR A
717 02B43 164          D0=D0+ 5
718 02B46 101          R1=A
719 02B49 14A          A=DAT0 B
720 02B4C 3100          LC(2) =tCOMMA
721 02B50 966          ?A#C B
722 02B53 C1          GOYES LIST06
723 02B55 161          LIST05 D0=D0+ 2
724 02B58 15A3          A=DAT0 4
725 02B5C 103          R3=A
726 02B5F 119          C=R1
727 02B62 8BE          ?C<=A A
728 02B65 01          GOYES LIST07
729 02B67 330000          LC(4) =eIVARG
730 02B6D 02          RTNSC
731 02B6F 111          LIST06 A=R1
732 02B72 103          R3=A
733 02B75 111          LIST07 A=R1
734 02B78 8F00000          GOSBVL =DECHEX
735 02B7F 101          R1=A
736 02B82 113          A=R3
737 02B85 8F00000          GOSBVL =DECHEX
738 02B8C 103          R3=A
739 02B8F 133          AD1EX
740 02B92 8F00000          GOSBVL =FILSK+
741 02B99 D7          D=C A
742 02B9B 174          D1=D1+ 5
743 02B9E 137          CD1EX
744 02BA1 10A          R2=C
745 02BA4 111          A=R1
746 02BA7 7360          GOSUB FRCRDn
747 02BAB 5C7          GONC FRCRD2
748 02BAE 133          AD1EX
749 02BB1 123          AR3EX
```

```
749 02BB4 11A          C=R2
750 02BB7 7350        GOSUB  FRCRDn
751 02BBB 560         GONC   LIST40
752 02BBE 7E60        GOSUB  RCDSKP
753 02BC2 1B00000     LIST40 D0=(5) =S-R1-1
754 02BC9 144         DAT0=C  A
755 02BCC 8F00000     GOSBVL =POPUPD
756 02BD3 11B         C=R3
757 02BD6 1F00000     LIST50 D1=(5) =OUTBS
758 02BDD 145         DAT1=C  A
759 02BE0 135         D1=C
760 02BE3 7940        GOSUB  RCDSKP
761 02BE7 108         R0=C
762 02BEA 8F00000     GOSBVL =PRPSND
763 02BF1 8F00000     GOSBVL =CK"ON"
764 02BF8 5E0         GONC   BYPASS
765 02BFB 135         D1=C
766 02BFE 173         D1=D1+ 4
767 02C01 137        CD1EX
768 02C04 51D         GONC   LIST50
769 02C07 8D00000     BYPASS GOVLNG =NXTSTM
770 02C0E CC          FRCRDn A=A-1  A
771 02C10 400        RTNC
772 02C13 101        R1=A
773 02C16 135        D1=C
774 02C19 8BF        ?D<=C  A
775 02C1C 21         GOYES  FRCRD3
776 02C1E 7110       GOSUB  FRCRD4
777 02C22 111        A=R1
778 02C25 58E        GONC   FRCRDn
779 02C28 821        FRCRD2 XM=0
780 02C2B 137        CD1EX
781 02C2E 03         FRCRD3 RTNCC
782 02C30 1C3        RCDSKP D1=D1- 4
783 02C33 15B3       FRCRD4 A=DAT1 4
784 02C37 8F00000     GOSBVL =SWPBYT
785 02C3E 24         P=      4
786 02C40 A80        A=0     P
787 02C43 23         P=      3
788 02C45 B14        A=A+1   WP
789 02C48 4B1        GOC     FRCRD5
790 02C4B A1C        A=A-1   WP
791 02C4E 173        D1=D1+ 4
792 02C51 D8         B=A     A
793 02C53 E4         A=A+1   A
794 02C55 81C        ASRB
795 02C58 C4         A=A+A   A
796 02C5A C4         A=A+A   A
797 02C5C 137        CD1EX
798 02C5F 135        D1=C
799 02C62 C2         C=C+A   A
800 02C64 20         FRCRD5 P= 0
801 02C66 01        RTN
```

***** end of changes

802
803
804
805

STITLE pWTKY handler

```
806
807          *** pWTKY handler
808 02C68 1F199F2 Wtky   D1=(5) RESJPC   in resv memory
809 02C6F 1530          A=DAT1 P
810 02C73 304          LCHEX  4
811 02C76 0E06          A=A&C  P
812 02C7A 908          ?A=0   P
813 02C7D 43           GOYES  Wtky2
814
815          *** margin management
816
817 02C7F 32D38          LC(3)  bMARGI
818 02C84 8F00000       GOSBVL =I/OFND  buffer oÈ est
819          *          stockÀe la marge.
820 02C8B 552           GONC   Wtky2
821 02C8E 14B           A=DAT1 B      A(B) = sa valeur
822 02C91 1F00000       D1=(5) =CURSOR D1 ^ CURSOR
823 02C98 14F           C=DAT1 B      C(B) = position du
824          *          *          curseur ( de 0 È 95 ).
825 02C9B 966           ?C#A  B      La position du curseur #
826          *          *          margination ?
827 02C9E 31           GOYES  Wtky2
828 02CA0 DB            C=D  A
829 02CA2 10B           R3=C
830 02CA5 8F00000       GOSBVL =CHIRP  Autrement on alerte...
831 02CAC 11B           C=R3
832 02CAF D7            D=C  A
833
834
835          STITLE Key repeat management
```

```
836
837     *** key repeat management
838
839     USRSTA EQU #01B80 is not supported
840
841     *****
842     * Routine KEYR
843     * Cette routine scrute le clavier: elle attend que
844     * l'on appuie sur une touche, puis place le code
845     * de la touche sur laquelle on a appuyé dans R0(B)
846     * KEYR boucle indéfiniment sur elle-même si l'on
847     * n'appuie sur aucune touche.
848     *****
849
850 02CB1 24      Wtky2 P= 4      Sauvegarde en RAM les 5
851     *          *          derniers adresses placées
852     *          *          dans la pile de
853 02CB3 8F00000 GOSBVL =R<RSTK  Retour (RSTK)
854 02CBA 7160    GOSUB RPTKY  L'action d'une touche
855     *          *          doit-elle se répéter ?
856 02CBE 482     GOC DEFKY   Oui; on retourne le code
857     *          *          de cette touche a la
858     *          *          routine qui a mis le pol
859     *** changes
860     * rev D
861     *          GOSUB usrsta * removed in rev X
862     * rev X: replaced by this... not supported!
863 02CC1 8F08B10 GOSBVL =USRSTA  Construit l'affichage
864     *          *          pour qu'il soit la
865     *          *          réplic du buffer d'écran
866     *** end of changes
867 02CC8 8F00000 GOSBVL =BLDDSP
868 02CCF 8F00000 GOSBVL =SETTMO  Autorise la mise
869     *          *          hors-tension du 71 au bout
870     *          *          de 10' si pendant ce temps
871     *          *          Aucune touche n'a été
872     *          *          pressée
873
874     *****
875     * Début de la boucle infinie
876     * Elle est du type:
877     *   0 0 DO
878     *   TRY-TO-SLEEP
879     *   KEY? IF LEAVE THEN
880     *   TIMED-OUT IF LEAVE THEN
881     *   LOOP
882     *****
883     fTMOUT EQU #3
884 02CD6 8F00000 KEYR10 GOSBVL =SLEEP  Place le 71 dans un état
885     *          *          de faible consommation
886     *          *          d'énergie. Ne retourne
887     *          *          que lorsqu'une touche a
888     *          *          été enfoncée ou lorsqu'une
889     *          *          alarme vient à terme
890 02CDD 4C1     GOC KEYR80  Nous sommes dans ce
891     *          *          dernier cas si carry est
892     *          *          armé
```

```
893 02CE0 8F00000      GOSBVL =POPBUF  et dans le 1er si elle
894      *      *      est d'alarme
895 02CE7 D9      DEFKY  C=B  A      Place le code de la
896      *      *      touche en C(B)
897 02CE9 108     DEFKY+  R0=C      puis en R0
898 02CEC 24      P=      4      Restaurons les 5 adresses
899      *      *      pr'admettent conservées en
900      *      *      RAM
901 02CEE 8F00000      GOSBVL =RSTK<R
902 02CF5 821      XM=0      et indiquons que le
903      *      *      pol-process doit se
904      *      *      terminer
905 02CF8 03      RTNCC
906
907 02CFA 7310     KEYR80 GOSUB  USRREQ  V'arifie qu'aucun
908      *      *      p'ariph'arique n'a besoin
909      *      *      d'aide
910 02CFE 8F00000      GOSBVL =ALMSRV  V'arifie qu'aucune alarme
911      *      *      n'est à terme
912 02D05 863      ?ST=0  =ftMOUT  10' sans activit' ne se
913      *      *      sont-elles pas acoules ?
914 02D08 EC      GOYES  KEYR10  Non; essayons de nouveau
915      *      *      de nous endormir
916 02D0A 3100     LC(2)  =k#OFF  Oui; faisons comme si
917      *      *      quelqu'un avait appuyé sur
918      *      *      la touche OFF (f ON)
919 02D0E 5AD      GONC   DEFKY+  (B.E.T.)
920
921      *****
922      * Routine utilis' par KEYR et RPTKY
923      *****
924 02D11      USRREQ
925
926      ***** changes *****
927      * rev. D:
928      *      GOSUB  usrsta
929      *      GOVLNG =CKSREQ
930
931      *****
932      * RPTKY est une routine HP. Je n'en ai modifi' que
933      * le d'but. Seule la partie modifi' sera
934      * comment' . R'f'erez-vous aux IDS 3 pour la partie
935      * non comment'.
936      *
937      * Algorithme simplifi':
938      * Si une touche commence seulement à r'p'ater son
939      *      son action (flRPTD abaiss')
940      *      alors on attend un temps T avant que cette
941      *      cette action ne se r'p'ete
942      *      puis on arme flRPTD
943      * Si la touche a d'jà r'p'at' son action
944      *      on n'attend pas avant de r'p'ater son action
945      * Si l'on appuie sur une nouvelle touche
946      *      alors on d'arme flRPTD
947      *****
948
949      *RPTKY
```

```
950      *      LC(2)  =f1RPTD
951      *      GOSBVL =SFLAG?  L'action de la touche
952      *      *      s'est-elle d'ajÈ rApÅtÅe ?
953      *      LCHEX  700000
954      *      GOC    RPTK10  Oui; on n'attends pas
955      *      *      avant de rApÅter de
956      *      *      nouveau celle-ci
957      *      LCHEX  4B
958      *RPTK10 GOSUB  WRTTM1  La routine ci-dessous est
959      *      *      commentÅe dans les IDS 3
960      *      LC(2)  =f1RPTD
961      *      GOSBVL =SFLAGS
962      *RPTK20 P=      8
963      *      GOSBVL =DEBNCE
964      *      D1=(5) (=KEYBUF)+2*14
965      *      A=0     A
966      *      A=DAT1 B
967      *      A=A-1  A
968      *      C=0     A
969      *      LC(2)  14
970      *      GOSBVL =IDIVA
971      *      P=      0
972      *      LC(5)  =KCOL0
973      *      C=C-B  A
974      *      D1=C
975      *      C=DAT1 XS
976      *      LCHEX  3
977      *      A=A&C  P
978      *RPTK30 A=A-1  P
979      *      GOC    RPTK40
980      *      C=C+C  XS
981      *      GOTO   RPTK30
982      *RPTK40 C=C+C  XS
983      *      GOC    RPTK45
984      *RPTKCC LC(2)  =f1RPTD
985      *      GOSBVL =SFLAGC
986      *TMRRST C=0     W
987      *      C=C+1  XS
988      *      GOTO   WRTTM1
989      *
990      *RPTK45 D0=(5) (=TIMER1)+5
991      *      D1=(5) =KEYPTR
992      *      A=DAT0 P
993      *      LCHEX  7
994      *      ?A=C   P
995      *      GOYES  RPTK47
996      *      GOSUB  TMRRST
997      *      D0=(5) (=KEYBUF)+2*14
998      *      A=DAT1 XS
999      *      ?A#0   XS
1000     *      GOYES  RPTKCC
1001     *      A=DAT0 B
1002     *      B=0     A
1003     *      B=A     B
1004     *      RTNSC
1005     *
1006     *RPTK47 A=DAT1  XS
```

```
1007      *      ?A#0   XS
1008      *      GOYES  RPTKCC
1009      *      GOSUB  USRREQ  o2C25
1010      *      GOTO   RPTK20
1011      *
1012      *WRTTM1  D0=(5) =TIMER1
1013      *      P=      5
1014      *      DAT0=C  6
1015      *      A=DAT0  6
1016      *      ?A=C   WP
1017      *      GOYES  WRTTMX
1018      *      DAT0=C  6
1019      *WRTTMX  P=      0
1020      *      RTNCC
1021      *
1022      ** routine added in rev D to replace
1023      ** unsupported USRSTA entry point (#01B80):
1024      *usrsta  D0=(5) =DSPSTA
1025      *      CSTEX
1026      *      C=DAT0  X
1027      *      CSTEX
1028      *      P=      0
1029      *      RTNCC
1030
1031      *****
1032      * rev. X:
1033 02D11 8F08B10      GOSBVL =USRSTA  not a supported entry point!
1034 02D18 8D00000      GOVLNG =CKSREQ
1035 02D1F 3100      RPTKY   LC(2)  =f1RPTD
1036 02D23 8F00000      GOSBVL =SFLAG?
1037 02D2A 35800007      LCHEX  700008  KSPD REP RATE
1038 02D32 460      GOC     RPTK10
1039 02D35 3109      LCHEX  90      KSPD INITIAL DELAY
1040 02D39 8D1D251  RPTK10  GOVLNG #152D1 not supported!
1041
1042      ***** end of changes *****
1043
1044
1045      STITLE pCONFIG handler
```



```
1046
1047          bMARGI EQU #0083D MARGIN buffer
1048          bENDUP EQU #0083E
1049          bXEQ EQU #00802
1050
1051          *** pCONFIG entry
1052 02D40          Config
1053 02D40 32D38          LC(3) bMARGI
1054 02D45 7E30          GOSUB i/ores
1055          * reset bit 0 of RESJPC
1056 02D49 30E          LCHEX E
1057 02D4C 1B199F2          D0=(5) RESJPC
1058 02D53 1520          A=DAT0 P
1059 02D57 0E06          A=A&C P
1060 02D5B 1500          DAT0=A P
1061          *
1062 02D5F 32E38          LC(3) bENDUP
1063 02D64 7F10          GOSUB i/ores Evite È bENDUP d'Átre
1064          *          d'Átruit È la prochaine reconfigu.
1065
1066          * call ROMAN module
1067 02D68 32BF3          LCHEX 3FB
1068 02D6D 8E0000          GOSUBL =ACBFND find ALTCH buffer
1069 02D73 411          GOC rtn4
1070 02D76 8E0000          GOSUBL =WRTBLA write ROMAN table address
1071 02D7C 32000          LC(3) =bALTCH
1072 02D81 7200          GOSUB i/ores
1073 02D85 00          rtn4 RTNSXM
1074 02D87 8D00000 i/ores GOVLNG =I/ORES
1075
1076
1077
1078          STITLE pPWROF handler
```

```

1079
1080
1081 02D8E 118      out      C=R0          Ces lignes, placées ici pr
1082              *                               gagner de la place,
1083 02D91 D7        D=C      A          restaurent le registre
1084              *                               D(A) si chers aux FPOLL
1085 02D93 32E38    savebf  LC(3)  bENDUP
1086 02D98 7BEF     GOSUB   i/ores     Evite à bENDUP d'être
1087              *                               détruit à la prochaine reconfigu.
1088 02D9C          rtncc
1089 02D9C D0        A=0  A          Désume carry
1090 02D9E E4        A=A+1 A
1091 02DA0 00        RTNSXM
1092
1093
1094              *** pPWROF handler entry
1095 02DA2          Pwrof
1096 02DA2 32E38    LC(3)  bENDUP
1097 02DA7 8F00000  GOSBVL =I/OFND    Mets D1 @ le début du
1098              *                               buffer (après l'en-tête)
1099 02DAE 5DE      GONC   rtncc     (Du moins s'il existe)
1100 02DB1 DB      C=D    A          D(A)
1101              **   B=C    A          -> B(A);
1102 02DB3 108     R0=C          -> R0;
1103 02DB6 3100    LC(2)  =flPWDN
1104 02DBA 8F00000 GOSBVL =SFLAG?   flPWDN est-il atteint?
1105 02DC1 5CC     GONC   out  o2D86
1106 02DC4 3100    LC(2)  =flTNOF
1107 02DC8 8F00000 GOSBVL =SFLAG?   flTNOF est-il allumé?
1108 02DCF 4EB     GOC    out
1109 02DD2 3100    LC(2)  =flTNOF
1110 02DD6 8F00000 GOSBVL =SFLAGS   Allumons flTNOF que l'on
1111              *                               n'intercepte pas pPWROF 1 2nd *
1112 02DDD 137     linep  CD1EX     D1 -> C(A) (D1 est
1113              *                               inchangé depuis =I/OFND)
1114 02DE0 8D00000 GOVLNG =LINEP+   Compile la ligne BASIC @
1115              *                               par C(A), puis l'écrit
1116
1117
1118              STITLE pFYPE handler

```

```
1119
1120      * *****
1121      * Non standard file type support
1122      * Written by Jan Buitenhuis and Janick Taillandier
1123      * *****
1124
1125      *** pFTYPE handler
1126 02DE7 7AD1      Ftype  GOSUB  eTBL
1127 02DEB 8          CON(1) #8
1128 02DEC 8          CON(1) #8
1129 02DED 0          CON(1) #0
1130 02DEE 50         CON(2) #05
1131 02DF0 4313A375 NIBASC '41:WA'
      14
1132 02DFA 1          CON(1) #1
1133 02DFB 040E       CON(4) #E040
1134 02DFE 8          CON(1) #8
1135 02E00 8          CON(1) #8
1136 02E01 0          CON(1) #0
1137 02E02 50         CON(2) #05
1138 02E04 4313A3B4 NIBASC '41:KE'
      54
1139 02E0E 1          CON(1) #1
1140 02E0F 050E       CON(4) #E050
1141 02E13 8          CON(1) #8
1142 02E14 8          CON(1) #8
1143 02E15 0          CON(1) #0
1144 02E16 50         CON(2) #05
1145 02E18 4313A335 NIBASC '41:ST'
      45
1146 02E22 1          CON(1) #1
1147 02E23 060E       CON(4) #E060
1148 02E27 2          CON(1) #2
1149 02E28 2          CON(1) #2
1150 02E29 0          CON(1) #0
1151 02E2A 50         CON(2) #05
1152 02E2C 4313A3D4 NIBASC '41:ML'
      C4
1153 02E36 1          CON(1) #1
1154 02E37 070E       CON(4) #E070
1155 02E3B 8          CON(1) #8
1156 02E3C 8          CON(1) #8
1157 02E3D 0          CON(1) #0
1158 02E3E 50         CON(2) #05
1159 02E40 4313A305 NIBASC '41:PR'
      25
1160 02E4A 1          CON(1) #1
1161 02E4B 080E       CON(4) #E080
1162 02E4F 8          CON(1) #8
1163 02E50 8          CON(1) #8
1164 02E51 0          CON(1) #0
1165 02E52 50         CON(2) #05
1166 02E54 4313A334 NIBASC '41:CA'
      14
1167 02E5E 1          CON(1) #1
1168 02E5F 020E       CON(4) #E020
1169 02E63 8          CON(1) #8
```

1170	02E64	8	CON(1)	#8
1171	02E65	0	CON(1)	#0
1172	02E66	50	CON(2)	#05
1173	02E68	4313A385 D4	NIBASC	'41:XM'
1174	02E72	1	CON(1)	#1
1175	02E73	030E	CON(4)	#E030
1176	02E77	8	CON(1)	#8
1177	02E78	8	CON(1)	#8
1178	02E79	0	CON(1)	#0
1179	02E7A	50	CON(2)	#05
1180	02E7C	4313A324 55	NIBASC	'41:BU'
1181	02E86	1	CON(1)	#1
1182	02E87	0B0E	CON(4)	#E0B0
1183	02E8B	8	CON(1)	#8
1184	02E8C	8	CON(1)	#8
1185	02E8D	0	CON(1)	#0
1186	02E8E	50	CON(2)	#05
1187	02E90	7353A3C4 02	NIBASC	'75:L '
1188	02E9A	1	CON(1)	#1
1189	02E9B	980E	CON(4)	#E089
1190	02E9F	8	CON(1)	#8
1191	02EA0	8	CON(1)	#8
1192	02EA1	0	CON(1)	#0
1193	02EA2	50	CON(2)	#05
1194	02EA4	7353A324 02	NIBASC	'75:B '
1195	02EAE	1	CON(1)	#1
1196	02EAF	880E	CON(4)	#E088
1197	02EB3	8	CON(1)	#8
1198	02EB4	8	CON(1)	#8
1199	02EB5	0	CON(1)	#0
1200	02EB6	50	CON(2)	#05
1201	02EB8	7353A314 02	NIBASC	'75:A '
1202	02EC2	1	CON(1)	#1
1203	02EC3	350E	CON(4)	#E053
1204	02EC7	8	CON(1)	#8
1205	02EC8	8	CON(1)	#8
1206	02EC9	0	CON(1)	#0
1207	02ECA	50	CON(2)	#05
1208	02ECC	7353A345 02	NIBASC	'75:T '
1209	02ED6	1	CON(1)	#1
1210	02ED7	250E	CON(4)	#E052
1211	02EDB	8	CON(1)	#8
1212	02EDC	8	CON(1)	#8
1213	02EDD	0	CON(1)	#0
1214	02EDE	50	CON(2)	#05
1215	02EE0	7353A375 02	NIBASC	'75:W '
1216	02EEA	1	CON(1)	#1
1217	02EEB	A80E	CON(4)	#E08A
1218	02EEF	8	CON(1)	#8
1219	02EF0	8	CON(1)	#8

1220	02EF1	0	CON(1)	#0
1221	02EF2	50	CON(2)	#05
1222	02EF4	7353A325	NIBASC	'75:R '
		02		
1223	02EFE	1	CON(1)	#1
1224	02EFF	B80E	CON(4)	#E08B
1225	02F03	8	CON(1)	#8
1226	02F04	8	CON(1)	#8
1227	02F05	0	CON(1)	#0
1228	02F06	50	CON(2)	#05
1229	02F08	7353A374	NIBASC	'75:G '
		02		
1230	02F12	1	CON(1)	#1
1231	02F13	850E	CON(4)	#E058
1232	02F17	0	CON(1)	#0
1233	02F18	0	CON(1)	#0
1234	02F19	0	CON(1)	#0
1235	02F1A	50	CON(2)	#05
1236	02F1C	64F42545	NIBASC	'FORTH'
		84		
1237	02F26	2	CON(1)	#2
1238	02F27	812E	CON(4)	#E218
1239	02F2B	912E	CON(4)	#E219
1240	02F2F	0	CON(1)	#0
1241	02F30	0	CON(1)	#0
1242	02F31	0	CON(1)	#0
1243	02F32	50	CON(2)	#05
1244	02F34	F424A402	NIBASC	'OBJ '
		02		
1245	02F3E	2	CON(1)	#2
1246	02F3F	C22E	CON(4)	#E22C
1247	02F43	D22E	CON(4)	#E22D
1248	02F47	0	CON(1)	#0
1249	02F48	0	CON(1)	#0
1250	02F49	0	CON(1)	#0
1251	02F4A	50	CON(2)	#05
1252	02F4C	3595D402	NIBASC	'SYM '
		02		
1253	02F56	2	CON(1)	#2
1254	02F57	E22E	CON(4)	#E22E
1255	02F5B	F22E	CON(4)	#E22F
1256	02F5F	0	CON(1)	#0
1257	02F60	0	CON(1)	#0
1258	02F61	0	CON(1)	#0
1259	02F62	50	CON(2)	#05
1260	02F64	25F4D402	NIBASC	'ROM '
		02		
1261	02F6E	4	CON(1)	#4
1262	02F6F	C12E	CON(4)	#E21C
1263	02F73	D12E	CON(4)	#E21D
1264	02F77	E12E	CON(4)	#E21E
1265	02F7B	F12E	CON(4)	#E21F
1266	02F7F	0	CON(1)	#0
1267	02F80	0	CON(1)	#0
1268	02F81	0	CON(1)	#0
1269	02F82	D1	CON(2)	#1D
1270	02F84	14442535	NIBASC	'ADRS '

```

      02
1271 02F8E 2          CON(1) #2
1272 02F8F 422E      CON(4) #E224
1273 02F93 522E      CON(4) #E225
1274 02F97 0          CON(1) #0
1275 02F98 0          CON(1) #0
1276 02F99 0          CON(1) #0
1277 02F9A 50        CON(2) #05
1278 02F9C 44D2C454  NIBASC 'D-LEX'
      85
1279 02FA6 1          CON(1) #1
1280 02FA7 FF00      CON(4) #00FF
1281 02FAB 0          CON(1) #0
1282 02FAC 0          CON(1) #0
1283 02FAD 0          CON(1) #0
1284 02FAE A0        CON(2) #0A
1285 02FB0 74251405  NIBASC 'GRAPH'
      84
1286 02FBA 2          CON(1) #2
1287 02FBB 222E      CON(4) #E222
1288 02FBF 322E      CON(4) #E223
1289 02FC3 FF        CON(2) #FF
1290 02FC5 07        eTBL  C=RSTK
1291 02FC7 135       D1=C
1292 02FCA 8F00000   GOSBVL =FTBSCH
1293 02FD1 5F0       GONC   rtn5
1294 02FD4 D9        C=B    A
1295 02FD6 135       D1=C
1296 02FD9 AC4       A=B    S
1297 02FDC 821       XM=0
1298 02FDF 03        RTNCC
1299 02FE1 00        rtn5  RTNSXM
1300
1301
1302
1303          STITLE pCLDST handler
```

```

1304
1305      * pCLDST handler
1306      *
1307      * Le poll-handler qui suit intercepte le poll
1308      * pCLDST, qui est envoy e apr es
1309      * chaque MEMORY LOST (Ce qui, tout programmeur en
1310      * assembleur le sait bien,
1311      * arrive des dizaines de fois par jour, si ce
1312      * n'est plus. Je m'arr ete l e car on
1313      * va me reprocher de vous effrayer). Car il est
1314      * insupportable de devoir mettre
1315      * un contraste de 7,  teindre le sonnette, passez
1316      * en RADIANS/USER, charger
1317      * quelques LEX (Comme REPEAT, CONTRAST...)... le
1318      * tout manuellement. Aussi j'ai
1319      * cr e CONFIGLX qui fait tout cela pour moi et
1320      * peut, qui plus est, afficher
1321      * Bonjour: c'est extraordinaire ! En un mot, le
1322      * poll pCONFIG est envoy e; CONFIGLX
1323      * l'attrape au vol;  x ute le sous-programme
1324      * BASIC: CONFIG (Merci   Mr Hernest
1325      * Planque !); retourne   l'assembleur; rends le
1326      * bienheureux poll; et voil  mon
1327      * 71 qui me dis bonjour, avec un contraste de
1328      * 7,...
1329      * Nous n' tonnerons personne en pr isant que le
1330      * sous-programme CONFIG doit
1331      *  tre ds un port ind pendant, ainsi que le LEX:
1332      * CONFIGLX (Ou mieux: ds 1 EEPROM HHP)
1333      * Un mot encore: sachez que si vs terminez votre
1334      * ss-prgm CONFIG par ENDALL, le
1335      * fameux poll pCONFIG ne sera pas envoy    d'autre
1336      * lexs. Par ailleurs, le
1337      * proc d  que j'ai d velopp  ds mon LEX pr le
1338      * poll pCONFIG peut- tre  tendu  
1339      * tous les polls rapides (Pour les autres
1340      * r f rez-vs aux IDS vol 1 (un peu
1341      * avant le chap tre sur la tokenization)). Sur,
1342      * ce bon MEMORY LOST.
1343      *
1344      * (Note: CONFIG replaced by ML)
1345
1346 02FE3 DB      Cldst  C=D    A      Sauvegarde ce sacr  D(A)
1347      *          *          ds la pile de retour BASIC
1348 02FE5 DA          A=C    A      (C'est celle qu'utilise
1349      *          *          les ordres GOSUB et POP)
1350 02FE7 8F00000  GOSBVL =PSHUPD (C'est cette routine qui
1351      *          *          se charge de tt le travail)
1352 02FEE 07      C=RSTK      Sauvegarde l'adr de retour
1353      *          *          vers la routine =FPOLL ds la
1354 02FF0 DA          A=C    A      m me pile
1355 02FF2 8F00000  GOSBVL =PSHUPD
1356 02FF9 07      C=RSTK      On recommence avec l'adr
1357      *          *          de retour   la routine =CLDST
1358 02FFB DA          A=C    A
1359 02FFD 8F00000  GOSBVL =PSHUPD
1360 03004 850      ST=1   =PgmRun Puis on allume le drapeau

```

```
1361          *          *          13 qui indique au routines
1362          *          *          internes qu'on ÅxÅcute un
1363          *          *          programme (BASIC ou BIN)
1364 03007 8F00000      GOSBVL =CALBIN  Et l'on se branche sur
1365          *          *          cette routine qui, È peu de
1366          *          *          choses prÉS celle qui est
1367          *          *          chargÅe d'ÅxÅxuter, en BASIC
1368          *          *          un sous-programme. Ci suit
1369          *          *          la description du dit ss-prog
1370 0300E C0          CON(2) 12      Longueur du chaÑnage
1371 03010 00          CON(2) =tCALL  Tokenization de CALL ML
1372 03012 00          CON(2) =tLITRL .
1373 03014 D4C4      NIBASC 'ML'   .ML
1374 03018 00          CON(2) =tPRMEN .
1375 0301A 00          CON(2) =tEOL  .
1376
1377          *
1378          * A ce point du LEX on espere que l'utilisateur
1379          * n'a pas trop perturbÅ le 71
1380          * pour que tout aille comme il faudrait (en
1381          * faisant des POP, ou des POKE ...
1382          *
1383
1384 0301C 8F00000      GOSBVL =POPUPD  Remettons ds la pile de
1385          *          *          retour (interne) les 2 adrs
1386 03023 DB          C=D      A      stockÅes ds la pile de
1387          *          *          retour BASIC
1388 03025 06          RSTK=C
1389 03027 8F00000      GOSBVL =POPUPD  (C'est cette routine qui
1390          *          *          fait le travail, car ns sommes
1391 0302E DB          C=D      A      bien fatiguÅs aujourd'hui)
1392 03030 06          RSTK=C
1393 03032 8F00000      GOSBVL =POPUPD  Puis on retaure D(A), que
1394          *          *          tout bon chasseur de FPOLL se
1395          *          *          doit d'attraper puis, en
1396          *          *          bon chasseur qu'il est,
1397          *          *          remettre intact È sa place)
1398 03039 D2          C=0 A      DÅsarmons carry
1399 0303B E6          C=C+1 A
1400 0303D 00          RTNSXM      Et renvoyons le poll en
1401          *          *          faisant croire que ns n'avons
1402          *          *          fait (XM#0) (Un bon
1403          *          *          chasseur ne doit pas se faire
1404          *          *          remarquer). Disons, +
1405          *          *          sÅrieusement, qu'il ne faut
1406          * EN AUCUN CAS retourner avec XM=0.
1407          * Le poll doit atteindre TOUS les LEX.
1408
1409
1410          STITLE pVER$ handler
```



```
1411
1412          *** pVER$ handler
1413 0303F 11B  VER$      C=R3
1414 03042 135          D1=C
1415 03045 112          A=R2
1416          *** changes
1417          *          D1=D1- 12          rev. D
1418 03048 1CD          D1=D1- 14          rev. X
1419 0304B 137          CD1EX
1420 0304E 8B6          ?C<A A
1421 03051 C1          GOYES VER00
1422 03053 135          D1=C
1423 03056 10B          R3=C
1424          *          LCASC ' JPC:D'      rev. D
1425          *          DAT1=C 12          rev. D
1426 03059 3D8754A3    LCASC ' JPC:Ex'  rev. X
          3405A402
1427 03069 15DD          DAT1=C 14          rev. X
1428          *** end of changes
1429
1430 0306D 00          VER00  RTNSXM
1431
1432
1433          STITLE pEDIT & pMRGE2 handlers
```

```
1434
1435 *****
1436 *
1437 * EDIT F1
1438 * F1 LEX 2326 01/01/00 00:00
1439 * MERGE F2
1440 *
1441 *****
1442
1443 0306F 3400000 LEX? LC(5) =fLEX
1444 03076 8A2 ?C=A A
1445 03079 00 RTNYES Retour avec Cy=1 si Lex
1446 0307B 01 RTN Retour avec Cy=0 sinon
1447
1448 *** pEDIT handler
1449 0307D 7EEF Edit GOSUB LEX? Fichier Lex ?
1450 03081 440 GOC rtn6 Oui. On autorise EDIT
1451 03084 00 RTNSXM Non. On ignore
1452 03086 03 rtn6 RTNCC Cy=0, pas d'erreur
1453
1454
1455 *
1456 * P = 0
1457 * D1 = ^ en-tete de F2
1458 * A(A) = type de F2
1459 * CURRST = ^ en-tete de F1
1460 *
1461
1462 *** pMRGE2 handler
1463 03088 73EF Mrge2 GOSUB LEX? F2 = LEX ?
1464 0308C 440 GOC hMRG20 oui
1465 0308F 00 RTNSXM non
1466 03091 17F hMRG20 D1=D1+ 16
1467 03094 173 D1=D1+ 4 D1 := ^ protection de F1
1468 03097 1530 A=DAT1 P
1469 0309B 1C3 D1=D1- 4
1470 0309E 1CF D1=D1- 16
1471 030A1 302 LC(1) 2
1472 030A4 0E06 A=A&C P
1473 030A8 90C ?A#0 P
1474 030AB D2 GOYES fprot
1475 *
1476 * F2 n'est pas prive
1477 *
1478 030AD 1B00000 D0=(5) =CURRST CURRST = adresse de F1
1479 030B4 146 C=DAT0 A
1480 030B7 134 D0=C D0 := ^ F1
1481 030BA 16F D0=D0+ 16 D0 := type de F1
1482 030BD D0 A=0 A
1483 030BF 15A3 A=DAT0 4 A(A) := type de F1
1484 030C3 78AF GOSUB LEX?
1485 030C7 440 GOC hMRG30
1486 030CA 00 RTNSXM F1 n'est pas un LEX
1487 *
1488 * F2 est bien du type LEX
1489 *
1490
```

```

1491 030CC 163      hMRG30  D0=D0+ 4      D0 := ^ protection de F1
1492 030CF 1524     A=DAT0 S
1493 030D3 948      ?A=0 S
1494 030D6 C0       GOYES hMRG35
1495
1496 030D8 330000   fprot  LC(4) =eFPROT File protect
1497 030DE 6C60     GOTO bserr
1498
1499 * F1 n'est ni prive, ni securise
1500 *
1501 030E2 133      hMRG35  AD1EX      R1 := D1 (^ entete de F2)
1502 030E5 101      R1=A      :
1503 030E8 16B      D0=D0+ 12  D0 := ^ FILEEND de F1
1504 030EB 136      CD0EX     R0 := D0 (^ FILEEND de F1)
1505 030EE 108      R0=C
1506 030F1 8F00000   GOSBVL =PSHUPD adr. de F2 a reactualiser
1507 030F8 111      A=R1      A(A) := ^ en-tete de F2
1508 030FB D2       C=0 A
1509 030FD 3102     LC(2) 32  Offset de FILEEND
1510 03101 CA       A=A+C A
1511 03103 131      D1=A      D1 := FILEEND de F2
1512 03106 143      A=DAT1 A
1513 03109 3150     LC(2) 5
1514 0310D EA       A=A-C A   Taille de F2 - 5
1515 0310F D8       B=A A
1516 03111 102     R2=A
1517
1518 * D1 = ^ FILEEND de F2
1519 * B(A) = A(A) = Longueur exacte de ce qu'on ajoute
1520 * R0 = ^ FILEEND de F1
1521 * R1 = ^ en-tete de F2
1522 * R2 = Longueur exacte de ce qu'on rajoute
1523 *
1524 03114 110      A=R0      D0 := ^ FILEEND de F1
1525 03117 130      D0=A      :
1526 0311A 11A      C=R2      RSTK := taille a ajouter
1527 0311D 06      RSTK=C    :
1528 0311F D6      C=A A     RSTK := ^ FILEEND de F1
1529 03121 06      RSTK=C    :
1530 03123 142     A=DAT0 A  A(A) := taille de F1
1531 03126 136     CD0EX     C(A) := ^ FILEEND de F1
1532 03129 134     D0=C      :
1533 0312C C2      C=C+A A  C(A) := ^ fichier après F1
1534 0312E 18F     D0=D0- 16 D0 := ^ en-tete de F1
1535 03131 18F     D0=D0- 16 :
1536 03134 132     AD0EX     A(A) := ^ en-tete de F1
1537 03137 DE      CAEX A
1538 03139 103     R3=A      Sauvegarde de la fin de F1
1539
1540 * A(A) = ^ fin du fichier F1
1541 * B(A) = longueur a deplacer
1542 * C(A) = ^ en-tete de F1
1543 *
1544 0313C 8F00000   GOSBVL =MGOSUB
1545 03143 00000     CON(5) =MVMEM+
1546 03148 590      GONC hMRG50
1547 0314B 8D00000   bserr  GOVLNG =BSERR

```

```
1548      *
1549      * A ce stade, un trou a ete cree a la fin de F1
1550      * pour y loger F2. Le travail suivant consiste
1551      * a y amener F2.
1552      * Note pour la comprehension generale: le
1553      * fichier F1 n'a pas bouge. Seul F2, peut-être...
1554      * C'est la raison pour laquelle l'adresse
1555      * de F2 a ete placee dans la pile des GOSUBS.
1556      *
1557 03152 07      hMRG50  C=RSTK          C(A) := ^ FILEEND de F1
1558 03154 108      R0=C          R0 := ^FILEEND de F1
1559 03157 07      C=RSTK          C(A) := taille de F2
1560 03159 10A      R2=C          R2 := taille de F2
1561 0315C 8F00000  GOSBVL =POPUPD  l'adresse de F2
1562 03163 DB      C=D A
1563 03165 109      R1=C          R1 := ^ FILEEND de F2
1564      *
1565      * R0 = ^ FILEEND de F1
1566      * R1 = ^ FILEEND de F2
1567      * R3 = ^ debut du trou (^ fin de F1)
1568      *
1569      * Deplacement du fichier a merger
1570      *
1571 03168 11B      C=R3
1572 0316B 135      D1=C          D1 := Start of des.
1573 0316E D2      C=0 A
1574 03170 3152    LC(2) 37
1575 03174 111      A=R1
1576 03177 CA      A=A+C A
1577 03179 130      D0=A          D0 := Start of source (F2)
1578 0317C 11A      C=R2          Block length (long. de F2)
1579 0317F 8F00000 GOSBVL =MOVEU3
1580      *
1581      * Une partie du fichier F2 a ete maintenant amenee
1582      * dans le trou laisse a la fin de F1. Il ne reste
1583      * plus qu'a actualiser le chainage des lex, et ce
1584      * sera termine
1585      *
1586 03186 118      C=R0          C(A) := ^ FILEEND de F1
1587 03189 134      D0=C
1588 0318C 16A      D0=D0+ 11      D0 := ^ Lex Chain
1589 0318F 5D0      GONC hMRG70  B.E.T
1590      *
1591      * La boucle suivante parcourt la "lex-chain"
1592      * jusqu'a trouver une entree nulle, signifiant
1593      * ainsi la fin de la recherche. Le dernier lex
1594      * de F1 est trouve.
1595      *
1596 03192 136      hMRG60  CD0EX          C(A) := ^ Lex-Chain
1597 03195 C2      C=C+A A
1598 03197 134      D0=C
1599 0319A 165      D0=D0+ 6          D0 := Next-Lex du suivant
1600 0319D 142      hMRG70  A=DAT0 A
1601 031A0 8AC      ?A#0 A
1602 031A3 FE      GOYES hMRG60
1603      *
1604      * La boucle est terminee, D0 pointe sur le
```

```
1605          * "Next-Lex offset" du dernier lex de F1
1606          *
1607 031A5 132          AD0EX          A(A) := ^ next-lex
1608 031A8 130          D0=A
1609 031AB 11B          C=R3          Adresse de la fin de F1
1610 031AE E2          C=C-A A      Offset dernier Lex-Chain
1611 031B0 144          DAT0=C A     Actualisation
1612          *
1613          * Retour a l'appelant, avec la saine sensation
1614          * du travail bien fait. Alleluia !
1615          *
1616 031B3 821          XM=0
1617 031B6 03          RTNCC
1618
1619 031B8          END
```

ACBFND	Ext		-	1068				
ALMSRV	Ext		-	910				
Asfld	Abs	10068	#00002754	-	147	78		
BLDDSP	Ext		-	867				
BSERR	Ext		-	1547				
BYPASS	Abs	11271	#00002C07	-	769	764		
BitsOK	Abs	1	#00000001	-	306	573		
CALBIN	Ext		-	1364				
CHIRP	Ext		-	830				
CK"ON"	Ext		-	763				
CKSREQ	Ext		-	1034				
COMF	Abs	24	#00000018	-	136	201		
CSLC5	Ext		-	277	279			
CSRC5	Ext		-	283	285			
CURRST	Ext		-	1478				
CURSOR	Ext		-	193	238	335	547	822
Cldst	Abs	12259	#00002FE3	-	1346	24		
Confg	Abs	11584	#00002D40	-	1052	22		
DECHEX	Ext		-	733	736			
DEFADR	Ext		-	206	663			
DEFKY	Abs	11495	#00002CE7	-	895	856		
DEFKY+	Abs	11497	#00002CE9	-	897	919		
DSPBFE	Ext		-	474				
DSPBFS	Ext		-	177				
DSPCHX	Ext		-	596	643			
DSPMSK	Ext		-	342	476			
DSPSTA	Ext		-	568				
ESCSTA	Ext		-	607	615			
Edit	Abs	12413	#0000307D	-	1449	70		
EscSt0	Abs	0	#00000000	-	308	616		
EscSt1	Abs	1	#00000001	-	309	608		
FILSK+	Ext		-	739				
FIRSTC	Ext		-	561				
FLGREG	Ext		-	147				
FRCRD2	Abs	11304	#00002C28	-	779	746		
FRCRD3	Abs	11310	#00002C2E	-	781	775		
FRCRD4	Abs	11315	#00002C33	-	783	776		
FRCRD5	Abs	11364	#00002C64	-	800	789		
FRCRDn	Abs	11278	#00002C0E	-	770	745	750	778
FST#0	Abs	10804	#00002A34	-	561	559		
FTBSCH	Ext		-	1292				
FUNCR0	Ext		-	234	245	260		
FXQ	Abs	10150	#000027A6	-	171	154		
Fok	Abs	10137	#00002799	-	167	164		
Fon	Abs	10130	#00002792	-	165	161		
Ftype	Abs	11751	#00002DE7	-	1126	25		
HXDCW	Ext		-	241				
I/OFND	Ext		-	818	1097			
I/ORES	Ext		-	1074				
=ILFART	Abs	10837	#00002A55	-	595	454		
ILSEND	Abs	10920	#00002AA8	-	642	612		
KEYR10	Abs	11478	#00002CD6	-	884	914		
KEYR80	Abs	11514	#00002CFA	-	907	890		
Kydf	Abs	9925	#000026C5	-	81	33		
Kydf9	Abs	10042	#0000273A	-	121	101		
LDCSPC	Ext		-	680				
LEX?	Abs	12399	#0000306F	-	1443	1449	1463	1484

LINEP+	Ext		-	1114					
LIST05	Abs	11093	#00002B55	-	722				
LIST06	Abs	11119	#00002B6F	-	730	721			
LIST07	Abs	11125	#00002B75	-	732	712	727		
LIST40	Abs	11202	#00002BC2	-	753	751			
LIST50	Abs	11222	#00002BD6	-	757	768			
LSTTXT	Abs	11018	#00002B0A	-	699	73			
LSTtxt	Abs	9917	#000026BD	-	73	52			
Left	Abs	10451	#000028D3	-	271	128			
MGOSUB	Ext			-	1544				
MNEMF	Abs	8	#00000008	-	134	195			
MODF	Abs	15	#0000000F	-	135	198			
MOVEU3	Ext			-	1579				
MVMEM+	Ext			-	1545				
Mrge2	Abs	12424	#00003088	-	1463	71			
NXTSTM	Ext			-	769				
NoChFC	Abs	11	#0000000B	-	307	572			
OUTBS	Ext			-	757				
POL40	Abs	10420	#000028B4	-	256	249			
=POLHND	Abs	9814	#00002656	-	31				
POPBUF	Ext			-	893				
POPUPD	Ext			-	755	1384	1389	1393	1561
PRPSND	Ext			-	762				
PSHUPD	Ext			-	1350	1355	1359	1506	
PgmRun	Ext			-	1360				
Pwrof	Abs	11682	#00002DA2	-	1095	23			
R<RSTK	Ext			-	605	853			
RCDSKP	Abs	11312	#00002C30	-	782	752	760		
RESJPC	Abs	194961	#0002F991	-	17	96	155	171	808 1057
RIEN	Abs	10673	#000029B1	-	403	577			
RPTK10	Abs	11577	#00002D39	-	1040	1038			
RPTKY	Abs	11551	#00002D1F	-	1035	854			
RSTK<R	Ext			-	620	901			
RTnsxm	Abs	10681	#000029B9	-	407	356	375	387	
S-R1-1	Ext			-	753				
SETTMO	Ext			-	868				
SFLAG?	Ext			-	229	1036	1104	1107	
SFLAGS	Ext			-	1110				
SKPTBL	Abs	10304	#00002840	-	214	211			
SLEEP	Ext			-	884				
SP05	Abs	10186	#000027CA	-	180	188			
SP06	Abs	10214	#000027E6	-	190	185			
SP08	Abs	10223	#000027EF	-	193	182			
SP10	Abs	10259	#00002813	-	204	197	200		
SWPBYT	Ext			-	784				
SYSFLG	Ext			-	271				
USRREQ	Abs	11537	#00002D11	-	924	907			
USRSTA	Abs	7040	#00001B80	-	839	863	1033		
VER\$	Abs	12351	#0000303F	-	1413	72			
VER00	Abs	12397	#0000306D	-	1430	1421			
VIEWD1	Ext			-	261				
View	Abs	10322	#00002852	-	226	127			
WINDLN	Ext			-	556				
WIPOUT	Ext			-	237				
WRTBLA	Ext			-	1070				
Wtky	Abs	11368	#00002C68	-	808	21			
Wtky2	Abs	11441	#00002CB1	-	850	813	820	827	

asfld	Abs	9921	#000026C1	-	78	84	93		
bALTCH	Ext			-	1071				
bENDUP	Abs	2110	#0000083E	-	1048	1062	1085	1096	
bMARGI	Abs	2109	#0000083D	-	1047	817	1053		
bXEQ	Abs	2050	#00000802	-	1049				
bserr	Abs	12619	#0000314B	-	1547	1497			
cldst	Abs	9802	#0000264A	-	24	45			
cfg	Abs	9790	#0000263E	-	22	39			
eFPROT	Ext			-	1496				
eIVARG	Ext			-	728				
eTBL	Abs	12229	#00002FC5	-	1290	1126			
edit	Abs	9899	#000026AB	-	70	56			
fLEX	Ext			-	1443				
fTEXT	Ext			-	699				
fTMOUT	Abs	3	#00000003	-	883	912			
flPWDN	Ext			-	1103				
flRPTD	Ext			-	1035				
flTNOF	Ext			-	1106	1109			
flUSER	Ext			-	228				
fprot	Abs	12504	#000030D8	-	1496	1474			
ftype	Abs	9808	#00002650	-	25	48			
hKD000	Abs	10044	#0000273C	-	123	114			
hKD001	Abs	10048	#00002740	-	124	117			
hKD002	Abs	10052	#00002744	-	125	120			
hKD003	Abs	10056	#00002748	-	126	108			
hKD010	Abs	10539	#0000292B	-	327	123			
hKD050	Abs	10683	#000029BB	-	432	124			
hKD060	Abs	10715	#000029DB	-	489	525			
hKD062	Abs	10730	#000029EA	-	499	495			
hKD065	Abs	10753	#00002A01	-	521	508			
hKD070	Abs	10759	#00002A07	-	523	477			
hKD080	Abs	10767	#00002A0F	-	536	517			
hKD090	Abs	10775	#00002A17	-	546	540			
hKD100	Abs	10937	#00002AB9	-	663	125	679		
hKD110	Abs	10960	#00002AD0	-	669	667			
hKD150	Abs	10973	#00002ADD	-	677	126			
hKD160	Abs	11012	#00002B04	-	688	686			
hMRG20	Abs	12433	#00003091	-	1466	1464			
hMRG30	Abs	12492	#000030CC	-	1491	1485			
hMRG35	Abs	12514	#000030E2	-	1501	1494			
hMRG50	Abs	12626	#00003152	-	1557	1546			
hMRG60	Abs	12690	#00003192	-	1596	1602			
hMRG70	Abs	12701	#0000319D	-	1600	1589			
i/ores	Abs	11655	#00002D87	-	1074	1054	1063	1072	1086
k#BKSP	Ext			-	295				
k#OFF	Ext			-	916				
kcCALC	Ext			-	91	152			
kcEOL	Ext			-	105				
kcFRT	Ext			-	115				
kcLFT	Ext			-	88				
kcRT	Ext			-	112				
kcRUN	Ext			-	118				
kcVIEW	Ext			-	85				
left	Abs	10064	#00002750	-	128	90			
linep	Abs	11741	#00002DDD	-	1112				
list01	Abs	11032	#00002B18	-	703	701			
mrge2	Abs	9905	#000026B1	-	71	59			

m\$	Abs	10623	#0000297F	-	370	365	368		
out	Abs	11662	#00002D8E	-	1081	1105	1108		
pCLDST	Ext			-	43				
pCONFIG	Ext			-	37				
pEDIT	Ext			-	54				
pFTYPE	Ext			-	46				
pKYDF	Ext			-	31				
pMRGE2	Ext			-	57				
p\$rof	Abs	9796	#00002644	-	23	42			
reste	Abs	10645	#00002995	-	382	362			
rst10	Abs	10654	#0000299E	-	385	390			
rst20	Abs	10667	#000029AB	-	389	384			
rtn	Abs	10148	#000027A4	-	170	150	176	192	
rtn1	Abs	10212	#000027E4	-	189	203			
rtn2	Abs	10320	#00002850	-	225	232			
rtn3	Abs	10537	#00002929	-	298	274	290		
rtn4	Abs	11653	#00002D85	-	1073	1069			
rtn5	Abs	12257	#00002FE1	-	1299	1293			
rtn6	Abs	12422	#00003086	-	1452	1450			
rtncc	Abs	11676	#00002D9C	-	1088	1099			
rtnxm0	Abs	10315	#0000284B	-	218	169			
savebf	Abs	11667	#00002D93	-	1085				
tCALL	Ext			-	1371				
tCOMMA	Ext			-	710	719			
tEOL	Ext			-	1375				
tLITRL	Ext			-	1372				
tPRMEN	Ext			-	1374				
ver\$	Abs	9911	#000026B7	-	72	61			
view	Abs	10060	#0000274C	-	127	87			
wtky	Abs	9786	#0000263A	-	21	36			

Input Parameters

Source file name is polhnd.a
Listing file name is polhnd.l
Object file name is polhnd.o
Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE FINPUT
2
3          * *****
4          * FINPUT module
5          *
6          * Keyword(s):
7          *   FINPUT
8          * Author(s):
9          *   Pierre David & Janick Taillandier
10         * History:
11         *   Published in JPC#43, apr 1987
12         *   Re-typed from paper issue of JPC JOurnal, JFG jan 2006
13         *   Included the changes as implemented in JPC ROM rev D
14         *   One minor change in rev X
15         *
16         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
17         * *****
18
19 031B8          ABS          #31B8
20
21         RESJPC EQU          #2F991          JPC reserved location
22
23         trfmbf EQU          #2F8C5          =TRFMBF
24         stmtr0 EQU          #2F871          =STMTR0
25
26         *
27         * format des "dope vecteuuuurs" made in France
28         *
29         * 00-03 : nombre d'elements (1..n)
30         * 04-08 : pointeur vers les donnees
31         * 09-12 : longueur max des elements
32         *
33         DOPEB$ EQU          (trfmbf)+00          13 quartets
34         DOPEd$ EQU          (trfmbf)+13          13 quartets
35         DOPEP$ EQU          (trfmbf)+26          13 quartets
36         DOPEA EQU          (trfmbf)+26          27 quartets
37         n EQU          (trfmbf)+53          5 quartets
38         *
39         * il reste 2 quartets a (=TRFMBF)+58 qui sont utilises lors
40         * de la compilation (fichier comp.as)
41         *
42
43         DOPEI$ EQU          (stmtr0)+00          13 quartets
44         DOPEM$ EQU          (stmtr0)+13          13 quartets
45         DOPEU$ EQU          (stmtr0)+26          13 quartets
46         * il reste 3 quartets a (=STMTR0)+39
47         * correspondant a (=STMTD1)+02
48
49
50         *****
51         * ELMCPY
52         *
53         * But: copier l'element Y$(I) dans X$(I)
54         * Entree:
55         *   - C(A) = I
56         *   - D0 = dope vecteuuuur de Y$
57         *   - D1 = dope vecteuuuur de X$
```

```

58      * Sortie:
59      *   - A(A) = adresse de Y$(I)
60      *   - B(A) = (LEN(Y$(I)) + 2)*2
61      *   - C(A) = adresse de X$(I)
62      *   - P = 0
63      * Abime: A-D, D0, D1
64      * Appelle: GETx$I, MOVE*M
65      * Niveaux: 1
66      * Historique:
67      *   86/09/05: P.D. & J.T. conception & codage
68      *   86/11/24: P.D.      integration dans bas.as
69      * *****
70
71 031B8 D7      ELMCPY  D=C   A           D(A) := I
72 031BA 7621      GOSUB =GETx$I   D0 := ^ Y$(I)
73 031BE 136      CD0EX
74 031C1 137      CD1EX           D1 := ^ Y$(I)
75 031C4 134      D0=C           D0 := dope vecteuuur de X$
76 031C7 DB      C=D A
77 031C9 7711      GOSUB =GETx$I   C(A) = D0 := ^ X$(I)
78      *
79      * C(A) = D0 = ^ X$(I) = dest. address.
80      * D1 = ^ Y$(I)
81      * et il faut faire X$(I) := Y$(I)
82      *
83
84      *
85      * B(A) := LEN(Y$(I)) + 2)*2
86      *
87 031CD D0      A=0   A
88 031CF 15B3      A=DAT1 4
89 031D3 E4      A=A+1  A
90 031D5 E4      A=A+1  A
91 031D7 C4      A=A+A  A
92 031D9 D8      B=A A
93
94 031DB 133      AD1EX           A(A) := source address (Y$(I))
95
96 031DE 8D0000      GOVLNG =MOVE*M
97
98
99      * *****
100     * evlary
101     *
102     * But: evaluer ce qui est apres la virgule pointee par D0.
103     * Entree:
104     *   - D0 = PC
105     *   - MTHSTK = somment de la Math-Stack
106     * Sortie:
107     *   - D0 = ^ passee l'expression
108     *   - B(W) = ^ dope vecteuuur fabrique
109     *   - MTHSTK actualisee pour prendre en compte une eventuelle
110     *     longueur d'element de tableau, ou pour "oublier" le
111     *     "dope vecteuuur"
112     * Abime: A-D, R0-R4, DST, D0, D1, Function Scratch
113     * Appelle: EXPEXC, RCL4, RCL5, BSL4, BSL5, POP1S
114     * Niveaux: 5 (EXPEXC)

```

```
115          * Historique:
116          *   86/08/30: P.D. & J.T. conception & codage
117          *   86/11/24: P.D.      ajout de documentation
118          * *****
119
120 031E5 8F00000  evlary  GOSBVL =EXPEXC
121 031EC AF1      B=0     W
122 031EF 31F1    LCHEX  1F      tableau alpha?
123 031F3 966    ?C#A   B      non
124 031F6 03     GOYES  evry20   (2 avenue du Lac, en fait...)
125          *
126          * string array descriptor
127          *
128 031F8 172     D1=D1+ 3      D1 := ^ maxlen
129 031FB 7631    GOSUB  RCL4
130 031FF 177     D1=D1+ 8      D1 := ^ pointer
131 03202 7D31    GOSUB  RCL5
132 03206 1C3    D1=D1- 4      D1 := ^ dim
133 03209 7821    GOSUB  RCL4
134 0320D 1C4    D1=D1- 5      D1 := ^ option base
135 03210 1574    C=DAT1 S
136 03214 94E    ?C#0   S
137 03217 90     GOYES  evry10
138 03219 23     P=     3
139 0321B B15    B=B+1  WP
140 0321E 20     P=     0
141 03220 17D    evry10 D1=D1+ 14
142 03223 5C3    GONC   evry90   B.E.T.
143
144          *
145          * Attention :
146          * Cette erreur ne devrait jamais arriver...
147          *
148
149 03226 31F0    evry20 LCHEX  0F      chaine alpha ?
150 0322A 966    ?C#A   B
151 0322D 76     GOYES  datap   Non. mais je ne sais pas comment
152          *                      cela serait possible
153          *
154          * On a trouve une expression alphanumerique sur la pile.
155          * Nous allons la transformer en vecteur de un element.
156          *
157 0322F 8F00000  GOSBVL =POP1S      D1 := ^ sommet de la Math-Stack
158          *
159          * La longueur de la chaine doit tenir sur 2 octets, d'ou le
160          * test de la longueur <20000 quartets
161          *
162 03236 3400002  LC(5) #20000
163 0323D 8BE    ?C<=A A
164 03240 B5     GOYES  strovf
165          *
166          * C'est une chaine, et elle est de bonne longueur
167          *
168 03242 81C    ASRB      A(A) := longueur en octets
169 03245 1C3    D1=D1- 4      D1 := ^ longueur de V$(1)
170 03248 1593   DAT1=A 4      LEN(V$(1)) := LEN(chaine)
171 0324C D8     B=A A
```

```

172 0324E 72D0          GOSUB  BSL5
173 03252 137          CD1EX
174 03255 135          D1=C          C(A) = D1 := ^ math-stack
175 03258 D5           B=C A
176 0325A 79C0         GOSUB  BSL4
177 0325E E5           B=B+1  A
178
179
180 *
181 * terminaison de evlary. On attend dans B(W) le "dope
182 * vecteruuuuur" et dans D1 l'adresse du sommet de la
183 * Math-Stack. Cette adresse sera sauvegardee en MTHSTK
184 03260 137          evry90  CD1EX          Evry Cedex en fait
185 03263 1F00000      D1=(5) =MTHSTK
186 0326A 145          DAT1=C A          Sommet de la Math-Stack
187
188 *
189 * Verification du nombre d'elements du tableau en cours
190 * avec I$
191 0326D 1FAF8F2      D1=(5) n
192 03274 143          A=DAT1 A
193 03277 23           P= 3
194 03279 914          ?A#B  WP
195 0327C 90           GOYES  erdim
196 0327E 20           P= 0
197
198 *
199 * On remet les choses en place
200 03280 135          D1=C          D1 := sommet de la Math-Stack
201 03283 01           RTN
202
203 03285 20           erdim  P= 0
204 03287 33601E       LC(4)  #E106      (=id)~(=eIDIM)  "#Dims"
205 0328D 8D00000      GOVLNG =BSERR
206
207 03294 8D00000      datap  GOVLNG =RDATTY
208 0329B 3100         strovf  LC(2)  =eSTROV
209 0329F 8D00000      GOVLNG =MFERR
210
211 *****
212 * GETM$I, GETI$I, GETU$I, GETB$I, GETD$I, GETP$I, GETx$I
213 *
214 * But: renvoyer l'adresse de l'element I du tableau M$
215 * (resp. I$, U$, B$, D$, P$).
216 * Entree:
217 * - C(A) = I
218 * Sortie:
219 * - C(A) = D0 = ^ long. de T$(I) (T$=M$,I$,U$,B$,D$,P$)
220 * Abime: A-C, D0
221 * Appelle: MPY
222 * Niveaux: 1
223 * Detail: La routine GETx$I permet de retrouver l'element I
224 * du tableau dont le dope vecteuuuuuur est pointe par D0
225 * Historique:
226 * 86/08/31: P.D. & J.T. conception & codage
227 *****
228

```

```

229 032A6 1BFD8F2  GETP$I  D0=(5) DOPEP$
230 032AD 6630      GOTO   GET010
231 032B1 1B2D8F2  GETD$I  D0=(5) DOPED$
232 032B8 6B20      GOTO   GET010
233 032BC 1B5C8F2  GETB$I  D0=(5) DOPEB$
234 032C3 6020      GOTO   GET010
235 032C7 1BB88F2  GETU$I  D0=(5) DOPEU$
236 032CE 6510      GOTO   GET010
237 032D2 1BE78F2  GETM$I  D0=(5) DOPEM$
238 032D9 6A00      GOTO   GET010
239 032DD 1B178F2  GETI$I  D0=(5) DOPEI$
240 032E4      GETx$I
241 032E4 CE      GET010  C=C-1  A
242 032E6 D5          B=C  A      !
243 032E8 AF2      C=0  W      ! C(15-5) := 0
244 032EB D9      C=B  A      !
245 032ED 168      D0=D0+ 9    D0 := ^ maxlen
246 032F0 AF0      A=0  W
247 032F3 15A3      A=DAT0 4    A(W) :=maxlen
248 032F7 E4      A=A+1  A
249 032F9 E4      A=A+1  A    A(W) :=maxlen + longueur
250 032FB 7A92      GOSUB mpy  A,B,C := offset en octets
251 032FF C4      A=A+A  A    A(A) := offset en quartets
252 03301 184      D0=D0- 5    D0 := ^ pointeur
253 03304 146      C=DAT0 A    C(A) := ^ T$(1)
254 03307 C2      C=C+A  A    C(A) := ^ T$(I)
255 03309 134      D0=C      D0 := ^ T$(I)
256 0330C 01      RTN
257
258
259 *****
260 * GET$ST
261 *
262 * But: renvoyer l'adresse du premier caractere d'un element
263 * de tableau
264 * Entree:
265 * - D0 = adresse de la longueur de l'element
266 * Sortie:
267 * - C(A) = D0 = adresse du premiere caractere de l'element
268 * sus-cite. (cf Math-Stack, il faut faire D0=D0+ 2 avant
269 * utilisation
270 * - A(A) = D0 en entree
271 * Abime: A(A), C(A), D0
272 * Niveaux: 0
273 * Historique:
274 * 86/09/02: P.D. & J.T. conception & codage
275 *****
276 0330E D2      GET$ST  C=0  A
277 03310 15E3      C=DAT0 4
278 03314 E6      C=C+1  A
279 03316 E6      C=C+1  A
280 03318 C6      C=C+C  A
281 0331A 132      AD0EX
282 0331D C2      C=C+A  A
283 0331F 134      D0=C
284 03322 01      RTN
285

```

```
286 *****
287 * BSL4, BSL5
288 *
289 * But: decaler B a gauche de 4 (resp. 5) quartets
290 * Entree:
291 * - B(W)
292 * Sortie:
293 * - B(W)
294 * Abime: B(W)
295 * Niveaux: 0
296 * Historique:
297 * 86/08/31: P.D. & J.T. conception & codage
298 *****
299
300 03324 BF1 BSL5 BSL W
301 03327 BF1 BSL4 BSL W
302 0332A BF1 BSL W
303 0332D BF1 BSL W
304 03330 BF1 BSL W
305 03333 01 RTN
306
307 *****
308 * RCL4, RCL5
309 *
310 * But: decale B(W) de 4 (resp. 5) quartets, et y placer les 4
311 * (resp. 5) quartets pointes par D1.
312 * Entree:
313 * - B(W) = registre a decalage
314 * - D1 = zone a inserer dans B
315 * Sortie:
316 * - B(W) decale
317 * Abime: B(W), C(A)
318 * Appelle: BSL4 (resp. BSL5)
319 * Niveaux: 1
320 * Historique:
321 * 86/08/30: P.D. & J.T. conception & codage
322 *****
323
324 03335 7EEF RCL4 GOSUB BSL4
325 03339 D2 C=0 A
326 0333B 15F3 C=DAT1 4
327 0333F C1 B=B+C A
328 03341 01 RTN
329 03343 7DDF RCL5 GOSUB BSL5
330 03347 147 C=DAT1 A
331 0334A D5 B=C A
332 0334C 01 RTN
333
334
335 *****
336 * FINPUe
337 *
338 * But: implementer un gestionnaire de masque d'ecran
339 * Abime: tout ce qui est abimable par un statement
340 * Niveaux: tous ceux qui sont abimables par un statement
341 * Algorithmme:
342 * evaluer I$, creer le dope vecteuuuuuur
```



```
343      *   evaluer M$, creer le dope vecteuuuuuur
344      *   si P$ n'existe pas
345      *       alors creer un pseudo P$ : P$(1) :=STR$(LEN(M$(1))&"F"
346      *       sinon evaluer P$, creer le dope vecteuuuuuur
347      *   finsi
348      *   initialiser le tableau D$ (D$(1) := M$(I))
349      *   initialiser le tableau U$ (U$(1) := NULL$)
350      *   compiler le format dans le tableau B$ (Bit Map)
351      *   evaluer A
352      *   verifier que M$ ne contient pas de car. non affichable
353      *   I := 1
354      *   repeter
355      *       afficher la chaine D$(I)
356      *       (-)
357      *       placer B$(I) dans DSPMSK
358      *       editer la ligne
359      *       selon la derniere touche appuyee :
360      *           [^]: I :- si I>1 alors I=1 ;
361      *           [v]: I :- si I>1 alors I=1 ;
362      *           [g][^]: I := 1 ;
363      *           [g][v]: I := n ;
364      *           [OFF]: sortie avec A := 0
365      *           [ATTN]: si display = M$(I)
366      *                   alors sortie avec A := 0 ;
367      *                   sinon
368      *                       afficher la chaine M$(I) ;
369      *                       reprendre a (-) ;
370      *                   finsi
371      *           [ENDLINE]: U$(I) := display
372      *                       si I=1 alors continuer dans [RUN]
373      *           [RUN]: U$(I) := display
374      *                   sortir avec A := 1 ;
375      *   fin selon
376      *   fin repeter
377      * Historique:
378      *   86/08/30: P.D. & J.T. conception & codage
379      *   86/11/24: P.D. ajout de documentation
380      * *****
381
382
383      * *****
384      * FINPUT entry
385      * *****
386 0334E 00000      REL(5) =FINPUd
387 03353 00000      REL(5) =FINPUp
388 03358           =FINPUe
389      *
390      * Traitement de I$
391      *
392 03358 8F00000      GOSBVL =ADDRSS
393      *
394      * A la sortie de ADDRSS,
395      * si Cy = 1, la variable n'est pas trouvee
396      * si CY = 0, la variable est trouvee, et :
397      *   D0 = B(A) = adresse du registre de la variable
398      *   A(A) = ^ passee la tokenisation de la variable
399      *
```

```

400 0335F 5F0          GONC   I$005
401 03362 33701E      LC(4)  #E107          "Var Not Found"
402 03368 8D00000     GOVLNG =BSERR
403
404 0336F 136         I$005  CD0EX          C(A) := ^ registre de la variable
405 03372 135         D1=C          D1 := ^ registre de la variable
406 03375 130         D0=A          D0 := PC
407
408 *
409 * Boucle pour trouver le registre reel de la variable
410 03378 30E         LCHEX E
411 0337B 170         I$010  D1=D1+ 1
412 0337E 1530       A=DAT1 P          A(P) := type de la variable
413 03382 906         ?C#A  P          Tableau indirect
414 03385 E0          GOYES  I$020
415 03387 170         D1=D1+ 1          D1 := ^ "pointer"
416 0338A 143         A=DAT1 A          A(A) := ^ reg. de la var. suivant
417 0338D 131         D1=A          D1 := ^ reg. de la var. suivant
418 03390 5AE         GONC   I$010          B.E.T.
419
420 *
421 * Fin de chainage
422 03393 1C0         I$020  D1=D1- 1          D1 := ^ deux premier quartets
423 03396 AF1         B=0    W
424 03399 14B         A=DAT1 B          A(B) := type de la variable
425 0339C 31F1       LCHEX  1F          Tableau alphanumerique
426 033A0 966         ?C#A  B
427 033A3 C3          GOYES  I$040
428
429 *
430 * Tableau (direct ou indirect)
431 033A5 172         D1=D1+ 3
432 033A8 798F       GOSUB  RCL4          Maxlen
433 033AC 177         D1=D1+ 8          D1 := ^ "pointer"
434 033AF 709F       GOSUB  RCL5          relative pointer
435 033B3 133         AD1EX
436 033B6 131         D1=A          A(A) := adresse du relative pointer
437 033B9 E0         A=A-B  A          A(A) := ^ premier element
438 033BB D8         B=A  A          B(8-5) := maxlen ; B(4-0) := pointer
439 033BD 1C3         D1=D1- 4          D1 := ^ dim
440 033C0 717F       GOSUB  RCL4
441 033C4 1C4         D1=D1- 5          D1 := ^ option base
442 033C7 1574       C=DAT1 S
443 033CB 94E        ?C#0  S
444 033CE 90         GOYES  I$030
445 033D0 23         P=     3
446 033D2 B15        B=B+1  WP          = 1 en option base 0
447 033D5 20         P=     0
448 033D7 6E40       I$030  GOTO   I$100
449
450 033DB 68BE       Datap  GOTO   datap
451
452 033DF 31F0       I$040  LCHEX  0F
453 033E3 966        ?C#A  B
454 033E6 B1         GOYES  I$050
455
456 *
* variable directe

```

```

457          *
458 033E8 172          D1=D1+ 3          D1 := ^ maxlen
459 033EB 764F        GOSUB RCL4
460 033EF 177          D1=D1+ 8          D1 := ^ relative pointer
461 033F2 7D4F        GOSUB RCL5          relative pointer
462 033F6 133          AD1EX          A(A) := adresse du relative pointer
463 033F9 E0          A=A-B A          A(A) := ^ premier element
464 033FB D8          B=A A          B(8-5) := maxlen ; B(4-0) := pointer
465 033FD 6A10        GOTO I$060
466
467          *
468          * Cette erreur ne devrait jamais arriver !
469          *
470 03401 31FF        I$050 LCHEX FF
471 03405 966          ?C#A B
472 03408 3D          GOYES Datap
473          *
474          * variable indirecte
475          *
476 0340A 176          D1=D1+ 7          D1 := ^ maxlen
477 0340D 742F        GOSUB RCL4
478 03411 1C4          D1=D1- 5          D1 := ^ absolute pointer
479 03414 7B2F        GOSUB RCL5
480 03418 BF1          I$060 BSL W
481 0341B BF1          BSL W
482 0341E BF1          BSL W
483 03421 BF1          BSL W
484 03424 E5          B=B+1 A          Dim := 1
485
486 03426 AF9          I$100 C=B W
487 03429 1F178F2    D1=(5) =DOPEI$
488 03430 15DC          DAT1=C 13
489 03434 24          P= 4
490 03436 A82          C=0 P
491 03439 20          P= 0
492 0343B 1EAF8F      D1=(4) n          Tant que =n reste en TRFMBF+53,
493 03441 145          DAT1=C A          D1=(2) aurait suffi
494          *
495          * A ce stade, la variable I$ est traitee (ouf)
496          *
497          * J.T. : "YapuquaM$" (je cite)
498          *
499 03444 8F00000      GOSBVL =COLLAP      Vide la Math-Stack
500 0344B 161          D0=D0+ 2
501 0344E 739D        GOSUB evlary          evaluation de M$
502 03452 1FE78F2    D1=(5) DOPEM$
503 03459 AF4          A=B W
504 0345C 159C          DAT1=A 13
505          *
506          * P$ ou A ?
507          *
508 03460 161          D0=D0+ 2          On passe tCOMMA
509 03463 1524          A=DAT0 S          A(S) := quartet de reconnaissance
510 03467 160          D0=D0+ 1
511 0346A 948          ?A=0 S          A ?
512 0346D A1          GOYES PP$010          Pseudo P$
513          *

```

```

514          * On est devant P$
515          *
516 0346F 727D          GOSUB  evlary          evaluation de P$
517 03473 161          D0=D0+ 2          J.T. & P.D. bugfix <861123.1415>
518 03476 136          CD0EX          Sauvegarde de PC dans R3 pour la
519 03479 10B          R3=C          compilation
520 0347C 69C0          GOTO   compilerP$
521
522 03480 8D00000  stkchr  GOVLNG =STKCHR
523          *
524          * On est devant A. Il faut creer un pseudo P$ par la boucle:
525          *   FOR I = nombre-d'elements(M$) STEP -1
526          *     P$(I) = STR$(LEN(M$(I))&"PU"
527          *   NEXT I
528          * Ce tableau P$ est construit sur la Math-Stack
529          *
530
531          *****
532          * PP$010
533          *
534          * But: creer un pseudo-P$ dans le cas ou ce tableau est
535          *   omis par l'utilisateur
536          * Historique:
537          *   86/08/30: P.D. & J.T. conception & codage
538          *   87/01/25: P.D. & J.T. remplace S-R0-0 par TRFMBF
539          *****
540
541 03487 136          PP$010  CD0EX          Sauvegarde de PC dans R3
542 0348A 10B          R3=C
543 0348D 8F00000          GOSBVL  =D1MSTK
544 03494 1BE78F2          D0=(5) DOPEM$
545 0349B D2          C=0   A
546 0349D 15E3          C=DAT0 4          C(A) := nombre d'elements de M$
547 034A1 1900          D0=(2) =TRFMBF
548          *
549          * Debut de la boucle ci-dessus expliquee.
550          * D0 = TRFMBF
551          * TRFMBF = I
552          * P$(I) sera depose sur la pile (pointee par D1)
553          *
554 034A5 144          PP$050  DAT0=C A          C(A) := I
555 034A8 762E          GOSUB  GETM$I          D0 := ^ M$(I) (sur la longueur)
556 034AC D0          A=0 A
557 034AE 15A3          A=DAT0 4          A(A) := longueur en hexa
558 034B2 8F00000          GOSBVL  =HEXDEC          A(W) = B(W) = C(W) := long. en dec.
559 034B9 04          SETHEX
560 034BB 814          ASRC
561 034BE 814          ASRC
562 034C1 814          ASRC
563 034C4 814          ASRC
564
565 034C7 8F00000          GOSBVL  =D=AVMS          D(A) := AVMEMS
566 034CE 304          LC(1) 4
567 034D1 816          CSRC          C(S) := 4
568          *
569          * Boucle interne de formattage ascii de la longueur de M$(I)
570          * On aura, si M$="AREUH" : P$ := "00005PU"

```

```

571          *
572 034D4 3103  PP$070  LCASC  '0'
573 034D8 A62          C=C+A  B
574 034DB D0          A=0    A
575 034DD 7F9F        GOSUB  stkchr
576 034E1 810          ASLC
577 034E4 A4E          C=C-1  S
578 034E7 5CE          GONC  PP$070
579 034EA 3105        LCASC  'P'
580 034EE 7E8F        GOSUB  stkchr
581 034F2 3155        LCASC  'U'
582 034F6 768F        GOSUB  stkchr
583          *
584          * On met la longueur de P$(I) sur la pile
585          *
586 034FA D2          C=0    A
587 034FC 307          LC(1)  7          LEN(P$(I)) := 7
588 034FF 1C3          D1=D1- 4
589 03502 15D3        DAT1=C  4
590          *
591          * On passe a l'element suivant de M$ (M$(I+1))
592          *
593 03506 1B00000      D0=(5) =TRFMBF
594 0350D 146          C=DAT0  A
595 03510 CE          C=C-1  A
596 03512 8AE          ?C#0   A
597 03515 09          GOYES  PP$050
598          *
599          * Creation du "dope vecteuuur"
600          *
601
602 03517 AF1          B=0    W
603 0351A 307          LC(1)  7
604 0351D A85          B=C    P          B(3-0):=maxlen
605
606 03520 137          CD1EX
607 03523 135          D1=C
608 03526 7AFD        GOSUB  BSL5
609 0352A D5          B=C    A          B(0-5):=maxlen;B(4-0):=pointer
610
611 0352C 1BE78F2      D0=(5) DOPEM$
612 03533 D2          C=0    A
613 03535 15E3        C=DAT0  4
614 03539 7AED        GOSUB  BSL4
615 0353D C1          B=B+C  A          B(12-9):=maxlen ;
616          *          B(8-4):=pointer ;
617          *          B(3-0):=dim
618 0353F 8F00000      GOSBVL =AVE=D1
619
620          *****
621          * compilerP$
622          *
623          * But: compiler le format donnee par l'utilisateur (ou
624          *   construit par SINPUT) en une Bit Map utilisable par le
625          *   display driver du HP71 (DSPMSK)
626          * Entree:
627          *   - DOPEP$ contient le dope vecteuuur de P$

```

```
628          * Sortie:
629          *   - B$ contient le tableau des Bit Map
630          * Abime:
631          * Appelle: fichier comp.as
632          * Niveaux: 1
633          * Historique:
634          *   86/11/24: P.D.          ajout de documentation
635          *****
636
637 03546      compilerP$
638 03546 1BFD8F2      D0=(5) DOPEP$
639 0354D AF4          A=B      W
640 03550 158C          DAT0=A 13
641
642          *
643          * Preparation des tableaux
644          * B$ (bit map) : Bit mask de DSPMSK
645          *   chargement : lors de la compilation (seulement lors)
646          *   utilisation: POKE dans DSPMASK
647          *   DIM B$(n)[12]
648          * D$ (Display) : copie des caracteres du DSPBUF
649          *   chargement : apres appui sur [ENDLINE]
650          *   utilisation: reaffichage du champ par une routine de
651          *   type DSPCMA, mais inversee
652          *   DIM D$(n)[96]
653          * U$ (User) : donnee saisie par l'utilisateur
654          *   chargement : apres appui sur [ENDLINE], par DSP$00
655          *   utilisation: lors de l'appui sur [RUN], chargement de
656          *   I$
657          *   DIM U$(n)[96]
658          *
659
660          * Les tableaux sont ranges entre OUTBS et AVMEMS
661          *
662 03554 8F00000      GOSBVL =OBCOLL
663          *
664          * Tout d'abord, y-a-t'il assez d'espace ?
665          *
666 0355B AF2          C=0      W
667 0355E 324A1      LC(3) (96+2+96+2+12+2)*2
668 03563 1BAF8F2      D0=(5) n
669 0356A AF0          A=0      W
670 0356D 142          A=DAT0 A
671 03570 7520          GOSUB mpy          A, B, C := taille memoire necessaire
672 03574 D1           B=0      A
673 03576 97D          ?B#0    W
674 03579 91           GOYES memerr
675          *
676          * Oui, il y a suffisamment d'espace
677          *
678 0357B 1B00000      D0=(5) =AVMEMS
679 03582 146          C=DAT0 A
680 03585 C2           C=C+A A          C(A) := nouvel AVMEMS
681 03587 164          D0=D0+ 5          D0=(5) AVMEME
682 0358A 142          A=DAT0 A          A(A) := AVMEME
683 0358D 8B6          ?C<A A
684 03590 01           GOYES cmp10      Ok
```

```

685
686 03592 8D00000 memerr GOVLNG =MEMERR
687 03599 8D00000 mpy GOVLNG =MPY
688
689 035A0 184 cmp10 D0=D0- 5 D0=(5) AVMEMS
690 035A3 142 A=DAT0 A A(A) := adresse de B$
691 035A6 144 DAT0=C A AVMEMS := adresse de la fin de U$
692 035A9 1BAF8F2 D0=(5) n
693 035B0 1F5C8F2 D1=(5) DOPEB$
694
695 * A(A) = ^ premier element de B$
696
697 035B7 AF1 B=0 W
698 035BA D2 C=0 A
699 035BC 31C0 LC(2) 12 Taille d'un element
700 035C0 D5 B=C A
701 035C2 7E5D GOSUB BSL5
702 035C6 D8 B=A A B(A) := adresse de B$(1)
703 035C8 7B5D GOSUB BSL4
704 035CC 146 C=DAT0 A C(A) := n
705 035CF C1 B=B+C A
706 035D1 AF9 C=B W
707 035D4 15DC DAT1=C 13
708 035D8 17C D1=D1+ 13
709
710 035DB AF2 C=0 W
711 035DE 31C1 LC(2) (12+2)*2
712 035E2 AFE ACEX W
713 035E5 D7 D=C A D(A) := adresse de B$(I)
714 035E7 AF2 C=0 W
715 035EA 146 C=DAT0 A
716 035ED 78AF GOSUB mpy
717 035F1 CB C=C+D A
718 035F3 DA A=C A
719 * D$
720 035F5 AF1 B=0 W
721 035F8 D2 C=0 A
722 035FA 3106 LC(2) 96
723 035FE D5 B=C A
724 03600 702D GOSUB BSL5
725 03604 D8 B=A A B(A) := adresse de D$(1)
726 03606 7D1D GOSUB BSL4
727 0360A 146 C=DAT0 A C(A) := n
728 0360D C1 B=B+C A
729 0360F AF9 C=B W
730 03612 15DC DAT1=C 13
731
732 03616 AF2 C=0 W
733 03619 314C LC(2) (96+2)*2
734 0361D AFE CAEX W
735 03620 D7 D=C A D(A) := adresse de D$(1)
736 03622 AF2 C=0 W
737 03625 146 C=DAT0 A
738 03628 7D6F GOSUB mpy
739 0362C CB C=C+D A
740 0362E DA A=C A
741 * U$

```

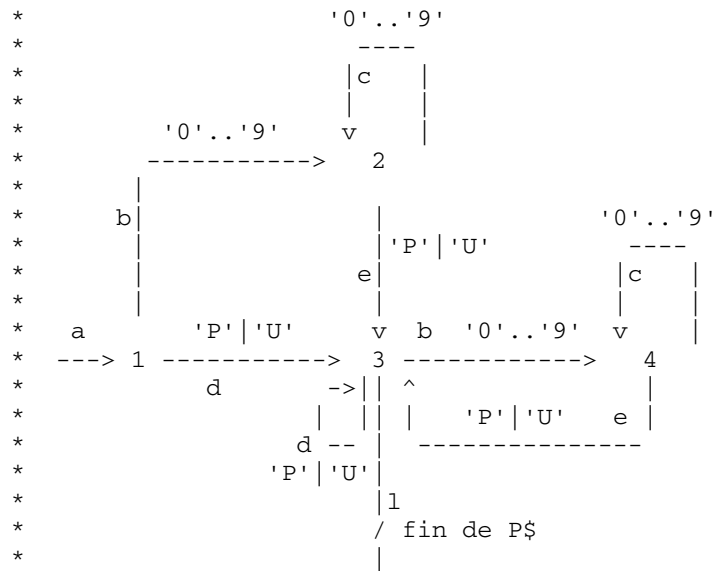
```
742 03630 1DB8          D1=(2) DOPEU$
743 03634 AF1          B=0    W
744 03637 D2           C=0    A
745 03639 3106         LC(2)  96
746 0363D D5           B=C    A
747 0363F 71EC         GOSUB  BSL5
748 03643 D8           B=A    A      B(A) := adresse de U$(1)
749 03645 7EDC         GOSUB  BSL4
750 03649 146          C=DAT0 A      C(A) := n
751 0364C C1           B=B+C  A
752 0364E AF9          C=B    W
753 03651 15DC         DAT1=C 13
754
755                    * end of bas.as
756
757                    * start of comp.as
758
759                    STITLE Compilation du format
```



```

760      *
761      * La boucle suivante est la boucle de compilation de P$
762      *
763      * FOR I = 1 TO nombre-d'elements(P$)
764      *   appliquer l'automate ;
765      *   D$(I) := M$(I)
766      * NEXT I
767      *
768
769      funcr0 EQU    #2F89B          =FUNCR0
770
771      i      EQU    (funcr0)+00     5 quartets
772      ctp$   EQU    (funcr0)+05     5 quartets
773      addsp$ EQU    (funcr0)+10     5 quartets
774      addsu$ EQU    (funcr0)+15     5 quartets
775      cptm$  EQU    (funcr0)+20     2 quartets
776      addsm$ EQU    (funcr0)+22     5 quartets
777      mask   EQU    (funcr0)+27     1 quartet
778      admask EQU    (funcr0)+28     5 quartets
779      cptmsk EQU    (funcr0)+33     2 quartets
780      mult   EQU    (funcr0)+35     2 quartets
781      lastUP EQU    (funcr0)+37     1 quartet
782      * il reste 1 quartet a (=FUNCR0)+41
783      credit EQU    (trfmbf)+58     2 quartets
784      * il ne reste plus rien en TRFMBF !
785
786      *
787      * Les variables k et p utilisees dans "bitmap" sont en
788      * vark : (=FUNCR0)+38 sur 2 quartets
789      * varp : (=FUNCR0)+40 sur 1 quartet
790      *
791
792      *
793      *
794      *
795      *
796      *
797      *
798      *
799      *
800      *
801      *
802      *
803      *
804      *
805      *
806      *
807      *
808      *
809      *
810      *
811      *
812      *
813      * Actions de l'automate :
814      *
815      * a: Initialisation de l'automate
816      *

```



```
817      *   ctp$ := LEN(GETP$I) ;
818      *   addsp$ := GETFST (GETPS$I) ;
819      *   mask := 2^3 ;
820      *   cptmsk := 96 ;
821      *   credit := min (MAXLEN(I$),96) ;
822      *   admask := GETB$I + (12+2 +4 +1) quartets ;
823      *   (admask) := 0 ;
824      *   addsu$ := GETU$I + 4 quartets ;
825      *   l := LEN(GETM$I) ;
826      *   si l>96 alors "String Overflow" finsi ;
827      *   cptm$ := 1 ;
828      *   addsm$ := GETFST (GETM$I) ;
829      *
830      * b: Initialisation explicite du multiplicateur
831      *
832      *   mult := carlu - '0' ;
833      *
834      * c: Calcul du multiplicateur
835      *
836      *   si mult>10 alors "Invalid format" finsi ;
837      *   mult := (mult * 10) + (carlu - '0') ;
838      *
839      * d: Initialisation implicite du multiplicateur
840      *
841      *   mult := 1 ;
842      *   executer e ;
843      *
844      * e: Protection
845      *
846      *   p := si carlu='P' alors 1 sinon 0 ;
847      *   lastUP := p ;
848      *   bitmap (mult, p) ;
849      *
850      * i: Sortie de l'automate
851      *
852      *   bitmap (cptmsk, lastUP) ;
853      *   U$(I) := REV$(U$(I)) ;
854      *
855      *
856      * bitmap (n, p)
857      *   si n=0 alors retour
858      *   pour k = n jusque 1
859      *     faire
860      *       si cptmsk = 0 alors "Invalid format" ; finsi
861      *       cptmsk-- ;
862      *       si credit = 0 alors p := 1 ; finsi
863      *       si p
864      *         alors (admask) := (admask) ou mask ;
865      *         sinon
866      *           si cptm$>0 alors (addsu$++) := (addsm$) ; finsi;
867      *           credit--;
868      *       finsi
869      *       addsm$++ ;
870      *       si cptm$>0 alors cptm$-- ; finsi
871      *       mask := mask >> 2 ;
872      *       si mask=0
873      *         alors
```

```
874          *          mask := 2^3 ;
875          *          admask-- ;
876          *          (admask) := 0 ;
877          *          finsi
878          *          finpour
879          *
880
881 03655 1FAF8F2          D1=(5) n
882 0365C 147            C=DAT1 A
883 0365F 1DB9          D1=(2) i
884          *
885          * la boucle "compiler" attend dans
886          * D1 = i adresse de la variable i
887          * C(A) = la valeur de cette variable
888
889 03663          compiler
890 03663 145          DAT1=C A          i := C(A)
891
892          *
893          * Action a : initialisation de l'automate
894          *
895          * ctp$ := LEN(GETP$I) ;
896          *
897 03666 1FB98F2          D1=(5) i
898 0366D 147            C=DAT1 A
899 03670 723C          GOSUB GETP$I          Commentaire
900 03674 1F0A8F2          D1=(5) ctp$
901 0367B D2            C=0 A
902 0367D 15E3          C=DAT0 4
903 03681 145          DAT1=C A
904          *
905          * addsp$ := GETFST (GETP$I) ;
906          *
907 03684 768C          GOSUB GET$ST          D0 est toujours comme apres GETP$I
908 03688 1D5A          D1=(2) addsp$
909 0368C 145          DAT1=C A
910          *
911          * mask := 2^3
912          *
913 0368F 308          LC(1) 8          2^3 (ou encore 8, ou plutot %1000)
914 03692 1D6B          D1=(2) mask
915 03696 1550          DAT1=C P
916          *
917          * cptmsk := 96
918          *
919 0369A D2            C=0 A          pour l'instruction suivante
920 0369C 3106          LC(2) 96
921 036A0 1DCB          D1=(2) cptmsk
922 036A4 14D          DAT1=C B
923          *
924          * credit := MIN (MAXLEN(I$), 96) ;
925          *
926 036A7 1BA78F2          D0=(5) (DOPEI$)+9          D0 := ^ MAXLEN(I$)
927 036AE D0            A=0 A
928 036B0 15A3          A=DAT0 4          A(A) := MAXLEN(I$)
929 036B4 8B2          ?C>A A
930 036B7 40            GOYES actiona-05          on ne change rien
```

```
931 036B9 DA          A=C A
932 036BB             actiona-05
933 036BB 1DFF        D1=(2) credit
934 036BF 149         DAT1=A B          credit := MIN(MAXLEN(I$),96) ;
935 *
936 * admask := GETB$I + (12*2 +4 -1) quartets ;
937 *
938 036C2 1BB98F2     D0=(5) i
939 036C9 146         C=DAT0 A
940 036CC 7CEB        GOSUB GETB$I
941 036D0 DA          A=C A
942 036D2 D2          C=0 A
943 036D4 31B1        LC(2) 12*2+4-1
944 036D8 C2          C=C+A A
945 036DA 1D7B        D1=(2) admask
946 036DE 145         DAT1=C A
947 *
948 * (admask) := 0 ;
949 *
950 * Bug fix <861129.1115>
951 036E1 134         D0=C
952 036E4 AC2         C=0 S
953 036E7 1544        DAT0=C S
954 *
955 * addsu$ := GETU$I + 4 quartets
956 *
957 036EB 1BB98F2     D0=(5) i
958 036F2 146         C=DAT0 A
959 036F5 7ECB        GOSUB GETU$I
960 036F9 E6          C=C+1 A
961 036FB E6          C=C+1 A
962 036FD E6          C=C+1 A
963 036FF E6          C=C+1 A
964 03701 1DAA        D1=(2) addsu$
965 03705 145         DAT1=C A
966 *
967 * i := LEN (GETM$I) ;
968 *
969 03708 1BB98F2     D0=(5) i
970 0370F 146         C=DAT0 A
971 03712 7CBB        GOSUB GETM$I
972 03716 D0          A=0 A
973 03718 15A3        A=DAT0 4
974 0371C D2          C=0 A
975 0371E 3106        LC(2) 96
976 03722 8BA         ?C>=A A
977 03725 D0          GOYES actiona-10
978
979 03727 3100        LC(2) =eSTROV
980 0372B 8D00000     GOVLNG =MFERR
981
982 03732             actiona-10
983 *
984 * cptm$ := i
985 *
986 03732 1DFA        D1=(2) cptm$
987 03736 149         DAT1=A B
```

```
988          *
989          * addsm$ := GETFST (GETM$I) ;
990          *
991 03739 71DB          GOSUB  GET$ST
992 0373D 1D1B          D1=(2) addsm$
993 03741 145          DAT1=C A
994
995          *****
996          * Etat 1
997          *****
998
999 03744 7460  etat1  GOSUB  getchr
1000 03748 AA0          REL(3) erreur          EOL
1001 0374B 600          REL(3) t1-3          P | U
1002 0374E B00          REL(3) t1-2          0..9
1003
1004 03751 7801  t1-3   GOSUB  actiond
1005 03755 6020          GOTO   etat3
1006
1007 03759 79B0  t1-2   GOSUB  actionb
1008          *          GOTO   etat2
1009          *
1010          * ATTENTION : LE CODE CONTINUE
1011          *
1012
1013          *****
1014          * Etat 2
1015          *****
1016
1017 0375D 7B40  etat2  GOSUB  getchr
1018 03761 190          REL(3) erreur          EOL
1019 03764 E00          REL(3) t2-3          P | U
1020 03767 300          REL(3) t2-2          0..9
1021
1022 0376A 7BB0  t2-2   GOSUB  actionc
1023 0376E 6EEF          GOTO   etat2
1024 03772 77F0  t2-3   GOSUB  actions
1025          *          GOTO   etat3
1026          *
1027          * ATTENTION : LE CODE CONTINUE
1028          *
1029
1030          *****
1031          * Etat 3
1032          *****
1033
1034 03776 7230  etat3  GOSUB  getchr
1035 0377A B42          REL(3) actionl          EOL
1036 0377D 600          REL(3) t3-3          P | U
1037 03780 B00          REL(3) t3-4          0..9
1038
1039 03783 76D0  t3-3   GOSUB  actiond
1040 03787 6EEF          GOTO   etat3
1041 0378B 7780  t3-4   GOSUB  actionb
1042          *          GOTO   etat4
1043          *
1044          * ATTENTION : LE CODE CONTINUE
```

```
1045      *
1046
1047      *****
1048      * Etat 4
1049      *****
1050
1051 0378F 7910  etat4  GOSUB  getchr
1052 03793 F50   REL(3) erreur      EOL
1053 03796 600   REL(3) t4-3        P | U
1054 03799 B00   REL(3) t4-4        0..9
1055
1056 0379C 7DC0  t4-3   GOSUB  actions
1057 037A0 65DF          GOTO  etat3
1058 037A4 7180  t4-4   GOSUB  actionc
1059 037A8 66EF          GOTO  etat4
1060
1061      *****
1062      * getchr
1063      *
1064      * But: lit un caractere et branche sur erreur s'il n'est pas
1065      * reconnu, puis le classe en trois categories et opere un
1066      * branchement a une adresse fournie par le module appelant
1067      * Entree:
1068      * - RSTK = adresse de la table de branchement
1069      * - addsp$ pointe sur deux quartets plus haut que le
1070      * caractere a lire.
1071      * - ctp$ = nombre de caracteres restant a lire (1..n)
1072      * Sortie:
1073      * - A(B) = caractere lu
1074      * - P = 0
1075      * - ctp$ et addsp$ ajustes
1076      * Abime: C(A)
1077      * Appelle: CONVUC, DRANGE, TBLJMC
1078      * Niveaux: 2 (CONVUC)
1079      * Detail:
1080      * Les trois classes de lexemes dont :
1081      * - Fin de ligne
1082      * - P / U
1083      * - 0..9
1084      * Donc, l'appel doit etre de la forme :
1085      * GOSUB getchr
1086      * REL(3) transition si fin de ligne
1087      * REL(3) transition si P ou U
1088      * REL(3) transition si 0..9
1089      * Historique:
1090      * 86/08/31: J.T. & P.D. conception & codage
1091      *****
1092
1093 037AC 1F0A8F2 getchr  D1=(5) ctp$
1094 037B3 147      C=DAT1 A
1095 037B6 CE       C=C-1 A
1096 037B8 145      DAT1=C A
1097 037BB 415      GOC      saut0      Fin de ligne
1098
1099 037BE 1D5A      D1=(2) addsp$
1100 037C2 147      C=DAT1 A
1101 037C5 CE       C=C-1 A
```

```
1102 037C7 CE          C=C-1  A
1103 037C9 145        DAT1=C  A
1104 037CC 135        D1=C
1105                  *** change
1106                  * rev D:
1107                  *      A=DAT1 B
1108                  *      GOSBVL =CONVUC
1109                  * rev X:
1110 037CF 8F00000     GOSBVL =CNVUCR
1111                  *** end of change
1112 037D6 3105        LCASC  'P'
1113 037DA 962        ?C=A   B
1114 037DD 92         GOYES  saut1
1115 037DF 3155        LCASC  'U'
1116 037E3 962        ?C=A   B
1117 037E6 02         GOYES  saut1
1118 037E8 8F00000     GOSBVL =DRANGE
1119 037EF 5F0        GONC   saut2
1120
1121 037F2 33501E      erreur  LC(4)  #E105      (id)~(=eIFMT)  "Invalid Format"
1122 037F8 8D00000     GOVLNG =BSERR
1123
1124 037FF D2         saut2   C=0    A
1125 03801 E6         C=C+1  A
1126 03803 540        GONC   st          B.E.T.
1127 03806 D2         saut1   C=0    A
1128 03808 E6         st      C=C+1  A
1129 0380A 540        GONC   saut
1130 0380D D2         saut0   C=0    A
1131 0380F 8D00000     saut   GOVLNG =TBLJMC
1132
1133                  *****
1134                  * actionb
1135                  *****
1136
1137 03816              actionb
1138 03816 3103        LCASC  '0'
1139 0381A B6A         A=A-C  B
1140 0381D 1FEB8F2     D1=(5) mult
1141 03824 149        DAT1=A  B
1142 03827 01         RTN
1143
1144                  *****
1145                  * actionc
1146                  *****
1147
1148 03829              actionc
1149                  *
1150                  * si mult>10 alors "invalid format" finsi ;
1151                  *
1152 03829 AE8         B=A    B
1153 0382C 1FEB8F2     D1=(5) mult
1154 03833 14B        A=DAT1 B
1155 03836 31A0        LC(2)  10
1156 0383A 9E6        ?C<A  B
1157 0383D 5B         GOYES  erreur
1158                  *
```

```
1159      * mult := (mult*10)
1160      *
1161 0383F A64      A=A+A  B
1162 03842 AE6      C=A    B
1163 03845 A66      C=C+C  B      C(A) := mult * 4
1164 03848 A66      C=C+C  B      C(A) := mult * 8
1165 0384B A6A      A=A+C  B      A(A) := mult * 10
1166      *
1167      *      + (carlu-'0') ;
1168      *
1169 0384E 3103      LCASC  '0'
1170 03852 B61      B=B-C  B
1171 03855 A60      A=A+B  B
1172 03858 149      DAT1=A  B
1173 0385B 01      RTN
1174
1175      *****
1176      * actiond
1177      *****
1178
1179 0385D      actiond
1180      *
1181      * mult := 1 ;
1182      *
1183 0385D AE2      C=0    B
1184 03860 B66      C=C+1  B
1185 03863 1FEB8F2  D1=(5) mult
1186 0386A 14D      DAT1=C  B
1187      *      GOTO  actions      (en fait: gosub ; rtn)
1188      *
1189      * ATTENTION : LE CODE CONTINUE :
1190      *
1191
1192      *****
1193      * actions
1194      *****
1195
1196 0386D      actions
1197      *
1198      * P := si carlu = 'P' alors 1 sinon 0 ;
1199      *
1200 0386D 3155      LCASC  'U'
1201 03871 B62      C=C-A  B
1202 03874 96A      ?C=0  B
1203 03877 80      GOYES  actions-010
1204 03879 AE2      C=0    B
1205 0387C B66      C=C+1  B
1206 0387F      actions-010
1207      *
1208      * lastUP := p ;
1209      *
1210 0387F 1F0C8F2  D1=(5) lastUP
1211 03886 15D0      DAT1=C  1
1212      *
1213      * bitmap (mult, p) ;
1214      *
1215 0388A DA      A=C    A
```



```
1216 0388C 1DEB          D1=(2) mult
1217 03890 14F          C=DAT1 B
1218 03893 6700        GOTO  bitmap          (en fait: gosub ; rtn)
1219
1220 03897 6A5F        Erreur GOTO  erreur
1221
1222
1223                    *****
1224                    * bitmap
1225                    *
1226                    * But: voir algorithme en debut de fichier
1227                    * Entree:
1228                    *   - A(0) = p
1229                    *   - C(B) = n
1230                    * Abime: A(A), C(A), D0, D1
1231                    * Niveaux: 0
1232                    * Historique:
1233                    *   86/08/31: J.T. & P.D. conception & codage
1234                    *   86/11/29: J.T. & P.D. ajout de credit
1235                    *****
1236                    vark   EQU   (funcr0)+38
1237                    varp   EQU   (funcr0)+40
1238
1239 0389B              bitmap
1240                    *
1241                    * si n=0 alors retiur ; finsi
1242                    *
1243 0389B 96A          ?C=0   B
1244 0389E 00          RTNYES
1245                    *
1246                    * pour k := n jusque 1
1247                    *
1248 038A0 1F3C8F2      D1=(5) varp
1249 038A7 1590        DAT1=A 1
1250 038AB 1D1C        D1=(2) vark
1251
1252 038AF 14D          bm010  DAT1=C B
1253                    *
1254                    * faire
1255                    *   si cptmsk=0 alors "invalid format" ; finsi
1256                    *
1257 038B2 1FCB8F2      D1=(5) cptmsk
1258 038B9 14F          C=DAT1 B
1259 038BC 96A          ?C=0   B
1260 038BF 8D          GOYES  Erreur
1261                    *
1262                    *   cptmsk-- ;
1263                    *
1264 038C1 A6E          C=C-1  B
1265 038C4 14D        DAT1=C  B
1266                    *
1267                    *   si credit = 0 alors p := 1 ;
1268                    *
1269 038C7 1FFF8F2      D1=(5) credit
1270 038CE 14F          C=DAT1 B          C(B) := credit en "Unprotected"
1271 038D1 1F3C8F2      D1=(5) varp
1272 038D8 96E          ?C#0   B
```

```
1273 038DB C0          GOYES  bm015
1274 038DD A80        A=0    P
1275 038E0 B04        A=A+1  P
1276 038E3 1590      DAT1=A  1
1277                  *
1278                  *      finsi
1279                  *
1280 038E7            bm015
1281                  *
1282                  *      si p
1283                  *
1284 038E7 15B0        A=DAT1  1
1285 038EB 908        ?A=0    P
1286 038EE 72         GOYES  bm020
1287                  *
1288                  *      alors (admask) := (admask) ou mask ;
1289                  *
1290 038F0 1F6B8F2     D1=(5) mask
1291 038F7 1574        C=DAT1  S
1292 038FB 1D7B        D1=(2) admask
1293 038FF 147         C=DAT1  A
1294 03902 135        D1=C
1295 03905 1534        A=DAT1  S
1296 03909 0E4E        A=A!C   S
1297 0390D 1514        DAT1=A  S
1298 03911 6940        GOTO   bm035
1299                  *
1300                  *      sinon
1301                  *      si cptm$>0
1302                  *
1303 03915 1FFA8F2     bm020 D1=(5) cptm$
1304 0391C 14F         C=DAT1  B
1305 0391F 96A        ?C=0    B
1306 03922 92         GOYES  bm030
1307                  *
1308                  *      alors (addsu$++) := (addsm$)
1309                  *
1310 03924 1D1B        D1=(2) addsm$
1311 03928 147         C=DAT1  A
1312 0392B 135        D1=C
1313 0392E 1C1        D1=D1- 2
1314 03931 14B        A=DAT1  B      (addsm$)
1315 03934 1FAA8F2     D1=(5) addsu$
1316 0393B 147         C=DAT1  A
1317 0393E 134        D0=C
1318 03941 E6         C=C+1  A
1319 03943 E6         C=C+1  A
1320 03945 145        DAT1=C  A      addsu$++
1321 03948 148        DAT0=A  B
1322
1323                  *
1324                  *      finsi
1325                  *
1326 0394B            bm030
1327                  *
1328                  *      credit-- ;
1329                  *
```

```
1330 0394B 1FFF8F2          D1=(5) credit
1331 03952 14F             C=DAT1 B
1332 03955 A6E            C=C-1 B
1333 03958 14D            DAT1=C B
1334                      *
1335                      *      finsi
1336                      *
1337 0395B                bm035
1338                      *
1339                      *      addsm$++ ;
1340                      *
1341 0395B 1F1B8F2          D1=(5) addsm$
1342 03962 147             C=DAT1 A
1343 03965 CE              C=C-1 A
1344 03967 CE              C=C-1 A
1345 03969 145            DAT1=C A
1346                      *
1347                      *      si cptm$>0 alors cptm$-- ; finsi
1348                      *
1349 0396C 1DFA            D1=(2) cptm$
1350 03970 14F             C=DAT1 B
1351 03973 A6E            C=C-1 B
1352 03976 450            GOC      bm040
1353 03979 14D            DAT1=C B
1354 0397C                bm040
1355                      *
1356                      *      mask := mask >> 2 ;
1357                      *
1358 0397C 1D6B            D1=(2) mask
1359 03980 D2              C=0      A
1360 03982 15F0           C=DAT1 1
1361 03986 81E            CSRB          C(0) := mask
1362                      *
1363                      *      si mask = 0
1364                      *      alors
1365                      *      mask := 2^3 ;
1366                      *
1367 03989 90E             ?C#0    P
1368 0398C B1             GOYES   bm050
1369 0398E 308           LC(1)   8      C(0) := 2^3
1370                      *
1371                      *      admask-- ;
1372                      *
1373 03991 1D7B            D1=(2) admask
1374 03995 143            A=DAT1 A
1375 03998 CC             A=A-1 A
1376 0399A 141            DAT1=A A
1377                      *
1378                      *      (admask) := 0 ;
1379                      *
1380 0399D 131             D1=A
1381 039A0 AC0            A=0 S
1382 039A3 1514           DAT1=A S
1383                      *
1384                      *      finsi
1385                      *
1386 039A7 1F6B8F2        bm050   D1=(5) mask
```

```

1387 039AE 15D0          DAT1=C 1
1388                    *
1389                    *   finpour
1390                    *
1391 039B2 1D1C          D1=(2) vark
1392 039B6 14F          C=DAT1 B
1393 039B9 A6E          C=C-1 B
1394 039BC 96A          ?C=0 B
1395 039BF 00           RTNYES
1396 039C1 6DEE          GOTO   bm010
1397
1398                    *****
1399                    * actionl
1400                    *****
1401
1402 039C5          actionl
1403                    *
1404                    *   bitmap (cptmsk, lastUP) ;
1405                    *
1406 039C5 1F0C8F2      D1=(5) lastUP
1407 039CC 15B0          A=DAT1 1
1408 039D0 1FCB8F2      D1=(5) cptmsk
1409 039D7 14F          C=DAT1 B
1410 039DA 7DBE          GOSUB  bitmap
1411                    *
1412                    *   U$(I) := REV$(U$(I)) ;
1413                    *
1414                    *
1415                    *   len(U$(I)) := ( addsu$ - GETU$I - 4) / 2
1416                    *
1417 039DE 1FB98F2      D1=(5) i
1418 039E5 147          C=DAT1 A
1419 039E8 7BD8          GOSUB  GETU$I
1420 039EC 1DAA          D1=(2) addsu$
1421 039F0 143          A=DAT1 A
1422 039F3 131          D1=A
1423 039F6 EE           C=A-C  A          addsu$ - GETU$I
1424 039F8 CE           C=C-1  A
1425 039FA CE           C=C-1  A
1426 039FC CE           C=C-1  A
1427 039FE CE           C=C-1  A
1428 03A00 AD0          A=0    M
1429 03A03 DA           A=C    A          A(A) := 2 * LEN(U$(I)) ;
1430 03A05 81C          ASRB                    A(A) := LEN(U$(I)) ;
1431 03A08 1583          DAT0=A 4
1432 03A0C 163          D0=D0+ 4
1433                    *
1434                    *   U$(I) := REV$(U$(I)) ;
1435                    *
1436 03A0F AF1          B=0    W
1437 03A12 D8           B=A    A
1438 03A14 822          SB=0
1439 03A17 81D          BSRB
1440 03A1A 136          CD0EX
1441 03A1D C9           C=C+B  A
1442 03A1F C9           C=C+B  A
1443 03A21 134          D0=C

```

```
1444 03A24 135          D1=C
1445 03A27 832          ?SB=0
1446 03A2A 71           GOYES  pair
1447 03A2C 511          GONC   impair
1448 03A2F 1C1         rptrev D1=D1- 2
1449 03A32 14A          A=DAT0 B
1450 03A35 14F          C=DAT1 B
1451 03A38 149          DAT1=A B
1452 03A3B 14C          DAT0=C B
1453 03A3E 161         impair D0=D0+ 2
1454 03A41 CD          pair   B=B-1  A
1455 03A43 5BE          GONC   rptrev
1456
1457 03A46 1FB98F2      D1=(5) i
1458 03A4D 147          C=DAT1 A
1459 03A50 1F2D8F2      D1=(5) DOPEd$
1460 03A57 1BE78F2      D0=(5) DOPEM$
1461 03A5E 8E457F      GOSUBL ELMCPY      D$(I) := M$(I)
1462
1463 03A64 1FB98F2      D1=(5) i
1464 03A6B 147          C=DAT1 A
1465 03A6E CE           C=C-1  A
1466 03A70 8AA          ?C=0   A
1467 03A73 60           GOYES  compfin
1468 03A75 6DEB         GOTO   compiler
1469
1470 03A79              compfin
1471
1472                    * end of comp.as, start of boucle.as
1473
1474                    *
1475                    * i contient le numero de la ligne en cours (1..n)
1476                    *
1477                    *** JFG, 2006: renamed i to i1,
1478                    *** because sasm doesn't allow to redefine a symbol
1479                    i1      EQU      #2F6C1      (=LDCSPC)
1480                    *
1481                    * <861202.2120> : merci HP !
1482                    * La variable i etait placee en FUNCRO. Comme la
1483                    * documentation d'HP ne l'indique pas, CREDIT ABIME TOUTE
1484                    * LA FUNCTION SCRATCH, ansi que DEUX QUARTETS DANS LA
1485                    * STATEMENT SCRATCH.
1486                    * Ceci expliquait un comportement "peu normal" de FINPUT.
1487                    *
1488
1489                    *
1490                    * menu est un flag pour indiquer si tout le display est
1491                    * protege.
1492                    *
1493                    *** menu      EQU      0          no more used (flag in D[S] now)
1494
1495                    *
1496                    * Si, si : Rappelez-vous. R3 contient toujours D0 :
1497                    *
1498                    * On va maintenant evaluer la variable A (dernier parametre:
1499                    * variable numerique). Les elements caracteristiques (pour
1500                    * utiliser apres DEST) vont etre places a l'endroit DOPEA.
```

```

1501          *
1502 03A79 11B          C=R3          C(A) := PC
1503 03A7C 134          D0=C          D0 := PC
1504 03A7F 8F00000     GOSBVL =EXPEXC
1505 03A86 8F00000     GOSBVL =AVE=D1
1506 03A8D 1BFD8F2     D0=(5) DOPEA
1507
1508 03A94 AF4          A=B      W
1509 03A97 1507         DAT0=A  W
1510 03A9B 16F          D0=D0+ 16      DOPEA + 0 := B(W) (16 q.)
1511
1512 03A9E 137          CD1EX
1513 03AA1 144          DAT0=C  A
1514 03AA4 164          D0=D0+ 5      DOPEA + 16 := D1 (5 q.)
1515
1516 03AA7 1F00000     D1=(5) =F-R1-0
1517 03AAE 147          C=DAT1  A
1518 03AB1 144          DAT0=C  A
1519 03AB4 164          D0=D0+ 5      DOPEA + 21 := F-R1-0 (5 q.)
1520
1521 03AB7 17E          D1=D1+ 15
1522 03ABA 15F0         C=DAT1  1
1523 03ABE 15C0         DAT0=C  1      DOPEA + 26 := F-R1-3 (1 q.)
1524          *
1525          * Voila, les elements caracteristiques de A sont sauves.
1526          *
1527
1528          *
1529          * Pour que [ATTN] se comporte bien, il faut verifier que
1530          * les elements de M$ ne comportent pas de caracteres non
1531          * affichables (NULL, CR, LF, ESC, BS). Rappelons que
1532          * [ATTN] comparera le contenu du display avec M$(1).
1533          *
1534
1535          *** added:
1536 03AC2 AC0          A=0      S
1537 03AC5 1BFB6F2     D0=(5) #2F6BF
1538 03ACC 1504         DAT0=A  S
1539          *** end of change
1540
1541 03AD0 1FAF8F2     D1=(5) n
1542 03AD7 147          C=DAT1  A
1543
1544 03ADA 108          ver10  R0=C
1545 03ADD 8EFE7F     GOSUBL GETM$I      D0 := ^ longueur M$(I)
1546 03AE3 D2          C=0      A
1547 03AE5 15E3         C=DAT0  4
1548 03AE9 D5          B=C      A
1549 03AEB D2          C=0      A
1550 03AED 3106         LC(2)  96
1551 03AF1 8B1          ?B>C  A
1552 03AF4 80          GOYES  invcar      "Invalid Prompt"
1553 03AF6 163          D0=D0+ 4
1554 03AF9 5E3         GONC   ver30      B.E.T.
1555
1556 03AFC 33401E     invcar  LC(4)  #E104      (id)~(eIPRMP) "Invalid Prompt"
1557 03B02 8D00000     GOVLNG =BSERR

```

```
1558 *
1559 * Routine interne de verification des caracteres un par un
1560 *
1561 03B09 14A ver20 A=DAT0 B
1562 03B0C 968 ?A=0 B NULL
1563 03B0F DE GOYES invcar "Invalid Prompt"
1564 03B11 31B1 LC(2) 27 C(B) := ESC
1565 03B15 962 ?C=A B
1566 03B18 4E GOYES invcar "Invalid Prompt"
1567 03B1A 31D0 LC(2) 13 C(B) := CR
1568 03B1E 962 ?C=A B
1569 03B21 BD GOYES invcar "Invalid Prompt"
1570 03B23 31A0 LC(2) 10 C(B) := LF
1571 03B27 962 ?C=A B
1572 03B2A 2D GOYES invcar "Invalid Prompt"
1573 03B2C 3180 LC(2) 08 C(B) := BS
1574 03B30 962 ?C=A B
1575 03B33 9C GOYES invcar "Invalid Prompt"
1576 03B35 161 D0=D0+ 2
1577 03B38 A6D ver30 B=B-1 B
1578 03B3B 5DC GONC ver20
1579 *
1580 03B3E 118 C=R0
1581 03B41 CE C=C-1 A
1582 03B43 8AE ?C#0 A
1583 03B46 49 GOYES ver10
1584 *
1585 * Maintenant, on prepare la boucle principale de FINPUT
1586 *
1587 03B48 1F1C6F2 D1=(5) i1
1588 03B4F D2 C=0 A
1589 03B51 E6 C=C+1 A
1590 03B53 145 DAT1=C A i := 1 (numero de la ligne courante)
1591
1592 *****
1593 * BP:, BP-
1594 *
1595 * But: Boucle principale de FINPUT
1596 * Entree:
1597 * - BP:
1598 * - i (FUNCR0) = numero de la ligne courante (0..n-1)
1599 * - BP-:
1600 * - i (FUNCR0) = numero de la ligne courante (0..n-1)
1601 * - D0 = dope vecteuuuuuuur du tableau a afficher
1602 * Sortie:
1603 * - suivant les touches :
1604 * [^], [v], [g][^], [g][v] : passage a la ligne suivant
1605 * la direction, sans changer la ligne courante.
1606 * [ENDLINE] : valide la ligne courante. Si l'ecran est
1607 * constitue d'une seule ligne, termine FINPUT (comme
1608 * (L)INPUT). Dans ce dernier cas, la variable A est
1609 * mise a la valeur 1.
1610 * [RUN] : valide la ligne courante, et termine FINPUT.
1611 * La variable A contiendra le numero de cette ligne
1612 * [ATTN] :
1613 * 1: remet la valeur par defaut de la ligne courante
1614 * 2: si la ligne courante contient deja la valeur
```

```
1615      *           par defaut, [ATTN] sort de FINPUT, la variable A
1616      *           est mise a 0. La variable/tableau I$ n'est pas
1617      *           modifiee par FINPUT.
1618      * Abime: CPU
1619      * Appelle: CHEDIT, DSPCHA, GETx$I, GETFST, CURSFL
1620      * Niveaux: 7 (CHEDIT)
1621      * Historique:
1622      *   86/09/03: J.T. & P.D. conception & codage
1623      *   86/11/22: J.T. & P.D. documentation
1624      *   86/11/22: J.T. & P.D. gestion des touches speciales
1625      *   86/11/24: P.D.      modification pour ajout de BP-
1626      *   87-88?   ?           use D[S] instead of ST for menu flag
1627      * *****
1628
1629 03B56      BP:
1630 03B56 1B2D8F2      D0=(5) DOPED$
1631 03B5D      BP-
1632      *
1633      * Mise a zero du masque avant d'envoyer D$(I) a l'affichage
1634      *
1635 03B5D 1F00000      D1=(5) =DSPMSK
1636 03B64 AF2          C=0      W
1637 03B67 15DF          DAT1=C 16
1638 03B6B 17F          D1=D1+ 16
1639 03B6E 15D7          DAT1=C 8
1640
1641      *
1642      * Affichage du tableau D$(I). Les caracteres de D$(I) sont
1643      * ranges en sens inverse, ce qui oblige a utiliser une
1644      * boucle speciale.
1645      *
1645 03B72 1F1C6F2      D1=(5) i1
1646 03B79 147          C=DAT1 A
1647      *
1648      * D0 := DOPEx$
1649      *
1650 03B7C 8E267F          GOSUBL GETx$I
1651 03B82 D2            C=0      A
1652 03B84 15E3          C=DAT0 4
1653 03B88 D7            D=C      A           D(A) := LEN(x$(I))
1654 03B8A 8EE77F          GOSUBL GET$ST
1655 03B90 DB            C=D      A           C(A) := LEN(x$(I))
1656 03B92 132          AD0EX
1657 03B95 6020          GOTO   anpc10
1658      *
1659      * Invariant de cette boucle :
1660      *   A(A) = pointeur dans x$(I)
1661      *   C(A) = nombre de caracteres restant a afficher
1662      *
1663      *
1664      *
1665      * ancpsd = rev$("dspcna")
1666      *
1667 03B99 06          ancpsd RSTK=C           RSTK := LEN(x$(I))
1668 03B9B CC          A=A-1  A
1669 03B9D CC          A=A-1  A
1670 03B9F D6          C=A      A
1671 03BA1 06          RSTK=C           RSTK := ^ x$(I)
```



```
1672 03BA3 130          D0=A
1673 03BA6 14A          A=DAT0 B
1674 03BA9 8F00000      GOSBVL =DSPCHA
1675 03BB0 07           C=RSTK          C(A) := ^ x$(I)
1676 03BB2 DA          A=C    A
1677 03BB4 07           C=RSTK          C(A) := LEN(x$(I))
1678 03BB6 CE          anpc10 C=C-1 A
1679 03BB8 50E          GONC    ancpsd
1680
1681                    *
1682                    * Injection de B$(I) dans DSPMSK :
1683                    *
1684                    *** change:
1685                    *          ST=0    menu
1686                    *          D=0      S
1687                    *** end of change
1688 03BBE 1B1C6F2        D0=(5) i1
1689 03BC5 146           C=DAT0 A
1690 03BC8 8EEE6F        GOSUBL GETB$I
1691 03BCE 163           D0=D0+ 4
1692 03BD1 1F00000        D1=(5) =DSPMSK
1693 03BD8 15EF          C=DAT0 16
1694 03BDC 15DF          DAT1=C 16
1695 03BE0 B76           C=C+1  W
1696 03BE3 550           GONC    inj10
1697                    * les 64 premiers caracteres sont tous proteges
1698                    *** change:
1699                    *          ST=1    menu
1700 03BE6 B47           D=D+1  S
1701                    *** end of change
1702 03BE9                inj10
1703 03BE9 16F           D0=D0+ 16
1704 03BEC 17F           D1=D1+ 16
1705 03BEF 15E7          C=DAT0 8
1706 03BF3 15D7          DAT1=C 8
1707
1708 03BF7 27            P=      7
1709 03BF9 B16           C=C+1  WP
1710 03BFC 20            P=      0
1711 03BFE 450           GOC    inj20
1712                    * les 32 derniers caracteres ne sont pas proteges
1713                    *** change:
1714                    *          ST=0    menu
1715 03C01 AC3           D=0      S
1716                    *** end of change
1717 03C04                inj20
1718                    *** added:
1719 03C04 ACB           C=D      S
1720 03C07 1B00000        D0=(5) =STSAVE
1721 03C0E 1544          DAT0=C  S
1722                    *** end of change
1723                    *
1724                    * Curseur a gauche avant l'edition
1725                    *
1726 03C12 8F00000        GOSBVL =CURSFL
1727
1728                    *****
```

```
1729      * edit
1730      *
1731      * But: realiser l'edition de la ligne deja affichee.
1732      * Entree:
1733      *   - la ligne est deja affichee
1734      * Sortie:
1735      *   - par les routines particulieres :
1736      *     deplacement du curseur : kUP, kDOWN, kTOP, kBOT
1737      *     validation et envoi : kENDLN, kRUN
1738      *     sortie : kATTN, kOFF
1739      * Appelle: BP10 ou BP20
1740      * Historique:
1741      *   86/11/23: J.T. & P.D. conception & codage
1742      *   87-88?   ?           use D[S] instead of ST for menu flag
1743      *   *****
1744
1745 03C19      edit
1746      *** change:
1747      *       ?ST=1 menu           tous les car. sont proteges ?
1748 03C19 1B00000      D0=(5) =STSAVE
1749 03C20 1564          C=DAT0 S
1750 03C24 94E          ?C#0 S           tous les car. sont proteges ?
1751      *** end of change
1752 03C27 07           GOYES BP20      oui : affichage sans introduction
1753      *                                           non : INPUT normal
1754
1755      *****
1756      * BP10
1757      *
1758      * But: realiser l'edition de la ligne deja affichee, et
1759      *   envoyer sur les routines particulieres.
1760      * Entree:
1761      * Sortie:
1762      *   - par les routines particulieres :
1763      *     Abime: A-D, D0, D1, R0, R3, ST, INSIMP (flag pour le poll)
1764      *     DEFADR, USRSTA, 32 nibbles at scratch
1765      * Appelle: CHEDIT, FINDA
1766      * Historique:
1767      *   86/11/23: J.T. & P.D. conception & codage
1768      *   *****
1769
1770 03C29      BP10
1771      *
1772      * Previent le Poll handler que c'est nous qu'on travaille !
1773      *
1774 03C29 1B199F2      D0=(5) RESJPC
1775 03C30 15A0          A=DAT0 1
1776 03C34 301          LC(1) 1
1777 03C37 0E0E          A=A!C P
1778 03C3B 1580          DAT0=A 1
1779
1780      *
1781      * Y-a-queq'un ?
1782      *
1783 03C3F 8F00000      GOSBVL =CHEDIT
1784
1785      *
```

```

1786          * Previent le Poll handler que c'est nous qu'on travaille
1787          * plus !
1788          *
1789 03C46 30E          LC(1)  #E
1790 03C49 816          CSRC
1791 03C4C 1B199F2      D0=(5) RESJPC
1792 03C53 1524          A=DAT0 S
1793 03C57 0E46          A=A&C S          Carry is not affected
1794 03C5B 1504          DAT0=A S          A(B) n'est pas modifie
1795          *
1796          * La carry est encore dans l'etat apres CHEDIT :
1797          *   Cy = 1 : touche speciale [ATTN], [^], etc.
1798          *   Cy = 0 : user terminating key
1799          *
1800 03C5F 59B          GONC  edit
1801          *
1802          * Et dans A(A) : le code de la touche
1803          *
1804 03C62 8F00000      GOSBVL =FINDA
1805 03C69 D0           CON(2) 13
1806 03C6B 081          REL(3) kENDLN
1807 03C6E E0           CON(2) 14
1808 03C70 201          REL(3) kATTN
1809 03C73 F0           CON(2) 15
1810 03C75 F81          REL(3) kRUN
1811 03C78 21           CON(2) 18
1812 03C7A D80          REL(3) kUP
1813 03C7D 31           CON(2) 19
1814 03C7F 570          REL(3) kDOWN
1815 03C82 41           CON(2) 20
1816 03C84 6B0          REL(3) kTOP
1817 03C87 51           CON(2) 21
1818 03C89 3C0          REL(3) kBOT
1819 03C8C 81           CON(2) 24
1820 03C8E 351          REL(3) kOFF
1821 03C91 00           CON(2) #00
1822          *
1823          * Aucune de celles-ci : on ne reconnait pas, on ignore.
1824          *
1825 03C93 658F          GOTO  edit
1826          *
1827          *****
1828          * BP20
1829          *
1830          * But: traiter le cas ou la ligne est entierement protegee
1831          *   C'est le cas particulier de MENU.
1832          * Entree:
1833          * Sortie:
1834          *   - par les routines particulieres :
1835          * Abime: A-D, D0, D1, 32 nibbles at scratch
1836          * Appelle: CKSREQ, SCRLLR, POPBUF, FINDA
1837          * Niveau: 6 (SCRLLR)
1838          * Historique:
1839          *   86/11/23: J.T. & P.D. conception & codage
1840          *   86/11/29: J.T. & P.D. ajout de RPTKY pour la repetition
1841          *****
1842

```

```
1843 03C97          BP20
1844 03C97 8F00000   GOSBVL =RPTKY
1845 03C9E 4E1       GOC     BP25          Go if repeated key
1846 03CA1 8F00000   BP12    GOSBVL =SCRLLR
1847 03CA8 5D0       GONC    BP15
1848 03CAB 8F00000   GOSBVL =CKSREQ
1849 03CB2 6EEF      GOTO    BP12
1850 03CB6 8F00000   BP15    GOSBVL =POPBUF
1851 03CBD D4         BP25    A=B      A          A(B) := keycode
1852 03CBF 8F00000   GOSBVL =FINDA
1853 03CC6 62        CON(2) 38
1854 03CC8 321      REL(3) kENDLN
1855 03CCB B2        CON(2) 43
1856 03CCD 5A0      REL(3) kATTN
1857 03CD0 E2        CON(2) 46
1858 03CD2 231      REL(3) kRUN
1859 03CD5 23        CON(2) 50
1860 03CD7 030      REL(3) kUP
1861 03CDA 33        CON(2) 51
1862 03CDC 810      REL(3) kDOWN
1863 03CDF 2A        CON(2) 162
1864 03CE1 950      REL(3) kTOP
1865 03CE4 3A        CON(2) 163
1866 03CE6 660      REL(3) kBOT
1867 03CE9 36        CON(2) 99
1868 03CEB 6F0      REL(3) kOFF
1869 03CEE 00        CON(2) #00
1870
1871          *
1872          * La touche n'est pas reconnue
1873          *
1873 03CF0 682F      GOTO    edit
1874
1875          *****
1876          * kUP, kDOWN, kBOT, kTOP
1877          *
1878          * But: traiter les touches de deplacement du curseur.
1879          * Entree:
1880          * Sortie:
1881          *   - par "curs"
1882          *   - la variable i contient le numero de la nouvelle ligne.
1883          * Abime: A-D, D0, D1, ST, i
1884          * Appelle: finign, curs
1885          * Niveau: 5 (finign)
1886          * Historique:
1887          *   86/11/23: J.T. & P.D. conception & codage
1888          *   86/11/24: J.T. & P.D. ajout de documentation
1889          *****
1890
1891          *****
1892          * Touche [v]
1893          *****
1894
1895 03CF4 7D60      kDOWN   GOSUB   finign
1896 03CF8 1F1C6F2  kDOWN-  D1=(5)  i1
1897 03CFF 143      A=DAT1  A
1898 03D02 E4        A=A+1   A
1899 03D04 521      GONC    curs          B.E.T.
```

```

1900
1901 *****
1902 * Touche [^]
1903 *****
1904
1905 03D07 7A50 kUP GOSUB finign
1906 03D0B 1F1C6F2 D1=(5) i1
1907 03D12 143 A=DAT1 A
1908 03D15 CC A=A-1 A
1909 *
1910 * Le code continue dans "curs"
1911 *
1912
1913 *****
1914 * curs
1915 *
1916 * But: gerer les deplacements dans les differentes lignes
1917 * d'une entree.
1918 * Entree:
1919 * - A(A) = nombre de lignes
1920 * Sortie:
1921 * - par BP: (Boucle Principale)
1922 * Abime: A-D, C(A), D0
1923 * Niveau: 0
1924 * Historique:
1925 * 86/09/03: P.D. & J.T. conception & codage
1926 * 86/11/22: P.D. & J.T. documentation
1927 *****
1928
1929 03D17 D2 curs C=0 A
1930 03D19 E6 C=C+1 A
1931 03D1B 8BE ?C<=A A
1932 03D1E 40 GOYES curs10
1933 03D20 DA A=C A A(A) := 1
1934 03D22 1BAF8F2 curs10 D0=(5) n
1935 03D29 146 C=DAT0 A
1936 03D2C 8BA ?C>=A A
1937 03D2F 40 GOYES curs20
1938 03D31 DA A=C A
1939 03D33 141 curs20 DAT1=A A
1940 03D36 6F1E GOTO BP:
1941
1942 *****
1943 * Touche [g][^]
1944 *****
1945
1946 03D3A 7720 kTOP GOSUB finign
1947 03D3E 1F1C6F2 D1=(5) i1
1948 03D45 D0 A=0 A
1949 03D47 E4 A=A+1 A
1950 03D49 59E GONC curs20 B.E.T.
1951
1952 *****
1953 * Touche [g][v]
1954 *****
1955
1956 03D4C 7510 kBOT GOSUB finign

```

```
1957 03D50 1F1C6F2          D1=(5) i1
1958 03D57 1BAF8F2          D0=(5) n
1959 03D5E 142              A=DAT0 A
1960 03D61 61DF             GOTO curs20
1961
1962
1963 * finign
1964 *
1965 * But: terminer physiquement (pour l'affichage) une ligne
1966 * saisie.
1967 * Entree: -
1968 * Sortie: -
1969 * Abime: A-D, D0, D1, ST, R3
1970 * Appelle: ILFART, CRLFOF
1971 * Niveaux: voir la documentation de ILFART
1972 * Detail:
1973 * La routine systeme FINLIN n'est pas utilisee. Ceci est
1974 * du a son mode de fonctionnement : FINLIN "deprotege" le
1975 * dernier caractere (le 96eme) et fait un CURSFR (cursor
1976 * far right), ce qui a un desagreable effet sur la video.
1977 * Le curseur balaye les 96 caracteres.
1978 * Cette routine laisse le dernier caractere intact, l'HPIL
1979 * ne balayera donc que ce qui est necessaire.
1980 * Historique:
1981 * 86/11/29: P.D. & J.T. conception & codage
1982 * 86/11/29: P.D. & J.T. documentation
1983 *****
1984
1985 03D65 8E0000 finign GOSUBL =ILFART          (in polhnd code)
1986 03D6B 8D00000 GOVLNG =CRLFOF
1987
1988 *****
1989 * kATTN, kOFF
1990 *
1991 * But: traiter les touches de sortie de FINPUT.
1992 * Entree: -
1993 * Sortie:
1994 * - par EXIT, ou BP.
1995 * - la variable A contiendra 0.
1996 * Appelle: finign, BP, ou EXIT
1997 * Historique:
1998 * 86/11/23: P.D. & J.T. conception & codage
1999 * 86/11/24: P.D. ajout de documentation
2000 *****
2001
2002 *****
2003 * Touche [ATTN]
2004 *****
2005
2006 03D72 7FEF kATTN GOSUB finign
2007 03D76 8F00000 GOSBVL =ATNCLR          C'est magique ! Mais ca evite au
2008 * cursor de se mal retrouver.
2009 *
2010 * if display = M1(I) then EXIT avec 0 ;
2011 *
2012 03D7D 1B1C6F2          D0=(5) i1
2013 03D84 146              C=DAT0 A
```



```

2071      * Detail: si il n'y a qu'une seule ligne, FINPUT traite
2072      *   la touche [ENDLINE] comme [RUN].
2073      * Historique:
2074      *   86/11/23: P.D. & J.T. conception & codage
2075      *   86/11/24: P.D.      ajout de documentation
2076      *
2077      *
2078      *
2079      * Touche [ENDLINE]
2080      *
2081      *
2082 03DEB      kENDLN
2083 03DEB 7250      GOSUB  VALIDE
2084 03DEF 1BAF8F2      D0=(5) n
2085 03DF6 146      C=DAT0 A
2086 03DF9 CE      C=C-1 A
2087 03DFB 8AA      ?C=0 A
2088 03DFE A0      GOYES  kRUN00
2089 03E00 67FE      GOTO  kDOWN-
2090      *
2091      *
2092      * Touche [RUN]
2093      *
2094      *
2095 03E04      kRUN
2096 03E04 7930      GOSUB  VALIDE
2097 03E08      kRUN00
2098      *
2099      * Copier le tableau U$ dans le tableau I$
2100      *
2101      * Algorithmme:
2102      * for x:= n downto 1 do
2103      *   begin
2104      *     I$(x) := U$(x) ;
2105      *   end ;
2106      *
2107 03E08 1FAF8F2      D1=(5) n
2108 03E0F 147      C=DAT1 A
2109      *
2110      * for x := n downto 0 do
2111      *
2112 03E12      kRUN10
2113      *
2114      * Assertion :
2115      *   C(A) := x
2116      *
2117 03E12 108      R0=C
2118 03E15 1BB88F2      D0=(5) DOPEU$      D0 := dope vecteuuuuuur source
2119 03E1C 1F178F2      D1=(5) DOPEI$      D1 := dope vecteuuuuuur destination
2120 03E23 8EF83F      GOSUBL ELMCPY
2121      *
2122 03E29 118      C=R0
2123 03E2C CE      C=C-1 A
2124 03E2E 8AE      ?C#0 A
2125 03E31 1E      GOYES  kRUN10
2126      *
2127      * end ;

```



```
2128 *
2129
2130 *
2131 * Renvoyer le numero de la ligne courante dans la var. A
2132 * et revenir a Basic.
2133 *
2134 03E33 1F1C6F2          D1=(5) i1
2135 03E3A 143             A=DAT1 A
2136 03E3D 6EC0           GOTO  EXIT
2137
2138 *****
2139 * VALIDE
2140 *
2141 * But: valider une ligne, c'est a dire copier l'entree de
2142 * l'utilisateur dans U$(I)
2143 * Entree:
2144 * - DSPBFS contient l'entree de l'utilisateur
2145 * Sortie:
2146 * - U$(I) contient les seuls caracteres entres
2147 * - D$(I) conient le display buffer tel quel
2148 * Abime:
2149 * Appelle: GETD$I, finign, DSP$00, POP15, GETU$I, DIMSTK
2150 * Niveaux:
2151 * Detail:
2152 * Algorithme:
2153 * adresse de D$(I)
2154 * recopier DSPBFS-DSPBFE dans D$(I)
2155 * finign
2156 * DSP$00
2157 * enlever le CR de la fin
2158 * adresse de U$(I)
2159 * mettre LEN(U$(I)) a jour
2160 * recopier Math-Stack dans U$(I)
2161 * Historique:
2162 * 86/11/23: P.D. & J.T. conception & codage
2163 * 86/11/24: P.D. ajout de documentation
2164 *****
2165
2166 03E41 VALIDE
2167 *
2168 * adresse de D$(I)
2169 *
2170 03E41 1B1C6F2          D0=(5) i1
2171 03E48 146             C=DAT0 A
2172 03E4B 8E064F          GOSUBL GETD$I
2173 *
2174 * recopier DSPBFS-DSPBFE dans D$(I)
2175 *
2176 03E51 1F00000          D1=(5) (=DSPBFE)-2
2177 03E58 D2              C=0 A
2178 03E5A 3106            LC(2) 96          Nb de boucles
2179 03E5E 5D0             GONC vld20          B.E.T.
2180 *
2181 * boucle pour determiner la longueur de D$(I)
2182 *
2183 03E61 14B            vld10 A=DAT1 B
2184 03E64 96C           ?A#0 B
```

```

2185 03E67 B0          GOYES vld30      sortie des qu'un caractere est
2186 03E69 1C1        D1=D1- 2        trouve
2187 03E6C A6E        vld20  C=C-1  B
2188 03E6F 51F        GONC   vld10
2189 03E72 B66        vld30  C=C+1  B
2190
2191 * C(A) = longueur en octets de la chaine lue dans DSPBUF
2192 *
2193 03E75 15C3        DAT0=C 4        LEN(D$(I)) := C(B)
2194 03E79 163        D0=D0+ 4
2195 03E7C 5E0        GONC   vld50
2196
2197 * boucle de recopie
2198 *
2199 03E7F 14B        vld40  A=DAT1 B
2200 03E82 148        DAT0=A B
2201 03E85 1C1        D1=D1- 2
2202 03E88 161        D0=D0+ 2
2203 03E8B A6E        vld50  C=C-1  B
2204 03E8E 50F        GONC   vld40
2205
2206 * finign : DSP$00
2207 *
2208 03E91 70DE        GOSUB  finign
2209 03E95 8F00000    GOSBVL =D1MSTK
2210 03E9C 850        ST=1  0        Pour retour apres DSP$00
2211 03E9F 8F00000    GOSBVL =DSP$00
2212 03EA6 8F00000    GOSBVL =POP1S
2213
2214 * Enlever le CR de la fin
2215 *
2216 03EAD 171        D1=D1+ 2
2217 03EB0 CC         A=A-1  A
2218 03EB2 CC         A=A-1  A        LEN(DSPBFS) := LEN(DSPBFS)-1 (octets)
2219 03EB4 100        R0=A        sauver la longueur dans R0
2220
2221 03EB7 1BA78F2    D0=(5) (DOPEI$)+9  D0 := longueur max des elements
2222 03EBE D2         C=0  A
2223 03EC0 15E3        C=DAT0 4
2224 03EC4 C6         C=C+C  A        C(A) := LEN(DISP$) en quartets
2225 03EC6 8B6        ?C<A  A
2226 03EC9 83        GOYES  strov2
2227
2228 * adresse de U$(I)
2229 *
2230 03ECB 1B1C6F2    D0=(5) i1
2231 03ED2 146        C=DAT0 A
2232 03ED5 8ECE3F    GOSUBL GETU$I    D0 := ^ LEN(U$(I))
2233
2234 * mettre LEN(U$(I)) a jour
2235 *
2236 03EDB 118        C=R0        restaurer la longueur de la chaine
2237 03EDE AD0        A=0  M
2238 03EE1 DA         A=C  A        A(5) := 0
2239 03EE3 81C        ASRB        A(A) := A(B) := longueur en octets
2240 03EE6 1583        DAT0=A 4
2241 03EEA 163        D0=D0+ 4

```

```
2242      *
2243      * recopier Math-Stack dans U$(I)
2244      *
2245      * recapitulatif :
2246      *   D0 = start of dest
2247      *   D1 = start of source
2248      *   A(A) = longueur en octets
2249      *
2250 03EED 132      AD0EX
2251 03EF0 133      AD1EX
2252 03EF3 DE      CAEX   A
2253 03EF5 136      CD0EX
2254 03EF8 C6      C=C+C  A
2255      *
2256      *   D0 = start of source
2257      *   D1 = start of dest
2258      *   C(A) = longueur en octets
2259      *
2260 03EFA 8D00000      GOVLNG =MOVEU3
2261
2262 03F01 3100      strov2  LC(2)  =eSTROV
2263 03F05 8D00000      GOVLNG =MFERR
2264
2265      *****
2266      * EXIT
2267      *
2268      * But: stoker la valeur de retour dans la variable A, et
2269      *   revenir a Basic par NXTSTM.
2270      * Entree:
2271      *   - A(A) = valeur a renvoyer (en hexa)
2272      *   - DOPEA = informations sauvees lors de l'evaluation de A
2273      *   - AVMEME = sauvegarde du pointeur de Math-Stack sauve
2274      *     lors de l'evaluation de A.
2275      * Sortie:
2276      *   - par NXTSTM
2277      * Abime: ouh la la !
2278      * Appelle: DEST, STORE, NXTSTM, HDFLT, AVE-D1, ATNCLR, NOSCLR
2279      * Niveaux: 6 (STORE)
2280      * Historique:
2281      *   86/11/22: P.D. & J.T. conception & codage
2282      *   86/11/24: P.D.      ajout de documentation
2283      *   *****
2284
2285 03F0C      EXIT
2286      *
2287      * Lors de l'evaluation de A, on avait sauvegarde ses
2288      * elements caracteristiques dans DOPEA. Il est maintenant
2289      * temps d'aller les rechercher, pour utiliser la sequence
2290      * DEST, STORE, etc. cf IDS I, page 13-13
2291      *
2292 03F0C 1B9F8F2      D0=(5) (DOPEA)+26      Variable A
2293 03F13 1F00000      D1=(5) =F-R1-3
2294 03F1A 15E0      C=DAT0 1
2295 03F1E 15D0      DAT1=C 1
2296
2297 03F22 184      D0=D0- 5
2298 03F25 1CE      D1=D1- 15
```

```
2299 03F28 146          C=DAT0 A
2300 03F2B 145          DAT1=C A
2301
2302 03F2E 184          D0=D0- 5
2303 03F31 146          C=DAT0 A
2304 03F34 135          D1=C
2305
2306 03F37 18F          D0=D0- 16
2307 03F3A 1567         C=DAT0 W
2308 03F3E AF5          B=C    W
2309
2310
2311
2312
2313 03F41 8F00000      GOSBVL =DEST          N'abime pas A(A)
2314 03F48 8F00000      GOSBVL =D1MSTK
2315
2316
2317
2318 03F4F 1B00000      D0=(5) =S-R1-2
2319 03F56 D2           C=0    A
2320 03F58 144          DAT0=C A
2321
2322 03F5B 8F00000      GOSBVL =HDFLT         A(W) := i en sortie (12 digits form)
2323 03F62 04           SETHEX
2324 03F64 1CF          D1=D1- 16
2325 03F67 1517         DAT1=A W              Math-Stack := A(W)
2326 03F6B 8F00000      GOSBVL =AVE=D1
2327 03F72 8F00000      GOSBVL =STORE        variable A := 1 en sortie
2328
2329
2330
2331 03F79 8F00000      GOSBVL =ATNCLR        On n'a jamais appuye sur [ATTN]
2332 03F80 8F00000      GOSBVL =NOSCRL        Le curseur est disponible
2333 03F87 8D00000      GOVLNG =NXTSTM        Au revoir...
2334
2335
2336
2337
2338 03F8E              END
```

ADDRSS	Ext		-	392						
ATNCLR	Ext		-	2007	2331					
AVE=D1	Ext		-	618	1505	2326				
AVMEMS	Ext		-	678						
BP-	Abs	15197	#00003B5D	-	1631	2049				
BP10	Abs	15401	#00003C29	-	1770					
BP12	Abs	15521	#00003CA1	-	1846	1849				
BP15	Abs	15542	#00003CB6	-	1850	1847				
BP20	Abs	15511	#00003C97	-	1843	1752				
BP25	Abs	15549	#00003CBD	-	1851	1845				
BP:	Abs	15190	#00003B56	-	1629	1940				
BSERR	Ext		-	205	402	1122	1557			
BSL4	Abs	13095	#00003327	-	301	176	324	614	703	726 749
BSL5	Abs	13092	#00003324	-	300	172	329	608	701	724 747
CHEDIT	Ext		-	1783						
CKSREQ	Ext		-	1848						
CNVUCR	Ext		-	1110						
COLLAP	Ext		-	499						
CRLF0F	Ext		-	1986						
CURSFL	Ext		-	1726						
D1MSTK	Ext		-	543	2209	2314				
D=AVMS	Ext		-	565						
DEST	Ext		-	2313						
DOPEA	Abs	194783	#0002F8DF	-	36	1506	2292			
DOPEB\$	Abs	194757	#0002F8C5	-	33	233	693			
DOPED\$	Abs	194770	#0002F8D2	-	34	231	1459	1630		
DOPEI\$	Abs	194673	#0002F871	-	43	239	487	926	2119	2221
DOPEM\$	Abs	194686	#0002F87E	-	44	237	502	544	611	1460 2048
DOPEP\$	Abs	194783	#0002F8DF	-	35	229	638			
DOPEU\$	Abs	194699	#0002F88B	-	45	235	742	2118		
DRANGE	Ext		-	1118						
DSP\$00	Ext		-	2211						
DSPBFE	Ext		-	2176						
DSPBFS	Ext		-	2029						
DSPCHA	Ext		-	1674						
DSPMSK	Ext		-	1635	1691					
Datap	Abs	13275	#000033DB	-	450	472				
ELMCPY	Abs	12728	#000031B8	-	71	1461	2120			
EXIT	Abs	16140	#00003F0C	-	2285	2059	2136			
EXPEXC	Ext		-	120	1504					
Erreur	Abs	14487	#00003897	-	1220	1260				
F-R1-0	Ext		-	1516						
F-R1-3	Ext		-	2293						
FINDA	Ext		-	1804	1852					
FINPUd	Ext		-	386						
=FINPUe	Abs	13144	#00003358	-	388					
FINPUp	Ext		-	387						
GET\$ST	Abs	13070	#0000330E	-	276	907	991	1654	2018	
GET010	Abs	13028	#000032E4	-	241	230	232	234	236	238
GETB\$I	Abs	12988	#000032BC	-	233	940	1689			
GETD\$I	Abs	12977	#000032B1	-	231	2172				
GETI\$I	Abs	13021	#000032DD	-	239					
GETM\$I	Abs	13010	#000032D2	-	237	555	971	1545	2014	
GETP\$I	Abs	12966	#000032A6	-	229	899				
GETU\$I	Abs	12999	#000032C7	-	235	959	1419	2232		
GETx\$I	Abs	13028	#000032E4	-	240	72	77	1650		
HDFLT	Ext		-	2322						

HEXDEC	Ext		-	558							
I\$005	Abs	13167	#0000336F	-	404	400					
I\$010	Abs	13179	#0000337B	-	411	418					
I\$020	Abs	13203	#00003393	-	422	414					
I\$030	Abs	13271	#000033D7	-	448	444					
I\$040	Abs	13279	#000033DF	-	452	427					
I\$050	Abs	13313	#00003401	-	470	454					
I\$060	Abs	13336	#00003418	-	480	465					
I\$100	Abs	13350	#00003426	-	486	448					
ILFART	Ext			-	1985						
MEMERR	Ext			-	686						
MFERR	Ext			-	209	980	2263				
MOVE*M	Ext			-	96						
MOVEU3	Ext			-	2260						
MPY	Ext			-	687						
MTHSTK	Ext			-	185						
NOSCRL	Ext			-	2332						
NXTSTM	Ext			-	2333						
OBCOLL	Ext			-	662						
POP1S	Ext			-	157	2212					
POPBUF	Ext			-	1850						
PP\$010	Abs	13447	#00003487	-	541	512					
PP\$050	Abs	13477	#000034A5	-	554	597					
PP\$070	Abs	13524	#000034D4	-	572	578					
RCL4	Abs	13109	#00003335	-	324	129	133	432	440	459	477
RCL5	Abs	13123	#00003343	-	329	131	434	461	479		
RDATTY	Ext			-	207						
RESJPC	Abs	194961	#0002F991	-	21	1774	1791				
RPTKY	Ext			-	1844						
S-R1-2	Ext			-	2318						
SCRLLR	Ext			-	1846						
STKCHR	Ext			-	522						
STORE	Ext			-	2327						
STSAVE	Ext			-	1720	1748					
TBLJMC	Ext			-	1131						
TRFMBF	Ext			-	547	593					
VALIDE	Abs	15937	#00003E41	-	2166	2083	2096				
actiona-05	Abs	14011	#000036BB	-	932	930					
actiona-10	Abs	14130	#00003732	-	982	977					
actionb	Abs	14358	#00003816	-	1137	1007	1041				
actionc	Abs	14377	#00003829	-	1148	1022	1058				
actiond	Abs	14429	#0000385D	-	1179	1004	1039				
actionl	Abs	14789	#000039C5	-	1402	1035					
actions	Abs	14445	#0000386D	-	1196	1024	1056				
actions-010	Abs	14463	#0000387F	-	1206	1203					
addsm\$	Abs	194737	#0002F8B1	-	776	992	1310	1341			
addsp\$	Abs	194725	#0002F8A5	-	773	908	1099				
addsu\$	Abs	194730	#0002F8AA	-	774	964	1315	1420			
admask	Abs	194743	#0002F8B7	-	778	945	1292	1373			
ancpsd	Abs	15257	#00003B99	-	1667	1679					
anpc10	Abs	15286	#00003BB6	-	1678	1657					
bitmap	Abs	14491	#0000389B	-	1239	1218	1410				
bm010	Abs	14511	#000038AF	-	1252	1396					
bm015	Abs	14567	#000038E7	-	1280	1273					
bm020	Abs	14613	#00003915	-	1303	1286					
bm030	Abs	14667	#0000394B	-	1326	1306					
bm035	Abs	14683	#0000395B	-	1337	1298					

memerr	Abs	13714	#00003592	-	686	674						
mpy	Abs	13721	#00003599	-	687	250	671	716	738			
mult	Abs	194750	#0002F8BE	-	780	1140	1153	1185	1216			
n	Abs	194810	#0002F8FA	-	37	191	492	668	692	881	1541	
					1934	1958	2084	2107				
pair	Abs	14913	#00003A41	-	1454	1446						
rptrev	Abs	14895	#00003A2F	-	1448	1455						
saut	Abs	14351	#0000380F	-	1131	1129						
saut0	Abs	14349	#0000380D	-	1130	1097						
saut1	Abs	14342	#00003806	-	1127	1114	1117					
saut2	Abs	14335	#000037FF	-	1124	1119						
st	Abs	14344	#00003808	-	1128	1126						
stkchr	Abs	13440	#00003480	-	522	575	580	582				
stmtr0	Abs	194673	#0002F871	-	24	43	44	45				
strov2	Abs	16129	#00003F01	-	2262	2226						
strovf	Abs	12955	#0000329B	-	208	164						
t1-2	Abs	14169	#00003759	-	1007	1002						
t1-3	Abs	14161	#00003751	-	1004	1001						
t2-2	Abs	14186	#0000376A	-	1022	1020						
t2-3	Abs	14194	#00003772	-	1024	1019						
t3-3	Abs	14211	#00003783	-	1039	1036						
t3-4	Abs	14219	#0000378B	-	1041	1037						
t4-3	Abs	14236	#0000379C	-	1056	1053						
t4-4	Abs	14244	#000037A4	-	1058	1054						
trfmbf	Abs	194757	#0002F8C5	-	23	33	34	35	36	37	783	
vark	Abs	194753	#0002F8C1	-	1236	1250	1391					
varp	Abs	194755	#0002F8C3	-	1237	1248	1271					
ver10	Abs	15066	#00003ADA	-	1544	1583						
ver20	Abs	15113	#00003B09	-	1561	1578						
ver30	Abs	15160	#00003B38	-	1577	1554						
vld10	Abs	15969	#00003E61	-	2183	2188						
vld20	Abs	15980	#00003E6C	-	2187	2179						
vld30	Abs	15986	#00003E72	-	2189	2185						
vld40	Abs	15999	#00003E7F	-	2199	2204						
vld50	Abs	16011	#00003E8B	-	2203	2195						

Input Parameters

Source file name is finput.a
Listing file name is finput.l
Object file name is finput.o
Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE ADFCN
2
3          * *****
4          * ADxxx functions
5          *
6          * Keyword(s):
7          *   ADSIZE, ADFIND, ADCREATE, ADDELETE, ADGET, ADPUT
8          * Author(s): Pierre David
9          * History:
10         *   No source available (never published)
11         *   Source file rebuilt from JPC LEX disassembly
12         *
13         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14         * *****
15
16 03F8E          ABS #3F8E
17
18         * ADxxx support routines
19 03F8E 133      o3F87 AD1EX
20 03F91 D2       C=0 A
21 03F93 3153    LCHEX 35
22 03F97 CA      A=A+C A
23 03F99 133    AD1EX
24 03F9C 15F3    C=DAT1 4
25 03FA0 C9      C=C+B A
26 03FA2 15D3    DAT1=C 4
27 03FA6 173    D1=D1+ 4
28 03FA9 D2      C=0 A
29 03FAB 15F3    C=DAT1 4
30 03FAF CC      A=A-1 A
31 03FB1 8B6    ?C<A A
32 03FB4 00      RTNYES
33 03FB6 C9      C=C+B A
34 03FB8 15D3    DAT1=C 4
35 03FBC 01      RTN
36 03FBE 1F00000 o3FB7 D1=(5) =STMTD0
37 03FC5 147    C=DAT1 A
38 03FC8 D5      B=C A
39 03FCA 174    D1=D1+ 5
40 03FCD 143    A=DAT1 A
41 03FD0 D2      C=0 A
42 03FD2 3153    LCHEX 35
43 03FD6 CA      A=A+C A
44 03FD8 131    D1=A
45 03FDB D2      C=0 A
46 03FDD 15F3    C=DAT1 4
47 03FE1 177    D1=D1+ 8
48 03FE4 8BD    ?B<=C A
49 03FE7 A1      GOYES o3FFA
50 03FE9 D2      C=0 A
51 03FEB 3163    LCHEX 36
52 03FEF 6555    GOTO o453E
53 03FF3 D2      o3FEC C=0 A
54 03FF5 15F3    C=DAT1 4
55 03FF9 133    AD1EX
56 03FFC CA      A=A+C A
57 03FFE 131    D1=A
```

```
58 04001 CD      o3FFA B=B-1 A
59 04003 8AD     ?B#0 A
60 04006 DE      GOYES o3FEC
61 04008 01      RTN
62 0400A D5      o4003 B=C A
63 0400C 6D00    GOTO o4013
64 04010 8F00000 o4009 GOSBVL =CNVUCR
65 04017 149     DAT1=A B
66 0401A 1C1     o4013 D1=D1- 2
67 0401D CD      B=B-1 A
68 0401F 50F     GONC o4009
69 04022 01      RTN
70 04024 1F00000 o401D D1=(5) =FUNCDO
71 0402B 143     A=DAT1 A
72 0402E 174     D1=D1+ 5
73 04031 147     C=DAT1 A
74 04034 135     D1=C
75 04037 D8      B=A A
76 04039 31F2    LCHEX 2F
77 0403D 5A0     GONC o4041
78 04040 14B     o4039 A=DAT1 B
79 04043 962     ?C=A B
80 04046 00      RTNYES
81 04048 1C1     o4041 D1=D1- 2
82 0404B CD      B=B-1 A
83 0404D 52F     GONC o4039
84 04050 03      RTNCC
85 04052 1F00000 o404B D1=(5) =FUNCDO
86 04059 147     C=DAT1 A
87 0405C D7      D=C A
88 0405E 14A     A=DAT0 B
89 04061 D1      B=0 A
90 04063 AE8     B=A B
91 04066 161     D0=D0+ 2
92 04069 174     D1=D1+ 5
93 0406C 147     C=DAT1 A
94 0406F 135     D1=C
95 04072 849     ST=0 9
96 04075 582     GONC o4097
97 04078 14A     o4071 A=DAT0 B
98 0407B 161     D0=D0+ 2
99 0407E 1C1     D1=D1- 2
100 04081 861    ?ST=0 1
101 04084 B0     GOYES o4088
102 04086 31F2    LCHEX 2F
103 0408A 962     ?C=A B
104 0408D D1      GOYES o40A3
105 0408F 8F00000 o4088 GOSBVL =CONVUC
106 04096 14F     C=DAT1 B
107 04099 966     ?C#A B
108 0409C E2      GOYES o40C3
109 0409E A6F     o4097 D=D-1 B
110 040A1 4D0     GOC o40A8
111 040A4 A6D     B=B-1 B
112 040A7 50D     GONC o4071
113 040AA 85A     o40A3 ST=1 10
114 040AD 01      RTN
```

```
115 040AF 861      o40A8 ?ST=0 1
116 040B2 E0      GOYES o40B9
117 040B4 14A     A=DAT0 B
118 040B7 31F2    LCHEX 2F
119 040BB 962     ?C=A B
120 040BE 70      GOYES o40BE
121 040C0 96D     o40B9 ?B#0 B
122 040C3 C0      GOYES o40C8
123 040C5 859     o40BE ST=1 9
124 040C8 01      RTN
125 040CA 9EA     o40C3 ?C>=A B
126 040CD DD      GOYES o40A3
127 040CF 84A     o40C8 ST=0 10
128 040D2 01      RTN
129 040D4 8F00000 o40CD GOSBVL =POP1S
130 040DB 8AC     ?A#0 A
131 040DE 80      GOYES o40DF
132 040E0 07      C=RSTK
133 040E2 6CD2    GOTO o43B8
134 040E6 137     o40DF CD1EX
135 040E9 C2      C=C+A A
136 040EB 135     D1=C
137 040EE 1B00000 D0=(5) =FUNCD1
138 040F5 144     DAT0=C A
139 040F8 184     D0=D0- 5
140 040FB AD2     C=0 M
141 040FE D6      C=A A
142 04100 81E     CSRB
143 04103 144     DAT0=C A
144 04106 700F    GOSUB o4003
145 0410A 08      CLRST
146 0410C 31E2    LCHEX 2E
147 04110 966     ?C#A B
148 04113 71      GOYES o4123
149 04115 850     ST=1 0
150 04118 1B00000 D0=(5) =FUNCD0
151 0411F 146     C=DAT0 A
152 04122 CE      C=C-1 A
153 04124 144     DAT0=C A
154 04127 5C0     GONC o412D
155 0412A 76FE    o4123 GOSUB o401D
156 0412E 450     GOC o412D
157 04131 851     ST=1 1
158 04134 1F00000 o412D D1=(5) =F-R0-1
159 0413B 143     A=DAT1 A
160 0413E D2      C=0 A
161 04140 174     D1=D1+ 5
162 04143 145     DAT1=C A
163 04146 8A8     ?A=0 A
164 04149 00      RTNYES
165 0414B E6      C=C+1 A
166 0414D 145     DAT1=C A
167 04150 1DB9    o4149 D1=(2) #9B
168 04154 147     C=DAT1 A
169 04157 134     D0=C
170 0415A 163     D0=D0+ 4
171 0415D 71FE    GOSUB o404B
```

```
172 04161 879      ?ST=1 9
173 04164 00      RTNYES
174 04166 86A     ?ST=0 10
175 04169 00      RTNYES
176 0416B 1F00000 D1=(5) =F-R0-0
177 04172 147     C=DAT1 A
178 04175 134     D0=C
179 04178 D0      A=0 A
180 0417A 15A3    A=DAT0 4
181 0417E C2      C=C+A A
182 04180 145     DAT1=C A
183 04183 174     D1=D1+ 5
184 04186 143     A=DAT1 A
185 04189 174     D1=D1+ 5
186 0418C 147     C=DAT1 A
187 0418F E6      C=C+1 A
188 04191 145     DAT1=C A
189 04194 8BE     ?C<=A A
190 04197 9B      GOYES o4149
191 04199 01      RTN
192 0419B 1F00000 o4194 D1=(5) =S-R0-0
193 041A2 147     C=DAT1 A
194 041A5 135     D1=C
195 041A8 01      RTN
196 041AA E6      o41A3 C=C+1 A
197 041AC E6      C=C+1 A
198 041AE C6      C=C+C A
199 041B0 133     AD1EX
200 041B3 CA      A=A+C A
201 041B5 D6      C=A A
202 041B7 133     AD1EX
203 041BA 01      RTN
204 041BC 1F00000 o41B5 D1=(5) =S-R0-1
205 041C3 147     C=DAT1 A
206 041C6 D7      D=C A
207 041C8 1C4     D1=D1- 5
208 041CB 147     C=DAT1 A
209 041CE 135     D1=C
210 041D1 DF      DCEX A
211 041D3 73DF    GOSUB o41A3
212 041D7 1F00000 D1=(5) =S-R0-0
213 041DE 145     DAT1=C A
214 041E1 135     D1=C
215 041E4 01      RTN
216 041E6 D2      o41DF C=0 A
217 041E8 31F1    LCHEX 1F
218 041EC 6853    GOTO o453E
219 041F0 161     o41E9 D0=D0+ 2
220 041F3 8F00000 GOSEVL =ADDRSS
221 041FA 4BE     GOC o41DF
222 041FD 136     CD0EX
223 04200 1B00000 D0=(5) =S-R0-0
224 04207 140     DAT0=A A
225 0420A 136     CD0EX
226 0420D 30E     LCHEX E
227 04210 160     o4209 D0=D0+ 1
228 04213 1520    A=DAT0 P
```

```
229 04217 906      ?C#A P
230 0421A E0      GOYES o4221
231 0421C 160     D0=D0+ 1
232 0421F 142     A=DAT0 A
233 04222 130     D0=A
234 04225 5AE     GONC o4209
235 04228 180     o4221 D0=D0- 1
236 0422B 14A     A=DAT0 B
237 0422E 31F1    LCHEX 1F
238 04232 966     ?C#A B
239 04235 1B      GOYES o41DF
240 04237 166     D0=D0+ 7
241 0423A D0      A=0 A
242 0423C 15A3    A=DAT0 4
243 04240 184     D0=D0- 5
244 04243 1564    C=DAT0 S
245 04247 94E     ?C#0 S
246 0424A 40      GOYES o4247
247 0424C E4      A=A+1 A
248 0424E D2      o4247 C=0 A
249 04250 307     LCHEX 7
250 04253 8B6     ?C<A A
251 04256 C0      GOYES o425B
252 04258 D2      C=0 A
253 0425A 3135    LCHEX 53
254 0425E 66E2    GOTO o453E
255 04262 160     o425B D0=D0+ 1
256 04265 D2      C=0 A
257 04267 15E3    C=DAT0 4
258 0426B D5      B=C A
259 0426D 167     D0=D0+ 8
260 04270 146     C=DAT0 A
261 04273 132     AD0EX
262 04276 EA      A=A-C A
263 04278 1B00000 D0=(5) =S-R0-0
264 0427F 146     C=DAT0 A
265 04282 140     DAT0=A A
266 04285 164     D0=D0+ 5
267 04288 D4      A=B A
268 0428A 140     DAT0=A A
269 0428D 134     D0=C
270 04290 01      RTN
271 04292 161     o428B D0=D0+ 2
272 04295 8F00000 GOBVL =EXPEX-
273 0429C 8F00000 GOBVL =RNDAXH
274 042A3 551     GONC o42B2
275 042A6 8A8     ?A=0 A
276 042A9 01      GOYES o42B2
277 042AB D6      C=A A
278 042AD 1F00000 D1=(5) =STMTD0
279 042B4 145     DAT1=C A
280 042B7 01      RTN
281 042B9 33B000 o42B2 LCHEX 000B
282 042BF 6582    GOTO o453E
283 042C3 14A     o42BC A=DAT0 B
284 042C6 161     D0=D0+ 2
285 042C9 8F00000 GOBVL =EOLXCK
```

```
286 042D0 445      GOC  o431E
287 042D3 8F00000  GOSBVL =EXPEX-
288 042DA 8F00000  o42D3 GOSBVL =POP1S
289 042E1 137      CD1EX
290 042E4 135      D1=C
291 042E7 C2        C=C+A A
292 042E9 135      D1=C
293 042EC D5        B=C A
294 042EE AF2      C=0 W
295 042F1 8A8      ?A=0 A
296 042F4 13       GOYES  o431E
297 042F6 CC       A=A-1 A
298 042F8 D6       C=A A
299 042FA F4       ASR A
300 042FC 8A8      ?A=0 A
301 042FF 70       GOYES  o42FF
302 04301 D2       C=0 A
303 04303 30F      LCHEX F
304 04306 133      o42FF AD1EX
305 04309 EA       A=A-C A
306 0430B 131      D1=A
307 0430E 1C0      D1=D1- 1
308 04311 80D0     P=C 0
309 04315 D2       C=0 A
310 04317 1571     C=DAT1 WP
311 0431B 20       P= 0
312 0431D D4       A=B A
313 0431F 131      D1=A
314 04322 550      GONC  o4321
315 04325 AF2      o431E C=0 W
316 04328 133      o4321 AD1EX
317 0432B 1F00000 D1=(5) =F-R0-0
318 04332 1557     DAT1=C W
319 04336 133      AD1EX
320 04339 01       RTN
321 0433B 17F      o4334 D1=D1+ 16
322 0433E D0       A=0 A
323 04340 15B3     A=DAT1 4
324 04344 34422E0 LCHEX 0E224
325 0434B 8A2      ?C=A A
326 0434E A0       GOYES  o4351
327 04350 D2       C=0 A
328 04352 31F3     LCHEX 3F
329 04356 03       RTNCC
330 04358 137      o4351 CD1EX
331 0435B 8F00000  GOSBVL =LOCADR
332 04362 5A0      GONC  o4366
333 04365 D2       o435E C=0 A
334 04367 31C3     LCHEX 3C
335 0436B 03       RTNCC
336 0436D 94B      o4366 ?D=0 S
337 04370 A0       GOYES  o4373
338 04372 A4F      D=D-1 S
339 04375 94F      ?D#0 S
340 04378 DE       GOYES  o435E
341 0437A 17F      o4373 D1=D1+ 16
342 0437D 174      D1=D1+ 5
```

```
343 04380 133      AD1EX
344 04383 1F00000  D1=(5) =F-R0-0
345 0438A 1577     C=DAT1 W
346 0438E 131      D1=A
347 04391 1537     A=DAT1 W
348 04395 978      ?A=0 W
349 04398 00       RTNYES
350 0439A 972      ?C=A W
351 0439D 00       RTNYES
352 0439F D2       C=0 A
353 043A1 31D3     LCHEX 3D
354 043A5 01       RTN
355
356
357
358 * *****
359 * ADSize entry
360 043A7 4412     NIBHEX 4412
361 * o43A4
362 043AB 301      =ADSIZE LCHEX 1
363 043AE 7C30     GOSUB o43E7
364 043B2 7460     GOSUB o4413
365 043B6 17F     D1=D1+ 16
366 043B9 D0       A=0 A
367 043BB 15B3     A=DAT1 4
368 043BF 7CC0     o43B8 GOSUB o4488
369 043C3 8F00000  GOBVL =HDFLT
370 043CA 868     ?ST=0 8
371 043CD 50       GOYES o43CB
372 043CF BCC     A=-A-1 S
373 043D2 04       o43CB SETHEX
374 043D4 1CF     D1=D1- 16
375 043D7 1517     DAT1=A W
376 043DB 8F00000  GOBVL =RSTK<R
377 043E2 07       C=RSTK
378 043E4 136     CD0EX
379 043E7 8D00000  GOVLNG =EXPR
380 043EE AFA     o43E7 A=C W
381 043F1 07       C=RSTK
382 043F3 108     R0=C
383 043F6 136     CD0EX
384 043F9 06       RSTK=C
385 043FB 8F00000  GOBVL =R<RSTK
386 04402 118     C=R0
387 04405 06       RSTK=C
388 04407 AC6     C=A S
389 0440A 814     ASRC
390 0440D 942     ?C=A S
391 04410 60       GOYES o440F
392 04412 67CE     GOTO o42D3
393 04416 6E0F     o440F GOTO o431E
394 0441A 7A60     o4413 GOSUB o4481
395 0441E 8F00000  GOBVL =R<RSTK
396 04425 8F00000  GOBVL =FILXQ$
397 0442C 463     GOC o445C
398 0442F 7C50     GOSUB o4488
399 04433 8F00000  GOBVL =POP1S
```



```
400 0443A 137      CD1EX
401 0443D C2       C=C+A A
402 0443F 135     D1=C
403 04442 7240    GOSUB o4481
404 04446 33A300  LCHEX 003A
405 0444C DA      o4445 A=C A
406 0444E 7F20    GOSUB o447A
407 04452 07      C=RSTK
408 04454 D6      C=A A
409 04456 858     ST=1 8
410 04459 F2      CSL A
411 0445B F6      CSR A
412 0445D DA      A=C A
413 0445F 6F5F    GOTO o43B8
414 04463 7120    o445C GOSUB o4481
415 04467 8F00000 GOSBVL =FINDF+
416 0446E 4DD     GOC o4445
417 04471 76CE    GOSUB o4334
418 04475 56D     GONC o4445
419 04478 848     ST=0 8
420 0447B 7200    GOSUB o447A
421 0447F 01      RTN
422 04481 8D00000 o447A GOVLNG =RSTK<R
423 04488 8D00000 o4481 GOVLNG =AVE=D1
424 0448F 8D00000 o4488 GOVLNG =D1MSTK
425
426
427                * *****
428                * ADFIND entry
429                * *****
430 04496 44423   NIBHEX 44423
431                * o4494
432 0449B 302     =ADFINE LCHEX 2
433 0449E 7C4F    GOSUB o43E7
434 044A2 137     CD1EX
435 044A5 1F00000 D1=(5) =F-R1-0
436 044AC 145     DAT1=C A
437 044AF 135     D1=C
438 044B2 8F00000 GOSBVL =POP1S
439 044B9 137     CD1EX
440 044BC C2      C=C+A A
441 044BE 135     D1=C
442 044C1 755F    GOSUB o4413
443 044C5 17F    D1=D1+ 16
444 044C8 D2      C=0 A
445 044CA 15F3   C=DAT1 4
446 044CE 177    D1=D1+ 8
447 044D1 133    AD1EX
448 044D4 1F00000 D1=(5) =F-R0-0
449 044DB 141    DAT1=A A
450 044DE 174    D1=D1+ 5
451 044E1 145    DAT1=C A
452 044E4 17A    D1=D1+ 11
453 044E7 147    C=DAT1 A
454 044EA 135    D1=C
455 044ED 73EB   GOSUB o40CD
456 044F1 1B00000 D0=(5) =F-R0-2
```

```
457 044F8 142      A=DAT0 A
458 044FB 879      ?ST=1 9
459 044FE 90       GOYES o4500
460 04500 870      ?ST=1 0
461 04503 40       GOYES o4500
462 04505 D0       A=0 A
463 04507 67BE     o4500 GOTO o43B8
464 0450B 850      o4504 ST=1 0
465 0450E 6600     GOTO o450E
466 04512 840      o450B ST=0 0
467 04515 8F00000  o450E GOSBVL =FSPECx
468 0451C 482      GOC o453E
469 0451F 978      ?A=0 W
470 04522 D1       GOYES o4538
471 04524 AC2      C=0 S
472 04527 943      ?D=C S
473 0452A 22       GOYES o4545
474 0452C B46      C=C+1 S
475 0452F 943      ?D=C S
476 04532 A1       GOYES o4545
477 04534 AC2      C=0 S
478 04537 A4E      C=C-1 S
479 0453A 943      ?D=C S
480 0453D F0       GOYES o4545
481 0453F D2       o4538 C=0 A
482 04541 31A3     LCHEX 3A
483 04545 8D00000  o453E GOVLNG =BSERR
484 0454C 860      o4545 ?ST=0 0
485 0454F 71       GOYES o455F
486 04551 1F00000  D1=(5) =S-R0-0
487 04558 1517     DAT1=A W
488 0455C 17F      D1=D1+ 16
489 0455F AFB      C=D W
490 04562 1557     DAT1=C W
491 04566 8F00000  o455F GOSBVL =FINDF
492 0456D D1       B=0 A
493 0456F 440      GOC o456D
494 04572 E5       B=B+1 A
495 04574 870      o456D ?ST=1 0
496 04577 00       RTNYES
497 04579 8AD      ?B#0 A
498 0457C 50       GOYES o457A
499 0457E 56C      GONC o453E
500 04581 137      o457A CD1EX
501 04584 1F00000  D1=(5) =STMTD1
502 0458B 145      DAT1=C A
503 0458E 137      CD1EX
504 04591 01       RTN
505 04593 61BF     o458C GOTO o453E
506
507
508
509
510
511 04597 00000     REL(5) =ADCREd
512 0459C 00000     REL(5) =ADCREp
513
*
* *****
* ADCREATE entry
* *****
* o459A
```

```
514 045A1 766F      =ADCREe GOSUB o4504
515 045A5 8A9       ?B=0 A
516 045A8 C0        GOYES o45AD
517 045AA D2        C=0 A
518 045AC 31B3      LCHEX 3B
519 045B0 649F      GOTO o453E
520 045B4 7B0D      o45AD GOSUB o42BC
521 045B8 1F00000   D1=(5) =S-R1-0
522 045BF 1577      C=DAT1 W
523 045C3 AF7       D=C W
524 045C6 D2        C=0 A
525 045C8 31D3      LCHEX 3D
526 045CC 8F00000   GOSBVL =CRETf+
527 045D3 4FB       GOC o458C
528 045D6 119       C=R1
529 045D9 135       D1=C
530 045DC 1B00000   D0=(5) =S-R0-0
531 045E3 1527      A=DAT0 W
532 045E7 1517      DAT1=A W
533 045EB 17F       D1=D1+ 16
534 045EE 33422E    LCHEX E224
535 045F4 15D3      DAT1=C 4
536 045F8 17F       D1=D1+ 16
537 045FB 174       D1=D1+ 5
538 045FE 19B9      D0=(2) #9B
539 04602 1567      C=DAT0 W
540 04606 1557      DAT1=C W
541 0460A 17F       D1=D1+ 16
542 0460D AF2       C=0 W
543 04610 15D7      DAT1=C 8
544 04614 8D00000   GOVLNG =NXTSTM
545 0461B 692F      o4614 GOTO o453E
546
547
548                * *****
549                * ADDELETE entry
550                * *****
551 0461F 00000      REL(5) =ADDELd
552 04624 00000      REL(5) =ADDELp
553                * o4622
554 04629 75EE       =ADDELe GOSUB o450B
555 0462D 716C       GOSUB o428B
556 04631 7E8C       GOSUB o42BC
557 04635 1F00000   D1=(5) =STMTD1
558 0463C 147       C=DAT1 A
559 0463F 135       D1=C
560 04642 75FC       GOSUB o4334
561 04646 54D       GONC o4614
562 04649 7179      GOSUB o3FB7
563 0464D D2        C=0 A
564 0464F 15F3      C=DAT1 4
565 04653 10B       R3=C
566 04656 133       AD1EX
567 04659 CA        A=A+C A
568 0465B 8F00000   GOSBVL =OBCOLL
569 04662 1F00000   D1=(5) =STMTD1
570 04669 147       C=DAT1 A
```

```
571 0466C 8F00000    GOSEVL =MGOSUB
572 04673 7F310      CON(5) #13F7
573 04678 42A        GOC  o4614
574 0467B 1F00000    D1=(5) =STMTD0
575 04682 143        A=DAT1 A
576 04685 174        D1=D1+ 5
577 04688 147        C=DAT1 A
578 0468B 135        D1=C
579 0468E D1         B=0 A
580 04690 CD         B=B-1 A
581 04692 78F8       GOSUB  o3F87
582 04696 8D00000    GOVLNG =NXTSTM
583 0469D D2         o4696 C=0 A
584 0469F 3152       LCHEX 25
585 046A3 61AE       o469C GOTO  o453E
586
587
588                 * *****
589                 * ADGET entry
590                 * *****
591 046A7 00000       REL(5) =ADGETd
592 046AC 00000       REL(5) =ADGETp
593                 * o46AA
594 046B1 7D5E       =ADGETe GOSUB  o450B
595 046B5 773B       GOSUB  o41E9
596 046B9 75DB       GOSUB  o428B
597 046BD 720C       GOSUB  o42BC
598 046C1 1F00000    D1=(5) =STMTD1
599 046C8 147        C=DAT1 A
600 046CB 135        D1=C
601 046CE 796C       GOSUB  o4334
602 046D2 50D        GONC  o469C
603 046D5 75E8       GOSUB  o3FB7
604 046D9 137        CD1EX
605 046DC D7         D=C A
606 046DE 134        D0=C
607 046E1 163        D0=D0+ 4
608 046E4 1F00000    D1=(5) =S-R0-1
609 046EB 143        A=DAT1 A
610 046EE 307        LCHEX 7
611 046F1 816        CSRC
612 046F4 D2         o46ED C=0 A
613 046F6 14E        C=DAT0 B
614 046F9 9E2        ?C>A B
615 046FC 1A         GOYES  o4696
616 046FE E6         C=C+1 A
617 04700 C6         C=C+C A
618 04702 132        AD0EX
619 04705 CA         A=A+C A
620 04707 132        AD0EX
621 0470A A4E        C=C-1 S
622 0470D 56E        GONC  o46ED
623 04710 DB         C=D A
624 04712 134        D0=C
625 04715 163        D0=D0+ 4
626 04718 7F7A       GOSUB  o4194
627 0471C 307        LCHEX 7
```

```
628 0471F 816      CSRC
629 04722 D0       o471B A=0 A
630 04724 14A     A=DAT0 B
631 04727 1593    DAT1=A 4
632 0472B D6      C=A A
633 0472D 797A    GOSUB o41A3
634 04731 6900    GOTO o4734
635 04735 14E     o472E C=DAT0 B
636 04738 14D     DAT1=C B
637 0473B 161     o4734 D0=D0+ 2
638 0473E 1C1     D1=D1- 2
639 04741 A6C     A=A-1 B
640 04744 50F     GONC o472E
641 04747 717A    GOSUB o41B5
642 0474B A4E     C=C-1 S
643 0474E 53D     GONC o471B
644 04751 8D00000 GOVLNG =NXTSTM
645 04758 6CED    o4751 GOTO o453E
646 0475C 6C11    o4755 GOTO o4872
647
648
649
650
651
652 04760 00000   REL(5) =ADPUTd
653 04765 00000   REL(5) =ADPUTp
654
655 0476A 74AD    =ADPUTe GOSUB o450B
656 0476E 7E7A    GOSUB o41E9
657 04772 7D4B    GOSUB o42BC
658 04776 1F00000 D1=(5) =STMTD1
659 0477D 147     C=DAT1 A
660 04780 135     D1=C
661 04783 74BB    GOSUB o4334
662 04787 50D     GONC o4751
663 0478A 1B00000 D0=(5) =S-R0-3
664 04791 D2      C=0 A
665 04793 15C0    DAT0=C 1
666 04797 8F00000 GOSBVL =OBCOLL
667 0479E 134     D0=C
668 047A1 163     D0=D0+ 4
669 047A4 136     CD0EX
670 047A7 1B00000 D0=(5) =AVMEMS
671 047AE 144     DAT0=C A
672 047B1 841     ST=0 1
673 047B4 73E9    o47AD GOSUB o4194
674 047B8 D0      A=0 A
675 047BA 15B3    A=DAT1 4
676 047BE D2      C=0 A
677 047C0 3195    LCHEX 59
678 047C4 8B6     ?C<A A
679 047C7 59      GOYES o4755
680 047C9 D6      C=A A
681 047CB E6      C=C+1 A
682 047CD C6      C=C+C A
683 047CF 8F00000 GOSBVL =MEMCKL
684 047D6 418     GOC o4751
```

```
685 047D9 7EB9      GOSUB o4194
686 047DD 1B00000   D0=(5) =AVMEMS
687 047E4 146      C=DAT0 A
688 047E7 134      D0=C
689 047EA D2       C=0 A
690 047EC 15F3     C=DAT1 4
691 047F0 14C     DAT0=C B
692 047F3 D7       D=C A
693 047F5 71B9     GOSUB o41A3
694 047F9 31F2     LCHEX 2F
695 047FD 6910     GOTO o4810
696 04801 1C1     o47FA D1=D1- 2
697 04804 14B     A=DAT1 B
698 04807 148     DAT0=A B
699 0480A 871     ?ST=1 1
700 0480D A0      GOYES o4810
701 0480F 966     ?C#A B
702 04812 50      GOYES o4810
703 04814 851     ST=1 1
704 04817 161     o4810 D0=D0+ 2
705 0481A CF      D=D-1 A
706 0481C 54E     GONC o47FA
707 0481F 871     ?ST=1 1
708 04822 80      GOYES o4823
709 04824 14C     DAT0=C B
710 04827 161     D0=D0+ 2
711 0482A 1F00000 o4823 D1=(5) =AVMEMS
712 04831 136     CD0EX
713 04834 143     A=DAT1 A
714 04837 145     DAT1=C A
715 0483A 871     ?ST=1 1
716 0483D A1      GOYES o4850
717 0483F 130     D0=A
718 04842 14A     A=DAT0 B
719 04845 3195    LCHEX 59
720 04849 962     ?C=A B
721 0484C D2      GOYES o4872
722 0484E B64     A=A+1 B
723 04851 148     DAT0=A B
724 04854 851     ST=1 1
725 04857 7169    o4850 GOSUB o41B5
726 0485B 1B00000 D0=(5) =S-R0-3
727 04862 15A0    A=DAT0 1
728 04866 B04     A=A+1 P
729 04869 1580    DAT0=A 1
730 0486D 307     LCHEX 7
731 04870 986     ?C<A P
732 04873 01      GOYES o487C
733 04875 6E3F    GOTO o47AD
734 04879 D2      o4872 C=0 A
735 0487B 3152    LCHEX 25
736 0487F 65CC    GOTO o453E
737 04883 1B00000 o487C D0=(5) =AVMEMS
738 0488A 142     A=DAT0 A
739 0488D 184     D0=D0- 5
740 04890 146     C=DAT0 A
741 04893 134     D0=C
```

```
742 04896 EA      A=A-C A
743 04898 1583    DAT0=A 4
744 0489C 163     D0=D0+ 4
745 0489F D0      A=0 A
746 048A1 14A     A=DAT0 B
747 048A4 1F00000 D1=(5) =FUNCD0
748 048AB 141     DAT1=A A
749 048AE D8      B=A A
750 048B0 8F00000 GOSEVL =COLLAP
751 048B7 1F00000 D1=(5) =FUNCD1
752 048BE 145     DAT1=C A
753 048C1 135     D1=C
754 048C4 8F00000 GOSEVL =D=AVMS
755 048CB 591     GONC 048DE
756 048CE 161     048C7 D0=D0+ 2
757 048D1 14A     A=DAT0 B
758 048D4 8F00000 GOSEVL =CONVUC
759 048DB AE6     C=A B
760 048DE 8F00000 GOSEVL =STKCHR
761 048E5 CD      048DE B=B-1 A
762 048E7 56E     GONC 048C7
763 048EA 08      CLRST
764 048EC 1F00000 D1=(5) =STMTD1
765 048F3 143     A=DAT1 A
766 048F6 D2      C=0 A
767 048F8 3153    LCHEX 35
768 048FC CA      A=A+C A
769 048FE 130     D0=A
770 04901 3180    LCHEX 08
771 04905 CA      A=A+C A
772 04907 15E3    C=DAT0 4
773 0490B 1DB9    D1=(2) #9B
774 0490F 141     DAT1=A A
775 04912 174     D1=D1+ 5
776 04915 145     DAT1=C A
777 04918 7818    GOSUB 0412D
778 0491C D2      C=0 A
779 0491E 10B     R3=C
780 04921 1B00000 D0=(5) =STMTD1
781 04928 146     C=DAT0 A
782 0492B 19B9    D0=(2) #9B
783 0492F 142     A=DAT0 A
784 04932 8F00000 GOSEVL =MGOSUB
785 04939 7F310   CON(5) #13F7
786 0493E 472     GOC 0495F
787 04941 1B00000 D0=(5) =STMTD1
788 04948 146     C=DAT0 A
789 0494B 135     D1=C
790 0494E 195A    D0=(2) #A5
791 04952 142     A=DAT0 A
792 04955 D1      B=0 A
793 04957 E5      B=B+1 A
794 04959 8EF26F  GOSUBL 03F87
795 0495F 8D00000 GOVLNG =NXTSTM
796 04966 6EDB    0495F GOTO 0453E
797
798 0496A      END
```

ADCREd	Ext		-	511						
=ADCREe	Abs	17825 #000045A1	-	514						
ADCREp	Ext		-	512						
ADDELd	Ext		-	551						
=ADDELe	Abs	17961 #00004629	-	554						
ADDELp	Ext		-	552						
ADDRSS	Ext		-	220						
=ADFINE	Abs	17563 #0000449B	-	432						
ADGETd	Ext		-	591						
=ADGETe	Abs	18097 #000046B1	-	594						
ADGETp	Ext		-	592						
ADPUTd	Ext		-	652						
=ADPUTe	Abs	18282 #0000476A	-	655						
ADPUTp	Ext		-	653						
=ADSIZE	Abs	17323 #000043AB	-	362						
AVE=D1	Ext		-	423						
AVMEMS	Ext		-	670	686	711	737			
BSERR	Ext		-	483						
CNVUCR	Ext		-	64						
COLLAP	Ext		-	750						
CONVUC	Ext		-	105	758					
CRETf+	Ext		-	526						
D1MSTK	Ext		-	424						
D=AVMS	Ext		-	754						
EOLXCK	Ext		-	285						
EXPEX-	Ext		-	272	287					
EXPR	Ext		-	379						
F-R0-0	Ext		-	176	317	344	448			
F-R0-1	Ext		-	158						
F-R0-2	Ext		-	456						
F-R1-0	Ext		-	435						
FILXQ\$	Ext		-	396						
FINDF	Ext		-	491						
FINDF+	Ext		-	415						
FSPECx	Ext		-	467						
FUNCD0	Ext		-	70	85	150	747			
FUNCD1	Ext		-	137	751					
HDFLT	Ext		-	369						
LOCADR	Ext		-	331						
MEMCKL	Ext		-	683						
MGOSUB	Ext		-	571	784					
NXTSTM	Ext		-	544	582	644	795			
OBCOLL	Ext		-	568	666					
POP1S	Ext		-	129	288	399	438			
R<RSTK	Ext		-	385	395					
RNDAHx	Ext		-	273						
RSTK<R	Ext		-	376	422					
S-R0-0	Ext		-	192	212	223	263	486	530	
S-R0-1	Ext		-	204	608					
S-R0-3	Ext		-	663	726					
S-R1-0	Ext		-	521						
STKCHR	Ext		-	760						
STMTD0	Ext		-	36	278	574				
STMTD1	Ext		-	501	557	569	598	658	764	780
				787						
o3F87	Abs	16270 #00003F8E	-	19	581	794				
o3FB7	Abs	16318 #00003FB E	-	36	562	603				

o3FEC	Abs	16371	#00003FF3	-	53	60			
o3FFA	Abs	16385	#00004001	-	58	49			
o4003	Abs	16394	#0000400A	-	62	144			
o4009	Abs	16400	#00004010	-	64	68			
o4013	Abs	16410	#0000401A	-	66	63			
o401D	Abs	16420	#00004024	-	70	155			
o4039	Abs	16448	#00004040	-	78	83			
o4041	Abs	16456	#00004048	-	81	77			
o404B	Abs	16466	#00004052	-	85	171			
o4071	Abs	16504	#00004078	-	97	112			
o4088	Abs	16527	#0000408F	-	105	101			
o4097	Abs	16542	#0000409E	-	109	96			
o40A3	Abs	16554	#000040AA	-	113	104	126		
o40A8	Abs	16559	#000040AF	-	115	110			
o40B9	Abs	16576	#000040C0	-	121	116			
o40BE	Abs	16581	#000040C5	-	123	120			
o40C3	Abs	16586	#000040CA	-	125	108			
o40C8	Abs	16591	#000040CF	-	127	122			
o40CD	Abs	16596	#000040D4	-	129	455			
o40DF	Abs	16614	#000040E6	-	134	131			
o4123	Abs	16682	#0000412A	-	155	148			
o412D	Abs	16692	#00004134	-	158	154	156	777	
o4149	Abs	16720	#00004150	-	167	190			
o4194	Abs	16795	#0000419B	-	192	626	673	685	
o41A3	Abs	16810	#000041AA	-	196	211	633	693	
o41B5	Abs	16828	#000041BC	-	204	641	725		
o41DF	Abs	16870	#000041E6	-	216	221	239		
o41E9	Abs	16880	#000041F0	-	219	595	656		
o4209	Abs	16912	#00004210	-	227	234			
o4221	Abs	16936	#00004228	-	235	230			
o4247	Abs	16974	#0000424E	-	248	246			
o425B	Abs	16994	#00004262	-	255	251			
o428B	Abs	17042	#00004292	-	271	555	596		
o42B2	Abs	17081	#000042B9	-	281	274	276		
o42BC	Abs	17091	#000042C3	-	283	520	556	597	657
o42D3	Abs	17114	#000042DA	-	288	392			
o42FF	Abs	17158	#00004306	-	304	301			
o431E	Abs	17189	#00004325	-	315	286	296	393	
o4321	Abs	17192	#00004328	-	316	314			
o4334	Abs	17211	#0000433B	-	321	417	560	601	661
o4351	Abs	17240	#00004358	-	330	326			
o435E	Abs	17253	#00004365	-	333	340			
o4366	Abs	17261	#0000436D	-	336	332			
o4373	Abs	17274	#0000437A	-	341	337			
o43B8	Abs	17343	#000043BF	-	368	133	413	463	
o43CB	Abs	17362	#000043D2	-	373	371			
o43E7	Abs	17390	#000043EE	-	380	363	433		
o440F	Abs	17430	#00004416	-	393	391			
o4413	Abs	17434	#0000441A	-	394	364	442		
o4445	Abs	17484	#0000444C	-	405	416	418		
o445C	Abs	17507	#00004463	-	414	397			
o447A	Abs	17537	#00004481	-	422	406	420		
o4481	Abs	17544	#00004488	-	423	394	403	414	
o4488	Abs	17551	#0000448F	-	424	368	398		
o4500	Abs	17671	#00004507	-	463	459	461		
o4504	Abs	17675	#0000450B	-	464	514			
o450B	Abs	17682	#00004512	-	466	554	594	655	

o450E	Abs	17685	#00004515	-	467	465				
o4538	Abs	17727	#0000453F	-	481	470				
o453E	Abs	17733	#00004545	-	483	52	218	254	282	468 499
					505	519	545	585	645	736 796
o4545	Abs	17740	#0000454C	-	484	473	476	480		
o455F	Abs	17766	#00004566	-	491	485				
o456D	Abs	17780	#00004574	-	495	493				
o457A	Abs	17793	#00004581	-	500	498				
o458C	Abs	17811	#00004593	-	505	527				
o45AD	Abs	17844	#000045B4	-	520	516				
o4614	Abs	17947	#0000461B	-	545	561	573			
o4696	Abs	18077	#0000469D	-	583	615				
o469C	Abs	18083	#000046A3	-	585	602				
o46ED	Abs	18164	#000046F4	-	612	622				
o471B	Abs	18210	#00004722	-	629	643				
o472E	Abs	18229	#00004735	-	635	640				
o4734	Abs	18235	#0000473B	-	637	634				
o4751	Abs	18264	#00004758	-	645	662	684			
o4755	Abs	18268	#0000475C	-	646	679				
o47AD	Abs	18356	#000047B4	-	673	733				
o47FA	Abs	18433	#00004801	-	696	706				
o4810	Abs	18455	#00004817	-	704	695	700	702		
o4823	Abs	18474	#0000482A	-	711	708				
o4850	Abs	18519	#00004857	-	725	716				
o4872	Abs	18553	#00004879	-	734	646	721			
o487C	Abs	18563	#00004883	-	737	732				
o48C7	Abs	18638	#000048CE	-	756	762				
o48DE	Abs	18661	#000048E5	-	761	755				
o495F	Abs	18790	#00004966	-	796	786				

Input Parameters

Source file name is adfcn.a

Listing file name is adfcn.l

Object file name is adfcn.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE ATTN
2
3          * *****
4          * ATTN key control command
5          *
6          * Keyword(s): ATTN
7          * Author(s):
8          *   Pierre David, Michel Martinet
9          * History:
10         *   Initially published in JPC#23
11         *
12         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
13         * *****
14
15 0496A          ABS          #496A
16
17         * *****
18         * ATTN entry
19         * *****
20 0496A 00000          REL(5) =ONOFFd
21 0496F 00000          REL(5) =ONOFFp
22 04974 1F00000 =ATTNe D1=(5) =ATNDIS
23 0497B 14A          A=DAT0 B
24 0497E 1590          DAT1=A 1
25         * above statement works, because the ON and OFF tokens
26         * are actually #E0 and #E1
27 04982 8D00000          GOVLNG =NXTSTM
28
29 04989          END
```

ATNDIS	Ext	-	22
=ATTNe	Abs	18804 #00004974	- 22
NXTSTM	Ext	-	27
ONOFFd	Ext	-	20
ONOFFp	Ext	-	21

Input Parameters

Source file name is attn.a

Listing file name is attn.l

Object file name is attn.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE BLIST
2
3          * *****
4          * Formatted listing commands
5          *
6          * Keyword(s):
7          *   PBLIST, DBLIST
8          * Author(s):
9          *   Jean-Pierre Bondu, Pierre David, Janick Taillandier
10         * History:
11         *   Published in JPC #53
12         *   Rev X (R. Rosenbaum):
13         *     changed default indent value from 0 to 2
14         *
15         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
16         * *****
17
18 04989          ABS      #4989
19
20         * system constants:
21         DISPt EQU      #00000
22         PRINTt EQU     #00001
23         LEOL EQU       #00002
24         oBSsod EQU     #00011
25         oFLENh EQU     #00020
26         oFLSTr EQU     #00031
27         oFTYPh EQU     #00010
28
29         id EQU        #E1      JPC LEX ID
30
31         *****
32         * Definitions empruntees a STRUC2
33         *****
34
35         * Quartets de reconnaissance
36         qENDL EQU      0        END LOOP
37         qENDS EQU      1        END SELECT
38         qENDI EQU      2        END IF
39
40         * Tokens de STRUC2
41         tEND2 EQU      66
42         tWHILE EQU     67
43         tREPEAT EQU    68
44         tUNTIL EQU     69
45         tLEAVE EQU     70
46         tLOOP EQU      96
47         tSELECT EQU    97
48         tCASE EQU      98
49         tIF2 EQU       99
50         tELSE2 EQU    100
51
52         *****
53         * ROUTINES D'EXECUTION
54         *****
55
56         * local storage address definitions
57         TRFMBF EQU     #2F8C5
```

```
58
59      INDVAL EQU    00+TRFMBF  5 q : valeur indentation
60      CURIND EQU    05+TRFMBF  5 q : indentation cour.
61      OUTYPE EQU    10+TRFMBF  1 q : 0: fichier, 1: std
62      OUTADR EQU    11+TRFMBF  5 q : adresse ds fichier
63      OUTHDR EQU    16+TRFMBF  5 q : header du fichier
64
65      LDEB  EQU    21+TRFMBF  5 q : ligne de debut
66      LFIN  EQU    26+TRFMBF  5 q : ligne de fin
67
68      INFILE EQU    31+TRFMBF  5 q : adresse ds fichier
69      INEND  EQU    36+TRFMBF  5 q : fin du fichier
70
71      INDARM EQU    41+TRFMBF  1 q : 0:std, 1:DATA, 2:!  
72
73      INDCOU EQU    42+TRFMBF  2 q : valeur courante
74      INDLIG EQU    44+TRFMBF  2 q : en debut de ligne
75
76      SAUTAV EQU    46+TRFMBF  1 q : sauter ligne avant
77      SAUTAP EQU    47+TRFMBF  1 q : sauter ligne apres
78
79      TMP    EQU    48+TRFMBF  au moins 2 q.
80
81      inDATA EQU    1          pour INDARM
82      inREM  EQU    2          pour INDARM
83
84      clRIEN EQU    0
85      clNDDF EQU    1
86      clSUB  EQU    2
87      clDEF1 EQU    3
88      clDEF2 EQU    4
89      clDATA EQU    5
90      clLBL  EQU    6
91      clREM  EQU    7
92      clIF   EQU    8
93
94
95
96
97      *****
98      * PBLIST / DBLIST
99      *
100     * But : produire un listing structure sur
101     * l'imprimante ou dans un fichier.
102     * Detail :
103     *   initialiser les variables globales
104     *   initialiser les options par default
105     *
106     *   tant que non fin de ligne tokenisee
107     *   faire
108     *     selon token
109     *       tCOMMA :
110     *         LDEB, LFIN := numeros de ligne
111     *         tester la validite
112     *       tTO :
113     *         evaluer le specificateur de fichier
114     *         chercher le fichier
```



```
115      *          si il existe
116      *          alors "File Exists"
117      *          fin si
118      *          stocker les parametres pour creation
119      *          tXWORD :
120      *          INDVAL := evaluer l'expression
121      *          autre :
122      *          evaluer le fichier
123      *          chercher le fichier
124      *          si il n'existe pas
125      *          alors "File Not Found"
126      *          fin si
127      *          verifier type et protections
128      *          INFILE := ^ fichier
129      *          fin selon
130      *          fin tantque
131      *          creer le fichier de sortie si necessaire
132      *          pour toutes les lignes du fichier
133      *          decompiler et lister
134      * Historique :
135      * 86/05/.. : JPB      conception & codage
136      * 88/01/10 : PD & JT ajout indentation & fichier
137      * *****
138
139      * *****
140      * PBLIST entry
141      * *****
142 04989 00000      REL(5) =PBLISd
143 0498E 00000      REL(5) =PBLISp
144 04993 31F1      =PBLISe LC(2) (PRINTt)*16+#F
145 04997 6810      GOTO LIST05
146
147 0499B 8D00000   bserr GOVLNG =BSERR
148
149
150      * *****
151      * DBLIST entry
152      * *****
153 049A2 00000      REL(5) =PBLISd
154 049A7 00000      REL(5) =PBLISp
155 049AC 31F0      =DBLISe LC(2) (DISPt)*16+#F
156
157 049B0 1F00000   LIST05 D1=(5) =MLFFLG Destination du xBLIST
158 049B7 14D      DAT1=C B
159      * Initialisation des variables globales
160      * INDVAL = 0 (pas d'indentation)          5 q
161      * CURIND = 0 (indentation courante = 0)   5 q
162      * OUTYPE = 0 (sortie std et pas fichier)  1 q
163
164      *** change
165      * rev D
166      *          D1=(5) INDVAL
167      *          C=0 W
168      *          DAT1=C 11
169      * LDEB = ligne de debut
170      * LFIN = ligne de fin
171      *          D1=(2) LDEB
```

```

172          *          C=C+1  A
173          *  rev X: changed default indent value from 0 to 2
174 049BA 1D5C          D1=(2)  INDVAL
175 049BE AF2          C=0    W
176 049C1 302         LCHEX   2
177 049C4 15DA         DAT1=C  11
178 049C8 1DAD         D1=(2)  LDEB
179 049CC CE          C=C-1  A
180          *** end of change
181
182 049CE 145          DAT1=C  A          LDEB := 0001 par default
183 049D1 174          D1=D1+ (LFIN)-(LDEB)
184 049D4 339999      LCHEX   9999
185 049DA 145          DAT1=C  A          LFIN := 0999 par default
186
187 049DD 1DEE         D1=(2)  INDARM    INDARM, INDCOU, INDLIG,
188 049E1 AF2          C=0    W          ... SAUTAV et SAUTAP
189 049E4 15D6         DAT1=C  7          ... := 0
190          * Chercher l'adresse du fichier courant
191 049E8 1E0000      D1=(4)  =CURRST
192 049EE 147          C=DAT1  A
193 049F1 1E4E8F      D1=(4)  INFILE
194 049F7 145          DAT1=C  A          INFILE := ^ debut fichier
195
196 049FA 14A          loop  A=DAT0  B          A(B) := token courant
197 049FD 8F00000     GOSBVL  =EOLXCK
198 04A04 560          GONC    loop10
199 04A07 6ED0          GOTO    endlop    Sortie si EOL trouve
200
201 04A0B 8F00000     loop10 GOSBVL  =FINDA
202 04A12 00           CON(2)  =tCOMMA  <line#>
203 04A14 830          REL(3)  LIST10
204 04A17 00           CON(2)  =tXWORD  INDENT <exp num>
205 04A19 360          REL(3)  LIST20
206 04A1C 00           CON(2)  =tTO     TO <filespec>
207 04A1E 080          REL(3)  LIST30
208 04A21 00           NIBHEX  00
209          * <filespec>
210 04A23 8F00000     GOSBVL  =FSPECx  specificateur valide ?
211 04A2A 4D1          GOC     bserR    non
212 04A2D 8F00000     GOSBVL  =FINDF+  trouve le fichier
213 04A34 431          GOC     bserR    non trouve
214 04A37 137          CD1EX   C(A) := ^ file start
215 04A3A 1F4E8F2     D1=(5)  INFILE
216 04A41 145          DAT1=C  A          INFILE := ^ file start
217 04A44 65BF          GOTO    loop
218
219 04A48 625F          bserR   GOTO    bserr
220          * <line#> [ , <line#> ]
221 04A4C 1FAD8F2     LIST10  D1=(5)  LDEB
222 04A53 7C56          GOSUB   getli#
223 04A57 141          DAT1=A  A          <line# 1>
224 04A5A 174          D1=D1+ (LFIN)-(LDEB)
225 04A5D 141          DAT1=A  A          <line# 2>
226 04A60 7F46          GOSUB   getli#
227 04A64 459          GOC     loop     pas de <line# 2>
228 04A67 141          DAT1=A  A

```

```
229 04A6A 1C4          D1=D1- (LFIN)-(LDEB)
230 04A6D 147          C=DAT1 A          C(A) := <line# 1>
231 04A70 8BE          ?C<=A A          1 <= 2
232 04A73 78           GOYES loop        oui : ok
233 04A75 8D00000     invarg GOVLNG =IVAERR non (ah bon ?)
234
235                   * INDENT <exp num>
236 04A7C 165          LIST20 D0=D0+ 6      sauter tXWORD id tINDENT
237 04A7F 8F00000     GOSBVL =EXPEXC
238 04A86 8F00000     GOSBVL =RNDAHX
239 04A8D 57E          GONC invarg       negatif
240 04A90 1F5C8F2     D1=(5) INDVAL
241 04A97 141          DAT1=A A
242 04A9A 6F5F         GOTO loop
243
244
245                   * TO <filespec>
246 04A9E 161          LIST30 D0=D0+ 2
247 04AA1 8F00000     GOSBVL =FSPECx      specificateur valide ?
248 04AA8 472          GOC Bserr non
249                   * Cy = 0 : mainframe recognisable file specifier found
250                   * A(W) = file name
251                   * D(S) = #F if device not specified
252                   * D(B) = details sur le port
253 04AAB 1F00000     D1=(5) =STMTR0      sauvegardes
254 04AB2 AFB          C=D W              Sauvegarde de D(W) (pour CRETF+)
255 04AB5 1557         DAT1=C W
256 04AB9 17F          D1=D1+ 16
257 04ABC 1517         DAT1=A W            Sauvegarde du nom du fichier
258
259 04AC0 8F00000     GOSBVL =FINDF+      Find file
260                   * A(W) = B(W) = file name
261                   * Cy = 0 : file found
262                   * Cy = 1 : file not found
263 04AC7 4C0          GOC LIST32          fichier non trouve : ok
264 04ACA 330000      LC(4) =eFEXST       "File Exists" / Beeeep !
265 04AD0 6ACE        Bserr GOTO bserr
266
267 04AD4 1FFC8F2     LIST32 D1=(5) OUTYPE
268 04ADB 301          LC(1) 1
269 04ADE 1550         DAT1=C P            Sortie := fichier
270 04AE2 671F         GOTO loop
271
272                   * Verification du type et des protections du
273                   * fichier a lister.
274 04AE6 1B4E8F2     endlop D0=(5) INFILE
275 04AED 146          C=DAT0 A
276 04AF0 135          D1=C
277 04AF3 17E          D1=D1+ (oFTYPh)-1
278 04AF6 143          A=DAT1 A
279 04AF9 F4           ASR A              A(A) := file type
280 04AFB 1CE          D1=D1- (oFTYPh)-1
281 04AFE 8F00000     GOSBVL =BASCHA      BASIC ?
282 04B05 4D4          GOC bsErr Non : "Invalid File Type"
283 04B08 8F00000     GOSBVL =GETPR1      PRIVATE ?
284 04B0F 434          GOC bsErr Oui : "File Protect"
285                   * D1 = ^ file type
```

```

286
287      * Preparer les pointeurs
288 04B12 17F      D1=D1+ (oFLENh)-(oFTYPh)
289 04B15 137      CD1EX
290 04B18 135      D1=C
291 04B1B 143      A=DAT1 A      A(A) := REL(5) FiLeNd
292 04B1E C2      C=C+A A
293 04B20 199E    D0=(2) INEND
294 04B24 144      DAT0=C A
295 04B27 D7      D=C A      D(A) := ^ fin fichier
296 04B29 19AD    D0=(2) LDEB
297 04B2D 142      A=DAT0 A      A(A) := <line1 #>
298 04B30 D8      B=A A
299 04B32 17E      D1=D1+ (oBSsod)-(lEOL)
300 04B35 137      CD1EX
301 04B38 135      D1=C
302 04B3B 171      D1=D1+ =lEOL      D0 = ^ line # (1ere ligne)
303 04B3E D0      A=0 A      La doc ne l'indique pas
304 04B40 8F00000  GOSBVL =FINDL0 Supporte !
305 04B47 4F0      GOC lst10
306 04B4A 860      ?ST=0 0      Il existe un numero de ligne
307 04B4D A0      GOYES lst10 plus grand que <line1 #>
308 04B4F 63C0     GOTO nxtstm
309
310 04B53 674E     bsErr GOTO bserr
311
312 04B57 1B4E8F2  lst10 D0=(5) INFILE
313 04B5E 137      CD1EX
314 04B61 144      DAT0=C A      INFILE := ^ line# courant
315      * Creation du fichier a partir des elements stockes
316      * dans STMTR0/1 pendant "TO <filespec>" si
317      * necessaire.
318 04B64 19FC      D0=(2) OUTYPE
319 04B68 1524     A=DAT0 S
320 04B6C 948      ?A=0 S
321 04B6F 96      GOYES nocret
322 04B71 1F00000  D1=(5) =STMTR0
323 04B78 1577     C=DAT1 W
324 04B7C AF7      D=C W      Restaurer pour CRETF+
325 04B7F D2      C=0 A
326 04B81 3192     LC(2) 37+4 header + EOF-mark
327 04B85 8F00000  GOSBVL =CRETF+ Le Lex ne doit pas bouger
328 04B8C 46C      GOC bsErr Probleme a la creation
329 04B8F 119      C=R1 C(A) := adresse du header
330 04B92 1F5D8F2  D1=(5) OUTHDR Header du fichier de sortie
331 04B99 145      DAT1=C A
332 04B9C 1F00000  D1=(5) =STMTR1
333 04BA3 1537     A=DAT1 W
334 04BA7 135      D1=C
335 04BAA 1517     DAT1=A W      Nom du fichier
336 04BAE 17F      D1=D1+ 16      D1 = ^ file type
337 04BB1 35100004 LCHEx 400001 text + copy code
338 04BB9 15D5     DAT1=C 6
339 04BBD 17F      D1=D1+ 16
340 04BC0 174      D1=D1+ 5
341      * EOF-Mark
342 04BC3 D2      C=0 A

```

```
343 04BC5 CE          C=C-1  A
344 04BC7 15D3        DAT1=C 4      D1 := ^ fin du fichier
345
346 04BCB 137         CD1EX
347 04BCE 1F0D8F2     D1=(5) OUTADR
348 04BD5 145         DAT1=C  A
349 04BD8             nocret
350
351 *****
352 * lst100
353 *
354 * But : debut de la boucle principale de D/PBLIST
355 * c'est a dire tests de sortie
356 * Entree : -
357 * Sortie :
358 * si sortie necessaire
359 * alors sortie par NXTSTM
360 * sinon D1 = ^ <line #>
361 * fin si
362 * Detail :
363 * test 1 : EOF ?
364 * test 2 : no ligne > no derniere ligne a lister
365 * test 3 : [ATTN] ?
366 * Appelle : CK"ON", NXTSTM
367 * Historique :
368 * 88/01/10 : PD & JT conception & codage
369 *****
370
371 04BD8             lst100
372 * 1 : atteint EOF ?
373 04BD8 1B4E8F2     D0=(5) INFILE
374 04BDF 142         A=DAT0 A      A(A) := ^ line#
375 04BE2 199E       D0=(2) INEND
376 04BE6 146         C=DAT0 A      C(A) := ^ FiLeNd
377 04BE9 8BE        ?A>=C  A
378 04BEC 72         GOYES  nxtstm
379 * 2 : atteint ligne > ligne fin ?
380 04BEE 131        D1=A          D1 := ^ line#
381 04BF1 D0         A=0      A
382 04BF3 15B3      A=DAT1 4      A(A) := line#
383 04BF7 19FD      D0=(2) LFIN
384 04BFB 146       C=DAT0 A
385 04BFE 8B6       ?A>C  A
386 04C01 21        GOYES  nxtstm
387 * 3 : [ATTN] ?
388 04C03 137       CD1EX
389 04C06 8F00000   GOSBVL =CK"ON"
390 04C0D 135       D1=C
391 04C10 490       GOC      lst110  Ok on a passe tous les tests
392 04C13 8D00000   nxtstm GOVLNG =NXTSTM
393
394 *****
395 * lst110
396 *
397 * But : analyse du premier statement et
398 * decompilation de la ligne.
399 * Entree :
```

```

400      *   - D1 = ^ <line #>
401      * Sortie :
402      *   - (OUTBS..AVMEMS) = ligne decompile
403      *     et indentation faite
404      * Appelle : gettok, LDCM10
405      *****
406
407      * Ok, on peut y aller
408      * D1 = ^ line#
409 04C1A 171      lst110 D1=D1+ 2      D1 := ^ Stlen - 2
410 04C1D 7CA4      GOSUB  gettok
411
412 04C21 AE8      B=A      B      B(B) := classe reconnue
413      * Analyse du premier statement pour voir s'il faut :
414      *   - indenter avant
415      *   - sauter une ligne avant
416      *   - marquer un SUB
417
418      * Indentation avant ?
419 04C24 119      C=R1      C(A) := indentation avant
420 04C27 8AA      ?C=0      A
421 04C2A 81      GOYES  lst120
422      * il faut indenter
423 04C2C 1BFE8F2      D0=(5) INDCOU      Indentation courante
424 04C33 14A      A=DAT0 B
425 04C36 B6A      A=A-C B
426 04C39 550      GONC  lst118      Ok, indentation valide
427 04C3C AE0      A=0      B      Indentation := 0
428 04C3F 148      lst118 DAT0=A B      Nouvelle indentation
429      * Cas autre que DATA ?
430      *
431      * Si token = DATA
432      *   alors
433      *     on est a l'interieur, on ne fait rien
434      *   sinon
435      *     si on etait avant a l'interieur de DATA
436      *     alors
437      *       on n'est plus dans des DATA (INDARM := 0)
438      *       sauter une ligne avant (SAUTAV := 1)
439      *     sinon
440      *       cas le plus frequent, on ne fait rien
441      *   fin si
442      * fin si
443 04C42 D4      lst120 A=B      A      A(B) := classe du token
444 04C44 3150      LC(2)  clDATA
445 04C48 962      ?A=C      B
446 04C4B 52      GOYES  lst122      cas "DATA"
447
448 04C4D 1BEE8F2      D0=(5) INDARM
449 04C54 15A0      A=DAT0 1      A(0) = 0, inDATA ou inREM
450
451 04C58 301      LC(1)  inDATA      On avait DATA avant ?
452 04C5B 906      ?A#C      P
453 04C5E 21      GOYES  lst122      Non : on ne fait rien (ah..)
454
455 04C60 D2      C=0      A      A(0) := 0 (no inDATA)
456 04C62 15C0      DAT0=C 1      INDARM := 0

```

```

457
458 04C66 193F          D0=(2) SAUTAV
459 04C6A E6           C=C+1  A      C(0) := 1
460 04C6C 15C0        DAT0=C  1
461
462                    * Si (INDARM = inREM) && (classe # clREM)
463                    *   alors
464                    *     INDARM := 0
465                    * fin si
466 04C70 19EE        lst122 D0=(2) INDARM
467 04C74 1520        A=DAT0 P
468 04C78 302         LC(1)  inREM
469 04C7B 906         ?A#C   P
470 04C7E 11          GOYES  lst125  INDARM # inREM
471 04C80 3170        LC(2)  clREM
472 04C84 961         ?C=B   B
473 04C87 80          GOYES  lst125  classe = clREM
474 04C89 D2          C=0    A
475 04C8B 1540        DAT0=C  P      INDARM := 0
476
477                    *****
478                    * Y-a-t'il des cas particuliers a traiter ?
479                    *****
480
481 04C8F D4           lst125 A=B    A      A(B) := classe du token
482 04C91 8F00000     GOSBVL =FINDA
483 04C98 20          CON(2) clSUB  SUB
484 04C9A D10        REL(3) l1SUB
485 04C9D 30          CON(2) clDEF1 DEF mono-ligne
486 04C9F D60        REL(3) l1DEF1
487 04CA2 40          CON(2) clDEF2 DEF multi-ligne
488 04CA4 0A0        REL(3) l1DEF2
489 04CA7 50          CON(2) clDATA DATA
490 04CA9 080        REL(3) l1DATA
491 04CAC 60          CON(2) clLBL  <label>
492 04CAE 690        REL(3) l1LBL
493 04CB1 00          NIBHEX 00
494 04CB3 62A0        GOTO   lst130  Pas de traitement special
495
496 04CB7 1FFE8F2     l1SUB  D1=(5) INDCOU  Indentation := 0
497 04CBE AE2         C=0    B
498 04CC1 14D         DAT1=C  B
499                    * sauter une ligne
500 04CC4 7903        GOSUB  sautln
501                    * une ligne de '-'
502 04CC8 1F00000     D1=(5) =OUTBS  Debut de la ligne
503 04CCF 147         C=DAT1 A
504 04CD2 135        D1=C      D1 := ^ debut ligne
505
506 04CD5 3FD2D2D2    LCASC  '-----'
      D2D2D2D2
      D2
507 04CE7 AFA         A=C    W      A(W) := pattern
508 04CEA D2          C=0    A
509 04CEC 3108        LC(2)  8*8*2   64 '-'
510 04CF0 8F00000     GOSBVL =STUFF  Remplissage de 64 '-'
511 04CF7 137         CD1EX  C(A) := adresse de fin

```

```
512 04CFA 1F00000          D1=(5) =AVMEMS
513 04D01 145             DAT1=C A      AVMEMS := fin
514 04D04 70D2           GOSUB print
515 04D08 6B30           GOTO 11skip   et terminer en sautant
516
517 04D0C 14B           11DEF1 A=DAT1 B      A(B) := tEOL ?
518 04D0F 3100           LC(2) =tEOL
519 04D13 966            ?A#C B
520 04D16 04             GOYES 1st130   DEF + autre chose
521 04D18 1F4F8F2       D1=(5) SAUTAP  Sauter apres
522 04D1F 301            LC(1) 1
523 04D22 15D0           DAT1=C 1      Sauter une ligne apres
524 04D26 5D1            GONC 11skip   B.E.T.
525
526 04D29 1FEE8F2       11DATA D1=(5) INDARM  DATA ou REM ?
527 04D30 15B0           A=DAT1 1
528 04D34 301            LC(1) inDATA
529 04D37 15D0           DAT1=C 1      INDARM := inDATA
530 04D3B 906            ?A#C P
531 04D3E 60             GOYES 11skip   Premier DATA
532 04D40 6510           GOTO 1st130   pas de saut
533
534 04D44                11DEF2
535 04D44                11LBL
536 04D44 1F3F8F2       11skip D1=(5) SAUTAV
537 04D4B 301            LC(1) 1      1 est different de 0
538 04D4E 1550           DAT1=C P      Sauter une ligne
539 04D52 6300           GOTO 1st130
540
541                      * Sauter une ligne ?
542 04D56 1B3F8F2       1st130 D0=(5) SAUTAV  Sauter avant ?
543 04D5D 1524           A=DAT0 S
544 04D61 948            ?A=0 S      Sauter la ligne ?
545 04D64 60             GOYES 1st140   Non
546 04D66 7762           GOSUB sautln  Sauter une ligne
547
548                      * Figer l'indentation courante
549 04D6A 1BFE8F2       1st140 D0=(5) INDCOU  Indentation courante
550 04D71 AF0            A=0 W
551 04D74 14A            A=DAT0 B
552
553 04D77 195C           D0=(2) INDVAL  Valeur de l'indentation
554 04D7B AF2            C=0 W
555 04D7E 146            C=DAT0 A
556
557 04D81 8F00000       GOSBVL =MPY    A,B,C := Ind. en octets
558
559 04D88 C6             C=C+C A      Indentation en quartets
560 04D8A 195F           D0=(2) TMP
561 04D8E 144            DAT0=C A      TMP := déplacement en q.
562                      * On voudrait C(A) quartets en plus. On peut ?
563 04D91 8F00000       GOSBVL =MEMCKL Memory Check with Leeway
564 04D98 560            GONC 1st145   Oui : on continue
565 04D9B 6FFB           GOTO bserr    Non : Insufficient Memory
566                      * B(A) := déplacement en quartets
567
568 04D9F 1A0000         1st145 D0=(4) =OUTBS
```



```

569 04DA5 146          C=DAT0 A          C(A) := OUTBS original
570 04DA8 06          RSTK=C          RSTK := OUTBS original
571
572 04DAA 135          D1=C          D1 := start of area
573 04DAD 3F020202    LCASC ' '
      02020202
      02
574 04DBF AFA          A=C W          A(W) := pattern a stuffer
575 04DC2 D9          C=B A          C(A) := longueur en quartets
576 04DC4 8F00000    GOSBVL =STUFF
577 04DCB 137          CD1EX          C(A) := pseudo OUTBS
578          * Laisser de la place pour le no de ligne
579 04DCE 134          D0=C          D0 := ^ pseudo OUTBS
580 04DD1 35020202    LCASC ' '
581 04DD9 15C5          DAT0=C 6
582
583          * Remettre D1 sur la ligne (<line #>)
584 04DDD 1F4E8F2      D1=(5) INFILE
585 04DE4 147          C=DAT1 A
586 04DE7 135          D1=C          D1 := ^ <line #>
587 04DEA D0          A=0 A
588 04DEC 15B3          A=DAT1 4      A(A) := no de ligne
589 04DF0 3400010    LCHEX 01000
590 04DF7 8BE          ?A>=C A
591 04DFA 91          GOYES lst160
592 04DFC F6          CSR A          C(A) := 100
593 04DFE 8BE          ?A>=C A
594 04E01 F0          GOYES lst151  1 blanc
595 04E03 F6          CSR A          C(A) := 10
596 04E05 8BE          ?A>=C A
597 04E08 50          GOYES lst152  2 blancs
598 04E0A 161          D0=D0+ 2
599 04E0D 161          lst152 D0=D0+ 2
600 04E10 161          lst151 D0=D0+ 2
601 04E13 136          lst160 CD0EX
602 04E16 1B00000    D0=(5) =OUTBS
603 04E1D 144          DAT0=C A
604          * D1 = ^ line#
605          * OUTBS decale pour LDCM10
606 04E20 8F00000    GOSBVL =LDCM10 Comme pour LIST
607          * R0 = ^ past tEOL
608          * B(A) = longueur en octets
609          * OUTBS collapsed
610 04E27 07          C=RSTK          C(A) := OUTBS original
611          * Decaler le numero de ligne et le remettre
612          * avant l'indentation
613 04E29 D7          D=C A          D(A) := OUTBS original
614 04E2B 1F5F8F2    D1=(5) TMP      juste avant no ligne
615 04E32 143          A=DAT1 A
616 04E35 CA          A=A+C A          A(A) := ^ no ligne
617 04E37 130          D0=A          D0 := ^ no ligne
618 04E3A 135          D1=C          D1 := ^ OUTBS original
619 04E3D 15A7          A=DAT0 8      Lire no de ligne
620 04E41 37020202    LCASC ' '
      02
621 04E4B 15C7          DAT0=C 8
622 04E4F 1597          DAT1=A 8      Nouveau no ligne

```

```

623 04E53 DB          C=D    A      C(A) := OUTBS original
624                * Placer AVMEMS a la bonne valeur
625 04E55 1F00000    D1=(5) =OUTBS  OUTBS modifie
626 04E5C 143        A=DAT1 A
627 04E5F 145        DAT1=C A      OUTBS original
628 04E62 C0         A=A+B  A
629 04E64 C0         A=A+B  A
630 04E66 174        D1=D1+ 5      =AVMEMS
631 04E69 141        DAT1=A A      nouveau AVMEMS
632                * Traitement du premier token a nouveau
633                * - !
634                * - indentation "apres"
635 04E6C 1B4E8F2    D0=(5) INFILE
636 04E73 142        A=DAT0 A
637 04E76 131        D1=A          D1 := ^ <line #>
638 04E79 171        D1=D1+ 2
639 04E7C 7D42       GOSUB gettok  A nouveau le 1er token
640 04E80 AE8        B=A    B      B(A) := classe du token
641                * Cas particulier du IF ?
642 04E83 3180       LC(2)  clIF
643 04E87 966        ?A#C  B
644 04E8A 60         GOYES  lst165  Non : on continue
645 04E8C 6BF0       GOTO   lst700  Ignorer le reste de la ligne
646
647 04E90 11A       lst165 C=R2          C(A) := indentation "apres"
648 04E93 8AA       ?C=0   A
649 04E96 F0        GOYES  lst170  Pas d'indentation
650
651 04E98 19FE       D0=(2) INDCOU
652 04E9C 14A       A=DAT0 B      A(B) := indentation courante
653 04E9F A6A       A=A+C  B
654 04EA2 148       DAT0=A B
655                * "!" ?
656 04EA5 3170       lst170 LC(2)  clREM
657 04EA9 965       ?B#C  B
658 04EAC 77        GOYES  lst190  Rentrer dans la boucle
659                * Sauver D1 pour etre remis apres
660 04EAE 137       CD1EX
661 04EB1 06        RSTK=C      RSTK := D1
662                * D1 := ^ no ligne dans la chaine decompilee
663 04EB3 1F00000    D1=(5) =OUTBS
664 04EBA 147        C=DAT1 A
665 04EBD 135        D1=C          D1 := ^ no ligne
666                * Mettre d'office 4 blancs
667 04EC0 37020202  LCASC  '  '
668 04ECA 15D7       DAT1=C 4*2
669                * Mettre un '-' ?
670 04ECE 175        D1=D1+ 3*2    D1 := ^ ('-' eventuel)
671 04ED1 19EE       D0=(2) INDARM
672 04ED5 1520       A=DAT0 P      A(0) := 0, inDATA ou inREM
673 04ED9 302        LC(1)  inREM
674 04EDC 1540       DAT0=C P      INDARM := inREM
675 04EE0 902        ?A=C  P      Deja une REM ?
676 04EE3 90         GOYES  lst180  Oui : ne rien faire
677 04EE5 31D2       LCASC  '-'     Non : ajouter un '-'
678 04EE9 14D        DAT1=C B

```

```

679 04EEC 3112      lst180 LCASC  '!'
680 04EF0 14B      lst182 A=DAT1 B
681 04EF3 962          ?A=C  B
682 04EF6 80        GOYES  lst185
683 04EF8 171        D1=D1+ 2
684 04EFB 54F        GONC   lst182  B.E.T.
685 04EFE 173      lst185 D1=D1+ 2+2  D1 := ^ debut du commentaire
686 04F01 133        AD1EX          A(A) := start of source
687 04F04 131        D1=A
688 04F07 1C3        D1=D1- 2+2    D1 := start of dest
689 04F0A 8F00000    GOSBVL =MOVEUA  AVMEMS := end of source
690
691 04F11 1B00000    D0=(5) =AVMEMS
692 04F18 137        CD1EX          C(A) := end of dest
693 04F1B 144        DAT0=C A
694 04F1E 07         C=RSTK          restaurer D1
695 04F20 135        D1=C
696
697 04F23           lst190
698
699 *****
700 * Debut de la boucle "dans la ligne"
701 *
702 * But : explorer les statements entre le deuxieme et
703 * le dernier de la ligne courante.
704 * Entree :
705 * - D1 = ^ Stlen - 2 du deuxieme Statement
706 * Sortie :
707 * - D1 = ^ tEOL de la ligne courante
708 * Historique :
709 * 88/01/24 : PD & JT conception & codage
710 *****
711
712 * Ce lst530 est une ruse (voir en fin de boucle)
713 04F23           lst530
714 04F23 76A1      lst500 GOSUB  gettok
715 04F27 AE8        B=A  B  B(A) := classe du token
716 04F2A 447        GOC   lst900
717
718 04F2D 3180      LC(2)  clIF  IF standard ?
719 04F31 962        ?A=C  B
720 04F34 45        GOYES  lst700  Ignorer le reste de la ligne
721 * Calculer la nouvelle indentation
722 04F36 1BFE8F2    D0=(5) INDCOU
723 04F3D 14E        C=DAT0 B  C(A) := indentation courante
724
725 04F40 111        A=R1          indentation avant (<= 0)
726 04F43 B62        C=C-A  B
727 04F46 550        GONC   lst510  Pas < 0
728 04F49 AE2        C=0  B
729 04F4C 112      lst510 A=R2          indentation apres (>= 0)
730 04F4F A62        C=C+A  B
731 04F52 14C        DAT0=C B
732 * Cas particuliers ?
733 04F55 3120      LC(2)  clSUB
734 04F59 965        ?B#C  B
735 04F5C 80        GOYES  lst520

```

```
736          * SUB
737 04F5E 148          DAT0=A B          indentation := R2
738 04F61 51C          GONC 1st530      B.E.T.
739 04F64 3110        1st520 LC(2)  clNDDF
740 04F68 965          ?B#C  B
741 04F6B 8B          GOYES 1st530
742          * END DEF
743 04F6D 14B          A=DAT1 B          A(B) = tEOL ?
744 04F70 3100        LC(2) =tEOL
745 04F74 966          ?A#C  B
746 04F77 CA          GOYES 1st530
747 04F79 301        LC(1) 1
748 04F7C 194F        D0=(2) SAUTAP
749 04F80 15C0        DAT0=C 1          Sauter apres := vrai
750
751          * Le 1st530 qui suit a ete deplace en debut de la
752          * boucle pour eviter le GOYES -> GOTO intempestif.
753          * 1st530
754 04F84 6E9F        GOTO 1st500
755
756 04F88 1F4E8F2      1st700 D1=(5) INFILE
757 04F8F 143          A=DAT1 A          A(A) := ^ <line #>
758 04F92 131          D1=A
759 04F95 8F00000      GOSBVL =NXTLIN D1 := ^ past tEOL
760 04F9C 550          GONC 1st910 B.E.T.
761          * Attention : le code continue en sequence !!!
762
763          *****
764          * Fin de la boucle "dans la ligne"
765          *
766          * But : recuperer le pointeur dans le flot tokenise
767          * pour la ligne suivante, et iterer
768          * Entree :
769          * - D1 = ^ tEOL de la ligne courante
770          * Sortie :
771          * - INFILE = ^ <line #> de la ligne suivante
772          * Historique :
773          * 88/01/24 : PD & JT conception & codage
774          *****
775
776 04F9F 171          1st900 D1=D1+ 2          D1 := ^ <line #> ligne suiv.
777 04FA2 137          1st910 CD1EX          C(A) := ^ <line #>
778 04FA5 1F4E8F2      D1=(5) INFILE
779 04FAC 145          DAT1=C A          Nouvelle ligne
780          * Prepare la ligne suivante
781          * Reporter le "saut apres" sur "saut avant" et
782          * effacer "saut apres"
783 04FAF 1F4F8F2      D1=(5) SAUTAP
784 04FB6 1534          A=DAT1 S
785 04FBA AC2          C=0 S
786 04FBD 1554          DAT1=C S
787 04FC1 1D3F        D1=(2) SAUTAV
788 04FC5 1514          DAT1=A S
789 04FC9 7B00        GOSUB print
790 04FCD 6A0C        GOTO 1st100
791
792          *****
```

```
793          * SOUS-PROGRAMMES
794          *****
795
796          *****
797          * sautln
798          *
799          * But : imprimer une ligne vide
800          * Entree : -
801          * Sortie : -
802          * Abime : A-D, R3
803          * Appelle : OBCOLL, print (tombe dedans)
804          * Niveaux :
805          * Historique :
806          *   88/01/24 : PD & JT conception & codage
807          *****
808
809 04FD1 8F00000  sautln GOSBVL =OBCOLL
810          * Attention : le code continue !!!
811
812          *****
813          * print
814          *
815          * But : imprimer la ligne comprise entre OUTBS et
816          *   AVMEMS sur le DISPLAY, le PRINTER ou dans un
817          *   fichier selon la valeur de OUTYPE.
818          * Entree :
819          *   - (OUTBS..AVMEMS) = la ligne
820          * Sortie : -
821          * Abime : A-D, R3
822          * Appelle : CKINFO, SNDWD+, SENDEL, SWPBYT, RPLLIN,
823          *   MOVED2
824          * Niveaux :
825          * Detail : l'execution de sautln continue en direct
826          * Historique :
827          *   88/01/24 : PD & JT isolement dans un sub
828          *****
829
830 04FD8 1BFC8F2  print D0=(5) OUTYPE
831 04FDF 1524      A=DAT0 S          0 : std, 1 : fichier
832 04FE3 94C      ?A#0 S
833 04FE6 33       GOYES prnt50
834          * Impression sur DISPLAY ou PRINTER
835 04FE8 8F00000  GOSBVL =CKINFO Prepare HPIL pour l'envoi
836 04FEF 1A0000  D0=(4) =AVMEMS
837 04FF5 142      A=DAT0 A
838 04FF8 184      D0=D0- 5          OUTBS
839 04FFB 146      C=DAT0 A          C(A) := ^ chaine
840 04FFE EA       A=A-C A          A(A) := longueur en quartets
841 05000 AF1      B=0 W
842 05003 D8       B=A A
843 05005 81D      BSRB           B(A) := longueur en octets
844 05008 850      ST=1 =InhEOL Inhibit EOL pour le cas ou
845 0500B 8F00000  GOSBVL =SNDWD+ Envoi proprement dit
846 05012 8D00000  GOVLNG =SENDEL EOL
847
848          * Impression dans un fichier
849 05019 1A0000  prnt50 D0=(4) =AVMEMS
```

```

850          * Insertion de la longueur LIF
851 0501F 142          A=DAT0 A          A(A) := end of source
852 05022 131          D1=A
853 05025 173          D1=D1+ 4          D1 := end of dest
854 05028 184          D0=D0- 5          D0=(5) OUTBS
855 0502B 146          C=DAT0 A          C(A) := start of source
856 0502E 8F00000     GOSBVL =MOVED2
857          * Calculer le padding LIF
858 05035 1B00000     D0=(5) =AVMEMS
859 0503C AF0          A=0 W
860 0503F 142          A=DAT0 A
861 05042 184          D0=D0- 5          OUTBS
862 05045 146          C=DAT0 A
863 05048 EA          A=A-C A          Longueur en quartets
864 0504A 81C          ASRB          Longueur en octets
865 0504D D8          B=A A          Sauvegarde
866 0504F 8F00000     GOSBVL =SWPBYT
867 05056 146          C=DAT0 A          C(A) := (OUTBS)
868 05059 135          D1=C          D1 := ^ LIF length
869 0505C 1593        DAT1=A 4
870 05060 822          SB=0
871 05063 81D          BSRB
872 05066 D0          A=0 A          Longueur a ajouter
873 05068 832          ?SB=0
874 0506B 40          GOYES prnt60     Longueur paire
875 0506D E4          A=A+1 A          1 octet a ajouter
876 0506F D2          prnt60 C=0 A
877 05071 304          LC(1) 4
878 05074 C4          A=A+A A
879 05076 CA          A=A+C A          Longueur totale a ajouter
880 05078 164          D0=D0+ 5          AVMEMS
881 0507B 146          C=DAT0 A
882 0507E C2          C=C+A A
883 05080 144          DAT0=C A          AVMEMS + 4 ou 6
884          * Insertion dans le fichier
885 05083 D2          C=0 A
886 05085 10B          R3=C          R3 := longueur vieille ligne
887 05088 1B5D8F2     D0=(5) OUTHDR
888 0508F 146          C=DAT0 A          C(A) := ^ file header
889 05092 190D        D0=(2) OUTADR
890 05096 142          A=DAT0 A          A(A) := ^ end of old line
891 05099 8F00000     GOSBVL =RPLLIN Ne peut pas bouger
892 050A0 4E0          GOC bSerr
893          * A(A) = end+1 of replaced line in file
894 050A3 1B0D8F2     D0=(5) OUTADR
895 050AA 140          DAT0=A A
896 050AD 01          RTN
897 050AF 6BE8        bSerr GOTO bserr
898
899          *****
900          * getli#
901          *
902          * But : evaluer un numero de ligne optionnel
903          * Entree :
904          * - D0 = ^ tCOMMA precedant <line #>
905          * Sortie :
906          * Cy = 1 : pas de <line #>

```

```
907      *   Cy = 0 :
908      *     - A(A) = numero de ligne en BCD
909      *     - D0 reactualise
910      * Niveaux : 0
911      * Utilise : C(B), A(A)
912      * Historique :
913      *   86/05/.. : JPB      conception & codage
914      *   88/01/10 : PD & JT modifications & documentation
915      *****
916
917 050B3 14A      getli# A=DAT0 B
918 050B6 3100      LC(2) =tCOMMA
919 050BA 966      ?A#C B
920 050BD 00      RTNYES
921 050BF 161      D0=D0+ 2
922 050C2 D0      A=0 A
923 050C4 15A3      A=DAT0 4
924 050C8 163      D0=D0+ 4
925 050CB 01      RTN      Cy = 0 a cause de D0=D0+ 4
926
927      *****
928      * gettok
929      *
930      * But : classer le token pointe par D1
931      * Entree :
932      *   - D1 = ^ Stlen - 2 du token a classer
933      * Sortie :
934      *   Cy = 1 : EOL trouvee
935      *   Cy = 0 : EOL non trouvee
936      *   - D1 = ^ Statement terminator
937      *   - A(B) = classe du token
938      *   R1 = indentation avant
939      *   R2 = indentation apres
940      * Niveaux : 1
941      * Utilise :
942      * Appelle : FINDA
943      * Historique :
944      *   88/01/23 : PD & JT & JJD conception & codage
945      *****
946
947 050CD AF2      gettok C=0 W
948 050D0 109      R1=C
949 050D3 10A      R2=C
950 050D6 14B      A=DAT1 B
951 050D9 3100      LC(2) =tEOL
952 050DD 962      ?A=C B
953 050E0 00      RTNYES
954 050E2 171      D1=D1+ 2      D1 := ^ Stlen
955 050E5 D2      C=0 A
956 050E7 14F      C=DAT1 B      C(A) := Stlen
957 050EA 133      AD1EX
958 050ED C2      C=C+A A      C(A) := ^ Stlen - 2 suivant
959 050EF 131      D1=A      D1 := ^ Stlen du courant
960 050F2 06      RSTK=C      RSTK := ^ tEOL ou t@ ou ...
961 050F4 171      D1=D1+ 2      D1 = ^ token courant
962 050F7 14B      A=DAT1 B
963
```

```

964 050FA 8F00000      GOSBVL =FINDA
965 05101 00          CON(2) =tXWORD  XWORD
966 05103 040        REL(3) gtXWRD
967 05106 00          CON(2) =tFOR    FOR
968 05108 A80        REL(3) gtFOR
969 0510B 00          CON(2) =tNEXT  NEXT
970 0510D 2C0        REL(3) gtNEXT
971 05110 00          CON(2) =tSUB    SUB
972 05112 6E0        REL(3) gtSUB
973 05115 00          CON(2) =tDEF    DEF FN
974 05117 3F0        REL(3) gtDEF
975 0511A 00          CON(2) =tDATA   DATA
976 0511C 611        REL(3) gtDATA
977 0511F 00          CON(2) =tLBLST  '...':
978 05121 C11        REL(3) gtLBL
979 05124 00          CON(2) =tEND    END
980 05126 9A0        REL(3) gtND
981 05129 00          CON(2) =tENDDF  END DEF
982 0512B BB0        REL(3) gtNDDF
983 0512E 00          CON(2) =tENDSB  END SUB
984 05130 F90        REL(3) gtNDSB
985 05133 00          CON(2) =t!      !
986 05135 311        REL(3) gtREM
987 05138 00          CON(2) =tIF     IF mono-ligne (standard)
988 0513A 911        REL(3) gtIF
989 0513D 00          NIBHEX 00
990
991 0513F 6E11      gtrien GOTO  gtRIEN
992
993 05143 171      gtXWRD D1=D1+ 2
994 05146 14B      A=DAT1 B
995 05149 311E      LC(2) =id
996 0514D 966      ?A#C B
997 05150 FE      GOYES gtrien
998
999 05152 171      D1=D1+ 2      D1 := ^ tooookken
1000 05155 14B     A=DAT1 B      A(B) := token
1001
1002 05158 8F00000  GOSBVL =FINDA
1003 0515F 34      CON(2) =tWHILE  WHILE
1004 05161 130      REL(3) gtWHIL
1005 05164 06      CON(2) =tLOOP   LOOP
1006 05166 C20      REL(3) gtLOOP
1007 05169 44      CON(2) =tREPEAT REPEAT
1008 0516B 720      REL(3) gtREPT
1009 0516E 54      CON(2) =tUNTIL  UNTIL
1010 05170 F50      REL(3) gtUNTL
1011 05173 36      CON(2) =tIF2    IF jpcrom
1012 05175 D10      REL(3) gtIF2
1013 05178 46      CON(2) =tELSE2  ELSE jpcrom
1014 0517A 320      REL(3) gtELS2
1015 0517D 16      CON(2) =tSELECT SELECT
1016 0517F C20      REL(3) gtSEL
1017 05182 26      CON(2) =tCASE   CASE
1018 05184 910      REL(3) gtCASE
1019 05187 24      CON(2) =tEND2   END IF/LOOP/SELECT/WHILE
1020 05189 E20      REL(3) gtENDj

```



```
1021 0518C 00          NIBHEX 00
1022 0518E 6FC0      GOTO  gtRIEN
1023
1024 *****
1025 * indentation avant : 0
1026 * indentation apres : +1
1027 *
1028 * FOR, WHILE, LOOP, REPEAT, IF
1029 *****
1030
1031 05192          gtFOR
1032 05192          gtIF2
1033 05192          gtLOOP
1034 05192          gtREPT
1035 05192 D2      gtWHIL C=0    A
1036 05194 E6      C=C+1  A
1037 05196 10A    R2=C          indentation apres := 1
1038 05199 64C0   GOTO  gtRIEN  Classe : normale
1039
1040 *****
1041 * indentation avant : -1
1042 * indentation apres : +1
1043 *
1044 * ELSE, CASE
1045 *****
1046
1047 0519D          gtCASE
1048 0519D D2      gtELS2 C=0    A
1049 0519F E6      C=C+1  A
1050 051A1 109    R1=C          indentation avant := 1
1051 051A4 10A    R2=C          indentation apres := 1
1052 051A7 66B0   GOTO  gtRIEN  Classe : normale
1053
1054 *****
1055 * indentation avant : 0
1056 * indentation apres : +2
1057 *
1058 * SELECT
1059 *****
1060
1061 051AB D2      gtSEL  C=0    A
1062 051AD 302    LC(1)  2
1063 051B0 10A    R2=C          indentation apres := 2
1064 051B3 6AA0   GOTO  gtRIEN  Classe : normale
1065
1066 051B7 171    gtENDj D1=D1+ 2      D1 = ^ q de reconnaissance
1067 051BA 14B    A=DAT1 B      A(B) = token suivant
1068 051BD 8F00000 GOSBVL =EOLXCK
1069 051C4 4A0    GOC  gtND  END WHILE
1070 * A(0) = quartet de reconnaissance
1071 051C7 301    LC(1) =qENDS
1072 051CA 902    ?A=C  P
1073 051CD D0     GOYES gtNDSL  END SELECT
1074 * END LOOP / IF
1075 * Attention : le code continue !!!
1076
1077 *****
```

```
1078      * indentation avant : -1
1079      * indentation apres : 0
1080      *
1081      * UNTIL, NEXT, END LOOP, END WHILE,
1082      * END IF, END, END SUB
1083      *****
1084
1085 051CF      gtND
1086 051CF      gtNDSB
1087 051CF      gtNEXT
1088 051CF D2      gtUNTL C=0      A
1089 051D1 E6      C=C+1      A
1090 051D3 109      R1=C      indentation avant := 1
1091 051D6 6780      GOTO      gtRIEN      Classe : normale
1092
1093      *****
1094      * indentation avant : -2
1095      * indentation apres : 0
1096      *
1097      * END SELECT
1098      *****
1099
1100 051DA D2      gtNDSL C=0      A
1101 051DC 302      LC(1)      2
1102 051DF 109      R1=C      indentation avant := 2
1103 051E2 6B70      GOTO      gtRIEN      Classe : normale
1104
1105      *****
1106      * indentation avant : -1
1107      * indentation apres : 0
1108      *
1109      * Cas particulier
1110
1111      * END DEF
1112      *****
1113
1114 051E6 D2      gtNDDF C=0      A
1115 051E8 E6      C=C+1      A
1116 051EA 109      R1=C      indentation avant := 1
1117 051ED 3110      LC(2)      clNDDF      Classe END DEF
1118 051F1 AEA      A=C      B
1119 051F4 6C60      GOTO      gtok99
1120
1121      *****
1122      * indentation avant : 0 (en fait : cas particulier)
1123      * indentation apres : +1
1124      *
1125      * Cas particulier
1126      *
1127      * SUB
1128      *****
1129
1130 051F8 D2      gtSUB C=0      A
1131 051FA E6      C=C+1      A
1132 051FC 10A      R2=C      indentation apres := 1
1133 051FF 3120      LC(2)      clSUB      Classe SUB
1134 05203 AEA      A=C      B
```

```
1135 05206 6A50          GOTO   gtok99
1136
1137 0520A 176          gtDEF  D1=D1+ 7
1138 0520D 14B          A=DAT1 B      A(B) := 0 si multi-ligne
1139 05210 968          ?A=0  B
1140 05213 D0            GOYES  gtDEF2  Multiligne
1141          * Attention : le code continue
1142          * DEF FN mono-ligne
1143
1144          *****
1145          * indentation avant : 0
1146          * indentation apres : 0
1147          *
1148          * Cas particulier
1149          *
1150          * DEF mono-ligne
1151          *****
1152
1153 05215 3130          gtDEF1 LC(2)  clDEF1  DEF mono-ligne
1154 05219 AEA          A=C    B
1155 0521C 6440          GOTO   gtok99
1156
1157          *****
1158          * indentation avant : 0
1159          * indentation apres : 1
1160          *
1161          * Cas particulier
1162          *
1163          * DEF multi-ligne
1164          *****
1165
1166 05220 D2            gtDEF2 C=0    A
1167 05222 E6            C=C+1  A
1168 05224 10A          R2=C      Indentation apres := 1
1169 05227 3140          LC(2)  clDEF2  DEF multi-ligne
1170 0522B AEA          A=C    B
1171 0522E 6230          GOTO   gtok99
1172
1173          *****
1174          * indentation avant : 0
1175          * indentation apres : 0
1176          *
1177          * Cas particulier
1178          *
1179          * DATA
1180          *****
1181
1182 05232 3150          gtDATA LC(2)  clDATA  DATA
1183 05236 AEA          A=C    B
1184 05239 6720          GOTO   gtok99
1185
1186          *****
1187          * indent          avant : 0
1188          * indent          apres : 0
1189          *
1190          * Cas particulier
1191          *
```

```
1192          * '...':
1193          *****
1194
1195 0523D 3160      gtLBL LC(2)  clLBL   label
1196 05241 AEA              A=C    B
1197 05244 6C10          GOTO   gtok99
1198
1199          *****
1200          * indentation avant : 0
1201          * indentation apres : 0
1202          *
1203          * Cas particulier
1204          *
1205          * !
1206          *****
1207
1208 05248 3170      gtREM LC(2)  clREM   !
1209 0524C AEA              A=C    B
1210 0524F 6110          GOTO   gtok99
1211
1212
1213          *****
1214          * indentation avant : 0
1215          * indentation apres : 0
1216          *
1217          * Cas particulier
1218          *
1219          * IF mono-ligne (standard)
1220          *****
1221
1222 05253 3180      gtIF   LC(2)  clIF    IF standard
1223 05257 AEA              A=C    B
1224 0525A 6600          GOTO   gtok99
1225
1226 0525E AE0      gtRIEN A=0    B
1227 05261 07      gtok99 C=RSTK
1228 05263 135          D1=C
1229 05266 03          RTNCC
1230
1231
1232          * *****
1233          * RENUMREM entry
1234          * *****
1235
1236          *****
1237          * RENUMREM
1238          *
1239          * But : renumeroter un programme Basic en tenant
1240          * compte des remarques.
1241          * Note : RENUMREM 100 , 10 , 200 , 500
1242          *          -v-   -v-   -v-   -v-
1243          * registres ->   R0   R1   @R2   R3
1244          * Historique :
1245          *   86/05/.. : JPB      conception & codage
1246          * *****
1247
1248 05268 00000      REL(5) =RENUMd
```

```
1249 0526D 00000      REL(5) =RENUMp
1250 05272 136        =RENUMe CD0EX
1251 05275 108        R0=C
1252 05278 7C02      GOSUB  CHKPSF
1253 0527C 7E12      GOSUB  GETSTe
1254 05280 8F00000   GOSBVL =PRSC00
1255
1256 05287 851        REN005 ST=1  1
1257 0528A 852        ST=1  2
1258 0528D 8F00000   GOSBVL =RENSUB
1259 05294 4B1        GOC    REN010
1260 05297 1F00000   D1=(5) =PCADDR
1261 0529E 11A        C=R2
1262 052A1 145        DAT1=C  A
1263 052A4 7702      GOSUB  UPDCRL
1264 052A8 3100      LC(2)  =eSTMNF "Statement Not Found"
1265 052AC 6BF1      GOTO   mferr
1266
1267 052B0 110        REN010 A=R0
1268 052B3 130        D0=A
1269 052B6 7FA1      GOSUB  LINE#1
1270 052BA 102        R2=A
1271 052BD D2         C=0    A
1272 052BF E6         C=C+1  A
1273 052C1 F2         CSL   A
1274 052C3 108        R0=C
1275 052C6 109        R1=C
1276 052C9 F2         CSL   A
1277 052CB F2         CSL   A
1278 052CD F2         CSL   A
1279 052CF 10B        R3=C
1280 052D2 7971      GOSUB  GETLN#
1281 052D6 436        GOC    REN100
1282 052D9 100        R0=A
1283 052DC 7F61      GOSUB  GETLN#
1284 052E0 495        GOC    REN100
1285 052E3 101        R1=A
1286 052E6 7561      GOSUB  GETLN#
1287 052EA 4F4        GOC    REN100
1288 052ED D6         C=A    A
1289 052EF 132        AD0EX
1290 052F2 102        R2=A
1291 052F5 8F00000   GOSBVL =FINDL
1292 052FC 860        ?ST=0  0
1293 052FF 60         GOYES  REN020
1294 05301 6119      GOTO   nxtstm
1295
1296 05305 137        REN020 CD1EX
1297 05308 12A        CR2EX
1298 0530B 134        D0=C
1299 0530E 7D31      GOSUB  GETLN#
1300 05312 472        GOC    REN100
1301 05315 05        SETDEC
1302 05317 23        P=    3
1303 05319 B14       A=A+1  WP
1304 0531C 20        P=    0
1305 0531E 04        SETHEX
```

```
1306 05320 491          GOC    REN100
1307 05323 D6           C=A    A
1308 05325 8F00000     GOSBVL =FINDL
1309 0532C D0           A=0    A
1310 0532E 15B3        A=DAT1 4
1311 05332 870         ?ST=1 0
1312 05335 50          GOYES  REN100
1313 05337 103         R3=A
1314
1315 0533A 7B21        REN100 GOSUB  LINE#1
1316 0533E 137         CD1EX
1317 05341 112         A=R2
1318 05344 8A2         ?A=C   A
1319 05347 32          GOYES  REN105
1320 05349 130         D0=A
1321 0534C 183         D0=D0- 4
1322 0534F 8F00000     GOSBVL =CPL#10
1323 05356 D2           C=0    A
1324 05358 15F3        C=DAT1 4
1325 0535C 110         A=R0
1326 0535F 8B6         ?A>C   A
1327 05362 80          GOYES  REN105
1328 05364 8CF07F      GOLONG invarg
1329
1330 0536A 1F00000     REN105 D1=(5) =CURREN
1331 05371 147         C=DAT1 A
1332 05374 D7           D=C    A
1333 05376 11A         C=R2
1334 05379 135         D1=C
1335 0537C D2           C=0    A
1336 0537E 15F3        C=DAT1 4
1337 05382 134         D0=C
1338 05385 118         C=R0
1339 05388 06          RSTK=C
1340
1341 0538A 11A         REN110 C=R2
1342 0538D 135         D1=C
1343 05390 111         A=R1
1344 05393 D8           B=A    A
1345
1346 05395 137         REN120 CD1EX
1347 05398 8BF         ?C>=D A
1348 0539B D2          GOYES  ren170
1349 0539D 135         D1=C
1350 053A0 D0           A=0    A
1351 053A2 15B3        A=DAT1 4
1352 053A6 11B         C=R3
1353 053A9 8BE         ?A>=C A
1354 053AC C1          GOYES  ren170
1355 053AE 110         A=R0
1356 053B1 8B2         ?A<C   A
1357 053B4 81          GOYES  REN140
1358
1359 053B6 04          REN130 SETHEX
1360 053B8 136         CD0EX
1361 053BB 108         R0=C
1362 053BE D2           C=0    A
```

1363 053C0 E6
 1364 053C2 109
 1365 053C5 54C
 1366 053C8 6170
 1367

C=C+1 A
 R1=C
 GONC REN110
 ren170 GOTO REN170

1368

1369

* ORGANIGRAMME DE REN140 a REN155:

1370

* -----

1371

* Entree: RSTK = Anc [Next REM Line#]

1372

* R0 = Nv [Next BASIC Line#]

1373

* B[B] = Inc [Increment]

1374

* -----

1375

* | ECRIRE Nv |

1376

* -----REN140

1377

* |

1378

* -----

1379

* <LIGNE REM ?>-----non-----

1380

* -----

1381

* |oui

1382

* -----

1383

* | ECRIRE Anc |

1384

* -----REN145

1385

* |

1386

* -----

1387

* |Anc=Anc+1| |Anc=Nv+1|

1388

* -----

1389

* |

1390

* -----

1391

* < Anc>NV ?>--non--

1392

* -----

1393

* |<-----

1394

* -----

1395

* |Nv=Nv+Inc|REN

1396

* -----150

1397

* |<-----

1398

* -----

1399

* |

1400

* #####

1401

* #REN 155#

1402

* ##### FIN

1403

1404

* ECRIRE Nv

1405

* REN140 DAT1=A 4 LINE# = Nv

1407 053CC 1593

D1=D1+ 6

1408 053D0 175

A=DAT1 B A[B] = token

1409 053D3 14B

D1=D1- 6

1410 053D6 1C5

LC(2) =t!

1411 053D9 3100

?A=C B Ligne REM ?

1412 053DD 962

GOYES REN145 oui

1413 053E0 C1

LC(2) =tREM

1414 053E2 3100

?A=C B Ligne REM ?

1415 053E6 962

GOYES REN145 oui

1416 053E9 31

A=R0 non: Anc = Nv+1

1417 053EB 110

C=RSTK

1418 053EE 07

C=A A

1419 053F0 D6

```
1420 053F2 05          SETDEC
1421 053F4 E6          C=C+1  A
1422 053F6 06          RSTK=C
1423 053F8 6910        GOTO   REN150
1424
1425 053FC 07          * ECRIRE Anc
      REN145 C=RSTK          C=Anc
1426 053FE 15D3        DAT1=C 4          Line# = Anc
1427 05402 05          SETDEC
1428 05404 E6          C=C+1  A          Anc = Anc+1
1429 05406 06          RSTK=C          Sauvegarde Anc
1430
1431 05408 110        * On est toujours en DEC
      A=R0          A=Nv
1432 0540B 23          P=      3
1433 0540D 99E        ?C<=A  WP          Anc <= Nv ?
1434 05410 70          GOYES  REN155      oui alors FIN
1435
1436 05412 C0          * Entree: A=Nv ; B=Inc ; DEC
      REN150 A=A+B  A          Nv = Nv+Inc
1437 05414 100        R0=A          Sauve Nv
1438
1439 05417 04          * FIN: on remet tout en ordre pour REN160
      REN155 SETHEX
1440 05419 20          P=      0
1441 0541B 173        D1=D1+ 4          D1 @ tLEN
1442 0541E D2          C=0    A
1443
1444 05420 14F          REN160 C=DAT1 B
1445 05423 133        AD1EX
1446 05426 CA          A=A+C  A
1447 05428 131        D1=A
1448 0542B 14B        A=DAT1 B
1449 0542E 171        D1=D1+ 2
1450 05431 90C        ?A#0  P
1451 05434 CE          GOYES  REN160
1452 05436 6E5F        GOTO   REN120
1453
1454 0543A 07          REN170 C=RSTK
1455 0543C 851        ST=1   1
1456 0543F 842        ST=0   2
1457 05442 8F00000    GOSBVL =RENSUB
1458 05449 8C8C7F    GOLONG nxtstm
1459
1460 0544F 14A        GETLN# A=DAT0 B
1461 05452 161        D0=D0+ 2
1462 05455 3100      LC(2)  =tCOMMA
1463 05459 966        ?A#C  B
1464 0545C 00        RTNYES
1465 0545E D0        A=0    A
1466 05460 15A3      A=DAT0 4
1467 05464 163      D0=D0+ 4
1468 05467 03        RTNCC
1469
1470 05469 1F00000    LINE#1 D1=(5) =CURREN
1471 05470 147        C=DAT1 A
1472 05473 D7        D=C    A
1473 05475 1CE      D1=D1- 15
1474 05478 143      A=DAT1 A
1475 0547B D2        C=0    A
1476 0547D 3113      LC(2)  =oFLSTr
```



```
1477 05481 CA          A=A+C  A
1478 05483 131        D1=A
1479 05486 01         RTN
1480
1481 05488 7210      CHKPSF GOSUB  GETSTe
1482 0548C 8F00000   GOSBVL =GETPRO
1483 05493 441       GOC    mferr
1484 05496 832       ?SB=0
1485 05499 00        RTNYES
1486 0549B 5C0       GONC   mferr
1487
1488 0549E 8F00000   GETSTe GOSBVL =GETSTC
1489 054A5 500       RTNNC
1490 054A8 8D00000   mferr  GOVLNG =MFERR
1491
1492 054AF 1F00000   UPDCRL D1=(5) =CURRST
1493 054B6 147       C=DAT1 A
1494 054B9 135       D1=C
1495 054BC 17E       D1=D1+ (=oFTYPh) -1
1496 054BF 143       A=DAT1 A
1497 054C2 F4        ASR    A
1498 054C4 3400000   LC(5) =fBASIC
1499 054CB 8A6       ?A#C  A
1500 054CE 61        GOYES  UPDCR1
1501 054D0 8F00000   GOSBVL =D0=PCA
1502 054D7 8F00000   GOSBVL =CPL#10
1503 054DE 147       C=DAT1 A
1504 054E1 540       GONC   UPDCR3
1505 054E4 D2        UPDCR1 C=0    A
1506 054E6 1B00000   UPDCR3 D0=(5) =CURRL
1507 054ED 15C3      DAT0=C 4
1508 054F1 03        RTNCC
1509
1510
1511 054F3           END
```

AVMEMS	Ext		-	512	691	836	849	858		
BASCHA	Ext		-	281						
BSERR	Ext		-	147						
Bserr	Abs	19152	#00004AD0	-	265	248				
CHKPSF	Abs	21640	#00005488	-	1481	1252				
CK"ON"	Ext		-	389						
CKINFO	Ext		-	835						
CPL#10	Ext		-	1322	1502					
CRETf+	Ext		-	327						
CURIND	Abs	194762	#0002F8CA	-	60					
CURREN	Ext		-	1330	1470					
CURRL	Ext		-	1506						
CURRST	Ext		-	191	1492					
D0=PCA	Ext		-	1501						
=DBLISe	Abs	18860	#000049AC	-	155					
DISPt	Abs	0	#00000000	-	21	155				
EOLXCK	Ext		-	197	1068					
EXPEXC	Ext		-	237						
FINDA	Ext		-	201	482	964	1002			
FINDF+	Ext		-	212	259					
FINDL	Ext		-	1291	1308					
FINDL0	Ext		-	304						
FSPECx	Ext		-	210	247					
GETLN#	Abs	21583	#0000544F	-	1460	1280	1283	1286	1299	
GETPR1	Ext		-	283						
GETPRO	Ext		-	1482						
GETSTC	Ext		-	1488						
GETSTe	Abs	21662	#0000549E	-	1488	1253	1481			
INDARM	Abs	194798	#0002F8EE	-	71	187	448	466	526	671
INDCOU	Abs	194799	#0002F8EF	-	73	423	496	549	651	722
INDLIG	Abs	194801	#0002F8F1	-	74					
INDVAL	Abs	194757	#0002F8C5	-	59	174	240	553		
INEND	Abs	194793	#0002F8E9	-	69	293	375			
INFILE	Abs	194788	#0002F8E4	-	68	193	215	274	312	373
				-	635	756	778			584
IWAERR	Ext		-	233						
InhEOL	Ext		-	844						
LDCM10	Ext		-	606						
LDEB	Abs	194778	#0002F8DA	-	65	178	183	221	224	229
LFIN	Abs	194783	#0002F8DF	-	66	183	224	229	383	296
LINE#1	Abs	21609	#00005469	-	1470	1269	1315			
LIST05	Abs	18864	#000049B0	-	157	145				
LIST10	Abs	19020	#00004A4C	-	221	203				
LIST20	Abs	19068	#00004A7C	-	236	205				
LIST30	Abs	19102	#00004A9E	-	246	207				
LIST32	Abs	19156	#00004AD4	-	267	263				
MEMCKL	Ext		-	563						
MFERR	Ext		-	1490						
MLFFLG	Ext		-	157						
MOVED2	Ext		-	856						
MOVEUA	Ext		-	689						
MPY	Ext		-	557						
NXTLIN	Ext		-	759						
NXTSTM	Ext		-	392						
OBCOLL	Ext		-	809						
OUTADR	Abs	194768	#0002F8D0	-	62	347	889	894		
OUTBS	Ext		-	502	568	602	625	663		

lst151	Abs	19984	#00004E10	-	600	594				
lst152	Abs	19981	#00004E0D	-	599	597				
lst160	Abs	19987	#00004E13	-	601	591				
lst165	Abs	20112	#00004E90	-	647	644				
lst170	Abs	20133	#00004EA5	-	656	649				
lst180	Abs	20204	#00004EEC	-	679	676				
lst182	Abs	20208	#00004EF0	-	680	684				
lst185	Abs	20222	#00004EFE	-	685	682				
lst190	Abs	20259	#00004F23	-	697	658				
lst500	Abs	20259	#00004F23	-	714	754				
lst510	Abs	20300	#00004F4C	-	729	727				
lst520	Abs	20324	#00004F64	-	739	735				
lst530	Abs	20259	#00004F23	-	713	738	741	746		
lst700	Abs	20360	#00004F88	-	756	645	720			
lst900	Abs	20383	#00004F9F	-	776	716				
lst910	Abs	20386	#00004FA2	-	777	760				
mferr	Abs	21672	#000054A8	-	1490	1265	1483	1486		
nocret	Abs	19416	#00004BD8	-	349	321				
nxtstm	Abs	19475	#00004C13	-	392	308	378	386	1294	1458
oBSsod	Abs	17	#00000011	-	24	299				
oFLENh	Abs	32	#00000020	-	25	288				
oFLSTr	Abs	49	#00000031	-	26	1476				
oFTYPh	Abs	16	#00000010	-	27	277	280	288	1495	
print	Abs	20440	#00004FD8	-	830	514	789			
prnt50	Abs	20505	#00005019	-	849	833				
prnt60	Abs	20591	#0000506F	-	876	874				
qENDI	Abs	2	#00000002	-	38					
qENDL	Abs	0	#00000000	-	36					
qENDS	Abs	1	#00000001	-	37	1071				
ren170	Abs	21448	#000053C8	-	1366	1348	1354			
sautln	Abs	20433	#00004FD1	-	809	500	546			
t!	Ext			-	985	1411				
tCASE	Abs	98	#00000062	-	48	1017				
tCOMMA	Ext			-	202	918	1462			
tDATA	Ext			-	975					
tDEF	Ext			-	973					
tELSE2	Abs	100	#00000064	-	50	1013				
tEND	Ext			-	979					
tEND2	Abs	66	#00000042	-	41	1019				
tENDDF	Ext			-	981					
tENDSB	Ext			-	983					
tEOL	Ext			-	518	744	951			
tFOR	Ext			-	967					
tIF	Ext			-	987					
tIF2	Abs	99	#00000063	-	49	1011				
tLBLST	Ext			-	977					
tLEAVE	Abs	70	#00000046	-	45					
tLOOP	Abs	96	#00000060	-	46	1005				
tNEXT	Ext			-	969					
tREM	Ext			-	1414					
tREPEAT	Abs	68	#00000044	-	43	1007				
tSELECT	Abs	97	#00000061	-	47	1015				
tSUB	Ext			-	971					
tTO	Ext			-	206					
tUNTIL	Abs	69	#00000045	-	44	1009				
tWHILE	Abs	67	#00000043	-	42	1003				
tXWORD	Ext			-	204	965				

Input Parameters

Source file name is blist.a

Listing file name is blist.l

Object file name is blist.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE COMBARR
2
3          * *****
4          * Combination and permutation functions
5          *
6          * Keyword(s):
7          *   ARR/PERM (rev D/X), COMB
8          * Author(s):
9          *   Guy TOUBLANC (P276) 4/12/86
10         * History:
11         *   Published in JPC #41
12         *   Rev X (R. Rosenbaum):
13         *     ARR (arrangement) renamed to PERM (permutation)
14         *
15         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
16         * *****
17
18 054F3          ABS      #54F3
19
20         * *****
21         * ARR/PERM entry
22         * *****
23 054F3 8822          NIBHEX 8822
24 054F7 840          =ARRe  ST=0  0
25 054FA 6A00          GOTO   START
26
27         * *****
28         * COMB entry
29         * *****
30 054FE 8822          NIBHEX 8822
31 05502 850          =COMBe ST=1  0
32 05505 7B11          START  GOSUB  POP
33 05509 8F00000       GOSBVL =STAB2
34 05510 17F          D1=D1+ 16
35 05513 979          ?B=0  W
36 05516 52           GOYES  C1
37 05518 7801          GOSUB  POP
38 0551C 8F00000       GOSBVL =STAB1
39 05523 8F00000       GOSBVL =RCCD2
40 0552A 22           P=      2
41 0552C 8F00000       GOSBVL =TST15
42 05533 541          GONC   P6
43 05536 860          ?ST=0  0
44 05539 F0           GOYES  P6
45 0553B AF2          C1     C=0  W
46 0553E 21           P=      1
47 05540 80CE          C=P    14
48 05544 68D0          GOTO   FNRTN
49 05548 26           P6     P=      6
50 0554A 8F00000       GOSBVL =TST15
51 05551 460          GOC    CONT
52 05554 66F0          GOTO   ERR
53 05558 860          CONT   ?ST=0  0
54 0555B 62           GOYES  ar
55 0555D A4E          C=C-1  S
56 05560 8F00000       GOSBVL =AD2-15
57 05567 8F00000       GOSBVL =RCCD2
```

```
58 0556E 21          P=      1
59 05570 8F00000    GOSBVL =TST15
60 05577 590        GONC   ar
61 0557A 8F00000    GOSBVL =EXAB2
62 05581 132        ar    AD0EX
63 05584 1B00000    D0=(5) (=FUNCD0)
64 0558B 140        DAT0=A  A
65 0558E AF0        A=0     W
66 05591 AF1        B=0     W
67 05594 2E        P=      14
68 05596 B05        B=B+1  P
69 05599 8F00000    GOSBVL =STSCR
70 055A0           LOOP
71 055A0 8F00000    GOSBVL =RCLW1
72 055A7 8F00000    GOSBVL =RCCD1
73 055AE 8F00000    GOSBVL =MP2-15
74 055B5 8F00000    GOSBVL =RCCD2
75 055BC 860        ?ST=0  0
76 055BF 90        GOYES  NODIV
77 055C1 8F00000    GOSBVL =DV2-15
78 055C8 8F00000    NODIV  GOSBVL =STSCR
79 055CF 8F00000    GOSBVL =EXAB2
80 055D6 8F00000    GOSBVL =SUBONE
81 055DD 979        ?B=0   W
82 055E0 22        GOYES  OUT
83 055E2 8F00000    GOSBVL =EXAB2
84 055E9 8F00000    GOSBVL =EXAB1
85 055F0 8F00000    GOSBVL =SUBONE
86 055F7 8F00000    GOSBVL =EXAB1
87 055FE 61AF       GOTO   LOOP
88 05602 8F00000    OUT    GOSBVL =RCLW1
89 05609 1B00000    D0=(5) (=FUNCD0)
90 05610 146        C=DAT0 A
91 05613 134        D0=C
92 05616 8F00000    GOSBVL =uRES12
93 0561D 8D00000    FNRTN  GOVLNG =FNRTN4
94 05624 8F00000    POP    GOSBVL =POP1R
95 0562B 94C        ?A#0   S
96 0562E D1        GOYES  ERR
97 05630 32110     LCHEX  011
98 05635 9B6        ?A>C   X
99 05638 31        GOYES  ERR
100 0563A 8F00000   GOSBVL =SPLITA
101 05641 8F00000   GOSBVL =CLRFRM
102 05648 400       RTNC
103 0564B 8D00000   ERR    GOVLNG =ARGERR
104
105 05652           END
```


AD2-15	Ext		-	56			
ARGERR	Ext		-	103			
=ARRe	Abs	21751 #000054F7	-	24			
C1	Abs	21819 #0000553B	-	45	36		
CLRFRC	Ext		-	101			
=COMBe	Abs	21762 #00005502	-	31			
CONT	Abs	21848 #00005558	-	53	51		
DV2-15	Ext		-	77			
ERR	Abs	22091 #0000564B	-	103	52	96	99
EXAB1	Ext		-	84	86		
EXAB2	Ext		-	61	79	83	
FNRTN	Abs	22045 #0000561D	-	93	48		
FNRTN4	Ext		-	93			
FUNCD0	Ext		-	63	89		
LOOP	Abs	21920 #000055A0	-	70	87		
MP2-15	Ext		-	73			
NODIV	Abs	21960 #000055C8	-	78	76		
OUT	Abs	22018 #00005602	-	88	82		
P6	Abs	21832 #00005548	-	49	42	44	
POP	Abs	22052 #00005624	-	94	32	37	
POP1R	Ext		-	94			
RCCD1	Ext		-	72			
RCCD2	Ext		-	39	57	74	
RCLW1	Ext		-	71	88		
SPLITA	Ext		-	100			
STAB1	Ext		-	38			
STAB2	Ext		-	33			
START	Abs	21765 #00005505	-	32	25		
STSCR	Ext		-	69	78		
SUBONE	Ext		-	80	85		
TST15	Ext		-	41	50	59	
ar	Abs	21889 #00005581	-	62	54	60	
uRES12	Ext		-	92			

Input Parameters

Source file name is combarr.a

Listing file name is combarr.l

Object file name is combarr.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE DATE
2
3          * *****
4          * Date calculation functions
5          *
6          * Keyword(s):
7          *   DATESTR$, DATEADD, DDAYS, DOW$, DOW, DMY, MDY
8          * Author(s):
9          *   Laurent Istria, Francois Legrand,
10         *   Janick Taillandier, Pierre David.
11         * History:
12         *   See below
13         *   Published in JPC #49
14         *   Bug fixes by Rodger Rosenbaum in rev. X
15         *   To do: 1999 leap year test bug fix, from rev.E2
16         *
17         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
18         * *****
19
20 05652          ABS          #5652
21
22         * Premiere version :
23         *   Laurent Istria
24         *   Parue dans JPC 28, Octobre 1985
25         *   Creation du Lex
26         *   Mots-clefs DDAYS, DMY, DOW$, DOW, et MDY
27         * Deuxieme version :
28         *   Francois Legrand
29         *   Parue dans JPC 35, Avril 1986
30         *   Ajout de DATE+
31         *   Retrait des commentaires
32         *   Non correction des defauts
33         * Troisieme version :
34         *   Janick Taillandier & Pierre David
35         *   16 au 18 Avril 1987
36         *   Re conception complete du Lex
37         *   Suppression du conflit de DATE+ par renommage
38         *   en DATEADD
39         *   Verification de la date corrige (117.041987)
40         *   Changement de la signification du flag fldate
41         *   (1 = DMY, 0 = MDY)
42         *   Ecriture des commentaires
43         *   Creation de la table de messages pour DOW$
44         *   Ajout du type de parametre alphanumerique
45         *   pour les dates ("aaaa/mm/jj" ou "aa/mm/jj")
46         *   Ajout de la fonction de conversion DATESTR$
47         *   DOW et DOW$ peuvent ne pas avoir de parametre
48         *   Amelioration de la verification des dates,
49         *   et correction de jj.991582, jj.00aaaa, et
50         *   mm.00aaaa)
51
52
53         * fldate EQU      (0-27)  Flag systeme pour la date (rev C)
54         fldate EQU      (0-53)  Flag systeme pour la date (rev D)
55         sDMY EQU        0       Flag temporaire
56         sDOW$ EQU       1       1 si DOW$, 0 si DOW
57
```

```
58             STITLE UTILITAIRES
59
60             *****
61             * getdat
62             *
63             * But: obtenir une date sous un format utilisable
64             *   a partir d'un objet sur la Math Stack
65             * Entree:
66             *   - D1 = ^ M.S.
67             * Sortie:
68             *   - A, B et C = numero du jour depuis le jour 0
69             *   - D1 reactualise
70             *   - ST(sDMY) = 1 si mode DMY, 0 si mode MDY
71             * Abime: A-D, R0, R1, FUNCD1, ST(0), ST(8)
72             * Niveaux: 3
73             * Appelle: POP1R, POP1S, conv2, chk/, verdat
74             * Algorithmme:
75             *
76             *   DECODAGE :
77             *
78             *   si type numerique
79             *     alors
80             *       x := IP(arg) ;
81             *       y := IP(FP(arg)*100) ;
82             *       A := IP(FP(arg*100)*10000) ;
83             *       si DMY
84             *         alors
85             *           D := x ;
86             *           B := y ;
87             *         sinon
88             *           D := y ;
89             *           B := x ;
90             *       fin si ;
91             *     sinon (type alphanumerique)
92             *       p := 3 ;
93             *       si arg$(p)="/"
94             *         alors
95             *           x := arg$(1,2) ;
96             *           si x < 60
97             *             alors A := 2000 + x ;
98             *             sinon A := 1900 + x ;
99             *           fin si ;
100            *         sinon
101            *           p := 5 ;
102            *           A := arg$(1,4) ;
103            *         fin si ;
104            *       si arg$(p) # "/" alors erreur ; fin si ;
105            *       B := arg$(p+1,p+2) ;
106            *       p := p+3 ;
107            *       si arg$(p) # "/" alors erreur ; fin si ;
108            *       D := arg$(p+1,p+2) ;
109            *       si il reste des caracteres alors erreur ;
110            *     fin si ;
111            *
112            *   VERIFICATION :
113            *
114            *   Modifications:
```

```
115      *   Ajout du parametre alphanumerique
116      *   Essayez 117.041987 avec l'ancienne version !
117      *   Separation du decodage et de la verification
118      *   Ajout des commentaires
119      *   Historique:
120      *   85/10/   : L.I.
121      *   87/04/16: J.T. & P.D. reconception & recodage
122      *****
123
124 05652      getdat
125      *
126      *   Lecture de fldATE pour avoir le mode DMY ou MDY
127      *
128 05652 31BC          LC(2)  fldATE  C(B) = flag number
129 05656 8F00000      GOSBVL =SFLAG?
130      *
131      *   En sortie de SFLAG?
132      *   Cy = flag teste (1 si DMY, 0 si MDY)
133      *   HEX mode
134      *   P=0
135      *
136 0565D 850          ST=1   sDMY   Mode DMY par defaut
137 05660 450          GOC    getd10  DMY, on ne change rien
138 05663 840          ST=0   sDMY   pour les ricains !
139 05666      getd10
140      *
141      *   Test du type
142      *
143 05666 1534          A=DAT1 S      Signature de l'element
144 0566A B44          A=A+1 S      Chaine = F ==> Cy := 1
145 0566D 4B5          GOC    getstr
146 05670      getnum
147 05670 8F00000      GOSBVL =POP1R
148 05677 17F          D1=D1+ 16    On passe le reel
149      *
150      *   En sortie de POP1R :
151      *   A = 12 digits form
152      *   DEC mode
153      *
154 0567A 20          P=      0      Apres POP1R, P = ?
155 0567C 04          SETHEX
156      *
157      *   Test du signe
158      *
159 0567E 94C          ?A#0  S      signe different de "+"
160 05681 44          GOYES  Ivarg
161      *
162      *   Test de l'exposant du nombre lu. Il doit valoir
163      *   0 ou 1. Le registre A a donc la forme suivante :
164      *   (exemple dans le cas DMY)
165      *
166      *   A(W) = 0jmmaaaa.....000
167      *   A(W) = 0jjmmaaaa.....001
168      *
169 05683 AB2          C=0    X
170 05686 932          ?A=C  X
171 05689 D0          GOYES  getn20
```

```
172 0568B B36          C=C+1  X
173 0568E 936          ?A#C   X
174 05691 43           GOYES  Ivarg
175 05693 BF0          ASL    W
176
177
178 * On a dans A :
179 * A(W) = jjmmaaaa..... ou encore
180 * A(W) = mmjjaaaa..... si MDY
181
182 05696          getn20
183 05696 810          ASLC
184 05699 810          ASLC
185 0569C AE6          C=A    B          C(B) := jj (si DMY)
186
187 * A(W) = mmaaaa.....jj si DMY
188 * A(W) = jjaaaa.....mm si MDY
189
190 0569F 810          ASLC
191 056A2 810          ASLC
192
193 *
194 * Si DMY
195 *   A(B) = mm
196 *   C(B) = jj
197 * Si MDY
198 *   A(B) = jj
199 *   C(B) = mm
200 * dans tous les cas, A(W) = aaaa.....yyxx
201 *
202
203 *
204 * Si MDY alors ACEX B
205
206 056A5 870          ?ST=1  sDMY
207 056A8 50           GOYES  getn30
208 056AA AEE          ACEX   B
209
210 056AD          getn30
211
212 *
213 * quelque soit le mode, on a maintenant :
214 * A(B) = mm
215 * C(B) = jj
216
217 056AD AE7          D=C    B          D(B) := jj
218 056B0 AE8          B=A    B          B(B) := mm
219
220 *
221 * Reste a isoler la date dans A(A) :
222 *
223 056B3 D0           A=0    A
224 056B5 810          ASLC
225 056B8 810          ASLC
226 056BB 810          ASLC
227 056BE 810          ASLC          A(A) := 0aaaa
228
229 *
230 * Nous avons donc maintenant :
231 *   A(A) = 0aaaa
232 *   B(B) = mm
233 *   D(B) = jj
234
235 056C1 67B0          GOTO   verdat
236
237
```

```

229 056C5 67C1      Ivarg GOTO   ivarg
230
231 056C9           getstr
232 056C9 8F00000   GOSBVL =POP1S
233 056D0 137       CD1EX
234 056D3 C2        C=C+A  A      C(A) := ^ item suivant
235 056D5 1F00000   D1=(5) =FUNCD1 Sauvegarde de D1
236 056DC 145       DAT1=C  A
237 056DF 135       D1=C      D1 := ^ debut de la chaine
238 056E2 25        P=        5
239 056E4 A80       A=0      P
240 056E7 20        P=        0
241 056E9 81C       ASRB           A(A) := longueur en octets
242 056EC D2        C=0      A
243 056EE 858       ST=1     8      8 caracteres pour la date
244 056F1 308       LC(1)    8
245 056F4 8A2       ?A=C    A
246 056F7 D0        GOYES   gets10
247 056F9 848       ST=0     8      10 caracteres pour la date
248 056FC 30A       LC(1)   10
249 056FF 8A6       ?A#C   A
250 05702 3C        GOYES   Ivarg
251 05704           gets10
252 05704 7741      GOSUB   conv2   convertit 2 caracteres
253 * C(B) = l'annee (ou le siecle)
254 05708 878       ?ST=1   8
255 0570B D1        GOYES   gets20
256 *
257 * L'annee est sur 4 chiffres, il faut lire les
258 * deux derniers.
259 *
260 0570D AF0       A=0      W
261 05710 AEA       A=C      B
262 05713 F0        ASL      A
263 05715 F0        ASL      A
264 05717 100       R0=A      R0 := 000000000000aa00
265 0571A 7131      GOSUB   conv2
266 0571E 110       A=R0
267 05721 AEA       A=C      B      A(W) := 000000000000aaaa
268 05724 6520      GOTO    gets50
269 *
270 * L'annee est sur 2 chiffres.
271 *   si <60 alors 20aa
272 *   sinon 19aa
273 *
274 05728           gets20
275 05728 AF0       A=0      W
276 0572B AEA       A=C      B
277 0572E D2        C=0      A
278 05730 3106      LCHEX   60
279 05734 9E2       ?A<C   B
280 05737 90        GOYES   gets30
281 05739 3191      LCHEX   19
282 0573D 560       GONC    gets40
283 05740 3102      gets30  LCHEX   20
284 05744 F2        gets40  CSL     A
285 05746 F2        CSL     A

```

```
286 05748 CA          A=A+C  A          A(W) := 000000000000aaaa
287 0574A 100        gets50 R0=A
288 0574D 7D21       GOSUB  chk/
289 05751 7AF0       GOSUB  conv2    C(B) := mm
290 05755 109        R1=C
291 05758 7221       GOSUB  chk/
292 0575C 7FE0       GOSUB  conv2
293 05760 AE7        D=C    B          D(B) := jj
294
295 *
296 * restauration de D1
297 05763 1F00000    D1=(5) =FUNCD1
298 0576A 147        C=DAT1 A
299 0576D 135        D1=C
300
301 *
302 * restauration du mois et de l'annee
303 05770 119        C=R1          C(B) := mm
304 05773 AE5        B=C    B          B(B) := mm
305 05776 110        A=R0          A(A) := 0aaaa
306
307 *
308 * Attention. Le code continue !!!
309
310 *****
311 * verdat
312 *
313 * But: verifier la validite d'une date
314 * Entree:
315 * - A(A) = 0aaaa
316 * - B(B) = mm
317 * - D(B) = jj
318 * Sortie:
319 * - A, B et C = numero du jour depuis le jour 0
320 * - HEX
321 * - P=0
322 * Abime: A-D
323 * Niveaux: 2
324 * Appelle: YMDDAY
325 * Algorithme:
326 *   erreur si j=0 ;
327 *   erreur si m=0 ;
328 *   erreur si a<1582 ;
329 *   erreur si m>12 ;
330 *   si a>1582
331 *     alors
332 *       si m#2
333 *         alors
334 *           jmax := dernier jour du mois ;
335 *           erreur si j>jmax ;
336 *           (date valide)
337 *         sinon
338 *           erreur si j>29 ;
339 *           si j=29
340 *             alors
341 *               erreur si a non divisible par 4 ;
342 *               si a divisible par 100
```



```
343          *          alors
344          *          erreur si non divis. par 400;
345          *          fin si ;
346          *          fin si ;
347          *          fin si ;
348          *          sinon (annee = 1582)
349          *          erreur si m<10 ;
350          *          erreur si m=10 et j<15 ;
351          *          jmax := dernier jour du mois ;
352          *          erreur si j>jmax ;
353          *          (date valide)
354          *          fin si ;
355          * Modifications:
356          * separation logique du reste du sous programme
357          * eclaireissement de l'algorithmme
358          * tests corrects pour 01.991582 (DMY)
359          * tests corrects pour j = 0 ou m = 0
360          * Historique:
361          * 87/04/18: J.T. & P.D. conception & codage
362          * 88/??/??: (PPC-Paris) attempt to fix leap year bug
363          * 92/??/??: Rodger Rosenbaum definitive leap year fix
364          * 06/01/11: J-F G. documentation
365          * *****
366
367
368 05779      verdat
369          *
370          * erreur si mois = 0 ;
371          *
372 05779 969          ?B=0   B
373 0577C 86          GOYES  erreur
374          *
375          * erreur si jour = 0 ;
376          *
377 0577E 96B          ?D=0   B
378 05781 36          GOYES  erreur
379          *
380          * erreur si mois > 12 ;
381          *
382 05783 3121          LCHEX  12
383 05787 9E1          ?B>C   B
384 0578A A5          GOYES  erreur
385          *
386          * erreur si annee < 1582 ;
387          *
388 0578C 3428510      LCHEX  01582
389 05793 8B2          ?A<C   A
390 05796 E4          GOYES  erreur
391          *
392          * si annee > 1582
393          *
394 05798 8A2          ?A=C   A
395 0579B D4          GOYES  verd50   annee = 1582
396          *
397          * alors
398          * si mois # 2
399          * alors verification normale
```

```
400          *
401 0579D 3120          LCHEX  02      Fevrier
402 057A1 965          ?C#B   B
403 057A4 41          GOYES  verd10   mois normal
404          *
405          *          sinon (mois = fevrier)
406          *          erreur si jour > 29 ;
407          *
408 057A6 3192          LCHEX  29
409 057AA 9E3          ?D>C   B
410 057AD 73          GOYES  erreur
411          *
412          *          si jour # 29
413          *          alors ok
414          *
415 057AF 963          ?D=C   B
416 057B2 A0          GOYES  verd15
417 057B4 6A70          GOTO   verd99
418 057B8 6640          verd10 GOTO   verd70   rallonge
419
420
421          *
422          *          sinon
423          *          erreur si a non divisible par 4 ;
424          *
425 057BC          verd15
426          ***** changes
427          ***** rev D
428          ** bug fix attempt, PPC-Paris 1988, for JPC LEX rev D:
429          ** replace:
430          **      C=A      A
431          ** with lines below to convert the BCD date in binary
432          ** before testing the divisibility by 4:
433          *          ASRC
434          *          C=A      P
435          *          C=C+A   P
436          *          C=C+A   P
437          *          C=C+A   P
438          *          C=C+A   P
439          *          C=C+A   P
440          *          C=C+A   P
441          *          C=C+A   P
442          *          C=C+A   P
443          *          C=C+A   P      C=C*10 P
444          *          ASLC
445          *          C=C+A   P
446          ** end of bug fix, now we can test if divisible by 4:
447          *          SB=0
448          *          CSRB
449          *          CSRB
450          *          ?SB=0
451          *          GOYES  verd20
452          * erreur GOTO   ivarg
453          **
454          **          erreur si a non divisible par 400 ;
455          * verd20
456          *          ?A#0   B
```

```
457      *      GOYES verd99   Ok, non divisible par 100
458      *      C=A      A
459      *      CSR      A
460      *      CSR      A      C(B) := siecle
461      ** note J-F G., jan 2006: here is a similar bug than above:
462      ** test of the century is done on a BCD, not binary, number!
463      ** It was not corrected with the bug above!!
464      ** JPCLEX rev D gives 1800 as leap year!
465      ** It should mentionned than the HP71
466      ** mainframe ROM 1BBBB has a similar bug (ref. ) that
467      ** was corrected in rev 2CCCC (quite a long story...)
468      *      SB=0      inutile ?
469      *      CSR      CSR
470      *      CSR      CSR
471      *      ?SB=0
472      *      GOYES verd99   Ok, divisible par 400
473      *      GONC      erreur B.E.T.
474      ***** rev X:
475      * definitive bug fix by Rodger Rosenbaum (~1992)
476      * that solves all leap year issues:
477 057BC 100      R0=A
478      *** 1999 leap year test bug fix, rev.E2 (not in rev. Ex)
479      **      LCHEX 0F0F0 *
480      **      C=A&C A *
481      ** to replace this:
482 057BF D6      C=A A * rev. Ex JFG
483      *** end of 1999 leap year test bug fix
484 057C1 C6      C=C+C A
485 057C3 F6      CSR A
486 057C5 CA      A=A+C A
487 057C7 3430300 LCHEX 00303
488 057CE 0EF2      C=A&C A
489 057D2 110      A=R0
490 057D5 96E      ?C#0 B
491 057D8 C0      GOYES erreur GO IF NOT DIV BY 4
492 057DA 8AA      ?C=0 A
493 057DD 25      GOYES verd99 OK, DIVISIBLE BY 400
494 057DF 96C      ?A#0 B
495 057E2 D4      GOYES verd99 OK, NOT DIV BY 100
496 057E4 68A0      erreur GOTO ivarg
497      ***** end of changes
498
499 057E8      verd50
500      *
501      * (annee 1582)
502      * erreur si mois < octobre
503      *
504 057E8 3101      LCHEX 10 Octobre
505 057EC 9E5      ?B<C B
506 057EF 5F      GOYES erreur
507      *
508      * erreur si mois = 10 et jour < 15 ;
509      *
510 057F1 965      ?B#C B
511 057F4 B0      GOYES verd70 mois normal
512 057F6 3151      LCHEX 15
513 057FA 9E7      ?D<C B
```

```
514 057FD 7E          GOYES erreur
515                  *
516                  * Attention ! le code continue !
517                  *
518
519
520                  *
521                  * (mois normal)
522                  *
523 057FF             verd70
524                  *
525                  * jmax := dernier jour du mois
526                  *
527 057FF AE9         C=B      B
528 05802 108        R0=C          Sauvegarde du mois
529 05805 3170      LCHEX  07
530 05809 9ED       ?B<=C  B
531 0580C 90        GOYES  verd80
532 0580E 05        SETDEC
533 05810 B65       B=B+1  B
534 05813 04        SETHEX
535
536 05815           verd80
537 05815 3110      LCHEX  01
538 05819 0E61      B=B&C  B      B := bit de poids faible
539 0581D 3103      LCHEX  30
540 05821 A69       C=C+B  B      C(B) := jmax
541
542                  *
543                  * erreur si jour > jmax ;
544                  *
545 05824 9E3       ?D>C  B
546 05827 DB        GOYES  erreur
547 05829 118      C=R0          sauvegarde du mois
548 0582C AE5      B=C      B
549
550 0582F           verd99
551                  *
552                  * Ok, c'est bon
553                  *
554 0582F D6        C=A      A
555 05831 AF0       A=0      W
556 05834 DA        A=C      A
557 05836 AE9       C=B      B
558 05839 AF1       B=0      W
559 0583C AE5       B=C      B
560 0583F AEB       C=D      B
561 05842 AF3       D=0      W
562 05845 AE7       D=C      B
563 05848 8D00000  GOVLNG =YMDDAY
564
565
566                  *****
567                  * conv2
568                  *
569                  * But: convertir deux caracteres en deux chiffres
570                  * BCD.
```

```
571      * Entree:
572      *   - D1 = ^ M.S.
573      * Sortie:
574      *   - C(B) = valeur lue et convertie
575      *   - D1 actualise
576      * Abime: A, B(B), C
577      * Niveaux: 2
578      * Appelle: conv1, DRANGE
579      * Historique:
580      *   87/04/17: J.T. & P.D. conception & codage
581      *****
582
583 0584F      conv2
584 0584F 7210      GOSUB conv1      poids fort
585 05853 AE8      B=A      B
586 05856 7B00      GOSUB conv1
587 0585A AE9      C=B      B
588 0585D BE2      CSL      B
589 05860 A62      C=A+C      B
590 05863 01      RTN
591
592 05865 1C1      conv1 D1=D1- 2
593 05868 14B      A=DAT1 B
594 0586B 8F00000      GOSBVL =DRANGE
595 05872 4A1      GOC      ivarg      byte not in range
596 05875 3103      LCASC '0'
597 05879 B6A      A=A-C      B      A(B) = 0d (d = 1..9)
598 0587C 01      RTN
599
600      *****
601      * chk/
602      *
603      * But: verifier que le caractere courant est bien
604      *   un slash.
605      * Entree:
606      *   - D1 = ^ M.S.
607      * Sortie:
608      *   - si le caractere etait bien un "/", D1 est
609      *     reactualise
610      *   - sinon erreur
611      * Abime: A(B), C(B)
612      * Niveaux: 0
613      * Historique:
614      *   87/04/17: J.T. & P.D. conception & codage
615      *****
616
617 0587E      chk/
618 0587E 1C1      D1=D1- 2
619 05881 14B      A=DAT1 B
620 05884 31F2      LCASC '/'
621 05888 962      ?A=C      B
622 0588B 00      RTNYES
623 0588D 8D00000      ivarg GOVLNG =ARGERR
624
625      *****
626      * send2
627      *
```

```
628          * But: fonction inverse de conv2 : envoie 2
629          *   chiffres BCD sur la M.S. en ASCII
630          * Entree:
631          *   - D1 = ^ M.S.
632          *   - D(A) = AVMEMS
633          *   - A(B) = les deux chiffres en BCD
634          * Sortie:
635          *   - D1 reactualise
636          * Abime: C(B)
637          * Niveaux: 2
638          * Appelle: STKCHR
639          * Historique:
640          *   87/04/17: J.T. & P.D. conception & codage
641          *****
642
643 05894          send2
644 05894 814          ASRC
645 05897 7300          GOSUB send1
646 0589B 810          ASLC
647 0589E          send1
648 0589E 3103          LCASC '0'
649 058A2 A86          C=A P
650 058A5 8D00000      stkchr GOVLNG =STKCHR
651
652
653          STITLE LES ORDRES BASIC
```

```
654
655 *****
656 * DATESTR$
657 *
658 * But: renvoyer la date alphanumerique au format
659 * HP71 ("aaaa/mm/jj") a partir de la date au
660 * format numerique jj.mmaaaa ou mm.jjaaaaa
661 * Note: La date renvoyee par DATE$ est de la forme
662 * "aa/mm/jj". La date renvoyee par DATESTR$ est
663 * de la forme "aaaa/mm/jj". DATESTR$(DATE$)
664 * convertit donc une date "aa" en date "aaaa".
665 * Syntaxe: DATESTR$ ( <date> )
666 * Historique:
667 * 87/04/17: P.D. & J.T. conception & codage
668 *****
669
670
671 * *****
672 * DATESTR$ entry
673 * *****
674 058AC C CON(1) 8+4 alpha ou num 1er param.
675 058AD 11 NIBHEX 11 2 parametres obligatoires
676 058AF =DATESe
677 058AF 7F9D GOSUB getdat
678 *
679 * A, B, C = date au format interne
680 *
681 058B3 8F00000 GOSBVL =DAYYMD
682 058BA 04 SETHEX
683 058BC 20 P= 0
684 *
685 * A(3-0) = aaaa
686 * B(B) = mm
687 * D(B) = jj
688 *
689 058BE 100 R0=A R0 := annee
690 058C1 AE9 C=B B
691 058C4 10A R2=C R2 := mois
692 058C7 AEB C=D B
693 058CA 10B R3=C R3 := jour
694
695 058CD 8F00000 GOSBVL =D=AVMS ne modifie pas A(W)
696 058D4 109 R1=C R1 := ^ bottom of M.S.
697
698 058D7 F4 ASR A
699 058D9 F4 ASR A A(B) := siecle
700 058DB 75BF GOSUB send2
701 058DF 110 A=R0 A(B) := annee
702 058E2 7EAF GOSUB send2
703 058E6 31F2 LCASC '/'
704 058EA 77BF GOSUB stkchr
705 058EE 112 A=R2 A(B) := mois
706 058F1 7F9F GOSUB send2
707 058F5 31F2 LCASC '/'
708 058F9 78AF GOSUB stkchr
709 058FD 113 A=R3 A(B) := jour
710 05900 709F GOSUB send2
```

```
711 05904 840          ST=0  0          No return desired
712 05907 8D00000      GOVLNG =ADHEAD
713
714 *****
715 * DATEADD
716 *
717 * But: renvoyer la date correspondant a : date + n
718 * Syntaxe: DATEADD ( <date> , <n> )
719 * Modifications:
720 *   Ajout du parametre de type alpha
721 *   Ajout des commentaires
722 *   Clarification du code
723 *   Extension des dates jusqu'au 31/12/9999
724 * Historique:
725 *   86/03/  : F.D.
726 *   87/04/17: P.D. & J.T. reconception & recodage
727 *****
728
729 * *****
730 * DATEADD entry
731 * *****
732 0590E 8             CON(1) 8          num          2eme param.
733 0590F C             CON(1) 8+4        alpha ou num  1er param.
734 05910 22           NIBHEX 22         2 parametres obligatoires
735 05912              =DATEAe
736 05912 8F00000     GOSBVL =RNDAHX
737 *
738 * Ce n'est pourtant pas si dur d'utiliser les
739 * registres dans leur totalite.
740 *
741 05919 AF2          C=0    W
742 0591C D6          C=A    A
743 0591E 470        GOC    DTAD10
744 *
745 * Parametre negatif
746 *
747 05921 FA          C=-C   A
748 05923 BFA        C=-C   W          parametre negatif sur 16 q.
749 05926            DTAD10
750 05926 17F        D1=D1+ 16
751 05929 10B        R3=C          R3 := n
752 0592C 722D      GOSUB  getdat
753 05930 11B        C=R3
754 05933 A72        C=A+C   W          C(W) := date + n
755 *
756 * Attention ! le code continue
757 *
758
759 *****
760 * rtndat
761 *
762 * But: convertir une date en format interne (nb de
763 * jours depuis le 1er janvier 0) en reel au
764 * format jj.mmaaaa (ou mm.jjaaaa), et retourner
765 * a Basic.
766 * Entree:
767 * - C(W) = date au format interne
```



```

768      * - ST(sDMY) indique le format (DMY ou MDY)
769      * Sortie: par FNRTN1
770      * Appelle: DAYYMD
771      * Historique:
772      *   87/04/17: P.D. & J.T. conception & codage
773      * *****
774
775 05936      rtndat
776 05936 8F00000      GOSBVL =DAYYMD
777      *
778      * A = aaaa
779      * B = mm
780      * D = jj
781      *
782 0593D 870      ?ST=1  sDMY
783 05940 B0      GOYES  DTAD20
784 05942 AE9      C=B      B      C := mm
785 05945 AEF      DCEX   B      D := mm ; C := jj
786 05948 AE5      B=C      B      B := jj
787 0594B      DTAD20
788      *
789      * Le nombre que l'on devra retourner doit etre de
790      * la forme :
791      *   0ddbbaaaa00000001  ou
792      *   0dbbaaaa00000000  si dd<10
793      * (dd = D(B), bb = B(B), aaaa = A(3-0))
794      *
795 0594B D6      C=A      A
796 0594D AF0      A=0      W
797 05950 24      P=      4
798 05952 A9A      A=C      WP
799 05955 20      P=      0
800      *
801      * A(W) = 000000000000aaaa
802      *
803 05957 814      ASRC
804 0595A 814      ASRC
805 0595D 814      ASRC
806 05960 814      ASRC
807      *
808      * A(W) = aaaa000000000000
809      *
810 05963 AE4      A=B      B
811 05966 814      ASRC
812 05969 814      ASRC
813      *
814      * A(W) = bbaaaa0000000000
815      *
816 0596C 3101      LCHEX  10
817 05970 AEF      DCEX   B      D := 10 ; C(B) := dd
818 05973 AEA      A=C      B      A(W) := bbaaaa00000000dd
819 05976 9E3      ?C<D  B      dd < 10
820 05979 50      GOYES  rtnd10  un seul shift, Cy := 1
821      * deux shifts, Cy := 0
822 0597B 814      ASRC
823 0597E 814      rtnd10 ASRC
824 05981 BF4      ASR      W

```

```
825      *
826      * A(W) = 0dbbaaaa00000000
827      * A(W) = 0dbbaaaa00000000
828      *
829 05984 450          GOC      rtn20
830 05987 B34          A=A+1  X
831 0598A AF6      rtn20 C=A    W
832 0598D 66B0      GOTO     fnrtn1
833
834      *****
835      * DDDAYS
836      *
837      * But: renvoyer date1-date2
838      * Syntaxe: DDDAYS ( <date1> , <date2> )
839      * Modifications:
840      *   Ajout des parametres de type alpha
841      *   Ajout des commentaires
842      * Historique:
843      *   85/10/   : L.I.
844      *   87/04/17: P.D. & J.T. reconception & recodage
845      *****
846
847      * *****
848      * DDDAYS entry
849      * *****
850 05991 C          CON(1) 8+4      alpha ou num
851 05992 C          CON(1) 8+4      alpha ou num
852 05993 22        NIBHEX 22      2 parametres obligatoires
853 05995          =DDAYS
854 05995 79BC      GOSUB  getdat
855 05999 103      R3=A          R3 := date2
856 0599C 72BC      GOSUB  getdat
857 059A0 11B      C=R3
858
859      * A(W) = date1
860      * C(W) = date2
861      *
862 059A3 AC3          D=0      S          signe := "+"
863 059A6 9FE          ?A>=C  W          date1 >= date2
864 059A9 C0          GOYES   DDAY10   oui : ok
865      * signe "-"
866 059AB AFE          ACEX    W          non : on echange
867 059AE 05          SETDEC
868 059B0 BCF          D=-D-1 S          et signe := "-"
869 059B3 04          SETHEX
870 059B5          DDAY10
871 059B5 B7E          C=A-C  W          C(W) := date1 - date2
872 059B8 8F00000     GOSBVL =HXDCW   full word hex-dec conv.
873 059BF 8F00000     GOSBVL =FLOAT
874      *
875      * A(W) = resultat, mode = DEC
876      *
877 059C6 AF6          C=A      W
878 059C9 ACB          C=D      S          C(S) := signe
879 059CC 6770        GOTO     fnrtn1
880
881      *****
```

```
882          * DOW$
883          *
884          * But: renvoyer le nom du jour
885          * Syntaxe: DOW$ ( [ <date> ] )
886          * Modifications:
887          *   Ajout du parametre de type alpha
888          *   Parametre optionnel = date d'aujourd'hui
889          *   Ajout des commentaires
890          *   Nom des jours en messages
891          * Historique:
892          *   85/10/   : L.I.
893          *   87/04/17: P.D. & J.T. reconception & recodage
894          *****
895
896          * *****
897          * DOW$ entry
898          * *****
899 059D0 C          CON(1) 8+4          alpha ou num
900 059D1 01        NIBHEX 01          1 parametre optionnel
901 059D3          =DOWse
902 059D3 851      ST=1    sDOW$
903 059D6 6900    GOTO    DOW00
904
905          *****
906          * DOW
907          *
908          * But: renvoyer le numero du jour
909          * Syntaxe: DOW ( [ <date> ] )
910          * Modifications:
911          *   Ajout du parametre de type alpha
912          *   Parametre optionnel = date d'aujourd'hui
913          *   Ajout des commentaires
914          * Historique:
915          *   85/10/   : L.I.
916          *   87/04/17: P.D. & J.T. reconception & recodage
917          *****
918
919          * *****
920          * DOW entry
921          * *****
922 059DA C          CON(1) 8+4          alpha ou num
923 059DB 01        NIBHEX 01          1 parametre optionnel
924 059DD          =DOWe
925 059DD 841      ST=0    sDOW$
926 059E0          DOW00
927          *
928          * Algorithmme :
929          * si nb parametre = 1
930          *   alors decoder la date
931          *   sinon obtenir la date d'ajourd'hui
932          * fin si ;
933          * jour := (date - 1) mod 7
934          *
935 059E0 94E      ?C#0    S
936 059E3 04      GOYES   DOW10
937          *
938          * Sauvegarde temporaire de D0 et D1
```

```
939          *
940 059E5 137          CD1EX
941 059E8 06          RSTK=C
942 059EA 136          CD0EX
943 059ED 06          RSTK=C
944 059EF 0B          CSTEX
945 059F1 10B         R3=C
946 059F4 8F00000     GOSBVL =CMPT      C = R1 := current time
947          *
948          * Restauration de D0 et D1 apres le monstre CMPT
949          *
950 059FB 11B         C=R3
951 059FE 0B          CSTEX
952 05A00 07          C=RSTK
953 05A02 134         D0=C
954 05A05 07          C=RSTK
955 05A07 135         D1=C
956          *
957          * Et on reprend le cours de nos investigations...
958          *
959 05A0A 119         C=R1
960 05A0D BF6         CSR      W
961 05A10 BF6         CSR      W
962 05A13 81E         CSRB           C / 512 (in seconds)
963 05A16 8F00000     GOSBVL =TODT      A = day number
964 05A1D 20          P=      0
965 05A1F 6700        GOTO      DOW20
966 05A23             DOW10
967 05A23 7B2C        GOSUB      getdat
968 05A27             DOW20
969 05A27 A7C         A=A-1      W
970 05A2A AF2         C=0        W
971 05A2D 307         LC(1)     7
972 05A30 8F00000     GOSBVL =IDIV      C := a-1 mod 7
973 05A37 20          P=      0
974          *
975          * C(0) = numero du jour (0:dimanche ... 6:samedi)
976          *
977 05A39 871         ?ST=1      sDOW$
978 05A3C F0          GOYES      DOW30
979          *
980          * Sortie numerique
981          *
982 05A3E 816         CSRC
983 05A41 816         CSRC           C(14) := a-1 mod 7
984 05A44 8D00000     fnrtn1 GOVLNG =FNRTN1
985          *
986          * Sortie alphanumerique
987          *
988 05A4B             DOW30
989 05A4B DA          A=C      A      A(A) := numero du jour
990 05A4D 34801E0     LC(5) #0E108     eSunday (=id)~(=eSUN)
991 05A54 C2          C=C+A    A      C(A) := numero du message
992 05A56 108         R0=C
993          *
994          * Pompe dans les IDS I, page 17-60, d'apres MSG$
995          *
```

```
996 05A59 8F00000          GOSBVL =R3=D10  Sauver D1 et D0
997 05A60 8F00000          GOSBVL =FPOLL
998 05A67 00                CON(2) =pTRANS
999 05A69 8F00000          GOSBVL =D0=AVS  D0 := (AVMEMS)
1000 05A70 118              C=R0
1001 05A73 8F00000          GOSBVL =TBMSG$
1002 05A7A 8D00000          GOVLNG =ERRM$f  Et c'est supporte !!!
1003
1004 * Fin du pompage...
1005 *
1006
1007 *****
1008 * DMY
1009 *
1010 * But: passer en mode jj.mmaaaa
1011 * Syntaxe: DMY
1012 * Modifications:
1013 * Utilisation de routines supportees
1014 * Historique:
1015 * 85/10/ : L.I.
1016 * 87/04/17: P.D. & J.T. reconception & recodage
1017 *****
1018
1019 * *****
1020 * DMY entry
1021 * *****
1022 05A81 00000          REL(5) =STDD
1023 05A86 00000          REL(5) =NOARGp
1024 05A8B                =DMYe
1025 05A8B 31BC          LC(2) =f1DATE
1026 05A8F 8F00000          GOSBVL =SFLAGS  Set system flag
1027 05A96 571          GONC  nxtstm  B.E.T.
1028
1029 *****
1030 * MDY
1031 *
1032 * But: passer en mode mm.jjaaaa
1033 * Syntaxe: MDY
1034 * Modifications:
1035 * Utilisation de routines supportees
1036 * Historique:
1037 * 85/10/ : L.I.
1038 * 87/04/17: P.D. & J.T. reconception & recodage
1039 *****
1040
1041 * *****
1042 * MDY entry
1043 * *****
1044 05A99 00000          REL(5) =STDD
1045 05A9E 00000          REL(5) =NOARGp
1046 05AA3                =MDYe
1047 05AA3 31BC          LC(2) =f1DATE
1048 05AA7 8F00000          GOSBVL =SFLAGC  Clear system flag
1049 05AAE 8D00000          nxtstm GOVLNG =NXTSTM
1050
1051 05AB5                END
```

ADHEAD	Ext		-	712						
ARGERR	Ext		-	623						
CMPT	Ext		-	946						
D0=AVS	Ext		-	999						
D=AVMS	Ext		-	695						
=DATEAe	Abs	22802	#00005912	-	735					
=DATESe	Abs	22703	#000058AF	-	676					
DAYMD	Ext			-	681	776				
DDAY10	Abs	22965	#000059B5	-	870	864				
=DDAYSe	Abs	22933	#00005995	-	853					
=DMYe	Abs	23179	#00005A8B	-	1024					
DOW00	Abs	23008	#000059E0	-	926	903				
DOW10	Abs	23075	#00005A23	-	966	936				
DOW20	Abs	23079	#00005A27	-	968	965				
DOW30	Abs	23115	#00005A4B	-	988	978				
=DOWe	Abs	23005	#000059DD	-	924					
=DOWse	Abs	22995	#000059D3	-	901					
DRANGE	Ext			-	594					
DTAD10	Abs	22822	#00005926	-	749	743				
DTAD20	Abs	22859	#0000594B	-	787	783				
ERRM\$f	Ext			-	1002					
FLOAT	Ext			-	873					
FNRTN1	Ext			-	984					
FPOLL	Ext			-	997					
FUNCD1	Ext			-	235	297				
HXDCW	Ext			-	872					
IDIV	Ext			-	972					
Ivarg	Abs	22213	#000056C5	-	229	160	174	250		
=MDYe	Abs	23203	#00005AA3	-	1046					
NOARGp	Ext			-	1023	1045				
NXTSTM	Ext			-	1049					
POP1R	Ext			-	147					
POP1S	Ext			-	232					
R3=D10	Ext			-	996					
RNDAHx	Ext			-	736					
SFLAG?	Ext			-	129					
SFLAGC	Ext			-	1048					
SFLAGS	Ext			-	1026					
STDd	Ext			-	1022	1044				
STKCHR	Ext			-	650					
TBMSG\$	Ext			-	1001					
TODT	Ext			-	963					
YMDDAY	Ext			-	563					
chk/	Abs	22654	#0000587E	-	617	288	291			
conv1	Abs	22629	#00005865	-	592	584	586			
conv2	Abs	22607	#0000584F	-	583	252	265	289	292	
erreur	Abs	22500	#000057E4	-	496	373	378	384	390	410 491
					506	514	546			
f1DATE	Abs	-53	#FFFFFFCB	-	54	128	1025	1047		
fnrtn1	Abs	23108	#00005A44	-	984	832	879			
getd10	Abs	22118	#00005666	-	139	137				
getdat	Abs	22098	#00005652	-	124	677	752	854	856	967
getn20	Abs	22166	#00005696	-	181	171				
getn30	Abs	22189	#000056AD	-	205	203				
getnum	Abs	22128	#00005670	-	146					
gets10	Abs	22276	#00005704	-	251	246				
gets20	Abs	22312	#00005728	-	274	255				

gets30	Abs	22336	#00005740	-	283	280			
gets40	Abs	22340	#00005744	-	284	282			
gets50	Abs	22346	#0000574A	-	287	268			
getstr	Abs	22217	#000056C9	-	231	145			
ivarg	Abs	22669	#0000588D	-	623	229	496	595	
nxtstm	Abs	23214	#00005AAE	-	1049	1027			
pTRANS	Ext			-	998				
rtnd10	Abs	22910	#0000597E	-	823	820			
rtnd20	Abs	22922	#0000598A	-	831	829			
rtndat	Abs	22838	#00005936	-	775				
sDMY	Abs	0	#00000000	-	55	136	138	202	782
sDOW\$	Abs	1	#00000001	-	56	902	925	977	
send1	Abs	22686	#0000589E	-	647	645			
send2	Abs	22676	#00005894	-	643	700	702	706	710
stkchr	Abs	22693	#000058A5	-	650	704	708		
verd10	Abs	22456	#000057B8	-	418	403			
verd15	Abs	22460	#000057BC	-	425	416			
verd50	Abs	22504	#000057E8	-	499	395			
verd70	Abs	22527	#000057FF	-	523	418	511		
verd80	Abs	22549	#00005815	-	536	531			
verd99	Abs	22575	#0000582F	-	550	417	493	495	
verdat	Abs	22393	#00005779	-	368	227			

Input Parameters

Source file name is date.a

Listing file name is date.l

Object file name is date.o

Flags set on command line
None

Errors

None


```
1          TITLE JPC LEX: MODULE DES
2
3          * *****
4          * LCD Drawing (dessin) module
5          *
6          * Keyword(s):
7          *   INVERSE, PAINT, CONTRAST
8          * Author(s):
9          *   Laurent Istria, Jean-Jacques Moreau
10         * History:
11         *   Initially published in JPC #24
12         *   INVERSE and PAINT rewritten as some stage
13         *
14         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
15         * *****
16
17 05AB5          ABS #5AB5
18
19         * *****
20         * INVERSE entry
21         * *****
22 05AB5 00000    REL(5) =STDD
23 05ABA 00000    REL(5) =INVERp
24 05ABF 14A      =INVERe A=DAT0 B
25 05AC2 8F00000  GOSBVL =EOLXCK
26 05AC9 D0      A=0 A
27 05ACB D2      C=0 A
28 05ACD 3138    LCHEX 83
29 05AD1 432     GOC o5B17
30 05AD4 8F00000  GOSBVL =EXPEXC
31 05ADB D2      C=0 A
32 05ADD 3138    LCHEX 83
33 05AE1 7760    GOSUB o5B6E
34 05AE5 101     R1=A
35 05AE8 D2      C=0 A
36 05AEA 3138    LCHEX 83
37 05AEE 7A50    GOSUB o5B6E
38 05AF2 119     C=R1
39 05AF5 8B6     o5B17 ?C<A A
40 05AF8 C6      GOYES o5B86
41 05AFA E2      C=C-A A
42 05AFC D7      D=C A
43 05AFE 7960    GOSUB o5B8D
44 05B02 14A     o5B24 A=DAT0 B
45 05B05 BEC     A=-A-1 B
46 05B08 148     DAT0=A B
47 05B0B 132     AD0EX
48 05B0E 3400000 LC(5) (=DD3END)-2
49 05B15 8A2     ?C=A A
50 05B18 71      GOYES o5B51
51 05B1A 3400000 LC(5) (=DD2END)-2
52 05B21 8A2     ?C=A A
53 05B24 51      GOYES o5B5B
54 05B26 130     D0=A
55 05B29 161     D0=D0+ 2
56 05B2C 531     GONC o5B62
57 05B2F 1B00000 o5B51 D0=(5) =DD2ST
```

```
58 05B36 490          GOC 05B62
59 05B39 1B00000     05B5B D0=(5) =DD1ST
60 05B40 CF          05B62 D=D-1 A
61 05B42 5FB          GONC 05B24
62 05B45 8D00000     GOVLNG =NXTSTM
63
64 05B4C 108          05B6E R0=C
65 05B4F 8F00000     GOSBVL =RNDAHX
66 05B56 5D0          GONC 05B86
67 05B59 17F          D1=D1+ 16
68 05B5C 118          C=R0
69 05B5F 8BA          ?C>=A A
70 05B62 00           RTNYES
71 05B64 8D00000     05B86 GOVLNG =ARGERR
72
73 05B6B D2           05B8D C=0 A
74 05B6D 31E2         LCHEX 2E
75 05B71 8B2          ?C>A A
76 05B74 B2           GOYES 05BC1
77 05B76 31E5         LCHEX 5E
78 05B7A 8B2          ?C>A A
79 05B7D 21           GOYES 05BB1
80 05B7F 31E5         LCHEX 5E
81 05B83 EA           A=A-C A
82 05B85 3400000     LC(5) =DD1ST
83 05B8C 591          GONC 05BC8
84 05B8F 31E2         05BB1 LCHEX 2E
85 05B93 EA           A=A-C A
86 05B95 3400000     LC(5) =DD2ST
87 05B9C 590          GONC 05BC8
88 05B9F 3400000     05BC1 LC(5) =DD3ST
89 05BA6 C4           05BC8 A=A+A A
90 05BA8 CA           A=A+C A
91 05BAA 130          D0=A
92 05BAD 01           RTN
93
94                    * *****
95                    * PAINT entry
96                    * *****
97 05BAF 88823         NIBHEX 88823
98 05BB4 136          =PAINTe CD0EX
99 05BB7 06           RSTK=C
100 05BB9 840          ST=0 0
101 05BBC 80DF         P=C 15
102 05BC0 892          ?P= 2
103 05BC3 50           GOYES 05BEA
104 05BC5 850          ST=1 0
105 05BC8 20           05BEA P= 0
106 05BCA D2           C=0 A
107 05BCC 3170         LCHEX 07
108 05BD0 787F         GOSUB 05B6E
109 05BD4 102          R2=A
110 05BD7 D2           C=0 A
111 05BD9 3138         LCHEX 83
112 05BDD 7B6F         GOSUB 05B6E
113 05BE1 101          R1=A
114 05BE4 860          ?ST=0 0
```

```
115 05BE7 F0          GOYES o5C18
116 05BE9 8F00000    GOSEVL =RNDAHX
117 05BF0 17F        D1=D1+ 16
118 05BF3 100        R0=A
119 05BF6 112        o5C18 A=R2
120 05BF9 3110       LCHEX 01
121 05BFD 6600       GOTO o5C26
122 05C01 A66        o5C23 C=C+C B
123 05C04 CC         o5C26 A=A-1 A
124 05C06 5AF        GONC o5C23
125 05C09 10A        R2=C
126 05C0C 111        A=R1
127 05C0F 785F      GOSUB o5B8D
128 05C13 14A        A=DAT0 B
129 05C16 103        R3=A
130 05C19 860        ?ST=0 0
131 05C1C 22         GOYES o5C60
132 05C1E 118        C=R0
133 05C21 8AA        ?C=0 A
134 05C24 D0         GOYES o5C53
135 05C26 11A        C=R2
136 05C29 0E6E      A=A!C B
137 05C2D 6D00       GOTO o5C5D
138 05C31 11A        o5C53 C=R2
139 05C34 BEE        C=-C-1 B
140 05C37 0E66      A=A&C B
141 05C3B 148        o5C5D DAT0=A B
142 05C3E 113        o5C60 A=R3
143 05C41 11A        C=R2
144 05C44 0E66      A=A&C B
145 05C48 AF2        C=0 W
146 05C4B 968        ?A=0 B
147 05C4E 90         GOYES o5C79
148 05C50 2E         P= 14
149 05C52 301        LCHEX 1
150 05C55 20         P= 0
151 05C57 AFA        o5C79 A=C W
152 05C5A 07         C=RSTK
153 05C5C 134        D0=C
154 05C5F AF6        C=A W
155 05C62 8D00000    GOVLNG =FNRTN1
156
157
158
159
160 05C69 00          NIBHEX 00
161 05C6B 136        =CONTRe CD0EX      Sauvegarde...
162 05C6E 108        R0=C
163 05C71 1B00000    D0=(5) =DCONTR    Adresse oE est stocke
164
165 05C78 D0          *
166 05C7A 15A0        A=0 A
167 05C7E 8F00000    A=DAT0 1          RAcupEre cette valeur
168 05C85 120        GOSEVL =HDFLT     Qui est convertie en dAcimal
169 05C88 118        AR0EX            Sauvegarde et fin...
170 05C8B 8D00000    C=R0
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Saturn Assembler JPC LEX: MODULE DES
Ver. 1.56, 12/20/89

Tue May 23 09:21:17 2006
des.a Page 4

172 05C92

END

ARGERR	Ext		-	71				
=CONTRe	Abs	23659	#00005C6B	-	161			
DCONTR	Ext			-	163			
DD1ST	Ext			-	59	82		
DD2END	Ext			-	51			
DD2ST	Ext			-	57	86		
DD3END	Ext			-	48			
DD3ST	Ext			-	88			
EOLXCK	Ext			-	25			
EXPEXC	Ext			-	30			
FNRTN1	Ext			-	155			
FNRTN2	Ext			-	170			
HDFLT	Ext			-	167			
=INVERe	Abs	23231	#00005ABF	-	24			
INVERp	Ext			-	23			
NXTSTM	Ext			-	62			
=PAINTe	Abs	23476	#00005BB4	-	98			
RNDAHx	Ext			-	65	116		
STDd	Ext			-	22			
o5B17	Abs	23285	#00005AF5	-	39	29		
o5B24	Abs	23298	#00005B02	-	44	61		
o5B51	Abs	23343	#00005B2F	-	57	50		
o5B5B	Abs	23353	#00005B39	-	59	53		
o5B62	Abs	23360	#00005B40	-	60	56	58	
o5B6E	Abs	23372	#00005B4C	-	64	33	37	108 112
o5B86	Abs	23396	#00005B64	-	71	40	66	
o5B8D	Abs	23403	#00005B6B	-	73	43	127	
o5BB1	Abs	23439	#00005B8F	-	84	79		
o5BC1	Abs	23455	#00005B9F	-	88	76		
o5BC8	Abs	23462	#00005BA6	-	89	83	87	
o5BEA	Abs	23496	#00005BC8	-	105	103		
o5C18	Abs	23542	#00005BF6	-	119	115		
o5C23	Abs	23553	#00005C01	-	122	124		
o5C26	Abs	23556	#00005C04	-	123	121		
o5C53	Abs	23601	#00005C31	-	138	134		
o5C5D	Abs	23611	#00005C3B	-	141	137		
o5C60	Abs	23614	#00005C3E	-	142	131		
o5C79	Abs	23639	#00005C57	-	151	147		

Input Parameters

Source file name is des.a

Listing file name is des.l

Object file name is des.o

Flags set on command line
None

Errors

None

```

1          TITLE JPC LEX: MODULE DESAL
2
3          * *****
4          * Desal module
5          *
6          * Keyword(s):
7          *   HTA$, ADBUF$, ASC$, ATH$, RED$
8          * Author(s):
9          *   Pierre David, Michel Martinet
10         * History:
11         *   Initially published in JPC #27
12         *   HTA$, ATH$ expanded to accept an optionnal 2nd argument
13         *   R. Rosenbaum, rev X: rewrite of RED$ function
14         *
15         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
16         * *****
17
18 05C92          ABS      #5C92
19
20         * *****
21         * HTA$ entry
22         * *****
23 05C92 8412          NIBHEX 8412
24 05C96 7F81          =HTAe  GOSUB veropt
25 05C9A 8F00000      GOSBVL =POP1S  Teste le paramÈtre de la fonction
26                                     *   et renvoie la longueur de la chaÑne
27                                     *   alpha en quartets dans A(A)
28 05CA1 AF2          C=0      W      Division de la longueur
29 05CA4 D6          C=A      A      de la chaÑne
30 05CA6 822          SB=0
31 05CA9 81E          CSRB      par
32 05CAC 81E          CSRB      4: deux dÅplacment de 1 bit
33                                     *   vers la droite
34 05CAF 832          ?SB=0      paire?
35 05CB2 50          GOYES HTA10  OK
36 05CB4 5D5          GONC Argerr Si le nombre est impair: erreur
37 05CB7 D7          HTA10 D=C   A      sauvegarde de N/4 dans D(A) qui
38                                     *   servira de compteur
39 05CB9 136          CD0EX      Sauvegarde du compteur programme (PC)
40 05CBC 06          RSTK=C     dans RSTK
41 05CBE 137          CD1EX      Addition de la longueur de
42 05CC1 C2          C=C+A   A      la chaÑne È D1
43 05CC3 134          D0=C      et on place D0 et
44 05CC6 135          D1=C      D1 sur la fin de la chaÑne
45 05CC9 DB          C=D      A      recupere N/4
46 05CCB C6          C=C+C   A      Multiplication par 2 pour
47 05CCD BF2          CSL      W      reconstitution de l'entÅte
48 05CD0 BF2          CSL      W      de la nouvelle
49 05CD3 30F          LC(1)  15     chaÑne
50 05CD6 109          R1=C      et sauvegarde dans le registre R1
51
52 05CD9 CF          Boucle D=D-1  A      On commence la boucle par dÅcrÅmenter D
53 05CDB 442          GOC      HTAFIN S'il ne reste plus de caractÈres
54                                     *   on branche en HTAFIN.
55 05CDE 183          D0=D0- 4      Autrement on dÅcrÅmente D0 de 4 pour
56 05CE1 15A3        A=DAT0 4      lire la valeur HEXA
57 05CE5 860          ?ST=0  0

```

```

58 05CE8 90          GOYES HTA20
59 05CEA 8F00000    GOSBVL =SWPBYT
60 05CF1 21          HTA20 P= 1      P= #-1 chr È convertir
61 05CF3 7220       GOSUB CONVAH    Voir encart
62 05CF7 1C1        D1=D1- 2      On dÅcrÅmente D1 pour
63 05CFA 149        DAT1=A B      placer le caractÈre ASCII
64 05CFD 5BD        GONC Boucle   Branchement inconditionnel
65
66 05D00 07          HTAFIN C=RSTK   Fin: restitution du
67 05D02 134        D0=C         compteur programme sauvegardÅ
68 *
69 05D05 1CF        D1=D1- 16    DÅcrÅmentation de D1 pour placer
70 05D08 119        C=R1         l'entÅte de la nouvelle chaÑne
71 *
72 05D0B 8D00000    GOVLNG =FNRTN4 Retour È BASIC
73
74 05D12 8D00000    Argerr GOVLNG =ARGERR
75
76
77 *****
78 *****
79 *
80 * CONVAH : CONvertit de l'Ascii en Hexa
81 *
82 * Objet: convertir une expression ASCII en HEXA
83 *
84 * EntrÅe:
85 * P= Nombre de caractÈre È convertir moins 1 (de 0 È 7)
86 * HEX
87 * A: valeurs HEXA correspondant au caractÈres ASCII
88 * Ex: 313445
89 * ----
90 * E 4 1 avec P= 2 en entrÅe.
91 *
92 * Sortie:
93 * P=0
94 * HEX
95 * Carry=0
96 * A(W) = B(W) = valeur HEXA
97 *
98 * Utilise: A(W), B(W), C(S), C(A), P
99 *
100 * Niveaux de pile des retours: 2
101 *
102 *****
103 *****
104
105 05D19 80CF       CONVAH C=P 15   Sauvegarde du nombre de caractÈres
106 *
107 05D1D 20          P= 0
108 05D1F 8F00000    BCL1 GOSBVL =DRANGE A(B) ? compris entre 30 et 39
109 05D26 5F1        GONC CVH10     Si oui: CVH10
110 05D29 8F00000    GOSBVL =CONVUC Convertit A(B) en majuscule
111 05D30 331464     LCASC 'FA'     Bornes test pour l'appel de RANGE
112 05D36 8F00000    GOSBVL =RANGE  VÅrifie si A(B) est entre A et F
113 05D3D 44D        GOC Argerr     Si non: ERREUR
114 05D40 309        LC(1) 9        On additionne 9 au quartet de poids

```



```

115 05D43 A0A          A=A+C  P      faible de A(B)
116 05D46 BF1          CVH10  BSL    W      Sauvegarde
117 05D49 A88          B=A    P      dans B(0)
118 05D4C BF4          ASR    W      On passe
119 05D4F BF4          ASR    W      au caractÉre suivant
120 05D52 A4E          C=C-1  S      C(S) = compteur
121 05D55 59C          GONC   BCL1   B.E.T.
122 05D58 AF4          A=B    W      A = B pour la sortie de routine
123 05D5B 03          RTNCC           Retour avec Carry = 0
124
125
126
127          * *****
128          * ADBUF$ entry
129          * *****
129 05D5D 411          NIBHEX 411   La fonction n'admet qu'un seul
130          *                               paramÉtre alphanumÁrique
131 05D60 8F00000      =ADBUFe GOSBVL =REVPOP  Teste le paramÉtre de la fonction et
132          *                               renvoie la longueur de la chaÑne alpha
133          *                               en quartets dans A(A). La chaÑne est
134          *                               renversÁe sur la Math Stack.
135 05D67 D2          C=0    A      ) On charge 6
136 05D69 306          LC(1)  6      ) dans C(A) pour
137 05D6C 8A6          ?A#C  A      ) tester si la longueur de la chaÑne
138          *                               est bonne.
139 05D6F 3A          GOYES  Argerr  Si elle est mauvaise: ERREUR.
140 05D71 15B5        A=DAT1 6      On charge la longueur de la chaÑne
141          *                               dans A(5-0)
142 05D75 175          D1=D1+ 6     D1 ^ maintenant la fin de la chaÑne.
143 05D78 137          CD1EX           Sauvegarde de D1
144 05D7B 06          RSTK=C           dans RSTK.
145 05D7D 22          P=      2     P= nombre de caractÉres de A(W)
146          *                               È convertir en HEXA.
147 05D7F 769F        GOSUB  CONVAH  Conversion !
148 05D83 D6          C=A    A      C(X) = valeur HEXA de la chaÑne
149          *                               introduite
150 05D85 8F00000      GOSBVL =IOFNDO Recherche du Buffer.
151 05D8C 581          GONC   FIN    Si Carry=0 :
152          *                               le buffer n'existe pas: FIN
153 05D8F 1C6          D1=D1- 7     On rÁajuste l'adresse pour pointer
154          *                               le dÁbut du buffer.
155 05D92 133          AD1EX           On charge l'adresse du buffer dans A(A)
156 05D95 24          P=      4     PrÁparation (comme pour CONVAH)
157          *                               pour l'appel de HEXASC.
158 05D97 AC2          C=0    S      ) La routine HEXASC demande que
159          *                               ) le nombre de quartets È
160 05D9A 80FF          CPEX   15     ) convertir soit dans C(S).
161 05D9E 8F00000      GOSBVL =HEXASC Conversion de l'adresse en ASCII.
162
163 05DA5 07          FIN     C=RSTK           ) Restitution
164 05DA7 135          D1=C           ) de D1.
165 05DAA 590          GONC   Adh    Si l'on vient de "GONC FIN"
166          *                               on branche en Adh.
167 05DAD 1C9          D1=D1- 10    Autrement on place le paramÉtre
168 05DB0 1599        DAT1=A 10     de retour sur la Math Stack.
169 05DB4 109          Adh     R1=C           Indispensable avant l'appel
170          *                               de la routine ADHEAD
171 05DB7 850          ST=1  0      On veut un retour aprÉs l'appel

```

```
172 * de ADHEAD.
173 05DBA 8F00000 GOSBVL =D=AVMS D(A) = AVMEMS
174 05DC1 8F00000 GOSBVL =ADHEAD Construction de l'en-tête de
175 * la nouvelle chaîne.
176 05DC8 8F00000 GOSBVL =REV$ On renverse le tout
177 05DCF 8D00000 GOVLNG =EXPR et retour È BASIC.
178
179
180 * *****
181 * ASC$ entry
182 * *****
183 05DD6 411 NIBHEX 411 Un paramètre alpha.
184 05DD9 137 =ASce CD1EX )
185 05DDC 135 D1=C ) Sauvegarde de D1
186 05DDF 06 RSTK=C ) dans RSTK. (opcode=1oct)
187 05DE1 8F00000 GOSBVL =POP1S
188 05DE8 AD1 B=0 M ) Division de la longueur
189 05DEB D8 B=A A ) de la chaîne par
190 05DED 81D BSRB ) 2. B=compteur de CHR.
191 05DF0 31E2 LCASC '.'
192 05DF4 0A ST=C ST(B)='.'
193 05DF6 3302E7 LCASC '~ '
194 05DFC D7 D=C A D(3-0)='~ '
195
196 05DFE CD BOUCLE B=B-1 A Reste-t-il des CHR ?
197 05E00 4C1 GOC SORTIE Si non: SORTIE.
198 05E03 14B A=DAT1 B A(B)=CHR È tester.
199 05E06 DB C=D A C(3-0)='~ '
200 05E08 8F00000 GOSBVL =RANGE 31<CHR<127 ?
201 05E0F 570 GONC ^D1 Si oui: B.E.T.
202 05E12 09 C=ST Autrement C(B)='.'
203 05E14 14D DAT1=C B Le CHR devient '.'
204 05E17 171 ^D1 D1=D1+ 2 Ajuste Carry.
205 05E1A 53E GONC BOUCLE B.E.T.
206
207 05E1D 07 SORTIE C=RSTK Restitution de
208 05E1F 135 D1=C D1.
209 05E22 8D00000 GOVLNG =EXPR et retour È BASIC.
210
211
212 * *****
213 * verifie si le parametre optionnel
214 * de HTA$ et ATH$ est #0
215 * *****
216 05E29 840 veropt ST=0 0
217 05E2C A4E C=C-1 S
218 05E2F A4E C=C-1 S
219 05E32 400 RTNC
220 05E35 8F00000 GOSBVL =POP1R
221 05E3C 04 SETHEX
222 05E3E 17F D1=D1+ 16
223 05E41 978 ?A=0 W
224 05E44 00 RTNYES
225 05E46 850 ST=1 0
226 05E49 01 RTN
227
228 * *****
```

```

229          * ATH$ entry
230          * *****
231 05E4B 8412          NIBHEX 8412
232 05E4F 76DF        =ATHe  GOSUB veropt
233 05E53 8F00000     GOSBVL =POP1S
234 05E5A 136         CD0EX          Sauvegarde de la valeur de
235 05E5D 06          RSTK=C          D0 dans RSTK pour rendre le pointeur D0
236          *          disponible
237 05E5F 137         CD1EX          On place
238 05E62 135         D1=C          le pointeur D0 È la
239 05E65 134         D0=C          valeur de D1
240 05E68 8F00000     GOSBVL =D=AVMS D(A) = AVMEMS
241 05E6F 1CF         D1=D1- 16      D1 pointe maintenant le dÅbut
242          *          du paramÈtre de ATH$
243 05E72 137         CD1EX          Calcul de la longueur
244 05E75 E2          C=C-A  A      de la future chaÑne
245 05E77 8B3         ?C<D  A      comparaison avec AVMEMS
246          *          pour savoir s'il y a
247 05E7A 35          GOYES Memerr  suffisamment de place en mÅmoire.
248          *          Si non: "In. Memory"
249 05E7C 135         D1=C          D1 ^ maintenant le futur paramÈtre
250 05E7F 06          RSTK=C          Sauvegarde dans RSTK
251 05E81 D6          C=A  A      Multiplication de la longueur
252 05E83 C6          C=C+C  A      de la chaÑne par 2 (N+N)
253 05E85 BF2        CSL  W      et construction de l'en-tÅte
254 05E88 BF2        CSL  W      de la
255 05E8B 30F        LC(1) 15      future chaÑne
256 05E8E 1557       DAT1=C  W      et mise en place sur la Math Stack
257 05E92 17F        D1=D1+ 16     D1 pointe maintenant la fin de la
258          *          future chaÑne
259 05E95 AF2        C=0  W      Mise È 0 pour permettre des dÅcalages
260          *          et des opÅrations dans C(S)
261 05E98 D6          C=A  A      Division de la longueur de la chaÑne
262 05E9A 81E        CSRB          par 2 pour initialiser le
263 05E9D D7          D=C  A      compteur de boucle D(A)
264 05E9F A4E        C=C-1  S      C(S)= 15
265
266 05EA2 CF          ATH10  D=D-1  A      Reste-t-il encore des caractÈres
267          *          È tester ?
268 05EA4 4F2        GOC  Fin      Non: branchement au label "Fin"
269 05EA7 14A        A=DAT0  B      A(B) = caractÈre ASCII È transformer
270          *          en HEXA
271 05EAA BCA        C=-C  S      Au retour de la routine HEXASC
272          *          C(S) = 15 . A l'appel de cette routine
273 05EAD 8F00000     GOSBVL =HEXASC C(S) = Nb. de quartets È transformer:
274          *          #-1
275 05EB4 860         ?ST=0  0
276 05EB7 90         GOYES ATH20
277 05EB9 8F00000     GOSBVL =SWPBYT
278 05EC0 1593       ATH20  DAT1=A  4      On place le rÅsultat de la
279          *          transformation sur la M.S.
280 05EC4 161         D0=D0+ 2      IncrÅmentation des
281 05EC7 173         D1=D1+ 4      compteurs D0 et D1
282 05ECA 57D        GONC  ATH10     Branchement inconditionnel (B.E.T.)
283
284 05ECD 8D00000     Memerr  GOVLNG =MEMERR
285

```

```

286 05ED4 07      Fin      C=RSTK      Restitution
287 05ED6 135     D1=C      des compteurs
288 05ED9 07      C=RSTK      D0 et
289 05EDB 134     D0=C      D1
290 05EDE 8D00000 GOVLNG =EXPR et retour È BASIC.
291
292
293              * *****
294              * RED$ entry
295              * *****
296 05EE5 411     NIBHEX 411
297 05EE8      =REDe
298              ***** changes
299              *** rev D
300              *      GOSUB  CNTBL  Suppression des blancs en d but
301              *      de cha ne.
302              *      GOSUB  ENTETE  Construction de l'en-t te de
303              *      la cha ne avant inversion.
304              *      GOSUB  CNTBL  Suppression des blancs en fin de cha ne.
305              *      GOC      FiN    On branche en: FiN
306              *
307              *CNTBL  GOSBVL =REVPOP "Pop" et renverse la cha ne.
308              *      B=0      M      ) Division de la longueur de la cha ne
309              *      B=A      A      ) par 2 pour avoir un compteur de
310              *      BSRB     ) caract res dans B(A).
311              *      LCASC   ' '    On charge le CHR$(32) dans C(B).
312              *BLCOUT  B=B-1  A      Reste-t-il encore des caract res
313              *      dans la cha ne ?
314              *      GOC      CHNULL Si non: on branche en cha ne nulle !
315              *      A=DAT1  B      ) Le caract re suivant est-il
316              *      ?A#C    B      ) diff rent de CHR$(32) ?
317              *      RTNYES   Si oui: fin de routine.
318              *      D1=D1+ 2      Autrement on incr mente D1 pour pointer
319              *      de CHR$ suivant.
320              *      GONC     BLCOUT B.E.T.
321              *
322              *ENTETE  A=B      A      Long de la cha ne restante -1 ) en
323              *      A=A+1    A      long de la cha ne ) octets
324              *      A=A+A    A      Long de la cha ne en quartets.
325              *      ASL      W      A = .....111110 ) avec 11111 =
326              *      ASL      W      A = ....1111100 ) long de la cha ne
327              *      A=A-1    P      A = ...111110F
328              *      D1=D1- 16     D cr mentation de D1 pour
329              *      DAT1=A  W      placer l'en-t te de la nouvelle cha ne
330              *      RTN      Retour
331              *
332              *CHNULL  C=RSTK     Suppression d'un niveau de sous-
333              *      programme si l'on vient de CNTBL.
334              *FiN     GOSUB  ENTETE Construction de l'en-t te de
335              *      la nouvelle cha ne.
336              *      GOVLNG =EXPR  Retour È BASIC.
337              ***
338
339              *** rev X
340 05EE8 8F00000  GOSBVL =POP1S
341 05EEF 7A00     GOSUB  REDIT
342 05EF3 840      ST=0    0

```

```
343 05EF6 8D00000          GOVLNG =ADHEAD
344
345          *
346 05EFD 137          =REDIT CD1EX
347 05F00 135          D1=C
348 05F03 C2          C=C+A  A
349 05F05 D5          B=C    A
350 05F07 136          CD0EX
351 05F0A 8F00000      GOSBVL =CSLC5
352 05F11 D9          C=B    A
353 05F13 AD1         B=0    M
354 05F16 D8          B=A    A
355 05F18 81D         BSRB
356 05F1B 3102        LCHEX  20
357 05F1F 1C1         D1=D1- 2
358 05F22 171         CEN1   D1=D1+ 2
359 05F25 14B         CEN2   A=DAT1 B
360 05F28 CD          B=B-1  A
361 05F2A 470         GOC    CEN2X
362 05F2D 962         ?A=C  B
363 05F30 2F          GOYES  CEN1
364 05F32 E5         CEN2X  B=B+1  A
365 05F34 D4          A=B    A
366 05F36 8A8         ?A=0  A
367 05F39 54          GOYES  EMPTY
368 05F3B C5          B=B+B  A
369 05F3D 137        CD1EX
370 05F40 C9          C=C+B  A
371 05F42 134        D0=C
372 05F45 135        D1=C
373 05F48 109        R1=C
374 05F4B 3102        LCHEX  20
375 05F4F 81D        BSRB
376 05F52 181        CEN3   D0=D0- 2
377 05F55 14A        A=DAT0 B
378 05F58 CD          B=B-1  A
379 05F5A 470        GOC    CEN4X
380 05F5D 962        ?A=C  B
381 05F60 2F          GOYES  CEN3
382 05F62 E5         CEN4X  B=B+1  A
383 05F64 D4          A=B    A
384 05F66 161        D0=D0+ 2
385 05F69 181        CEN5   D0=D0- 2
386 05F6C 1C1        D1=D1- 2
387 05F6F 14E        CEN6   C=DAT0 B
388 05F72 14D        DAT1=C B
389 05F75 CD          B=B-1  A
390 05F77 8AD        ?B#0  A
391 05F7A FE          GOYES  CEN5
392 05F7C C4          A=A+A  A
393 05F7E 8F00000    EMPTY  GOSBVL =CSRC5
394 05F85 134        D0=C
395 05F88 01         RTN
396          ***** end of changes
397
398 05F8A          END
```

=ADBUFe	Abs	23904	#00005D60	-	131			
ADHEAD	Ext			-	174	343		
ARGERR	Ext			-	74			
=ASCe	Abs	24025	#00005DD9	-	184			
ATH10	Abs	24226	#00005EA2	-	266	282		
ATH20	Abs	24256	#00005EC0	-	278	276		
=ATHe	Abs	24143	#00005E4F	-	232			
Adh	Abs	23988	#00005DB4	-	169	165		
Argerr	Abs	23826	#00005D12	-	74	36	113	139
BCL1	Abs	23839	#00005D1F	-	108	121		
BOUCLE	Abs	24062	#00005DFE	-	196	205		
Boucle	Abs	23769	#00005CD9	-	52	64		
CEN1	Abs	24354	#00005F22	-	358	363		
CEN2	Abs	24357	#00005F25	-	359			
CEN2X	Abs	24370	#00005F32	-	364	361		
CEN3	Abs	24402	#00005F52	-	376	381		
CEN4X	Abs	24418	#00005F62	-	382	379		
CEN5	Abs	24425	#00005F69	-	385	391		
CEN6	Abs	24431	#00005F6F	-	387			
CONVAH	Abs	23833	#00005D19	-	105	61	147	
CONVUC	Ext			-	110			
CSLC5	Ext			-	351			
CSRC5	Ext			-	393			
CVH10	Abs	23878	#00005D46	-	116	109		
D=AVMS	Ext			-	173	240		
DRANGE	Ext			-	108			
EMPTY	Abs	24446	#00005F7E	-	393	367		
EXPR	Ext			-	177	209	290	
FIN	Abs	23973	#00005DA5	-	163	151		
FNRTN4	Ext			-	72			
Fin	Abs	24276	#00005ED4	-	286	268		
HEXASC	Ext			-	161	273		
HTA10	Abs	23735	#00005CB7	-	37	35		
HTA20	Abs	23793	#00005CF1	-	60	58		
HTAFIN	Abs	23808	#00005D00	-	66	53		
=HTAe	Abs	23702	#00005C96	-	24			
IOFNDO	Ext			-	150			
MEMERR	Ext			-	284			
Memerr	Abs	24269	#00005ECD	-	284	247		
POP1R	Ext			-	220			
POP1S	Ext			-	25	187	233	340
RANGE	Ext			-	112	200		
=REDIT	Abs	24317	#00005EFD	-	346	341		
=REDe	Abs	24296	#00005EE8	-	297			
REV\$	Ext			-	176			
REVPOP	Ext			-	131			
SORTIE	Abs	24093	#00005E1D	-	207	197		
SWPBYT	Ext			-	59	277		
^D1	Abs	24087	#00005E17	-	204	201		
veropt	Abs	24105	#00005E29	-	216	24	232	

Input Parameters

Source file name is desal.a

Listing file name is desal.l

Object file name is desal.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE DIV
2
3          * *****
4          * Division module
5          *
6          * Keyword(s):
7          *   PHI, LCM, GCD, FPRIM, NPRIM, PRIM
8          * Author(s):
9          *   Guy Toublanc
10         * History:
11         *   Published in JPC #48
12         *   R. Rosenbaum, rev X:
13         *     remaned PPCM, PGCD to LCM, GCD
14         *     bug fixes: PHI(0), PRIM(1), LCM(0,x)
15         *
16         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
17         * *****
18
19 05F8A          ABS      #5F8A
20
21         * *****
22         * PHI entry
23         * *****
24 05F8A 811          NIBHEX 811
25 05F8D 7E21        =PHIe  GOSUB  popln
26         *** change rev X:
27 05F91 978          ?A=0   W
28 05F94 A6          GOYES  RESA
29         *** end of change
30 05F96 100          R0=A
31 05F99 102          R2=A
32 05F9C AF2          C=0    W
33 05F9F 809          C+P+1
34 05FA2 972          ?A=C   W
35 05FA5 95          GOYES  RESA
36 05FA7 71A2        HIO    GOSUB  GTES
37 05FAB 100          R0A    R0=A
38 05FAE 976          ?A#C   W
39 05FB1 C0          GOYES  LO0
40 05FB3 109          R1=C
41 05FB6 7610        GOSUB  PHI
42 05FBA 534          GONC   RESA
43 05FBD 7312        LO0    GOSUB  div
44 05FC1 119          C=R1
45 05FC4 979          ?B=0   W
46 05FC7 4E          GOYES  R0A
47 05FC9 7300        GOSUB  PHI
48 05FCD 59D          GONC   HIO
49 05FD0 112          PHI    A=R2
50 05FD3 119          C=R1
51 05FD6 7AF1        GOSUB  div
52 05FDA 119          C=R1
53 05FDD A7E          C=C-1  W
54         *
55 05FE0 102          R2A    R2=A          ( A(W)*C(W) en decimal
56 05FE3 8F00000     dh     GOSBVL =DCHXW  (
57 05FEA 05          SETDEC  (
```



```
58 05FEC 112          A=R2          (
59 05FEF 8F00000     GOSBVL =MPY   (
60 05FF6 102         R2=A
61 05FF9 01         RTN
62
63 05FFB AFA        RESC  A=C    W
64 05FFE 8F00000     RESA  GOSBVL =FLOAT
65 06005 AF6        C=A    W
66 06008 132        AD0EX
67 0600B 1B00000     D0=(5) =ATNFLG
68 06012 1504       DAT0=A S
69 06016 130        D0=A
70 06019 8D00000     OUT   GOVLNG =FNRTN1
71
72
73          * *****
74          * PPCM/LCM entry
75          * *****
76 06020 88888888     NIBHEX 88888888882A  param: mini 2 maxi 10
      882A
77 0602C 852         =PPCMe ST=1  2      repere pour PPCM
78 0602F 511         GONC   pgd
79
80
81          * *****
82          * PGCD/GCD entry
83          * *****
84 06032 88888888     NIBHEX 88888888882A  dito ci-dessus
      882A
85 0603E 842         =PGCDe ST=0  2      repere pour PGCD
86 06041 7770       pgd   GOSUB  DCS      pope et teste le dernier
87          *          param.
88 06045 101         R1=A          sauve en R1
89 06048 7370       GOSUB  popln    pope et teste ke param.
90          *          precedent
91 0604C 590         GONC   pgc
92 0604F 10B        LOOPG  R3=C          sauve le PGCD ou le PPCM
93 06052 7960       GOSUB  popln    pope et tteste le param.
94          *          precedent
95 06056 7800       pgc   GOSUB  PGC      calcul du PGCD
96 0605A 94F        ?D#0  S          encore 1 param. a poper ?
97 0605D 2F         GOYES  LOOPG   oui retour a la boucle
98 0605F 5B9        GONC   RESC   non alors resultat
99
100 06062 119        PGC    C=R1          calcul PGCD des param.
101 06065 97E        ?C#0  W
102 06068 50         GOYES  DIF0
103 0606A AFE        ACEX   W
104 0606D 103        DIF0   R3=A
105 06070 10A        R2=C
106 06073 7A51       EUCL   GOSUB  R1C
107 06077 979        ?B=0  W
108 0607A 80         GOYES  PGCD
109 0607C 111        A=R1
110 0607F 53F        GONC   EUCL
111 06082 119        PGCD   C=R1          C(W)=PGCD
112 06085 862        ?ST=0  2          seulement calcul du PGCD ?
```

```

113 06088 00          RTNYES          oui alors retour
114 0608A 113        A=R3            non alors calcul du PPCM
115 0608D 7341       GOSUB  div
116 06091 100        R0=A
117 06094 112        A=R2
118 06097 7E50       GOSUB  E12
119 0609B AFE        ACEX  W
120 0609E 7231       GOSUB  div
121 060A2 118        C=R0
122 060A5 9F2        ?A<C  W
123 060A8 B0         GOYES  pop
124 060AA 753F       GOSUB  dh
125 060AE 101        R1=A
126 060B1 01         RTN
127 060B3 07         pop  C=RSTK
128 060B5 8D00000    ERR  GOVLNG =ARGERR
129
130 060BC AC7        DCS  D=C  S      nombre de param. -> D(S)
131 060BF 8F00000    pop1n GOSBVL =POP1R  pope le parametre
132 060C6 17F        D1=D1+ 16
133 060C9 948        ?A=0  S      pas "." avec le 1er param.?
134 060CC 50         GOYES  C11    oui alors test C11
135 060CE 850        ST=1  0      non alors repere de "."
136 060D1 32110     C11  LCHEX  011
137 060D6 9B6        ?A>C  X      N >= 10^12 ?
138 060D9 AD         GOYES  pop    oui alors erreur
139 060DB 8F00000    GOSBVL =SPLITA
140 060E2 8F00000    GOSBVL =CLRFRC
141 060E9 59C        GONC  pop
142 060EC AD4        A=B  M
143 060EF A4F        D=D-1  S     actualise nombre param.
144 060F2 8D00000    GOVLNG =RJUST conversion deci. flot.->dec
145
146 060F9 21         E12  P=  1      C(W)=10^12 en decimal
147 060FB AF2        C=0  W
148 060FE 80CC       C=P  12
149 06102 20         P=  0
150 06104 01         RTN
151 06106 AF2        CONST C=0  W
152 06109 302        LCHEX  2
153 0610C AF7        D=C  W      D(W)=2 en decimal
154 0610F 3101       LCHEX  10
155 06113 10C        R4=C      R4=10 en decimal
156 06116 01         RTN
157
158
159
160
161
162 06118 8812       NIBHEX 8812  param. : 1 mini 2 maxi
163 0611C            =FPRIME
164 0611C 851        ST=1  1      repere pour FPRIM
165 0611F 590        GONC  prim
166
167
168
169

```

```

170 06122 8822          NIBHEX 8822  param. : 2 mini 2 maxi
171 06126              =NPRIME
172 06126 841          ST=0  1      repere pour NPRIME
173 06129 840          prim  ST=0  0      pas de param. negatif
174 0612C 7C8F          GOSUB  DCS    pope et teste le parametre
175 06130 AF2          C=0    W
176 06133 870          ?ST=1 0      repere de param. negatif
177 06136 60           GOYES  LIM2    oui alors teste de ce param.
178 06138 7DBF          GOSUB  E12    non alors C(W)=10^12
179 0613C 10A          LIM2  R2=C    R2=0 ou 10^12
180 0613F 94B          ?D=0  S      1 seul parametre ?
181 06142 C0           GOYES  LIM1    oui alors 1 limite
182 06144 102          R2=A      sauve le 2e parametre
183 06147 840          ST=0  0
184 0614A 717F          GOSUB  popln  pope et teste le 1er param.
185 0614E 11A          LIM1  C=R2    C(W)=limite sup. ou inf.
186 06151 9FA          ?A<=C W    n <= limite ?
187 06154 50           GOYES  N<n
188 06156 850          ST=1  0
189 06159 AF2          N<n  C=0    W
190 0615C 302          LCHEX  2
191 0615F 870          ?ST=1 0
192 06162 A0           GOYES  DESC
193 06164 9FE          ?A>=C W
194 06167 50           GOYES  DESC
195 06169 AFA          A=C    W
196 0616C 100          DESC  R0=A
197 0616F 112          A=R2
198 06172 860          ?ST=0 0
199 06175 A0           GOYES  ASC
200 06177 9FE          ?A>=C W
201 0617A 50           GOYES  ASC
202 0617C 10A          R2=C
203 0617F 738F          ASC  GOSUB  CONST
204 06183 AF0          A=0    W
205 06186 103          R3=A
206 06189 118          C=R0
207 0618C 108          LOOPF R0=C
208 0618F 112          A=R2
209 06192 860          ?ST=0 0
210 06195 50           GOYES  CRES
211 06197 AFE          ACEX  W
212 0619A 9F2          CRES ?C>A W
213 0619D D2           GOYES  RESN
214 0619F 7DA0          GOSUB  NTES
215 061A3 976          ?A#C  W
216 061A6 01           GOYES  CONT
217 061A8 871          ?ST=1 1
218 061AB 22           GOYES  RESU
219 061AD 11B          C=R3
220 061B0 B76          C=C+1 W
221 061B3 10B          R3=C
222 061B6 118          CONT  C=R0
223 061B9 860          ?ST=0 0
224 061BC 80           GOYES  INCR
225 061BE A7E          C=C-1 W
226 061C1 5AC          GONC  LOOPF

```

```
227 061C4 B76      INCR   C=C+1  W
228 061C7 54C      GONC   LOOPF
229 061CA 113      RESN   A=R3
230 061CD 603E     RESU   GOTO   RESA   resultat
231
232 061D1 109      R1C    R1=C
233                *** changes
234                * rev D
235                *div   GOSBVL =IDIV
236                *      RTNCC
237                * rev X:
238 061D4 97A      div    ?C=0  W
239 061D7 90        GOYES  FIXGCD
240 061D9 8D00000  GOVLNG =IDIV
241 061E0 AFA      FIXGCD A=C   W
242 061E3 AF5      B=C    W
243 061E6 E6       C=C+1  A
244 061E8 109      R1=C
245 061EB CE       C=C-1  A
246 061ED 2F       P=     15
247 061EF 03      RTNCC
248                *** end of changes
249
250
251                * *****
252                * PRIM entry
253                * *****
254 061F1 8812      NIBHEX 8812
255 061F5          =PRIME
256 061F5 73CE     GOSUB  DCS
257 061F9 94B      ?D=0   S
258 061FC 52       GOYES  TVAL
259 061FE AF2      C=0    W
260 06201 32999   LCHEX  999
261 06206 9F6     ?A>C  W
262 06209 F2      GOYES  err
263
264                *** changes
265                * rev D
266                *      C=C+1  A
267                *      R1=C
268                *      R0=A
269                *      GOSUB  popln
270                *      C=R1
271                *      GOSUB  R2A
272                *      A=R0
273                *      A=A+C  W
274                *TVAL
275                * rev X
276 0620B 100      R0=A
277 0620E 7DAE     GOSUB  popln
278 06212 118     C=R0
279 06215 BF0     ASL   W
280 06218 BF0     ASL   W
281 0621B BF0     ASL   W
282 0621E A7A     A=A+C  W
283 06221          TVAL
```

```
284          ** PRIM(1) bug fix, added JFG March 05, taken from ver.Ex
285 06221 AF6          C=A W
286 06224 A7E          C=C-1 W      test if arg = 1
287 06227 97E          ?C#0 W
288 0622A 60          GOYES skip5
289 0622C 61DD         GOTO RESA
290 06230          skip5
291          ** end of PRIM(1) fix
292          *** end of changes
293 06230 100          R0=A
294 06233 97C          ?A#0 W
295 06236 60          GOYES VAL
296 06238 6C7E        err GOTO ERR
297 0623C 7C00        VAL GOSUB GTES
298 06240 976          ?A#C W
299 06243 50          GOYES resc
300 06245 AF2          C=0 W
301 06248 62BD        resc GOTO RESC
302
303 0624C 76BE        GTES GOSUB CONST
304 06250 AFB        NTES C=D W
305 06253 7441        GOSUB TES0
306 06257 B66          C=C+1 B
307 0625A 7D31        GOSUB TES0
308 0625E 7631        GOSUB TES2
309 06262 7231        GOSUB TES2
310 06266 7B21        GOSUB TES4
311 0626A 7A21        LOOP GOSUB TES2
312 0626E 7321        GOSUB TES4
313 06272 7221        GOSUB TES2
314 06276 7B11        GOSUB TES4
315 0627A 7411        GOSUB TES6
316 0627E 7611        GOSUB TES2
317 06282 7C01        GOSUB TES6
318 06286 7B01        GOSUB TES4
319 0628A 7A01        GOSUB TES2
320 0628E 7301        GOSUB TES4
321 06292 7CF0        GOSUB TES6
322 06296 78F0        GOSUB TES6
323 0629A 7AF0        GOSUB TES2
324 0629E 70F0        GOSUB TES6
325 062A2 7FE0        GOSUB TES4
326 062A6 7EE0        GOSUB TES2
327 062AA 74E0        GOSUB TES6
328 062AE 73E0        GOSUB TES4
329 062B2 7CD0        GOSUB TES6
330 062B6 75D0        GOSUB TES8
331 062BA 77D0        GOSUB TES4
332 062BE 76D0        GOSUB TES2
333 062C2 7FC0        GOSUB TES4
334 062C6 7EC0        GOSUB TES2
335 062CA 77C0        GOSUB TES4
336 062CE 7DB0        GOSUB TES8
337 062D2 7CB0        GOSUB TES6
338 062D6 7BB0        GOSUB TES4
339 062DA 74B0        GOSUB TES6
340 062DE 76B0        GOSUB TES2
```

```
341 062E2 7FA0          GOSUB  TES4
342 062E6 78A0          GOSUB  TES6
343 062EA 7AA0          GOSUB  TES2
344 062EE 70A0          GOSUB  TES6
345 062F2 7C90          GOSUB  TES6
346 062F6 7B90          GOSUB  TES4
347 062FA 7A90          GOSUB  TES2
348 062FE 7390          GOSUB  TES4
349 06302 7C80          GOSUB  TES6
350 06306 7E80          GOSUB  TES2
351 0630A 7480          GOSUB  TES6
352 0630E 7380          GOSUB  TES4
353 06312 7280          GOSUB  TES2
354 06316 7B70          GOSUB  TES4
355 0631A 7A70          GOSUB  TES2
356 0631E 7A60          GOSUB  TES10
357 06322 7270          GOSUB  TES2
358 06326 86C           ?ST=0  12
359 06329 83           GOYES  PL10
360 0632B 132          AD0EX
361 0632E 1B00000      D0=(5) =ATNFLG
362 06335 1524         A=DAT0  S
363 06339 948         ?A=0    S
364 0633C 22          GOYES  D0A
365 0633E 04          SETHEX
366 06340 B44         A=A+1   S
367 06343 481         GOC    setdec
368 06346 1544        DAT0=C  S
369 0634A 130         D0=A
370 0634D 20          P=      0
371 0634F 33F01E      LC(4)  #E10F
372 06355 8D00000     GOVLNG =BSERR
373
374 0635C 05          setdec SETDEC
375 0635E 130         D0A    D0=A
376 06361 AF0        PL10   A=0    W
377 06364 21         P=     1
378 06366 B04        A=A+1  P
379 06369 A72        C=C+A  W
380 0636C 110        A=R0
381 0636F 7E5E       GOSUB  R1C
382 06373 119        C=R1
383 06376 27         P=     7
384 06378 919        ?B=0   WP
385 0637B E0         GOYES  rnp
386 0637D 9F2        ?C>A  W
387 06380 60         GOYES  RESP
388 06382 67EE       GOTO   LOOP
389 06386 118        RESP   C=R0
390 06389 494        rnp    GOC    RNP
391 0638C A1B        TES10  C=C+D  WP    dernier diviseur + 10
392 0638F A1B        TES8   C=C+D  WP
393 06392 A1B        TES6   C=C+D  WP
394 06395 A1B        TES4   C=C+D  WP
395 06398 A1B        TES2   C=C+D  WP    dernier diviseur + 2
396 0639B 109        TES0   R1=C   sauve nouveau diviseur
397 0639E 110        A=R0
```

```
398 063A1 20          P=      0
399 063A3 9FA        HI      ?C>=A  W
400 063A6 A0          GOYES   LO
401 063A8 BF2          CSL     W
402 063AB 0C          P=P+1
403 063AD 55F          GONC    HI
404 063B0 B7A        LO      A=A-C  W
405 063B3 5CF          GONC    LO
406 063B6 A7A          A=A+C  W
407 063B9 0D          P=P-1
408 063BB 480          GOC     LO1
409 063BE BF6          CSR     W
410 063C1 5EE          GONC    LO
411 063C4 119        LO1     C=R1
412 063C7 27          P=      7
413 063C9 91C          ?A#0   WP
414 063CC 00          RTNYES
415 063CE 07          C=RSTK
416 063D0 119          C=R1
417 063D3 110        RNP     A=R0
418 063D6 01          RTN
419
420 063D8            END
```

ARGERR	Ext		-	128					
ASC	Abs	24959	#0000617F	-	203	199	201		
ATNFLG	Ext		-	67	361				
BSERR	Ext		-	372					
C11	Abs	24785	#000060D1	-	136	134			
CLRFRC	Ext		-	140					
CONST	Abs	24838	#00006106	-	151	203	303		
CONT	Abs	25014	#000061B6	-	222	216			
CRES	Abs	24986	#0000619A	-	212	210			
D0A	Abs	25438	#0000635E	-	375	364			
DCHXW	Ext		-	56					
DCS	Abs	24764	#000060BC	-	130	86	174	256	
DESC	Abs	24940	#0000616C	-	196	192	194		
DIF0	Abs	24685	#0000606D	-	104	102			
E12	Abs	24825	#000060F9	-	146	118	178		
ERR	Abs	24757	#000060B5	-	128	296			
EUCL	Abs	24691	#00006073	-	106	110			
FIXGCD	Abs	25056	#000061E0	-	241	239			
FLOAT	Ext		-	64					
FNRTN1	Ext		-	70					
=FPRIME	Abs	24860	#0000611C	-	163				
GTES	Abs	25164	#0000624C	-	303	36	297		
HI	Abs	25507	#000063A3	-	399	403			
HI0	Abs	24487	#00005FA7	-	36	48			
IDIV	Ext		-	240					
INCR	Abs	25028	#000061C4	-	227	224			
LIM1	Abs	24910	#0000614E	-	185	181			
LIM2	Abs	24892	#0000613C	-	179	177			
LO	Abs	25520	#000063B0	-	404	400	405	410	
LO0	Abs	24509	#00005FBD	-	43	39			
LO1	Abs	25540	#000063C4	-	411	408			
LOOP	Abs	25194	#0000626A	-	311	388			
LOOPF	Abs	24972	#0000618C	-	207	226	228		
LOOPG	Abs	24655	#0000604F	-	92	97			
MPY	Ext		-	59					
N<n	Abs	24921	#00006159	-	189	187			
=NPRIME	Abs	24870	#00006126	-	171				
NTES	Abs	25168	#00006250	-	304	214			
OUT	Abs	24601	#00006019	-	70				
PGC	Abs	24674	#00006062	-	100	95			
PGCD	Abs	24706	#00006082	-	111	108			
=PGCDe	Abs	24638	#0000603E	-	85				
PHI	Abs	24528	#00005FD0	-	49	41	47		
=PHIe	Abs	24461	#00005F8D	-	25				
PL10	Abs	25441	#00006361	-	376	359			
POP1R	Ext		-	131					
=PPCMe	Abs	24620	#0000602C	-	77				
=PRIME	Abs	25077	#000061F5	-	255				
R0A	Abs	24491	#00005FAB	-	37	46			
R1C	Abs	25041	#000061D1	-	232	106	381		
R2A	Abs	24544	#00005FE0	-	55				
RESA	Abs	24574	#00005FFE	-	64	28	35	42	230
RESC	Abs	24571	#00005FFB	-	63	98	301		289
RESN	Abs	25034	#000061CA	-	229	213			
RESP	Abs	25478	#00006386	-	389	387			
RESU	Abs	25037	#000061CD	-	230	218			
RJUST	Ext		-	144					

RNP	Abs	25555	#000063D3	-	417	390					
SPLITA	Ext			-	139						
TES0	Abs	25499	#0000639B	-	396	305	307				
TES10	Abs	25484	#0000638C	-	391	356					
TES2	Abs	25496	#00006398	-	395	308	309	311	313	316	319
					323	326	332	334	340	343	347
					350	353	355	357			
TES4	Abs	25493	#00006395	-	394	310	312	314	318	320	325
					328	331	333	335	338	341	346
					348	352	354				
TES6	Abs	25490	#00006392	-	393	315	317	321	322	324	327
					329	337	339	342	344	345	349
					351						
TES8	Abs	25487	#0000638F	-	392	330	336				
TVAL	Abs	25121	#00006221	-	283	258					
VAL	Abs	25148	#0000623C	-	297	295					
dh	Abs	24547	#00005FE3	-	56	124					
div	Abs	25044	#000061D4	-	238	43	51	115	120		
err	Abs	25144	#00006238	-	296	262					
pgc	Abs	24662	#00006056	-	95	91					
pgd	Abs	24641	#00006041	-	86	78					
pop	Abs	24755	#000060B3	-	127	123	138	141			
popln	Abs	24767	#000060BF	-	131	25	89	93	184	277	
prim	Abs	24873	#00006129	-	173	165					
resc	Abs	25160	#00006248	-	301	299					
rnp	Abs	25481	#00006389	-	390	385					
setdec	Abs	25436	#0000635C	-	374	367					
skip5	Abs	25136	#00006230	-	290	288					

Input Parameters

Source file name is div.a

Listing file name is div.l

Object file name is div.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE DRIVE
2
3          * *****
4          * LEX drive commands
5          *
6          * Keyword(s):
7          *   LXON, LXOFF
8          * Author(s):
9          *   Michel Martinet
10         * History:
11         *   Published in JPC #30
12         *   R. Rosenbaum, rev X:
13         *     remaned LEX file ON, LEX file OFF to LXON file, LXOFF file
14         *
15         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
16         * *****
17
18 063D8          ABS      #63D8
19
20         * *****
21         * LEX ON/OFF entries
22         * *****
23         *      REL(5) =LXOFFd      rev D
24         *      REL(5) =LEXp        rev D
25 063D8 00000    REL(5) =LXDRVd     rev X
26 063DD 00000    REL(5) =LXDRVp     rev X
27 063E2 7F50     =LXOFFe GOSUB POSD1
28 063E6 D2       C=0      A
29 063E8 A6E      C=C-1    B
30 063EB 8A2      ?A=C     A
31 063EE 94       GOYES    Nxtstm
32 063F0 3400000 LC(5)    =fLEX
33 063F7 8A6      ?A#C     A
34 063FA 44       GOYES    Eftype
35 063FC D2       C=0      A
36 063FE A6E      C=C-1    B
37 06401 4A2      GOC      Config
38
39         *      REL(5) =LXOND      rev D
40         *      CON(5) 0          rev D
41 06404 00000    REL(5) =LXDRVd     rev X
42 06409 00000    REL(5) =LXDRVp     rev X
43 0640E 7330     =LXONE  GOSUB POSD1
44 06412 3400000 LC(5)    =fLEX
45 06419 8A2      ?A=C     A
46 0641C B1       GOYES    Nxtstm
47 0641E D5       B=C      A
48 06420 D2       C=0      A
49 06422 A6E      C=C-1    B
50 06425 8A6      ?A#C     A
51 06428 61       GOYES    Eftype
52 0642A D9       C=B      A
53 0642C 15D3     Config  DAT1=C 4
54 06430 8F00000 GOSBVL  =LEXBF+
55 06437 8D00000 Nxtstm  GOVLNG  =NXTSTM
56
57 0643E 3100     Eftype  LC(2)  =eFTYPE
```

```
58 06442 435          GOC   Mferr
59
60 06445 8F00000  POSD1  GOSBVL =FILXQ^
61 0644C 490          GOC   POSD2
62 0644F 3100         LC(2) =eFnFND
63 06453 524          GONC  Mferr
64 06456 8F00000  POSD2  GOSBVL =FINDF
65 0645D 443          GOC   EfNfnd
66 06460 AC2          C=0   S
67 06463 B46          C=C+1 S
68 06466 9C3          ?D>C S
69 06469 14           GOYES Efaccs
70 0646B 7000         GOSUB POSD3
71 0646F 07           POSD3  C=RSTK
72 06471 DA           A=C A
73
74                   *** changed for sasm compatibility, JFG jan2006
75                   * LC(5) (POSD3)- (=JPCLEX)
76                   * enable one of the two lines below:
77                   * LC(5) (POSD3)-8      if ROM header included
78                   * LC(5) (POSD3)+#25    if ROM header not included
79                   ***
79 0647A EA           A=A-C A
80 0647C 137          CD1EX
81 0647F 135          D1=C
82 06482 8A2          ?C=A A
83 06485 81           GOYES Edrive
84 06487 17F          D1=D1+ 16
85 0648A D0           A=0 A
86 0648C 15B3         A=DAT1 4
87 06490 01           RTN
88
89 06492 3100         EfNfnd LC(2) =eFnFND
90 06496 8D00000  Mferr  GOVLNG =MFERR
91
92 0649D 33101E       Edrive  LC(4)  #E101
93 064A3 8D00000       GOVLNG =BSERR
94
95 064AA 3100         Efaccs LC(2) =eFACCS
96 064AE 47E          GOC   Mferr
97
98 064B1              END
```

BSERR	Ext		-	93			
Config	Abs	25644	#0000642C	-	53	37	
Edrive	Abs	25757	#0000649D	-	92	83	
Efnfnd	Abs	25746	#00006492	-	89	65	
Efaccs	Abs	25770	#000064AA	-	95	69	
Eftype	Abs	25662	#0000643E	-	57	34	51
FILXQ^	Ext		-	60			
FINDF	Ext		-	64			
LEXBF+	Ext		-	54			
LXDRVd	Ext		-	25	41		
LXDRVp	Ext		-	26	42		
=LXOFFe	Abs	25570	#000063E2	-	27		
=LXONE	Abs	25614	#0000640E	-	43		
MFERR	Ext		-	90			
Mferr	Abs	25750	#00006496	-	90	58	63 96
NXTSTM	Ext		-	55			
Nxtstm	Abs	25655	#00006437	-	55	31	46
POSD1	Abs	25669	#00006445	-	60	27	43
POSD2	Abs	25686	#00006456	-	64	61	
POSD3	Abs	25711	#0000646F	-	71	70	76
eFACCS	Ext		-	95			
eFTYPE	Ext		-	57			
eFnFND	Ext		-	62	89		
fLEX	Ext		-	32	44		

Input Parameters

Source file name is drive.a

Listing file name is drive.l

Object file name is drive.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE ENDUP
2
3          * *****
4          * ENDUP module
5          *
6          * Keyword(s):
7          *   ENDUP, EXECUTE, STARTUP$, ENDUP$
8          * Author(s):
9          *   Jean-Jacques Moreau
10         * History:
11         *   Published in JPC #31
12         *
13         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14         * *****
15
16 064B1          ABS      #64B1
17
18          bENDUP EQU      #0083E
19          bXEQ   EQU      #00802
20
21
22         * *****
23         * ENDUP entry
24         * *****
25 064B1 00000          REL(5) =STDD
26 064B6 00000          REL(5) =STRNGp
27 064BB          =ENDUPe
28 064BB 8F00000      GOSBVL =EXPEXC  Mets la ligne È ÅxÅcutter sur la pile
29 064C2 32E38          LC(3)  bENDUP
30 064C7 7A30          GOSUB  end007  Stocke la ligne BASIC ds bENDUP
31
32 064CB 8D00000      nxtstm GOVLNG =NXTSTM
33
34
35         *
36         * ExÅcute une ligne BASIC. EXECUTE fut Åcrit pour
37         * rendre programmable des fonctions non programmables.
38
39         * Exemple d'utilisation:
40         *   EDIT TOTO
41         *   10 DISP 'Running...'
42         *   20 EXECUTE "FREE:PORT(.01) @ PURGE ABC
43         *               CONTRAST 7 @ RUN TOTO,30"
44         *   30 DISP 'Done'
45         *
46
47         * *****
48         * EXECUTE entry
49         * *****
50 064D2 00000          REL(5) =STDD
51 064D7 00000          REL(5) =STRNGp
52 064DC          =EXECUe
53 064DC 8F00000      GOSBVL =EXPEXC  Mets la ligne sur la pile
54 064E3 32208          LC(3)  bXEQ
55 064E8 7910          GOSUB  end007  Stocke la dite ligne ds bXEQ
56 064EC 5ED          GONC   nxtstm  Va lÈ-bas si elle ne contient
57         *               aucune instruction
```

```
58 064EF 32208          LC(3)  bXEQ
59 064F4 8F00000       GOSBVL =I/OFND  Mets D1 @ dÃbut de la ligne
60 064FB 137           CD1EX
61 064FE 8D00000       GOVLNG =LINEP+  ExÃcute l'Ã
62
63 06505 10A          end007 R2=C      Sauvegarde l'ID du buffer de stockage
64 06508 8F00000       GOSBVL =XXHEAD  EnlÃve l'en tÃte de la chaÃne placÃe
65 *                   *                   sur la pile
66 0650F 31D0          LCHEX  0D
67 06513 1C1           D1=D1- 2
68 06516 14D          DAT1=C B      Ajoute 1 CR derriÃre la ligne, il faudra
69 06519 850           ST=1  0      se rappeler qu'elle n'est pas tokenizÃe
70 0651C 78C0         GOSUB  ahead  Ajoute un nouvel en tÃte
71 06520 8F00000       GOSBVL =REVPOP  Puis inverse l'ordre des caractÃres
72 *                   *                   de la chaÃne
73 06527 137           CD1EX          Pointeur de pile -> C(A)
74 0652A 108           R0=C          -> R0;
75 0652D C2            C=C+A  A
76 0652F 8F00000       GOSBVL =AVE=C
77 06536 D2            C=0    A
78 06538 E6            C=C+1  A
79 0653A E6            C=C+1  A      2 -> C(A);
80 0653C 8B6           ?A>C  A      La ligne est-elle composÃe d'autre chose
81 *                   *                   que d'un CR?
82 0653F E0            GOYES  end010
83 06541 11A           C=R2
84 06544 8F00000       GOSBVL =I/ODAL  DÃtruit le buffer dont l'ID est ds C(A)
85 0654B 03            RTNCC
86
87 0654D D8          end010 B=A    A      Longueur de la ligne -> B(A);
88 0654F D2            C=0    A
89 06551 310C         LC(2)  192     96 carateres. Lg max autorisÃe -> C(A);
90 06555 8BA           ?A<=C  A      La ligne n'est-elle pas trop longue?
91 06558 D0            GOYES  end040  Non;
92 0655A 3100         LC(2)  =eSTROV
93 0655E 8D00000       GOVLNG =MFERR  Si.
94 06565 101          end040 R1=A    A      Longueur de la ligne -> R1;
95 06568 11A           C=R2
96 0656B 8F00000       GOSBVL =I/OALL  Contracte ou agrandit le buffer pr qu'il
97 06572 490           GOC    end050  ait la taille spÃcifiÃe en A(A)
98 06575 8D00000       GOVLNG =MEMERR  Erreur s'il n'y a pas assez de place
99 0657C 110          end050 A=R0
100 0657F 130          D0=A
101 06582 119          C=R1
102 06585 8F00000       GOSBVL =MOVEU3  DÃplace la ligne de la pile ds le buffer
103 0658C 02            RTNSC
104
105
106 * *****
107 * STARTUP$ entry
108 * *****
109 0658E 00            NIBHEX 00
110 06590              =STARTe
111 06590 32000         LC(3)  =bSTART
112 06595 590           GONC   run      (B.E.T.)
113
114
```



```
115          * *****
116          * ENDUP$ entry
117          * *****
118 06598 00          NIBHEX 00
119 0659A          =ENDUse
120 0659A 32E38      LC(3) bENDUP
121 0659F 133      run AD1EX          Sauvegarde D1 ds R1
122 065A2 101          R1=A
123 065A5 8F00000    GOSBVL =I/OFND
124 065AC 137          CD1EX
125 065AF 136          CD0EX          D0 @ ce qui suit l'en-tÁte
126 065B2 D5          B=C A          Sauvegarde ce pointeur ds B(A)
127 065B4 8F00000    GOSBVL =D=AVMS Mets (AVMEMS) ds D(A)
128 065BB 111          A=R1
129 065BE 131          D1=A          D1 = (FORSTK)
130 065C1 5E1          GONC bf2s20 Va si la chaÑne contenue ds le buffer
131          *          a 1 longueur nulle
132 065C4 14A          bf2s10 A=DAT0 B Prends 1 char. ds le buffer
133 065C7 31D0          LCHEX 0D
134 065CB 962          ?A=C B En a-t-on atteint la fin?
135 065CE 21          GOYES bf2s20 Non
136 065D0 AE6          C=A B Oui
137 065D3 8F00000    GOSBVL =STKCHR Mets le car. sur la pile;
138          *          vÅrifie qu'il y a suf de place
139 065DA 161          D0=D0+ 2 Passe au char. suivant
140 065DD 56E          GONC bf2s10 (B.E.T.)
141
142 065E0 D9          bf2s20 C=B A
143 065E2 134          D0=C
144 065E5 840          ST=0 0
145 065E8 8D00000    adhead GOVLNG =ADHEAD Ajoute un en-tÁte a la chaÑne.
146
147 065EF          END
```

ADHEAD	Ext		-	145		
AVE=C	Ext		-	76		
D=AVMS	Ext		-	127		
=ENDUPe	Abs	25787	#000064BB	-	27	
=ENDUSe	Abs	26010	#0000659A	-	119	
=EXECUe	Abs	25820	#000064DC	-	52	
EXPEXC	Ext		-	28	53	
I/OALL	Ext		-	96		
I/ODAL	Ext		-	84		
I/OFND	Ext		-	59	123	
LINEP+	Ext		-	61		
MEMERR	Ext		-	98		
MFERR	Ext		-	93		
MOVEU3	Ext		-	102		
NXTSTM	Ext		-	32		
REVPOP	Ext		-	71		
=STARTe	Abs	26000	#00006590	-	110	
STDD	Ext		-	25	50	
STKCHR	Ext		-	137		
STRNGp	Ext		-	26	51	
XXHEAD	Ext		-	64		
adhead	Abs	26088	#000065E8	-	145	70
bENDUP	Abs	2110	#0000083E	-	18	29 120
bSTART	Ext		-	111		
bXEQ	Abs	2050	#00000802	-	19	54 58
bf2s10	Abs	26052	#000065C4	-	132	140
bf2s20	Abs	26080	#000065E0	-	142	130 135
eSTROV	Ext		-	92		
end007	Abs	25861	#00006505	-	63	30 55
end010	Abs	25933	#0000654D	-	87	82
end040	Abs	25957	#00006565	-	94	91
end050	Abs	25980	#0000657C	-	99	97
nxtstm	Abs	25803	#000064CB	-	32	56
run	Abs	26015	#0000659F	-	121	112

Input Parameters

Source file name is endup.a

Listing file name is endup.l

Object file name is endup.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE EXIT
2
3          * *****
4          * EXIT module
5          *
6          * Keyword(s):  EXIT
7          * Author(s):
8          *   Janick Taillandier
9          * History:
10         *   Published in JPC #30
11         *
12         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
13         * *****
14
15 065EF          ABS      #65EF
16
17         *
18         * *****
19         * Statement: EXIT loop index *
20         * *****
21         *
22         * NOTE: this version does not allow EXIT from keyboard
23         *
24         * FOR-NEXT stack structure:
25         *
26         * Low memory
27         *
28         *   5 nibs      Return address (D0)
29         *  16 nibs      Step value
30         *  16 nibs      Limit
31         *   2 nibs      ASCII letter or alpha-digit
32         *   2 nibs      00 or ASCII letter
33         *
34         * High memory
35         *
36
37         * *****
38         * EXIT entry
39         * *****
40 065EF 00000          REL(5) =EXITd
41 065F4 00000          REL(5) =EXITp
42
43         * the following code from INDXSN (#0897A)
44         * unsupported entry point
45
46 065F9 858          =EXITe ST=1  8          alpha-digit variable flag
47 065FC 15A3          A=DAT0  4          read in index variable from prgm
48 06600 3314A5          LCASC  'ZA'
49 06606 8F00000          GOSBVL =RANGE
50 0660D 23            P=      3
51 0660F 490            GOC    indx02
52 06612 848            ST=0   8          single-character alpha digit
53 06615 D1            B=0     A
54 06617 21            P=      1
55 06619 A98            indx02 B=A     WP
56 0661C 23            P=      3
57 0661E 1F00000          D1=(5) =FORSTK
```

```

58 06625 147          C=DAT1 A
59 06628 108          R0=C          save ptr to top of FOR-NEXT stack
60 0662B 174          D1=D1+ 5
61 0662E 143          A=DAT1 A          ptr to bottom of FOR-NEXT stack
62 06631 135          indx04 D1=C
63 06634 8BA          ?C>=A A          end of search?
64 06637 C1           GOYES  indx07
65 06639 17F          D1=D1+ 16
66 0663C 17F          D1=D1+ 16
67 0663F 174          D1=D1+ 5
68 06642 147          C=DAT1 A          read variable from stack
69 06645 911          ?B=C  WP          ok?
70 06648 D0           GOYES  indx08
71 0664A 173          D1=D1+ 4
72 0664D 137          CD1EX
73 06650 50E          GONC   indx04      B.E.T.
74 06653 0D          indx07 P=P-1       clear carry
75 06655 20          indx08 P= 0
76
77          * End of INDXSN code
78
79 06657 595          GONC   efwo
80 0665A 108          R0=C          save variable name in R0
81 0665D 173          D1=D1+ 4      D1 @ bottom of entry
82 06660 133          AD1EX        collapse all preceeding entries
83 06663 1F00000     D1=(5) =AVMEME .
84 0666A 141          DAT1=A A     .
85 0666D 174          D1=D1+ 5     (=FORSTK) - (=AVMEME)
86 06670 141          DAT1=A A     .
87
88 06673 1F00000     D1=(5) =PRGMEN set D[A] for TKSCN7
89 0667A 147          C=DAT1 A     .
90 0667D D7           D=C  A       .
91
92 0667F 23           P= 3
93 06681 878          ?ST=1 8      alpha digit variable ?
94 06684 40           GOYES  exit01
95 06686 21           P= 1
96 06688 80FF        exit01 CPEX  15
97 0668C 183          D0=D0- 4     for tXWORD and lex ID
98 0668F 183          exit10 D0=D0- 4 D0 @ length byte (pass token and length)
99          * reuse LINSKP (skip to t@ or tEOL)
100 06692 D0           A=0  A
101 06694 14A          A=DAT0 B     line length
102 06697 D8           B=A  A
103 06699 132          AD0EX
104 0669C C0           A=A+B A     points to EOL or @
105 0669E 130          D0=A
106 066A1 20           P= 0
107 066A3 3100         LC(2) =tNEXT . in buffer limits ?
108 066A7 8F00000     GOSBVL =TKSCN7
109 066AE 4D0          GOC   exit02  token found
110 066B1 3100         efwo  LC(2) =eFwONX
111 066B5 8D00000     GOVLNG =MFERR
112 066BC 80DF        exit02 P=C  15
113 066C0 110          A=R0        restore EXIT variable name
114 066C3 161          D0=D0+ 2    point to current NEXT variable

```

```
115 066C6 1561      C=DAT0 WP
116 066CA 916      ?A#C  WP
117 066CD 2C       GOYES  exit10  try next NEXT
118 066CF 183      D0=D0- 4
119                * reuse NXTST2 (#08A58)
120 066D2 D0       A=0    A
121 066D4 14A     A=DAT0 B      read stmt length
122 066D7 136     CD0EX
123 066DA CA      A=A+C  A
124 066DC 130     D0=A
125 066DF 8D00000 GOVLNG =RUNRTN
126
127 066E6        END
```

AVMEME	Ext		-	83	
EXITd	Ext		-	40	
=EXITe	Abs	26105 #000065F9	-	46	
EXITp	Ext		-	41	
FORSTK	Ext		-	57	
MFERR	Ext		-	111	
PRGMEN	Ext		-	88	
RANGE	Ext		-	49	
RUNRTN	Ext		-	125	
TKSCN7	Ext		-	108	
eFwoNX	Ext		-	110	
efwo	Abs	26289 #000066B1	-	110	79
exit01	Abs	26248 #00006688	-	96	94
exit02	Abs	26300 #000066BC	-	112	109
exit10	Abs	26255 #0000668F	-	98	117
indx02	Abs	26137 #00006619	-	55	51
indx04	Abs	26161 #00006631	-	62	73
indx07	Abs	26195 #00006653	-	74	64
indx08	Abs	26197 #00006655	-	75	70
tNEXT	Ext		-	107	

Input Parameters

Source file name is exit.a

Listing file name is exit.l

Object file name is exit.o

Flags set on command line
None

Errors

None


```
1          TITLE JPC LEX: MODULE FKEY
2
3          * *****
4          * FKEY module
5          *
6          * Keyword(s):
7          *   FKEY
8          * Author(s):
9          *   Jean-Pierre Bondu
10         * History:
11         *   Published in JPC #24
12         *
13         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14         * *****
15
16 066E6          ABS      #66E6
17
18         * *****
19         * FKEY entry
20         * *****
21 066E6 00000          REL(5) =STDD
22 066EB 00000          REL(5) =STRNGp
23 066F0 8F00000 =FKEYe GOSBVL =GTKYCD
24 066F7 D9           C=B   A
25 066F9 108          R0=C
26 066FC 34C1000      LC(5) 28
27 06703 D5           B=C   A
28 06705 3400000      LC(5) =KEYBUF
29 0670C DA           A=C   A
30 0670E E6           C=C+1 A
31 06710 E6           C=C+1 A
32 06712 808F          INTOFF
33 06716 8F00000      GOSBVL =MOVE*M
34 0671D 110          A=R0
35 06720 1B00000      D0=(5) =KEYBUF
36 06727 148          DAT0=A B
37 0672A 180          D0=D0- 1
38 0672D 1524         A=DAT0 S
39 06731 B44          A=A+1 S
40 06734 550          GONC   LBL
41 06737 A4C          A=A-1 S
42 0673A 1504          LBL    DAT0=A S
43 0673E 8080          INTON
44 06742 8D00000      GOVLNG =NXTSTM
45
46 06749          END
```

=FKEYe	Abs	26352 #000066F0	-	23	
GTKYCD	Ext		-	23	
KEYBUF	Ext		-	28	35
LBL	Abs	26426 #0000673A	-	42	40
MOVE*M	Ext		-	33	
NXTSTM	Ext		-	44	
STDD	Ext		-	21	
STRNGp	Ext		-	22	

Input Parameters

Source file name is fkey.a

Listing file name is fkey.l

Object file name is fkey.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE FIND
2
3          * *****
4          * Find module
5          *
6          * Keyword(s):
7          *   FIND
8          * Author(s):
9          *   Jean-Jacques Moreau, Janick Taillandier
10         * History:
11         *   Initially published in JPC #45
12         *   Numerous undocumented attempts to fix bugs
13         *   Bug fix by Rodger Rosenbaum, rev. X
14         *
15         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
16         * *****
17
18 06749          ABS      #6749
19
20         * *****
21         * FIND <chaine-alphanumerique>
22         * permet de trouver, dans un fichier Basic la
23         * premiere occurence de la chaine specifiee
24         * apres la ligne courante.
25         * Historique:
26         *   01/86 J.J. Moreau
27         *     conception et premiere version
28         *   12/86 J. Taillandier
29         *     - debogage
30         *     - suppression du message d'erreur dans
31         *       cette version "autonome"
32         *     - positionement du curseur sur la
33         *       chaine trouvee
34         * *****
35
36         * *****
37         * FIND entry
38         * *****
39 06749 00000          REL(5) =STRNGp
40 0674E 8F00000 =FINDe  GOSBVL =EXPEXC  depose "search" sur la MS
41 06755 8F00000          GOSBVL =REVPOP  renverse search
42 0675C 8A8           ?A=0   A          LEN(search)=0 ?
43 0675F 84           GOYES  NStrng  oui, erreur
44 06761 1B00000      D0=(5) =S-R1-2  S-R1-2 := LEN(search)
45 06768 140          DAT0=A A
46 0676B 137          CD1EX          ^ search
47 0676E 135          D1=C
48 06771 164          D0=D0+ 5          D0 := ^ S-R1-3
49 06774 144          DAT0=C A          S-R1-3 := ^ search
50 06777 8F00000      GOSBVL =AVE=D1  AVMEME := D1
51
52 0677E 8F00000      GOSBVL =GETPRO  fichier prive ?
53 06785 401          GOC   Mferr    oui, erreur
54 06788 8F00000      GOSBVL =GETSTC  fichier Basic ?
55 0678F 5D0          GONC   Find20  oui
56 06792 3100        LC(2)  =eFTYPE  non, erreur
57 06796 8D00000      Mferr  GOVLNG  =MFERR
```

```
58 0679D 8F00000 Find20 GOSBVL =NULLP fichier vide ?
59 067A4 511 GONC Find30 non
60 067A7 20 NStrng P= 0 erreur 'Not Found
61
62 *** change
63 * LC(4) #0002 * rev D
64 067A9 33201E LC(4) #E102 * rev X
65 *** end of change
66
67 067AF 8D00000 GOVLNG =BSERR
68 067B6 1F00000 Find30 D1=(5) =CURRL positionnement ..
69 067BD D2 C=0 A .
70 067BF 15F3 C=DAT1 4 .
71 067C3 8F00000 GOSBVL =FINDL .. ligne courante
72
73 *** change
74 * rev D
75 * GOC Find05
76 * ?ST=1 0
77 * GOYES NStrng
78 067CA Find05
79 * rev X, same as finds
80 067CA 5CD GONC NStrng non trouvee
81 *** end of change
82
83 067CD 137 CD1EX C[A] := ^ numero de ligne
84 067D0 D5 B=C A sauve dans B[A]
85 067D2 137 CD1EX restaure D1
86 067D5 8F00000 GOSBVL =NXTLIN ligne suivante
87 067DC 8B3 ?C<D A avant fin de fichier ?
88 067DF 70 GOYES Find15 oui
89
90 *** change
91 * rev D
92 * GONC NStrng
93 * rev X, same as finds
94 067E1 D4 A=B A non, on commence a la ..
95 067E3 131 D1=A .. ligne courante
96 *** end of change
97
98 067E6 DB Find15 C=D A C[A] := ^ fin fichier
99 067E8 8F00000 GOSBVL =CSLW5
100 067EF 137 CD1EX C[A] := ^ debut de ligne
101 067F2 135 D1=C D1 := ^ debut de ligne
102 067F5 10B Findx R3=C sauvegarde des pointeurs
103 * de fichier dans R3
104 067F8 8F00000 GOSBVL =LDCM10 decompile la ligne
105 * A ce stade :
106 * (OUTBS) ^ ligne decompilee ("objet")
107 * B[A] = LEN(object) en octets
108 * P indetermine
109 *
110 * S-R1-2 = LEN(search)
111 * S-R1-3 = ^ search
112 * R3[9:5] = ^ debut de ligne courante
113 * R3[4:0] = ^ fin de fichier
114 067FF 20 Find40 P= 0 par securite
```

```

115          * Comme la recherche porte simplement sur la
116          * partie utile de la ligne, excluant le numero de
117          * ligne, il faut actualiser la longueur de la
118          * ligne
119 06801 AF2          C=0      W
120 06804 1B00000    D0=(5) =OUTBS
121 0680B 142        A=DAT0 A      A[A] := ^ debut de ligne
122 0680E 1A0000    D0=(4) =LDCSPC
123 06814 146        C=DAT0 A      C[A] := ^ espace apres numero de ligne
124 06817 E2        C=C-A  A
125 06819 81E        CSRB          longueur en octets
126 0681C E6        C=C+1 A      + 1 espace          added rev X/finds
127 0681E D4        A=B    A      A[A] longueur totale
128 06820 EA        A=A-C  A      A[A] := longueur sans en-tete
129 06822 D8        B=A    A      B[A] := longueur utile
130 06824 C4        A=A+A  A      A[A] := longueur en nibs
131 06826 146        C=DAT0 A      C[A] := ^ objet
132 06829 10A       R2=C          R2 := ^ objet
133          *      C=C-1  A      * removed in rev X, as in finds
134          *      C=C-1  A      * removed in rev X, as in finds
135 0682C 135       D1=C          D1 := ^ objet
136 0682F 1B00000    D0=(5) =S-R1-3  .
137 06836 146       C=DAT0 A      .
138 06839 109       R1=C          R1 := ^ search
139 0683C 184       D0=D0- 5      D0 ^ S-R1-2
140 0683F 146       C=DAT0 A      C[A] := LEN(search)
141          * A ce stade :
142          * A[A] = LEN(objet)
143          * R2 = ^ objet
144          * C[A] = LEN(search)
145          * R1 = ^ search
146          * Le code est inspire de POS et STREQ1 mais ces
147          * deux sous-programmes travaillent sur des chaines
148          * "inversees" (premier caractere en adresse haute)
149          * Ici le premier caractere est en adresse basse.
150          * La chaine est divisÃe en n blocs de 16 quartets
151          * et un reste.
152 06842 EA        A=A-C  A
153 06844 455       GOC      Pos5      LEN(objet)<LEN(search)
154          *** change in rev X: added, was NOT in finds, bug fix?
155 06847 AD1       B=0      M
156 0684A D8       B=A      A
157 0684C 81D      BSRB
158          *** end of change
159 0684F CE        C=C-1  A          LEN(search)-1
160 06851 80D0     P=C      0          RMD(Len(search$)-1,16) -> reste
161 06855 F6       CSR      A          DIV(Len(search$)-1,16)
162 06857 D7       D=C      A          -> n
163 06859 542     GONC     Pos3      B.E.T.
164 0685C D9       Pos2    C=B      A
165 0685E 06       RSTK=C          Scanindex -> Stack;
166 06860 137     CD1EX
167 06863 06       RSTK=C          Scanpt -> Stack;
168 06865 137     CD1EX
169 06868 DB       C=D      A
170 0686A D5       B=C      A          B[A] := n
171 0686C 7E01     GOSUB   streql   compare search avec la --

```

```

172          *          suite de objet          |
173 06870 07          C=RSTK                    |
174 06872 135        D1=C          restaure D1  |
175 06875 07          C=RSTK                    |
176 06877 D5          B=C      A      restaure B[A]          v
177 06879 562        GONC   Pos6  search incluse dans objet
178 0687C CD          B=B-1  A      utilise plus loin -----
179 0687E 119        Pos3  C=R1          ^ search          |
180 06881 134        D0=C          D0 := ^ search          |
181 06884 451        GOC     Pos5  objet est analyse <-----
182 06887 14A        A=DAT0 B      A[B] := premier caractere de search
183 0688A 171        Pos4  D1=D1+ 2  caractere suivant de objet
184 0688D 14F        C=DAT1 B      C[B] := caractere suivant
185 06890 962        ?A=C   B
186 06893 9C        GOYES  Pos2  debut d'egalite
187 06895 CD          B=B-1  A      reste-t-il des caracteres
188
189          *** changes:
190          * rev D
191          *          ?B#0 A
192          *          GOYES  Pos4
193          *Pos5      GOTO    Find50
194          *Pos6      P= 0
195          * rev X, as finds
196 06897 52F        GONC   Pos4      oui
197 0689A 112        Pos5      A=R2      sortie
198 0689D 131        D1=A          pour avoir 0 plus loin
199 068A0          Pos6
200          *** end of changes
201
202 068A0 112          A=R2
203 068A3 AF2          C=0      W
204 068A6 D6          C=A      A      A[A] := ^ objet
205 068A8 133        AD1EX          D1 := ^ premier caractere
206 068AB EA          A=A-C  A
207          * A[A] contient le deplacement (dans objet)
208          * pour lequel on a trouve search dans objet.
209          * Si cette valeur est non nulle on va afficher la
210          * ligne
211 068AD 20          P= 0          rev X, as finds
212 068AF 11B        C=R3
213 068B2 135        D1=C          D1 := ^ debut de ligne
214 068B5 8A8        ?A=0   A      a-t-on trouve search ?      rev X/finds
215 068B8 67        GOYES  Find60  non                          rev X/finds
216          * On affiche la ligne.
217          * Il faudra positionner le curseur sur le
218          * premier caractere ou on trouve search
219 068BA 103        R3=A          R3 := deplacement
220 068BD 8F00000    GOSBVL =LDCOMP  Decompile la ligne pointee par D1;
221          *          elle devient ligne courante
222 068C4 1B00000    D0=(5) =OUTBS
223 068CB 146        C=DAT0 A
224 068CE 134        D0=C          D0 := ^ premier caractere
225 068D1 1F00000    D1=(5) =LDCSPC
226 068D8 AD0        A=0      M
227 068DE 143        A=DAT1 A      A[A] := ^ espace apres numero de ligne
228 068DE EA          A=A-C  A

```

```

229 068E0 11B          C=R3          C[A] := deplacement
230 068E3 CA          A=A+C  A      ajoute a l'offset
231 068E5 81C        ASRB          converti en octets
232 068E8 100        R0=A          sauve dans R0
233 068EB D9         C=B    A      2*LEN(ligne)
234 068ED 06         RSTK=C        sauvegarde
235 068EF 7C00        GOSUB  Find*
236 068F3 B1C3        NIBHEX B1C3   curseur eteint
237 068F7 E3         NIBASC '>'     '>'
238 068F9 B1E3        NIBHEX B1E3   curseur allume
239 068FD FF         NIBHEX FF     fin de sequence
240 068FF 07         Find* C=RSTK     C[A] := ^ debut sequence
241 06901 135        D1=C
242 06904 8F00000    GOSBVL =BF2DSP affiche la sequence
243 0690B 07         C=RSTK     B[A] := 2*LEN(ligne)
244 0690D D5         B=C    A
245 0690F 8F00000    GOSBVL =DPCNO affiche buffer de sortie
246 06916 118        C=R0       nombre de curseur a droite
247 06919 8F00000    GOSBVL =CURSRT execution ...
248 06920 8F00000    GOSBVL =BLDDSP construit l'affichage
249 06927 8D00000    GOVLNG =MAIN30 retour a Basic
250                    * on n'a rien trouve sur la ligne courante, on
251                    * essaie de passer a la ligne suivante.
252                    *** changes
253                    * rev D, not in finds
254                    * Find50 P=    0
255                    *   C=R3
256                    *   D1=C
257                    *** end of change
258 0692E 8F00000    Find60 GOSBVL =CSRW5
259 06935 D7         D=C    A      D[A] := ^ fin de fichier
260 06937 8F00000    GOSBVL =NXTLIN passage ligne suivante
261                    *   P=    0      ver D and finds, removed in ver X !
262 0693E 8B3        ?C<D  A      dans le fichier ?
263 06941 60         GOYES  Find70 oui
264 06943 636E      GOTO  NStrng  fin de fichier et rien trouve
265                    * on va passer a la ligne suivante, on remet les
266                    * pointeurs en place et on verifie que
267                    * l'utilisateur n'a pas appuye sur [ON].
268 06947 11B        Find70 C=R3          .
269 0694A 137        CD1EX          .
270 0694D 10B        R3=C          restaure R3
271 06950 8F00000    GOSBVL =CK"ON" verifie [ON] -----
272 06957 11B        C=R3
273 0695A 135        D1=C          D1 := ^ numero de ligne |
274 0695D 490        GOC    Findx+  [ON] non demande <-----
275 06960 8D00000    GOVLNG =NXTSTM si demande, fin
276 06967 6D8E      Findx+ GOTO  Findx  recommencons ...
277 0696B 1527      morest A=DAT0 W
278 0696F 1577      C=DAT1 W
279 06973 976      ?A#C  W
280 06976 00        RTNYES          # retour Carry Set
281 06978 16F      D0=D0+ 16      ^ mot suivant
282 0697B 17F      D1=D1+ 16
283 0697E CD        streql B=B-1  A      termine avec les mots complets
284 06980 5AE      GONC  morest  non
285 06983 1521      tailst A=DAT0 WP  test sur le reste

```



```
286 06987 1571      C=DAT1 WP
287 0698B 916      ?A#C  WP
288 0698E 00      RTNYES
289 06990 03      RTNCC
290
291 06992          END
```

AVE=D1	Ext		-	50			
BF2DSP	Ext		-	242			
BLDDSP	Ext		-	248			
BSERR	Ext		-	67			
CK"ON"	Ext		-	271			
CSLW5	Ext		-	99			
CSRW5	Ext		-	258			
CURRL	Ext		-	68			
CURSRT	Ext		-	247			
DSPCNO	Ext		-	245			
EXPEXC	Ext		-	40			
FINDL	Ext		-	71			
=FINDe	Abs	26446	#0000674E	-	40		
Find*	Abs	26879	#000068FF	-	240	235	
Find05	Abs	26570	#000067CA	-	78		
Find15	Abs	26598	#000067E6	-	98	88	
Find20	Abs	26525	#0000679D	-	58	55	
Find30	Abs	26550	#000067B6	-	68	59	
Find40	Abs	26623	#000067FF	-	114		
Find60	Abs	26926	#0000692E	-	258	215	
Find70	Abs	26951	#00006947	-	268	263	
Findx	Abs	26613	#000067F5	-	102	276	
Findx+	Abs	26983	#00006967	-	276	274	
GETPRO	Ext		-	52			
GETSTC	Ext		-	54			
LDCM10	Ext		-	104			
LDCOMP	Ext		-	220			
LDCSPC	Ext		-	122	225		
MAIN30	Ext		-	249			
MFERR	Ext		-	57			
Mferr	Abs	26518	#00006796	-	57	53	
NStrng	Abs	26535	#000067A7	-	60	43	80 264
NULLP	Ext		-	58			
NXTLIN	Ext		-	86	260		
NXTSTM	Ext		-	275			
OUTBS	Ext		-	120	222		
Pos2	Abs	26716	#0000685C	-	164	186	
Pos3	Abs	26750	#0000687E	-	179	163	
Pos4	Abs	26762	#0000688A	-	183	196	
Pos5	Abs	26778	#0000689A	-	197	153	181
Pos6	Abs	26784	#000068A0	-	199	177	
REVPOP	Ext		-	41			
S-R1-2	Ext		-	44			
S-R1-3	Ext		-	136			
STRNGp	Ext		-	39			
eFTYPE	Ext		-	56			
morest	Abs	26987	#0000696B	-	277	284	
streql	Abs	27006	#0000697E	-	283	171	
tailst	Abs	27011	#00006983	-	285		

Input Parameters

Source file name is find.a

Listing file name is find.l

Object file name is find.o

Flags set on command line
None

Errors

None

```

1          TITLE JPC LEX: MODULE FORMA
2
3          * *****
4          * Formatting statement module
5          *
6          * Keyword(s):
7          *   SPACE$, REDUCE$, CENTER$, FORMAT$, CESURE$
8          * Author(s):
9          *   Pierre David, Michel Martinet
10         * History:
11         *   Published in JPC #26
12         *   SPACE$ changed to accept a 2nd optional parameter
13         *   R. Rosenbaum, rev X:
14         *   apply RED$ on strings before CENTER$
15         *
16         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
17         * *****
18
19 06992          ABS      #6992
20
21         * local symbols:
22         flag      EQU    0
23         boucle    EQU    1
24         caract    EQU    2
25
26
27         * *****
28         * reduire
29         * But: enlever tous les espaces superflus d'une
30         * chaÑne de caractÉres.
31         * *****
32 06992 04      reduire SETHEX      HEX pour POP1S
33 06994 8F00000      GOSBVL =POP1S
34 0699B 136      CD0EX      C=pointeur programme
35 0699E 8F00000      GOSBVL =CSLC5      C=.....D0.....
36 069A5 137      CD1EX      C=.....D0 D1
37 069A8 C2      C=C+A      A      C=.....D0 (D1+long)
38 069AA 109      R1=C      R1=...[D0][D1+A]
39 069AD 134      D0=C      D0=^dÅbut de la chaÑne
40 069B0 135      D1=C      D1= idem
41 069B3 D3      D=0      A      D[A]=Nb d'emplacements
42         *          *          (espaces de la chaÑne
43         *          *          rÅduite)
44 069B5 850      ST=1      flag      Il faut enlever les
45         *          *          blancs suivant
46 069B8 841      ST=0      boucle    aucun passage dans la
47         *          *          boucle
48 069BB 842      ST=0      caract    CaractÉre non blanc non
49         *          *          rencontrÅ
50 069BE 3102      LCASC      ' '      C=' '
51 069C2 AE5      B=C      B      B[B]=32 (espace)
52 069C5 CC      Enleve      A=A-1      A      Longueur en quartets
53 069C7 4E2      GOC      Fin      Saut si longueur nulle
54 069CA 851      ST=1      boucle    au moins un passage dans
55         *          *          la boucle
56 069CD CC      A=A-1      A      quartets...
57 069CF 181      D0=D0- 2      caractÉre suivant

```

```

58 069D2 14E          C=DAT0 B      Il est recopi  dans C
59 069D5 965          ?C#B  B      est-ce un espace ?
60 069D8 F0           GOYES noblan  non
61 069DA 870          ?ST=1 flag    Si non, est-ce le premier
62                   *          *          d'une s rie ?
63 069DD 8E           GOYES Enleve  non, alors on le n glige
64 069DF E7           D=D+1 A      oui: un emplacement
65                   *          *          suppl mentaire
66 069E1 850          ST=1  flag    C'est le premier d'une
67                   *          *          s rie
68 069E4 580          GONC  ajoute  B.E.T.
69 069E7 840          noblan ST=0  flag    Premier d'une s rie:
70                   *          *          aucun sens, puisque non
71                   *          *          blanc
72 069EA 852          ST=1  caract  Au moins un caract re non
73                   *          *          blanc dans la cha ne
74 069ED 1C1          ajoute D1=D1- 2   On range le caract re
75                   *          *          C[B] ds la cha ne r duite.
76 069F0 14D          DAT1=C B
77 069F3 51D          GONC  Enleve  B.E.T.
78
79 069F6 860          Fin  ?ST=0  flag
80 069F9 11           GOYES nonnul
81 069FB 861          ?ST=0 boucle
82 069FE C0           GOYES nonnul
83 06A00 862          ?ST=0 caract
84 06A03 70           GOYES nonnul
85 06A05 171          D1=D1+ 2     Si le dernier caract re
86                   *          *           tait un blanc, on
87                   *          *          l'enl ve,
88 06A08 CF           D=D-1  A     ainsi que du nombre
89                   *          *          d'emplacements
90 06A0A 01           nonnul RTN
91
92
93
94                   * *****
95                   * SPACE$ entry
96                   * *****
97 06A0C 8C12          NIBHEX 8C12
98 06A10              =SPACEe
99 06A10 850          ST=1  0
100 06A13 80DF         P=C 15
101 06A17 891          ?P= 1
102 06A1A 50           GOYES o69D6
103 06A1C 840          ST=0  0
104 06A1F              o69D6
105 06A1F 8F00000      GOSBVL =RNDAHX
106 06A26 440          GOC   positif
107 06A29 D0           A=0   A
108 06A2B              positif
109 06A2B 100          R0=A
110 06A2E 17F         D1=D1+ 16
111 06A31 860          ?ST=0 0
112 06A34 90           GOYES o69F4
113 06A36 3102        LCHEX 20
114 06A3A 591          GONC  o6A0B

```

```
115 06A3D 1534      o69F4 A=DAT1 S
116 06A41 B44       A=A+1 S
117 06A44 4F1       GOC o6A1B
118 06A47 8F00000   GOSBVL =RNDAHX
119 06A4E AE6       C=A B
120 06A51 17F       D1=D1+ 16
121 06A54 1C1       o6A0B D1=D1- 2
122 06A57 14D       DAT1=C B
123 06A5A 171       D1=D1+ 2
124 06A5D D0        A=0 A
125 06A5F E4        A=A+1 A
126 06A61 5B1       GONC o6A34
127 06A64 8F00000   o6A1B GOSBVL =POP1S
128 06A6B 137       CD1EX
129 06A6E C2        C=C+A A
130 06A70 135       D1=C
131 06A73 D6        C=A A
132 06A75 AF0       A=0 W
133 06A78 DA        A=C A
134 06A7A 81C       ASRB
135 06A7D 102       o6A34 R2=A
136 06A80 137       CD1EX
137 06A83 135       D1=C
138 06A86 109       R1=C
139 06A89 8F00000   GOSBVL =D=AVMS
140 06A90 136       CD0EX
141 06A93 06        RSTK=C
142 06A95 118       C=R0
143 06A98 D5        B=C A
144 06A9A 6220      GOTO o6A74
145 06A9E 112       o6A55 A=R2
146 06AA1 119       C=R1
147 06AA4 134       D0=C
148 06AA7 6010      GOTO o6A6F
149 06AAB 181       o6A62 D0=D0- 2
150 06AAE 14E       C=DAT0 B
151 06AB1 8F00000   GOSBVL =STKCHR
152 06AB8 CC        o6A6F A=A-1 A
153 06ABA 50F       GONC o6A62
154 06ABD CD        o6A74 B=B-1 A
155 06ABF 5ED       GONC o6A55
156 06AC2 07        C=RSTK
157 06AC4 134       D0=C
158 06AC7 461       GOC ret2
159
160
161                * *****
162                * REDUCE$ entry
163                * *****
164 06ACA 411                NIBHEX 411
165 06ACD 71CE              =REDUCe GOSUB reduire
166
167                *****
168                * retour
169                * But: assurer le retour È l'environnement Basic,
170                * et prendre en charge la restauration des
171                * pointeurs.
```

```

172          *****
173 06AD1 119   retour  C=R1           C=....D0...
174 06AD4 8F00000 GOSBVL =CSRC5       C=.....D0
175 06ADB 136   CD0EX           D0=pgm counter
176 06ADE 840   ret2    ST=0    0       ADHEAD ne rend pas la main
177 06AE1 8F00000 GOSBVL =D=AVMS
178 06AE8 8D00000 GOVLNG =ADHEAD
179
180 06AEF 8D00000 argerr GOVLNG =ARGERR
181
182
183          * *****
184          * CENTER$ entry
185          * *****
186 06AF6 8422   NIBHEX 8422
187 06AFA       =CENTEe
188 06AFA 7E50   GOSUB  Argnum   Pop, teste et convertit en
189          *           *           HEXA le paramÈtre
190          *           *           numÁrique.
191 06AFE 8F00000 GOSBVL =REVPOP   "Pop", test et renverse
192          *           *           la chaÑne alphanumÁrique.
193          *           *           En sortie de routine, D1
194          *           *           pointe aprÈs l'en-tÁte de
195          *           *           la chaÑne.
196
197          *** changes
198          * rev D
199          *           CD1EX       Chargement dans C(A) de
200          **          *           la valeur de D1.
201          *           D1=C       Restitution.
202          *           C=C+A   A   C(A) = maintenant
203          **          *           l'adresse de fin de chaÑne
204          **          *           (utilisÁ plus loin par la
205          **          *           routine ADHEAD).
206          *           R1=C       Sauvegarde dans R1 pour
207          **          *           Átre utilisÁ dans ADHEAD.
208          * rev X
209 06B05 8E0000   GOSUBL =REDIT   (in desal module)
210 06B0B 8A8     ?A=0   A
211 06B0E 63     GOYES  FIN
212          *** end of changes
213
214 06B10 8F00000 GOSBVL =D=AVMS   D(A) = AVMEMS (AVailable
215          *           *           MEMory Start). L'opÁration
216          *           *           est faite maintenant car
217          *           *           la routine utilise C(A).
218 06B17 118     C=R0           On charge dans C(A) la
219          *           *           valeur du paramÈtre
220          *           *           numÁrique.
221 06B1A C6     C=C+C   A       C(A) = paramÈtre
222          *           *           numÁrique en quartets.
223 06B1C 42D     GOC    argerr   En cas de dÁpassement,
224          *           *           l'argument Átait supÁrieur
225          *           *           È la longueur maxi. d'une
226          *           *           chaÑne: ERREUR.
227
228          *** change: removed from rev D

```

```

229      *           ?A=0  A           La longueur de la chaîne
230      **          *                est-elle nulle ??
231      *           GOYES  FIN        Si oui: FIN.
232      *** end of change
233 06B1F 8BE      ?C<=A  A           La longueur de la chaîne
234      *           *                >= È la longueur demandée?
235 06B22 22      GOYES  FIN        Si oui: on renvoie
236      *           *                simplement la chaîne.
237 06B24 E2      C=C-A  A           Calcul du nombre de
238      *           *                blancs: C(A) = 4 fois le
239      *           *                nombre de blancs.
240 06B26 AD0     A=0    M           ) Division de C(A)
241 06B29 DA      A=C    A           ) par 4
242 06B2B 81C     ASRB                    ) et restitution dans A(A)
243 06B2E 81C     ASRB                    ) en octets.
244 06B31 3102    LCASC  ' '        On charge " " dans C(B)
245 06B35 590     GONC  test        Reste-t-il des caractères
246      *           *                È empiler ? (B.E.T.)
247 06B38 8F00000 bsp   GOSBVL =STKCHR Oui: on empile un blanc
248      *           *                devant la chaîne
249      *           *                alphanumérique.
250 06B3F CC      test  A=A-1  A      A(A) sert de compteur.
251 06B41 56F     GONC  bsp        Boucle si A>=0
252 06B44 850     FIN   ST=1  0      Obligatoire pour un bon
253      *           *                retour de ADHEAD.
254 06B47 8F00000 GOSBVL =ADHEAD Mise en place de
255      *           *                l'en-tête de la nouvelle
256      *           *                chaîne.
257 06B4E 8F00000 GOSBVL =REV$   On renverse È nouveau
258      *           *                avant le
259 06B55 8D00000 GOVLNG =EXPR   retour È BASIC.
260
261      *****
262      * Argnum
263      * But: renvoyer le paramètre qui est sur la pile,
264      * d'après la vérification et test (>0).
265      * Le nombre est renvoyé dans A(A) et R0(A) en hexa
266      * En sortie, on a Carry=0
267      *****
268 06B5C 8F00000 Argnum GOSBVL =RNDAXH  A(A)=nombre hexa
269 06B63 5B8      GONC  argerr  Argument Error si <0
270 06B66 8A8      ?A=0  A           Paramètre nul
271 06B69 68      GOYES  argerr  ... alors Arg. Error
272 06B6B 17F     D1=D1+ 16        POPer le paramètre
273 06B6E 100     R0=A                    et le mettre dans R0
274 06B71 01      RTN
275
276 06B73 8D00000 memerr GOVLNG =MEMERR
277
278
279      * *****
280      * FORMAT$ entry
281      * *****
282 06B7A 8422      NIBHEX 8422
283 06B7E           =FORMAe
284 06B7E 7ADF     GOSUB  Argnum  A[A],R0[A]=nombre hexa
285 06B82 7C0E     GOSUB  reduire

```



```

286 06B86 119          C=R1          C=.....(^dÅbut de la
287                   *          *          chaÑne)
288 06B89 110          A=R0          A=paramÉtre
289 06B8C E2           C=C-A  A
290 06B8E 44E          GOC  memerr
291 06B91 E2           C=C-A  A          C=^fin de la chaÑne
292                   *          *          formattÅe
293 06B93 4FD          GOC  memerr
294 06B96 10A          R2=C          R2=^fin de la chaÑne
295                   *          *          formattÅe
296 06B99 DA           A=C  A          Pour transfert ensuite
297                   *          *          dans D0
298 06B9B 134          D0=C          }
299 06B9E 16F          D0=D0+ 16     }+16 pour l'en-tÅte
300 06BA1 41D          GOC  memerr   }
301 06BA4 136          CD0EX         }
302 06BA7 1B00000     D0=(5) =AVMEMS
303 06BAE 132          AD0EX
304 06BB1 8BE          ?C<=A  A          a-t-on passÅ AVMEMS ?
305 06BB4 FB           GOYES  memerr   non: pas d'erreur
306 06BB6 119          C=R1          C=^dÅbut de la chaÑne
307 06BB9 AF0          A=0  W          }
308 06BBC DA           A=C  A          } A=00...000 ^dÅbut
309 06BBE 137          CD1EX         C=^fin de la chaÑne
310                   *          *          rÅduite
311 06BC1 135          D1=C          D1= idem
312 06BC4 EA           A=A-C  A          A=longueur rÅduite
313 06BC6 81C          ASRB          ... en octets
314 06BC9 118          C=R0          C=longueur formattÅe
315 06BCC 8B6          ?A>C  A          erreur si long(rÅduite) >
316                   *          *          long(formattÅe)
317 06BCF 34           GOYES  strovf
318 06BD1 8A2          ?A=C  A          Retour si chaÑne dÅjÈ
319                   *          *          formattÅe...
320 06BD4 A3           GOYES  retouR
321 06BD6 E4           A=A+1  A          long(chaÑne rÅduite)
322 06BD8 8AB          ?D=0  A          Peut-on mettre des blancs?
323 06BDB 33           GOYES  retouR   non: la chaÑne rÅduite
324                   *          *          est renvoyÅe
325 06BDD EA           A=A-C  A
326 06BDF F8           A=-A  A
327 06BE1 DB           C=D  A
328 06BE3 10B          R3=C          R3= nb total
329                   *          *          d'emplacements
330 06BE6 8F00000     GOSBVL =IDIVA  A/C
331 06BED 20           P= 0          P=15 aprÈs IDIVA
332                   *          *          A: Quotient  B:C: Reste
333                   *          *          Il faut mettre C fois
334                   *          *          A+1 blancs
335                   *          *          et (le reste) fois  A
336                   *          *          blancs
337 06BEF 108          R0=C          R0 <- C
338 06BF2 D8           B=A  A
339 06BF4 E5           B=B+1  A
340 06BF6 7320         GOSUB  rajout
341 06BFA 110          A=R0          A=longueur de la chaÑne
342                   *          *          formattÅe

```

```

343 06BFD 11B          C=R3          C=nb total d'emplacements
344 06C00 E2          C=C-A  A      C=(le reste)
345 06C02 CD          B=B-1  A      (un espace en moins par
346                   *          *          boucle)
347 06C04 7510        GOSUB  rajout
348 06C08 11A          C=R2
349 06C0B 135          D1=C          pointeur de pile-math
350 06C0E 62CE        retouR  GOTO   retour
351
352                   *****
353                   * strovf
354                   * But: renvoyer l'erreur "DÃpassement de chaÃne"
355                   * (pour les heureux possesseurs de FRALEX).
356                   * Pour les autres, il n'y a plus rien Ã faire...
357                   *****
358 06C12 3152        strovf  LC(2)  37      String Overflow
359 06C16 8D00000      GOVLNG =MFERR
360
361                   *****
362                   * rajout
363                   * But: rajouter C(A) espaces Ã la chaÃne
364                   *****
365 06C1D D7          rajout  D=C    A      D=compteur.
366 06C1F 3102          LCASC  ' '      C=' '
367 06C23 6620          GOTO   bool
368 06C27 14B          bb      A=DAT1 B      Recopier le dernier
369                   *          *          caractÃre de la chaÃne
370                   *          *          rÃduite
371 06C2A 171          D1=D1+ 2
372 06C2D 148          DAT0=A  B      dans la chaÃne formattÃe
373 06C30 161          D0=D0+ 2
374 06C33 966          ?C#A   B      est-ce un espace ?
375 06C36 1F          GOYES  bb      non: on continue
376 06C38 D4          A=B    A      Boucle interne sur A.
377                   *          *          Elle est exÃcutÃe A fois
378 06C3A 580          GONC   b2
379 06C3D 14C        recopie DAT0=C  B      Un espace de plus dans la
380                   *          *          chaÃne formattÃe
381 06C40 161          D0=D0+ 2
382 06C43 CC          b2      A=A-1  A
383 06C45 57F          GONC   recopie  On continue tant que A>=0
384 06C48 CF          D=D-1  A      Un emplacement de moins
385 06C4A 8AF        bool   ?D#0  A      Est-ce le dernier
386                   *          *          emplacement ?
387 06C4D AD          GOYES  bb      Non, on continue
388 06C4F 01          RTN                    Oui: retour Ã l'envoyeur
389
390
391
392                   * *****
393                   * CESURE$ entry
394                   * *****
395 06C51 8422          NIBHEX 8422
396 06C55 730F        =CESURE GOSUB Argnum  Retourne un nombre HEXA
397                   *          *          dans A(A) et R0(A)
398 06C59 D8          B=A    A      Sauvegarde de la cÃsure
399                   *          *          dans B(A).

```

```

400 06C5B 8F00000          GOSBVL =POP1S  VÅrifie que l'on est bien
401                        *          *          en prÅsence d'une chaÑne
402                        *          *          et retourne sa longueur
403                        *          *          dans A(A).
404 06C62 D6              C=A    A          Division de A(A) par 2.
405 06C64 AD0            A=0    M          "
406 06C67 DA             A=C    A          "
407 06C69 81C           ASRB          "
408 06C6C 137           CD1EX        C(A) ^ DÅbut de la
409                        *          *          chaÑne.
410 06C6F 135           D1=C          "
411 06C72 C2            C=C+A   A          "
412 06C74 C2            C=C+A   A          "
413 06C76 D7           D=C    A          On sauvegarde l'adresse de
414                        *          *          dÅbut de chaÑne dans D(A)
415 06C78 8B4           ?A<B   A          longueur(chaÑne) < cÅsure?
416 06C7B 97           GOYES   FinC        Oui, on renvoie la chaÑne
417                        *          *          tout simplement
418 06C7D E9           C=C-B   A          On place D1 sur le
419                        *          *          premier caractÈre concernÅ
420 06C7F E9           C=C-B   A          par la cÅsure
421 06C81 135           D1=C          "
422 06C84 3F02F312      LCASC   ') ; : , ! ? ' * Tous les
      E2C2A3B3
      92
423                        *          *          caractÈres de comparaison
424 06C96 0A           ST=C          On met ST(0) È 0: flag
425 06C98 14B          Boucle A=DAT1 B        A(B)=caractÈre È comparer.
426 06C9B 962           ?A=C   B          Est-ce un espace ?
427 06C9E 61           GOYES   TestP        Oui: on va en TestP pour
428                        *          *          comparaison avec les
429                        *          *          autres caractÈres de C(W)
430 06CA0 850           ST=1   flag        On a rencontrÅ au moins
431                        *          *          un caractÈre diffÅrent
432                        *          *          d'un espace
433 06CA3 CD           AutreB B=B-1 A          DÅcrÅment de la cÅsure
434 06CA5 480           GOC    Nobl        Reste-t-il encore des
435                        *          *          caractÈres È tester ?
436 06CA8 171           D1=D1+ 2        Oui, on place D1 sur le
437                        *          *          suivant
438 06CAB 5CE           GONC   Boucle        B.E.T.
439 06CAE 110          Nobl   A=R0          A(A) = cÅsure
440 06CB1 424           GOC    FinC          B.E.T.
441 06CB4 110          TestP A=R0          A(A) = Valeur de la cÅsure
442 06CB7 8A0           ?A=B   A          La position de l'espace =
443                        *          *          cÅsure ?
444 06CBA D1           GOYES   NotstP        Oui, on retourne chercher
445                        *          *          l'espace suivant en
446                        *          *          dÅcrÅmentant la cÅsure
447
448                        * La sÅquence suivante teste si A(B) est un des
449                        * caractÈres de ponctuation.
450                        * Elle prend moins de place dans ce Lex que
451                        * l'appel È la routine interne: MEMBER
452 06CBC 27            P=      7          Autrement test: P=
453                        *          *          compteur
454 06CBE 1C1           D1=D1- 2        D1 ^ le caractÈre suivant

```

```

455          *          *          l'espace
456 06CC1 14B          A=DAT1 B      A(B) = caractÈre È tester
457 06CC4 BF6      Bcl2  CSR      W      DÅcalage du registre C
458 06CC7 BF6          CSR      W      "
459 06CCA 962          ?A=C      B      Le caractÈre est-il Ågal ?
460 06CCD 71          GOYES Suite   Oui: on branche en "Suite"
461 06CCF 0D          P=P-1          DÅcrÅmentation du compteur
462 06CD1 52F          GONC      Bcl2   Il reste des caractÈres È
463          *          *          tester dans C
464 06CD4 4B1          GOC      Decr   Il n'en reste plus
465 06CD7 870      NotstP ?ST=1   flag   A-t-on dÅjÈ rencontrÅ un
466          *          *          caractÈre diffÅrent d'un
467          *          *          espace
468 06CDA 9C          GOYES AutreB  Oui: on ne dÅcrÅmente pas
469          *          *          la cÅsure (elle pointe ce
470          *          *          caractÈre !)
471 06CDC CC          A=A-1   A      On dÅcrÅmente la cÅsure
472 06CDE 100         R0=A          R0 <- la cÅsure
473 06CE1 51C         GONC      AutreB  B.E.T.
474 06CE4 118      Suite  C=R0          La position du blanc =
475          *          *          la cÅsure ?
476 06CE7 8A1         ?B=C      A      " " "
477 06CEA 60          GOYES Decr   Oui: on branche en fin de
478          *          *          programme directement
479 06CEC E5          B=B+1   A      Autrement on ajuste le
480          *          *          rÅsultat pour
481 06CEE E5          B=B+1   A      englober le caractÈre de
482          *          *          ponctuation
483 06CF0 D4          Decr   A=B      A      Passage de B(A) dans A(A)
484          *          *          pour l'appel de la
485          *          *          routine FLTDH
486 06CF2 CC          A=A-1   A      DÅcrÅment de A(A) pour
487          *          *          Åtre juste
488 06CF4 DB          FinC   C=D      A
489 06CF6 135         D1=C          D1 pointe maintenant le
490          *          *          dÅbut de la chaÑne.
491 06CF9 1CF         D1=D1- 16    D1-16 nous sommes prÅt È
492          *          *          placer le paramÈtre
493          *          *          numÅrique
494 06CFC 8F00000     GOSBVL =HDFLT Transformation de A(A) en
495          *          *          nombre dÅcimal dans A(W)
496 06D03 AF6          C=A      W
497 06D06 8D00000     GOVLNG =FNRTN4 Retour È BASIC.
498
499 06D0D          END

```

ADHEAD	Ext		-	178	254				
ARGERR	Ext		-	180					
AVMEMS	Ext		-	302					
Argnum	Abs	27484	#00006B5C	-	268	188	284	396	
AutreB	Abs	27811	#00006CA3	-	433	468	473		
Bcl2	Abs	27844	#00006CC4	-	457	462			
Boucle	Abs	27800	#00006C98	-	425	438			
=CENTEe	Abs	27386	#00006AFA	-	187				
=CESURE	Abs	27733	#00006C55	-	396				
CSLC5	Ext		-	35					
CSRC5	Ext		-	174					
D=AVMS	Ext		-	139	177	214			
Decr	Abs	27888	#00006CF0	-	483	464	477		
EXPR	Ext		-	259					
Enleve	Abs	27077	#000069C5	-	52	63	77		
FIN	Abs	27460	#00006B44	-	252	211	235		
FNRTN4	Ext		-	497					
=FORMAe	Abs	27518	#00006B7E	-	283				
Fin	Abs	27126	#000069F6	-	79	53			
FinC	Abs	27892	#00006CF4	-	488	416	440		
HDFLT	Ext		-	494					
IDIVA	Ext		-	330					
MEMERR	Ext		-	276					
MFERR	Ext		-	359					
Nobl	Abs	27822	#00006CAE	-	439	434			
NotstP	Abs	27863	#00006CD7	-	465	444			
POP1S	Ext		-	33	127	400			
REDIT	Ext		-	209					
=REDUCEe	Abs	27341	#00006ACD	-	165				
REV\$	Ext		-	257					
REVPOP	Ext		-	191					
RNDAHx	Ext		-	105	118	268			
=SPACEe	Abs	27152	#00006A10	-	98				
STKCHR	Ext		-	151	247				
Suite	Abs	27876	#00006CE4	-	474	460			
TestP	Abs	27828	#00006CB4	-	441	427			
ajoute	Abs	27117	#000069ED	-	74	68			
argerr	Abs	27375	#00006AEF	-	180	223	269	271	
b2	Abs	27715	#00006C43	-	382	378			
bb	Abs	27687	#00006C27	-	368	375	387		
bool	Abs	27722	#00006C4A	-	385	367			
boucle	Abs	1	#00000001	-	23	46	54	81	
bsp	Abs	27448	#00006B38	-	247	251			
caract	Abs	2	#00000002	-	24	48	72	83	
flag	Abs	0	#00000000	-	22	44	61	66	69 79 430
					465				
memerr	Abs	27507	#00006B73	-	276	290	293	300	305
noblan	Abs	27111	#000069E7	-	69	60			
nonnul	Abs	27146	#00006A0A	-	90	80	82	84	
o69D6	Abs	27167	#00006A1F	-	104	102			
o69F4	Abs	27197	#00006A3D	-	115	112			
o6A0B	Abs	27220	#00006A54	-	121	114			
o6A1B	Abs	27236	#00006A64	-	127	117			
o6A34	Abs	27261	#00006A7D	-	135	126			
o6A55	Abs	27294	#00006A9E	-	145	155			
o6A62	Abs	27307	#00006AAB	-	149	153			
o6A6F	Abs	27320	#00006AB8	-	152	148			

o6A74	Abs	27325	#00006ABD -	154	144	
positif	Abs	27179	#00006A2B -	108	106	
rajout	Abs	27677	#00006C1D -	365	340	347
recopie	Abs	27709	#00006C3D -	379	383	
reduire	Abs	27026	#00006992 -	32	165	285
ret2	Abs	27358	#00006ADE -	176	158	
retouR	Abs	27662	#00006C0E -	350	320	323
retour	Abs	27345	#00006AD1 -	173	350	
strovf	Abs	27666	#00006C12 -	358	317	
test	Abs	27455	#00006B3F -	250	245	

Input Parameters

Source file name is forma.a

Listing file name is forma.l

Object file name is forma.o

Flags set on command line
None

Errors

None

```

1          TITLE JPC LEX: MODULE FRAC
2
3          * *****
4          * Real to fraction conversion function
5          *
6          * Keyword(s):
7          *   FRAC$
8          * Author(s):
9          *   Guy Toublanc
10         * History:
11         *   Published in JPC #42
12         *
13         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14         * *****
15
16 06D0D          ABS      #6D0D
17
18         * *****
19         * FRAC$ entry
20         * *****
21 06D0D 8812          NIBHEX 8812      mini : 1 param. maxi 2
22         *          *          param. numÃ©riques
23 06D11 132          =FRACe AD0EX      sauvegarde D0
24 06D14 1B00000      D0=(5) =FUNCD0
25 06D1B 140          DAT0=A A
26 06D1E A4E          C=C-1 S      dÃ©crÃ©mente le nombre
27 06D21 A4E          C=C-1 S      de paramÃ©tres
28 06D24 AF2          C=0 W
29 06D27 10A          R2=C          D-1 = 0 -> R2
30 06D2A 21           P= 1
31 06D2C 80CE          C=P 14
32 06D30 109          R1=C          Do = 1 -> R1
33 06D33 108          R0=C          N-1 = 1 -> R0
34 06D36 840          ST=0 0      repÃ©re paramÃ©tre prÃ©cision
35 06D39 4B2          GOC POPN      1 seul paramÃ©tre alors POPN
36 06D3C 8F00000      GOSBVL =POP1R      pope et teste n (prÃ©cision
37         *          *          ou nombre d'itÃ©rations)
38 06D43 17F          D1=D1+ 16      actualise D1
39 06D46 948          ?A=0 S      paramÃ©tre prÃ©cision ?
40 06D49 50           GOYES A0S      oui alors on conserve le
41         *          *          repÃ©re prÃ©cision (S0=0)
42 06D4B 850          ST=1 0      repÃ©re itÃ©ration
43 06D4E AC0          AOS A=0 S
44 06D51 938          ?A=0 X      1 seul digit ?
45 06D54 50           GOYES aslc2      oui alors shift 2 fois
46 06D56 810          ASLC      non alors shift 3 fois
47 06D59 810          aslc2 ASLC
48 06D5C 810          ASLC      ici A(B) = n
49 06D5F AE6          C=A B      C(W) = 10^n
50 06D62 BBA          C=-C X      C(W) = 10^(-n)
51
52 06D65 8F00000      POPN GOSBVL =POP1R      pope et teste N
53 06D6C 96E          ?C#0 B      C(B) = n ?
54 06D6F 51           GOYES PARAM      oui alors prÃ©cision donnÃ©e
55         *          *          ou nombre d'itÃ©rations
56         *          *          demandÃ©
57 06D71 32994          LCHEX 499

```



```

58 06D76 B32          C=C-A X      C(X) = 499 - exposant de N
59 06D79 32099       LCHEX 990    C(W) = 10^(-10)
60 06D7E 550         GONC  PARAM  exposant de N > 0 alors
61                   *      *      C(W) = prÃcision
62 06D81 A32         C=C+A X      exposant < 0 alors prÃc.
63                   *      *      flottante 10^(-10 + expos.
64                   *      *      de N )
65
66 06D84 AC6         PARAM C=A    S      rÃcupÃre le signe de N
67 06D87 1557       DAT1=C W    sauve le signe de N et le
68                   *      *      paramÃtre prÃcision ou
69                   *      *      itÃrations sur la math
70                   *      *      stack
71 06D8B AC0         A=0    S      A(W) = ABS(N)
72 06D8E 7651       GOSUB  STO    sauvegarde dans la scratch
73                   *      *      math stack
74 06D92 7951       GOSUB  stscr  sauvegarde de ABS(Ho) =
75                   *      *      ABS(N)
76 06D96 7431       GOSUB  clrfrc A(W) = IP(ABS(N))
77 06D9A 103        R3=A      IP(ABS(N)) = ABS(No) -> R3
78
79 06D9D 860        LOOP  ?ST=0 0    option prÃcision ?
80 06DA0 11         GOYES  PREC   oui saut au test prÃcision
81 06DA2 14B        A=DAT1 B    A(B) = - paramÃtre
82                   *      *      itÃrations
83 06DA5 B64        A=A+1 B    incrÃmentation
84 06DA8 4E2        GOC   out    derniÃre itÃration
85                   *      *      alors rÃsultat
86 06DAB 149        DAT1=A B    sauve le nombre
87                   *      *      d'itÃrations restantes
88 06DAE 591        GONC  ITER  saute le test prÃcision
89 06DB1 7A51       PREC  GOSUB  A-1S  A(W) = -IP(ABS(N))
90                   *      *      ou -ABS(Ni/Di)
91                   *      *      X = ABS(N)
92                   *      *      A(W) = ABS(N)
93                   *      *      X = ABS(N) - ABS(Ni/Di)
94                   *      *      = delta
95                   *      *      A(W) = ABS(delta)
96 06DB5 1577       C=DAT1 W    C(W) = prÃcision et signe
97                   *      *      de N
98 06DB9 AC2        C=0    S      C(W) = prÃcision
99 06DEC 23         P=    3      pour TEST12A
100 06DBE 8F00000   GOSBVL =TST12A ABS(delta) <= prÃcision ?
101 06DC5 411       GOC   out    oui alors rÃsultat
102 06DC8 7911     ITER  GOSUB  AR3    Ni -> scratch math stack
103 06DCC 8F00000   GOSBVL =RCLW2  X = ABS(Hi)
104 06DD3 77F0     GOSUB  clrfrc  A(W) = IP(ABS(Hi))
105 06DD7 4F6      out   GOC   OUT    FP(Hi) = 0 alors rÃsultat
106 06DDA 7131     GOSUB  A-1S   A(W) = FP(ABS(Hi))
107 06DDE 8F00000   GOSBVL =SPLITA A(W) -> X
108 06DE5 8F00000   GOSBVL =1/X15  X = 1/FP(ABS(Hi))
109                   *      *      = ABS(Hi+1)
110 06DEC 7601     GOSUB  ROUND  ABS(Hi+1) arrondi Ã 12
111                   *      *      digits -> A(W)
112 06DF0 8F00000   GOSBVL =RCSCR  fait la place pour Hi+1
113 06DF7 7FC0     GOSUB  STOFrc  ABS(Hi+1) -> scr. math st.
114                   *      *      puis A(W) = IP(ABS(Hi+1))

```

```

115          *          *          = ai+1
116 06DFB 11B          C=R3          C(W) = ABS(Ni)
117 06DFE 78D0        GOSUB mp2-12  A(W) = ai+1 * ABS(Ni)
118 06E02 118          C=R0          C(W) = ABS(Ni-1)
119 06E05 7311        GOSUB AD          A(W) = ABS(Ni+1)
120          *          *          = ai+1*ABS(Ni)+ABS(Ni-1)
121 06E09 123          AR3EX         R3 = ABS(Ni+1)
122          *          *          A(W) = ABS(Ni)
123 06E0C 100          R0=A          RO = Ni replace Ni-1
124 06E0F 8F00000     GOSBVL =RCLW1    X = ABS(Hi+1)
125 06E16 74B0        GOSUB clrfrcc    A(W) = ai+1
126 06E1A 119          C=R1          C(W) = Di
127 06E1D 79B0        GOSUB mp2-12  A(W) = ai+1 * Di
128 06E21 11A          C=R2          C(W) = Di-1
129 06E24 74F0        GOSUB AD          A(W) = Di+1 = ai+1*Di+Di-1
130 06E28 121          AR1EX         R1= Di+1  A(W) = Di
131 06E2B 102          R2=A          R2 = Di replace Di-1
132 06E2E 73B0        GOSUB AR3        X = ABS(Ni+1) -> scr.mstack
133 06E32 AD4          A=B M          A(W) = ABS(Ni+1)
134 06E35 119          C=R1          C(W) = Di+1
135 06E38 8F00000     GOSBVL =DV2-12  X = ABS(Ni+1) / Di+1
136 06E3F 73B0        GOSUB ROUND     A(W) = ABS(Ni+1)/Di+1
137          *          *          arrondi È 12 digits
138          *          *          scrmstk -> ABS(Ni+1) -> R3
139 06E43 695F        GOTO LOOP
140
141 06E47 113          OUT  A=R3          A(W) = ABS(Ni+1)
142 06E4A 1577          C=DAT1 W        C(S) = signe de N
143 06E4E ACA          A=C S          A(W) = Ni+1
144 06E51 103          R3=A          R3 = Ni+1
145 06E54 1B00000     D0=(5) =FUNCD0 rÅcupÅration de D0
146 06E5B 146          C=DAT0 A
147 06E5E 134          D0=C
148 06E61 119          C=R1          C(W) = Di+1
149 06E64 2E          P= 14
150 06E66 A0E          C=C-1 P
151 06E69 A7E          C=C-1 W        si Di+1 = 1 alors carry
152 06E6C 20          P= 0          nÅcessaire pour STR$00
153 06E6E 841          ST=0 1        les blancs sont supprimÅs
154 06E71 5A0          GONC DIF1     Di+1 # 1 alors rÅsultat
155          *          *          avec Di+1
156 06E74 72B0        GOSUB A2STR    Ni+1 ou N en chaÑne
157 06E78 6630        GOTO expr     et fin
158
159 06E7C 111          DIF1 A=R1          A(W) = Di+1
160 06E7F 7330        GOSUB REVST    Di+1 sur la math stack ->
161          *          *          chaÑne alpha inversÅe
162          *          *          -> en-tÅte enlevÅe
163 06E83 119          C=R1          C(A) = D1 (fin de chaÑne)
164 06E86 06          RSTK=C        sauve D1 sur pile retours
165 06E88 31F2        LCASC '/'      C(B) = /
166 06E8C 1C1          D1=D1- 2      prÅpare la pile È recevoir
167          *          *          /
168 06E8F 14D          DAT1=C B      / sur la pile
169 06E92 113          A=R3          A(W) = Ni+1
170 06E95 1CF          D1=D1- 16     prÅpare la pile È recevoir
171          *          *          Ni+1

```

```

172 06E98 7A10          GOSUB  REVST  Ni+1 -> pile -> chaÑÑe
173                   *      *      alpha -> chaÑÑe inversÅe
174                   *      *      -> entÁte enlevÅe
175 06E9C 07           C=RSTK   on rÅcupÉre D1 (fin de
176                   *      *      chaÑÑe
177 06E9E 109          R1=C     R1(A) = D1 (fin de chaÑÑe)
178                   *      *      nÅcessaire pour ADHEAD
179 06EA1 8F00000       GOSBVL =ADHEAD ajoute l'entÁte
180 06EA8 8F00000       GOSBVL =REV$   renverse la chaÑÑe
181 06EAF 8D00000  expr  GOVLNG =EXPR  rÅsultat et retour au BASIC
182
183 06EB6 7070          REVST  GOSUB  A2STR  converti A(W)
184 06EBA 8F00000       GOSBVL =REV$   chaÑÑe inversÅe
185 06EC1 20            P=      0      nÅcessaire pour XXHEAD
186 06EC3 8D00000       GOVLNG =XXHEAD en-tÁte enlevÅe
187
188 06ECA 7A10          STOfrfc GOSUB  STO   A(W) -> scr. math st.
189 06ECE 8F00000  clrfrc GOSBVL =CLRFRFC X = IP(X)
190 06ED5 AD4           A=B     A=B     M     X -> A(W)
191 06ED8 01            RTN
192
193 06EDA 8F00000  mp2-12 GOSBVL =MP2-12 X = A(W) * C(W)
194 06EE1 63FF          GOTO   A=B     X -> A(W) et retour
195
196 06EE5 113           AR3     A=R3     A(W) = ABS(Ni)
197 06EE8 8F00000       STO     GOSBVL =SPLITA A(W) -> X
198 06EEF 8D00000  stscr  GOVLNG =STSCR  X -> scr. math st.
199
200 06EF6 8F00000  ROUND  GOSBVL =uRES12 X 15 digits -> C(W) arrondi
201                   *      *      È 12 digits
202 06EFD AFA           A=C     W
203 06F00 8F00000       GOSBVL =RCSCR  scr.mstack -> ABS(Ni) -> Y
204 06F07 ADB           C=D     M     Y -> C(W)
205 06F0A 10B          R3=C     ABS(Ni) -> R3
206 06F0D 01            RTN
207
208 06F0F A4C           A-1S   A=A-1   S     positif -> nÅgatif
209 06F12 8F00000       GOSBVL =RCLW2  X = 2e niveau scr.mstack
210 06F19 AD4           A=B     M     X -> A(W)
211 06F1C 8F00000  AD     GOSBVL =AD2-12 X = A(W) + C(W)
212 06F23 AC0           A=0     S     valeur absolue
213 06F26 6EAF          GOTO   A=B     X -> A(W) et retour
214
215 06F2A 1517          A2STR  DAT1=A W   nombre sur la pile
216 06F2E 133          AD1EX  AD1EX      sauve D1
217 06F31 1F00000       D1=(5) =DSPFMT
218 06F38 1574          C=DAT1 S     C(S) = format courant
219 06F3C 10C           R4=C     sauve dans R4(S)
220 06F3F D2            C=0     A     impose le ..
221 06F41 15D0          DAT1=C 1     .. format STD
222 06F45 133          AD1EX  AD1EX      restaure D1
223 06F48 8F00000       GOSBVL =STR$SB fait la conversion
224 06F4F 133          AD1EX  AD1EX      sauve D1
225 06F52 1F00000       D1=(5) =DSPFMT
226 06F59 11C           C=R4
227 06F5C 1554          DAT1=C  S     restaure le format courant
228 06F60 133          AD1EX  AD1EX      restaure D1

```

Saturn Assembler JPC LEX: MODULE FRAC
Ver. 1.56, 12/20/89

Tue May 23 09:21:17 2006
frac.a Page 5

229 06F63 01 RTN
230
231 06F65 END

1/X15	Ext		-	108			
A-1S	Abs	28431	#00006F0F	-	208	89	106
A0S	Abs	27982	#00006D4E	-	43	40	
A2STR	Abs	28458	#00006F2A	-	215	156	183
A=B	Abs	28373	#00006ED5	-	190	194	213
AD	Abs	28444	#00006F1C	-	211	119	129
AD2-12	Ext		-	211			
ADHEAD	Ext		-	179			
AR3	Abs	28389	#00006EE5	-	196	102	132
CLRFRC	Ext		-	189			
DIF1	Abs	28284	#00006E7C	-	159	154	
DSPFMT	Ext		-	217	225		
DV2-12	Ext		-	135			
EXPR	Ext		-	181			
=FRACe	Abs	27921	#00006D11	-	23		
FUNCD0	Ext		-	24	145		
ITER	Abs	28104	#00006DC8	-	102	88	
LOOP	Abs	28061	#00006D9D	-	79	139	
MP2-12	Ext		-	193			
OUT	Abs	28231	#00006E47	-	141	105	
PARAM	Abs	28036	#00006D84	-	66	54	60
POP1R	Ext		-	36	52		
POP1	Abs	28005	#00006D65	-	52	35	
PREC	Abs	28081	#00006DB1	-	89	80	
RCLW1	Ext		-	124			
RCLW2	Ext		-	103	209		
RCSCR	Ext		-	112	203		
REV\$	Ext		-	180	184		
REVST	Abs	28342	#00006EB6	-	183	160	172
ROUND	Abs	28406	#00006EF6	-	200	110	136
SPLITA	Ext		-	107	197		
STO	Abs	28392	#00006EE8	-	197	72	188
STOfrc	Abs	28362	#00006ECA	-	188	113	
STR\$SB	Ext		-	223			
STSCR	Ext		-	198			
TST12A	Ext		-	100			
XXHEAD	Ext		-	186			
aslc2	Abs	27993	#00006D59	-	47	45	
clrfrc	Abs	28366	#00006ECE	-	189	76	104 125
expr	Abs	28335	#00006EAF	-	181	157	
mp2-12	Abs	28378	#00006EDA	-	193	117	127
out	Abs	28119	#00006DD7	-	105	84	101
stscr	Abs	28399	#00006EEF	-	198	74	
uRES12	Ext		-	200			

Input Parameters

Source file name is frac.a
Listing file name is frac.l
Object file name is frac.o
Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE GRAPH
2
3          * *****
4          * Graphic commands for the GRAPH program
5          *   (graphics on ThinkJet)
6          *
7          * Keyword(s):
8          *   GPSET, GLINE
9          * Author(s):
10         *   Pierre David
11         * History:
12         *   Published in JPC #35
13         *
14         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
15         * *****
16
17 06F65          ABS          #6F65
18
19         * *****
20         * rndahx
21         *
22         * but: Åvaluer le parametre au sommet de la math-
23         *   stack
24         * entrÅe:
25         *   - D1 ^ sommet de la math-stack
26         * sortie:
27         *   - A(A) = valeur du paramÅtre, en hexa
28         *   - D1 ^ paramÅtre suivant
29         *   - mode = HEX
30         *   - P=0
31         * appelle: RNDAHX
32         * note: si le paramÅtre est nÅgatif, une erreur
33         *   ARGERR est gÅnÅrÅe
34         * *****
35 06F65 8F00000  rndahx GOSBVL =RNDAHX  A(A) := param. en hexa
36 06F6C 570      GONC  argerr  ParamÅtre < 0
37 06F6F 17F      D1=D1+ 16
38 06F72 01      RTN
39 06F74 8D00000  argerr GOVLNG =ARGERR
40
41
42         * *****
43         * initgr
44         *
45         * but: initialiser le nÅcessaire pour l'exÅcution
46         *   des ordres PSET et LINE
47         * entrÅe:
48         *   - D1 ^ sommet de la math-stack
49         * sortie:
50         *   - D0 = adresse de la chaÑne
51         *   - A(A) = longueur de la chaÑne en pixels
52         * *****
53 06F7B 8F00000  initgr GOSBVL =EXPEX-  Evaluation des paramÅtres
54 06F82 32760    LC(3)  'g'      Code de la variable
55 06F87 AB5      B=C      X      transfÅrÅ dans B(X)
56 06F8A 8F00000  GOSBVL =ADRS50
57 06F91 412      GOC      invvar  variable non trouvÅe
```

```

58 06F94 16A          D0=D0+ 11      on cherche l'adresse de la
59 06F97 142          A=DAT0 A       chaÑne (ad. relative)
60 06F9A 136          CD0EX
61 06F9D E2           C=C-A  A       calcul de l'adresse rÅelle
62 06F9F 134          D0=C           D0 := adresse de la chaÑne
63 06FA2 D0           A=0    A
64 06FA4 15A3         A=DAT0 4      A(A) := LEN(G$)
65 06FA8 C4           A=A+A  A
66 06FAA C4           A=A+A  A
67 06FAC C4           A=A+A  A       longueur en pixels
68 06FAE 163          D0=D0+ 4      D0 := adresse des donnÅes
69 06FB1 01           RTN          dans la chaÑne
70 06FB3 3135         invvar LC(2) 83
71 06FB7 8D00000     GOVLNG =MFERR
72
73
74
75
76
77 06FBE 00000        REL(5) =STDd
78 06FC3 00000        REL(5) =FIXp
79
80
81
82
83
84 06FC8              =GPSETe
85 06FC8 7FAF         GOSUB  initgr  initialise le graphique
86 06FCC 100          R0=A          sauvegarde de la longueur
87 06FCF 729F         GOSUB  rndahx
88 06FD3 118          C=R0          C(A) := xmax
89 06FD6 8AC          ?A#0  A
90 06FD9 70           GOYES  PSET10
91 06FDB E4           A=A+1  A       si x = 0 alors x := 1
92 06FDD 590          GONC   PSET20  B.E.T.
93 06FE0 8BA          PSET10 ?A<=C  A
94 06FE3 40           GOYES  PSET20
95 06FE5 DA           A=C    A       si x > 640 alors x := 640
96 06FE7 7400         PSET20 GOSUB  pset
97 06FEB 6D31         GOTO   LINE99  NXTSTM
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112 06FEF CC          pset  A=A-1  A      x := numÅro - 1
113 06FF1 D8          B=A    A
114 06FF3 AF0         A=0    W

```



```

115 06FF6 D4          A=B      A      A := 00000000000xxxxx
116 06FF8 81C        ASRB
117 06FFB 81C        ASRB
118 06FFE 81C        ASRB      A := x div 8
119 07001 D2          C=0      A
120 07003 307        LC(1)    7
121 07006 0EF1       B=B&C    A      B := x mod 8
122 0700A ED         B=C-B    A      C := 7 - (x mod 8)
123 0700C 31F4       LC(2)    79
124 07010 E2         C=C-A    A      C := offset de x en octets
125 07012 C6         C=C+C    A      C := " " " quartets
126 07014 132        AD0EX
127 07017 C2         C=C+A    A      C := adresse rAelle de x
128 07019 134        D0=C
129 0701C AE2        C=0      B
130 0701F B66        C=C+1    B
131 07022 6600       GOTO     pset20
132 07026 A66        pset10   C=C+C    B
133 07029 CD         pset20   B=B-1    A
134 0702B 5AF        GONC     pset10
135 0702E AE5        B=C      B
136 07031 14E        C=DAT0   B
137 07034 0E6D       C=C!B    B      Allumage du point
138 07038 14C        DAT0=C   B
139 0703B 130        D0=A
140 0703E 01         RTN      Restauration ancien D0
141
142
143
144 * *****
145 * GLINE entry
146 * *****
146 07040 00000       REL(5)  =STDd
147 07045 00000       REL(5)  =GLINEp
148 *****
149 * GLINEe
150 *
151 * but: tracer une ligne, pour le programme GRAPH
152 *****
153 0704A             =GLINEe
154 0704A 7D2F        GOSUB   initgr   initialisation du graphique
155 0704E 102         R2=A      R2 := LEN(G$)
156 07051 136        CD0EX     C := adresse de G$
157 07054 108        R0=C      R0 := @ G$
158 07057 305        LC(1)    5      5 paramÉtres È stocker
159 0705A 1B00000    D0=(5)  =STMTR0  a partir de STMTR0
160 07061 6310       GOTO     LINE20
161
162 07065 109        LINE10   R1=C      sauvegarde du compteur
163 07068 79FE       GOSUB   rndahx
164 0706C 140        DAT0=A    A      sauvegarde du paramÉtre
165 0706F 164        D0=D0+  5
166 07072 119        C=R1
167 07075 A0E        LINE20   C=C-1    P      restauration du compteur
168 07078 5CE        GONC     LINE10  tant que compteur >= 0
169
170
171 * 0 + STMTR0 : Åcart

```

```

172          * 5 + STMTR0 : taille
173          * 10 + STMTR0 : premier incrément
174          * 15 + STMTR0 : longueur
175          * 20 + STMTR0 : x
176
177 0707B 8AC          ?A#0 A x
178 0707E 40          GOYES LINE30
179 07080 E4          A=A+1 A si x = 0 alors x := 1
180 07082 118        LINE30 C=R0 C = adresse de G$
181 07085 134          D0=C D0 = " "
182 07088 101          R1=A R1 := valeur courante de x
183 0708B 706F        GOSUB pset allumer x
184 0708F 1F00000     D1=(5) 5*3+(=STMTR0) longueur
185 07096 111          A=R1 A := x
186 07099 147          C=DAT1 A C := longueur
187 0709C 8AE          ?C#0 A
188 0709F 70          GOYES LINE35
189 070A1 E6          C=C+1 A si longueur = 0 alors
190 070A3 145        DAT1=C A longueur := 1
191 070A6 CA          LINE35 A=C+A A A := point final + 1
192 070A8 CC          A=A-1 A A := point final
193 070AA 11A          C=R2 C=LEN(G$)
194 070AD 8BA          ?A<=C A
195 070B0 F0          GOYES LINE40
196 070B2 111          A=R1 A := x1
197 070B5 E2          C=C-A A
198 070B7 E6          C=C+1 A nouvelle longueur calculée
199 070B9 145        DAT1=C A
200 070BC 112          A=R2
201 070BF 100        LINE40 R0=A R0 := limite, nommons-la xf
202 070C2 792F        GOSUB pset allumer xf
203 070C6 1C4          D1=D1- 5 D1 = 10 + STMTD0
204 070C9 147          C=DAT1 A C(A) := premier incrément
205 070CC 111          A=R1
206 070CF CA          A=A+C A
207 070D1 101          R1=A
208 070D4 118          C=R0 xf
209 070D7 8BE          ?A>=C A si x+premier incrément>=xf
210 070DA F4          GOYES LINE99 pas la peine de tracer
211 070DC 1C4          D1=D1- 5 D1 = 5 + STMTR0 (taille)
212 070DF 147          C=DAT1 A
213 070E2 10A          R2=C
214 070E5 1C4          D1=D1- 5 D1 = 0 + STMTR0 (écart)
215          * R0 := xf
216          * R1 := x
217          * R2 := taille
218          * R3 := incrément dans la taille
219          * D1 pointe sur l'écart
220 070E8 D2        LINE50 C=0 A boucle externe
221 070EA E6          C=C+1 A le compteur de taille est
222 070EC 10B          R3=C initialisé avec 1
223 070EF 111        LINE60 A=R1 A := x
224 070F2 79FE        GOSUB pset
225 070F6 111          A=R1
226 070F9 E4          A=A+1 A x := x + 1
227 070FB 101          R1=A
228 070FE 118          C=R0

```

```
229 07101 8BE          ?A>=C  A      si x >= x2
230 07104 52          GOYES  LINE99    fini
231 07106 113          A=R3
232 07109 E4          A=A+1  A      compteur taille incrementé
233 0710B 103          R3=A
234 0710E 11A          C=R2      C := taille
235 07111 8B2          ?A<C   A      arriv  a la fin de taille
236 07114 BD          GOYES  LINE60
237 07116 111          A=R1
238 07119 147          C=DAT1 A      C :=  cart
239 0711C CA          A=A+C   A      on fait le grand  cart
240 0711E 101          R1=A     nouveau x
241 07121 118          C=R0     xf
242 07124 8B2          ?A<C   A      si x < xf
243 07127 1C          GOYES  LINE50    on recommence !
244
245 07129 8D00000 LINE99 GOVLNG =NXTSTM
246
247 07130          END
```

ADRS50	Ext		-	56				
ARGERR	Ext		-	39				
EXPEX-	Ext		-	53				
FIXp	Ext		-	78				
=GLINEe	Abs	28746	#0000704A	-	153			
GLINEp	Ext			-	147			
=GPSETe	Abs	28616	#00006FC8	-	84			
LINE10	Abs	28773	#00007065	-	162	168		
LINE20	Abs	28789	#00007075	-	167	160		
LINE30	Abs	28802	#00007082	-	180	178		
LINE35	Abs	28838	#000070A6	-	191	188		
LINE40	Abs	28863	#000070BF	-	201	195		
LINE50	Abs	28904	#000070E8	-	220	243		
LINE60	Abs	28911	#000070EF	-	223	236		
LINE99	Abs	28969	#00007129	-	245	97	210	230
MFERR	Ext			-	71			
NXTSTM	Ext			-	245			
PSET10	Abs	28640	#00006FE0	-	93	90		
PSET20	Abs	28647	#00006FE7	-	96	92	94	
RNDAHx	Ext			-	35			
STDD	Ext			-	77	146		
STMTR0	Ext			-	159	184		
argerr	Abs	28532	#00006F74	-	39	36		
initgr	Abs	28539	#00006F7B	-	53	85	154	
invvar	Abs	28595	#00006FB3	-	70	57		
pset	Abs	28655	#00006FEF	-	112	96	183	202 224
pset10	Abs	28710	#00007026	-	132	134		
pset20	Abs	28713	#00007029	-	133	131		
rndahx	Abs	28517	#00006F65	-	35	87	163	

Input Parameters

Source file name is graph.a

Listing file name is graph.l

Object file name is graph.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE HMS
2
3          * *****
4          * Time conversion and calculation functions
5          *
6          * Keyword(s):
7          *   HR, HMS, HMSSUB, HMSADD
8          * Author(s):
9          *   Michel Martinet, Guy Toublanc
10         * History:
11         *   Published in JPC #50
12         *
13         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14         * *****
15
16 07130          ABS      #7130
17
18         * *****
19         * HR entry
20         * *****
21 07130 811          NIBHEX 811
22 07133 840        =HRe   ST=0  0
23 07136 580          GONC   ST1
24
25         * *****
26         * HMS entry
27         * *****
28 07139 811          NIBHEX 811
29 0713C 850        =HMSe  ST=1  0
30 0713F 841        ST1    ST=0  1
31 07142 852          ST=1  2
32 07145 AF0        A0W    A=0   W
33 07148 AF1          B=0   W
34 0714B 8F00000    GOSBVL =STAB1
35 07152 8F00000    POPN   GOSBVL =ARGPRP
36 07159 872          ?ST=1 2
37 0715C 80          GOYES  CONV
38 0715E BCC        A=-A-1 S
39 07161 852          ST=1  2
40 07164 7631       CONV   GOSUB  stab2
41 07168 7061          GOSUB  rccd1
42 0716C 8F00000    GOSBVL =STAB1
43 07173 7531       GOSUB  exab2
44 07177 E4          A=A+1  A
45 07179 E4          A=A+1  A
46 0717B 870        ?ST=1  0
47 0717E 60          GOYES  six
48 07180 7A11       GOSUB  stab2
49 07184 71F0       six    GOSUB  SIX
50 07188 860        ?ST=0  0
51 0718B 60          GOYES  exa
52 0718D 7D01       GOSUB  stab2
53 07191 7711       exa    GOSUB  exab2
54 07195 870        ?ST=1  0
55 07198 60          GOYES  si
56 0719A 7BD0       GOSUB  SIX
57 0719E 77D0       si     GOSUB  SIX
```

```
58 071A2 8F00000      GOSBVL =RCCD2
59 071A9 7621         GOSUB  ad2-15
60 071AD CC          A=A-1  A
61 071AF CC          A=A-1  A
62 071B1 7711         GOSUB  rccd1
63 071B5 871         ?ST=1  1
64 071B8 00          RTNYES
65 071BA 8F00000      GOSBVL =uRES12
66 071C1 870         ?ST=1  0
67 071C4 90          GOYES  RNDTST
68 071C6 8D00000  OUT GOVLNG =FNRTN4
69
70
71 071CD 108         RNDTST R0=C
72 071D0 10C         R4=C
73 071D3 110         A=R0
74 071D6 AC0        A=0    S
75 071D9 AF2        C=0    W
76 071DC 2E         P=     14
77 071DE 306        LCHEX  6
78 071E1 A3E        C=C-1  X
79 071E4 109        R1=C
80 071E7 B34        A=A+1  X
81 071EA B34        A=A+1  X
82 071ED B34        A=A+1  X
83 071F0 B34        A=A+1  X
84 071F3 7CE0       GOSUB  IP
85 071F7 5B1        GONC   FP
86 071FA 7CD0       GOSUB  DV/100
87 071FE 460        GOC    NOFP
88 07201 7EE0       GOSUB  TEST
89 07205 71D0       NOFP   GOSUB  DV/100
90 07209 490        GOC    FP
91 0720C 73E0       GOSUB  TEST
92 07210 100        R0=A
93 07213 118        FP     C=R0
94 07216 114        A=R4
95 07219 AC6        C=A    S
96 0721C 69AF       GOTO   OUT
97
98
99
100
101
102 07220 8822       NIBHEX 8822
103 07224           =HMSSUe
104 07224 842       ST=0   2
105 07227 511       GONC   ST
106
107
108
109
110
111 0722A 88888888  NIBHEX 88888888882A
      882A
112 07236           =HMSADe
113 07236 852       ST=1   2
```

```
114 07239 851      ST      ST=1  1
115 0723C 840              ST=0  0
116 0723F A4E              C=C-1  S
117 07242 A4E              C=C-1  S
118 07245 10C              R4=C
119 07248 79FE              GOSUB  A0W
120 0724C 8F00000 STA      GOSBVL  =STAB1
121 07253 17F              D1=D1+ 16
122 07256 78FE              GOSUB  POPN
123 0725A 11C              C=R4
124 0725D A4E              C=C-1  S
125 07260 10C              R4=C
126 07263 58E              GONC   STA
127 07266 850              ST=1   0
128 07269 AF2              C=0    W
129 0726C 108              R0=C
130 0726F 109              R1=C
131 07272 841              ST=0   1
132 07275 6EEE              GOTO   CONV
133 07279 AF2      SIX     C=0    W
134 0727C 2E              P=     14
135 0727E 306              LCHEX  6
136 07281 AF7              D=C    W
137 07284 A82              C=0    P
138 07287 860              ?ST=0 0
139 0728A B0              GOYES  dv
140 0728C CC              A=A-1  A
141 0728E 8D00000 mp2-15 GOVLNG  =MP2-15
142 07295 E4      dv      A=A+1  A
143 07297 8D00000      GOVLNG  =DV2-15
144
145 0729E 8F00000 stab2  GOSBVL  =STAB2
146 072A5 8D00000 clrfrc GOVLNG  =CLRFRFC
147
148 072AC 8F00000 exab2  GOSBVL  =EXAB2
149 072B3 1517      frac15 DAT1=A  W
150 072B7 AF9              C=B    W
151 072BA AF7              D=C    W
152 072BD 74EF              GOSUB  clrfrc
153 072C1 BCC              A=-A-1 S
154 072C4 1577              C=DAT1 W
155 072C8 6A00              GOTO   ad2-15
156
157 072CC 8F00000 rccd1  GOSBVL  =RCCD1
158 072D3 8D00000 ad2-15 GOVLNG  =AD2-15
159
160 072DA A3C      DV/100 A=A-1  X
161 072DD A3C              A=A-1  X
162 072E0 100              R0=A
163 072E3 8F00000 IP      GOSBVL  =SPLITA
164 072EA 77BF              GOSUB  clrfrc
165 072EE AD4      A=B     A=B    M
166 072F1 01              RTN
167
168 072F3 120      TEST   AR0EX
169 072F6 8F00000      GOSBVL  =SPLITA
170 072FD 72BF              GOSUB  frac15
```



```
171 07301 AD4          A=B      M
172 07304 119          C=R1
173 07307 976          ?A#C     W
174 0730A C1           GOYES    NO60
175 0730C AF0          A=0      W
176 0730F 120          AR0EX
177 07312 8F00000      GOSBVL  =SPLITA
178 07319 8F00000      GOSBVL  =ADDONE
179 07320 AD4          A=B      M
180 07323 120          AR0EX
181 07326 118          NO60    C=R0
182 07329 8F00000      GOSBVL  =AD2-12
183 07330 6DBF         GOTO     A=B
184
185 07334              END
```

A0W	Abs	28997	#00007145	-	32	119		
A=B	Abs	29422	#000072EE	-	165	183		
AD2-12	Ext			-	182			
AD2-15	Ext			-	158			
ADDONE	Ext			-	178			
ARGPRP	Ext			-	35			
CLRFRC	Ext			-	146			
CONV	Abs	29028	#00007164	-	40	37	132	
DV/100	Abs	29402	#000072DA	-	160	86	89	
DV2-15	Ext			-	143			
EXAB2	Ext			-	148			
FNRTN4	Ext			-	68			
FP	Abs	29203	#00007213	-	93	85	90	
=HMSADe	Abs	29238	#00007236	-	112			
=HMSSUe	Abs	29220	#00007224	-	103			
=HMSe	Abs	28988	#0000713C	-	29			
=HRe	Abs	28979	#00007133	-	22			
IP	Abs	29411	#000072E3	-	163	84		
MP2-15	Ext			-	141			
NO60	Abs	29478	#00007326	-	181	174		
NOFP	Abs	29189	#00007205	-	89	87		
OUT	Abs	29126	#000071C6	-	68	96		
POPNI	Abs	29010	#00007152	-	35	122		
RCCD1	Ext			-	157			
RCCD2	Ext			-	58			
RNDTST	Abs	29133	#000071CD	-	71	67		
SIX	Abs	29305	#00007279	-	133	49	56	57
SPLITA	Ext			-	163	169	177	
ST	Abs	29241	#00007239	-	114	105		
ST1	Abs	28991	#0000713F	-	30	23		
STA	Abs	29260	#0000724C	-	120	126		
STAB1	Ext			-	34	42	120	
STAB2	Ext			-	145			
TEST	Abs	29427	#000072F3	-	168	88	91	
ad2-15	Abs	29395	#000072D3	-	158	59	155	
clrfrc	Abs	29349	#000072A5	-	146	152	164	
dv	Abs	29333	#00007295	-	142	139		
exa	Abs	29073	#00007191	-	53	51		
exab2	Abs	29356	#000072AC	-	148	43	53	
frac15	Abs	29363	#000072B3	-	149	170		
mp2-15	Abs	29326	#0000728E	-	141			
rccd1	Abs	29388	#000072CC	-	157	41	62	
si	Abs	29086	#0000719E	-	57	55		
six	Abs	29060	#00007184	-	49	47		
stab2	Abs	29342	#0000729E	-	145	40	48	52
uRES12	Ext			-	65			

Input Parameters

Source file name is hms.a

Listing file name is hms.l

Object file name is hms.o

Flags set on command line
None

Errors

None

```
1 TITLE JPC LEX: MODULE KA
2
3 * *****
4 * KA function
5 *
6 * Keyword(s):
7 * KA
8 * Author(s):
9 * Pierre David
10 * History:
11 * Source file from an early version (private communication)
12 * and updated from actual JPC LEX code disassembly
13 * R. Rosenbaum, rev X:
14 * change of the meaning of "." at the end of a search line
15 *
16 * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
17 * *****
18
19 07334 ABS #7334
20
21 * local symbols:
22 point EQU 0
23 slash EQU 1
24 trouve EQU 9
25 chpinf EQU 10
26 CHPMAX EQU 7
27
28 * ADRS file type
29 fADRS EQU #E224
30
31 RESJPC EQU #2F991 JPC reserved location
32
33 * KA support routines
34
35 07334 1B00000 seekC D0=(5) =S-R1-2 No de la ligne pointÃe par STMTD0
36 0733B 142 A=DAT0 A
37 0733E 8BA ?C>=A A No demandÃ >= No dÃja pointÃ ?
38 07341 A1 GOYES seek10
39 07343 1900 D0=(2) =STMTD1 Non: on repart du dÃbut.
40 07347 142 A=DAT0 A
41 0734A 132 AD0EX
42 0734D 16B D0=D0+ 12 DÃbut des donnÃes de la ligne No 1
43 * (passÃe la long.)
44 07350 132 AD0EX
45 07353 184 D0=D0- 5
46 07356 140 DAT0=A A Nouvelle adresse de la ligne
47 07359 D0 A=0 A Ligne No 0
48 0735B DE seek10 ACEX A -> A=No Ã atteindre. C=No dÃjÃ acquis.
49 0735D EA A=A-C A Nb de boucles
50 0735F D8 seekA B=A A ds A: Nb de lignes Ã sauter
51 07361 1B00000 D0=(5) =STMTD0
52 07368 146 C=DAT0 A
53 0736B 134 D0=C
54 0736E 183 D0=D0- 4 Pointe sur la long.
55 07371 501 GONC seek30 B.E.T.
56 07374 D2 seek20 C=0 A
57 07376 15E3 C=DAT0 4
```

```

58 0737A 132          AD0EX
59 0737D C2          C=C+A  A
60 0737F 134          D0=C          D0=dÅbut de la ligne
61 07382 CD          seek30 B=B-1  A
62 07384 5FE          GONC  seek20
63 07387 163          D0=D0+ 4      DÅbut des donnÅes
64 0738A 136          CD0EX
65 0738D 134          D0=C          Pour avoir la rÅponse dans D0 et C...
66 07390 01          RTN
67
68 07392 1B00000  afch05 D0=(5) =STMTD0
69 07399 146          C=DAT0  A
70 0739C 134          D0=C          Adresse de la ligne courante
71 0739F 6D00          GOTO  afch20
72 073A3 C6          afch10 C=C+C  A      Taille du champ en quartets
73 073A5 132          AD0EX
74 073A8 CA          A=A+C  A      Adresse de la long. du champ suivant
75 073AA 130          D0=A
76 073AD D2          afch20 C=0    A
77 073AF 14E          C=DAT0  B      Long. du champ en octets
78 073B2 161          D0=D0+ 2
79 073B5 A4E          C=C-1  S
80 073B8 5AE          GONC  afch10
81 073BB 01          RTN          En sortie: C(B)=long. du champ
82          *          È afficher
83
84 073BD 1B00000  affchp D0=(5) =S-R1-3  Afficher le champ S-R1-3
85 073C4 1564          C=DAT0  S
86 073C8 76CF          GOSUB  afch05
87 073CC 132          AD0EX
88 073CF 131          D1=A          Pour compatibilitÅ avec DSPCNA
89 073D2 DA          A=C    A
90 073D4 AE6          dspcna C=A    B
91 073D7 D0          dspcnc A=0   A
92 073D9 AEA          A=C    B
93 073DC 8D00000  GOVLNG =DSPCNA
94
95 073E3 1B00000  chpedt D0=(5) =STMTR1  Sortie: D0 pointe le champ dont le No
96          *          est ds C(A)
97 073EA 146          C=DAT0  A
98 073ED 816          CSRC          C(S)=No du champ
99 073F0 7264          GOSUB  ldcset
100 073F4 163          D0=D0+ 4      Debut (longueur) du premier champ
101 073F7 65BF          GOTO  afch20
102
103 073FB 23          o73B7 P= 3
104 073FD 8F00000  GOSBVL =R<RSTK
105 07404 7701          GOSUB  prpstr
106 07408 1B00000  D0=(5) =STMTR1
107 0740F 146          C=DAT0  A      C(A)=No du champ
108 07412 8AE          ?C#0  A
109 07415 C2          GOYES  edt095
110 07417 7351          GOSUB  find/
111 0741B 452          GOC    edt095
112 0741E 1B00000  D0=(5) =FUNCD0
113 07425 146          C=DAT0  A
114 07428 E6          C=C+1  A

```

```

115 0742A 144          DAT0=C A          1 caractÉre de plus
116 0742D C6          C=C+C A
117 0742F 164          D0=D0+ 5
118 07432 142          A=DAT0 A          Adresse de la chaÑne sur la M.S.
119 07435 EA          A=A-C A          Adresse de la fin de la chaÑne
120 07437 131          D1=A
121 0743A 31F2         LCASC '/'
122 0743E 14D          DAT1=C B          Et hop !
123 07441 7E9F         edt095 GOSUB chpedt D0 ^ chp È remplacer
124 07445 1F00000      D1=(5) =STMTR0
125 0744C 14D          DAT1=C B          Longueur de l'ancien champ
126 0744F 171          D1=D1+ 2
127 07452 132          AD0EX
128 07455 141          DAT1=A A          Adresse du champ en question
129 07458 1D00         D1=(2) =FUNCD0
130 0745C 143          A=DAT1 A          Taille de la chaÑne sur la M.S.
131                    *          (en octets)
132 0745F EE           C=A-C A          Nouveau - Ancien
133 07461 4A1          GOC edt100        Nouveau < Ancien
134 07464 8AA          ?C=0 A           Nouveau=Ancien : on ne modifie rien.
135 07467 36           GOYES edt150
136 07469 C6           C=C+C A          Taille supplÃmentaire (en octets)
137 0746B 8F00000      GOSBVL =MEMCKL   Memory Check With Leeway
138 07472 590          GONC edt100
139 07475 8D00000      GOVLNG =BSERR
140 0747C 1F00000      edt100 D1=(5) 2+(=STMTR0)
141 07483 147          C=DAT1 A          Adresse du champ
142 07486 1B00000      D0=(5) =FUNCD0
143 0748D 142          A=DAT0 A          Long de la nouvelle ligne en octets
144 07490 C4           A=A+A A          " " quartets
145 07492 C2           C=C+A A          Dest address
146 07494 06           RSTK=C
147 07496 143          A=DAT1 A          Adresse du champ
148 07499 1C1          D1=D1- 2          STMTR0
149 0749C D2           C=0 A
150 0749E 14F          C=DAT1 B          Long de l'ancienne ligne en octets
151 074A1 C6           C=C+C A          " " quartets
152 074A3 CA           A=A+C A          Source address
153 074A5 1F00000      D1=(5) =AVMEMS
154 074AC 147          C=DAT1 A
155 074AF E2           C=C-A A          Block length
156                    * A(A)=source add ; C(A)=Block Length ; RSTK=dest addr
157 074B1 D5           B=C A
158 074B3 07           C=RSTK
159 074B5 74A3         GOSUB move*m
160                    * A(A)=source ; C(A)=dest ; B(A)=length
161 074B9 E2           C=C-A A          dest-source
162 074BB 1F00000      D1=(5) =AVMEMS
163 074C2 143          A=DAT1 A
164 074C5 C2           C=C+A A
165 074C7 145          DAT1=C A          RÅactualisation de AVMEMS
166 074CA 1E0000      edt150 D1=(4) 2+(=STMTR0)
167 074D0 147          C=DAT1 A
168 074D3 134          D0=C
169 074D6 181          D0=D0- 2          Adresse de la longueur du champ
170 074D9 1D00         D1=(2) =FUNCD0
171 074DD 147          C=DAT1 A

```

```

172 074E0 14C          DAT0=C B
173 074E3 AE5          B=C   B      B(B)=Nb de caractÉres È transfÁrer
174 074E6 174          D1=D1+ 5
175 074E9 147          C=DAT1 A
176 074EC 135          D1=C
177 074EF 5E0          GONC   edt170  B.E.T.
178 074F2 1C1          edt160 D1=D1- 2
179 074F5 14F          C=DAT1 B
180 074F8 161          D0=D0+ 2
181 074FB 14C          DAT0=C B
182 074FE A6D          edt170 B=B-1 B
183 07501 50F          GONC   edt160
184 07504 23           P= 3
185 07506 8F00000     GOSBVL =RSTK<R
186 0750D 01          RTN
187
188 0750F 7C06        prpstr GOSUB   finlin
189 07513 850          ST=1   0
190 07516 8F00000     GOSBVL =COLLAP
191 0751D 135          D1=C
192 07520 8F00000     GOSBVL =DSP$00  Construit la chaÑne sur la Math-Stack
193 07527 8F00000     GOSBVL =POP1S  A(A)=longueur en quartets, y compris
194 0752E 137          CD1EX   le CR de la fin.
195 07531 C2          C=C+A   A      C(A)=adresse du premier caractÉre de
196 *                  *                  la chaÑne
197 07533 1B00000     D0=(5) =FUNCD1
198 0753A 144          DAT0=C A      FUNCD1=Adresse du premier caractÉre (-2)
199 0753D 135          D1=C
200 07540 CC          A=A-1   A
201 07542 CC          A=A-1   A      A(A)=longueur en quartets
202 07544 AD2          C=0     M
203 07547 D6          C=A     A
204 07549 81E          CSRB
205 0754C 184          D0=D0- 5      C(A)=longueur en octets
206 0754F 144          DAT0=C A      FUNCD0=longueur de la chaÑne en octets.
207 07552 01          RTN
208
209 07554 D5          uprc$  B=C   A      Entree: C(A)=long. en octets,
210 *                  *                  D1: ^ premier caract.
211 07556 6D00        GOTO   uprc20
212 0755A 8F00000     uprc10 GOSBVL =CNVUCR
213 07561 149          DAT1=A B
214 07564 1C1          uprc20 D1=D1- 2
215 07567 CD          B=B-1   A
216 07569 50F          GONC   uprc10
217 0756C 01          RTN      A(B) contient le dernier caractere
218
219 0756E 1F00000     find/  D1=(5) =FUNCD0
220 07575 143          A=DAT1 A
221 07578 174          D1=D1+ 5
222 0757B 147          C=DAT1 A
223 0757E 135          D1=C
224 07581 D8          B=A     A
225 07583 31F2        LCASC  '/'
226 07587 5A0          GONC   find20  B.E.T.
227 0758A 14B          find10 A=DAT1 B
228 0758D 962          ?A=C   B

```

```
229 07590 00          RTNYES
230 07592 1C1        find20 D1=D1- 2
231 07595 CD         B=B-1  A
232 07597 52F        GONC   find10
233 0759A 03         RTNCC
234
235 0759C 1F00000    SDlsve D1=(5) =STMTD1
236 075A3 147        C=DAT1  A
237 075A6 1CF        D1=D1- 16      S-R1-1
238 075A9 145        DAT1=C  A
239 075AC 01         RTN
240
241 075AE 1F00000    SDlres D1=(5) =S-R1-1
242 075B5 147        C=DAT1  A
243 075B8 17F        D1=D1+ 16
244 075BB 145        DAT1=C  A
245 075BE 01         RTN
246
247 075C0 D2         fkey   C=0    A      Entree: R3(B)=physical key-code
248 075C2 31C1       LC(2)  28
249 075C6 D5         B=C    A
250 075C8 3400000    LC(5)  =KEYBUF
251 075CF DA         A=C    A
252 075D1 E6         C=C+1  A
253 075D3 E6         C=C+1  A
254 075D5 808F       INTOFF
255 075D9 7082       GOSUB  move*m
256 075DD 11B        C=R3
257 075E0 1B00000    D0=(5) =KEYBUF
258 075E7 14C        DAT0=C  B
259 075EA 180        D0=D0- 1
260 075ED 1564       C=DAT0  S
261 075F1 B46        C=C+1  S
262 075F4 460        GOC    fkey10
263 075F7 1544       DAT0=C  S
264 075FB 8080       fkey10 INTON
265 075FF 01         RTN
266
267 07601 1B00000    pass  D0=(5) =DSPCHX  Sortie: Cy=1 D#0 : Non valide
268 07608 146        C=DAT0  A          Cy=1 D=0 : rien rentre
269 0760B D0         A=0    A          Cy=0 : Valide (D(W)=passwd)
270 0760D 140        DAT0=A  A
271 07610 1A0000    D0=(4) =S-R1-0
272 07616 144        DAT0=C  A
273 07619 7C10       GOSUB  pass10
274 0761D B41402     pass05 NIBASC 'KA '
275 07623 07163737   NIBASC 'password'
      77F62746
276 07633 02F302     NIBASC ' ? '
277 07639 07         pass10 C=RSTK
278 0763B 135        D1=C
279 0763E 31E0       LC(2)  ((pass10)-(pass05))/2
280 07642 719D       GOSUB  dspcnc
281 07646 7684       GOSUB  chedit
282 0764A D3         D=0    A
283 0764C CF         D=D-1  A
284 0764E 8AC        ?A#0  A
```



```

285 07651 A2          GOYES  pass30
286 07653 78C4      pass20 GOSUB  finlin
287 07657 1B00000  pass21 D0=(5) =S-R1-0
288 0765E 146          C=DAT0 A
289 07661 1A0000      D0=(4) =DSPCHX
290 07667 144          DAT0=C A
291 0766A 1A0000      D0=(4) =STMTD1
292 07670 146          C=DAT0 A
293 07673 134          D0=C
294 07676 18F          D0=D0- 16
295 07679 02          RTNSC
296 0767B 31D0      pass30 LC(2)  13
297 0767F 966          ?A#C   B
298 07682 1D          GOYES  pass20
299 07684 778E      GOSUB  prpstr  A(A)=long. en quartets, C(A) en octets,
300 * * * * *          D1 pointe sur le 1er caract de la chaÑne
301 07688 D7          D=C    A
302 0768A 8AB          ?D=0   A
303 0768D AC          GOYES  pass21
304 0768F D2          C=0    A
305 07691 3180        LC(2)  8
306 07695 8BF          ?D<=C A
307 07698 60          GOYES  pass35
308 0769A DA          A=C    A
309 0769C C4          A=A+A  A
310 0769E D6      pass35 C=A    A
311 076A0 CE          C=C-1  A
312 076A2 80D0        P=C    0
313 076A6 133          AD1EX
314 076A9 EA          A=A-C  A
315 076AB CC          A=A-1  A
316 076AD 131          D1=A
317 076B0 AF2          C=0    W
318 076B3 1571        C=DAT1 WP
319 076B7 20          P=     0
320 076B9 AF7          D=C    W
321 076BC 779F        GOSUB  pass21
322 076C0 03          RTNCC
323
324 076C2 8A8      Num2D1 ?A=0   A
325 076C5 41          GOYES  num15
326 076C7 8F00000  GOSBVL =HEXDEC  On attend le nb ds A(A)
327 076CE 04          SETHEX  RÅsultat dans A,B,C
328 076D0 27          P=     7          Max=1048575
329 076D2 0D      num10  P=P-1
330 076D4 908          ?A=0   P
331 076D7 BF          GOYES  num10
332 076D9 80CF      num15  C=P    15          C(S)=nb de chiffres a afficher - 1
333 076DD 814      num20  ASRC
334 076E0 0D          P=P-1
335 076E2 88F          ?P#    #F
336 076E5 8F          GOYES  num20
337 076E7 20          P=     0          A(W)=C1 C2 ... Cn 0 ... 0
338 076E9 3103        LCASC  '0'
339 076ED 80DF        P=C    15
340 076F1 810      num30  ASLC
341 076F4 A6A          A=A+C  B

```

```

342 076F7 149          DAT1=A B
343 076FA 171          D1=D1+ 2
344 076FD D0           A=0    A
345 076FF 0D           P=P-1
346 07701 5FE          GONC   num30
347 07704 20           P=    0      En sortie: A(W)=0, P=0, Cy=1
348 07706 01           RTN
349
350 07708 8F00000 keywt GOSBVL =SCRLLR
351 0770F 5D0          GONC   =kwt10
352 07712 8F00000          GOSBVL =CKSREQ
353 07719 6EEF          GOTO   keywt
354 0771D 8F00000 kwt10 GOSBVL =POPBUF
355 07724 7021          GOSUB atnclr  Abime A(A)
356 07728 D4           A=B    A
357 0772A 3136          LC(2) #63
358 0772E 966          ?C#A B
359 07731 00           RTNYES
360 07733 8CA9C0          GOLONG o838B
361
362 07739 1F00000 tststr D1=(5) =FUNCD0
363 07740 147          C=DAT1 A
364 07743 D7           D=C    A      D(A)=long de la chaîne sur la Math-Stack
365 07745 14A          A=DAT0 B
366 07748 D1           B=0    A
367 0774A AE8          B=A    B      B(A)=Long du champ "Nom" pointé par D0
368 0774D 161          D0=D0+ 2
369 07750 174          D1=D1+ 5
370 07753 147          C=DAT1 A
371 07756 135          D1=C
372 07759 849          ST=0   trouve
373 0775C 582          GONC   tst20   B.E.T.
374 0775F 14A          tst10 A=DAT0 B
375 07762 161          D0=D0+ 2
376 07765 1C1          D1=D1- 2
377 07768 861          ?ST=0 slash
378 0776B B0           GOYES  tst15
379 0776D 31F2          LCASC  '/'      Il faut arrêter la comparaison au "/"
380 07771 962          ?A=C    B
381 07774 D1           GOYES  chp<ms  "tot/" < "toto"
382 07776 8F00000 tst15 GOSBVL =CONVUC
383 0777D 14F          C=DAT1 B
384 07780 966          ?A#C    B      On teste enfin les deux chaînes !
385 07783 E2           GOYES  tst50
386 07785 A6F          tst20 D=D-1  B      Fin de la M.S.
387 07788 4D0          GOC    tst30   "    "
388 0778B A6D          B=B-1  B      Fin du champ
389 0778E 50D          GONC   tst10
390
391 07791 85A          chp<ms ST=1  chpinf
392 07794 01           RTN
393 07796 861          tst30 ?ST=0 slash
394 07799 E0           GOYES  tst35
395
396
397
398 0779B 14A          A=DAT0 B

```

```
399 0779E 31F2          LCASC  '/'
400 077A2 962          ?A=C  B      Est-ce un "/"
401 077A5 70           GOYES  chp=ms  Oui: victoire !
402                   * Non: alors, de toute maniÈre, le test suivant est faux.
403 077A7 96D          tst35  ?B#0  B
404 077AA C0           GOYES  chp>ms
405 077AC 859          chp=ms ST=1  trouve
406 077AF 01           RTN
407 077B1 9EA          tst50  ?A<=C  B
408 077B4 DD           GOYES  chp<ms
409 077B6 84A          chp>ms ST=0  chpinf
410 077B9 01           RTN
411
412 077BB 1B00000      search D0=(5) =STMTD0
413 077C2 146          C=DAT0 A
414 077C5 134          D0=C
415 077C8 7D6F         GOSUB  tststr
416 077CC 879          ?ST=1  trouve
417 077CF 00           RTNYES
418 077D1 D2           C=0    A      1Ère fiche
419 077D3 86A          ?ST=0  chpinf  Il faut recommencer depuis la 1Ère
420 077D6 C0           GOYES  srh10
421 077D8 1B00000      D0=(5) =S-R1-2
422 077DF 146          C=DAT0 A
423
424 077E2 06           srh10  RSTK=C
425 077E4 7C4B         GOSUB  seekC
426 077E8 1F00000      D1=(5) =STMTD0
427 077EF 145          DAT1=C A      Nouvelle adresse
428 077F2 07           C=RSTK
429 077F4 1C5          D1=D1- 6      S-R1-2
430 077F7 145          DAT1=C A
431 077FA 7B3F         GOSUB  tststr  Tester le champ pointÅ par D0
432 077FE 879          ?ST=1  trouve
433 07801 00           RTNYES
434 07803 86A          ?ST=0  chpinf
435 07806 00           RTNYES
436 07808 1F00000      D1=(5) =STMTD1
437 0780F 143          A=DAT1 A
438 07812 131          D1=A          D1 ^ nb max de fiches
439 07815 D0           A=0    A
440 07817 15B3         A=DAT1 4      A(A)=Nb max de fiches (1..m)
441 0781B 1F00000      D1=(5) =S-R1-2
442 07822 147          C=DAT1 A      No de la fiche courante.
443 07825 E6           C=C+1  A
444 07827 8B6          ?C<A  A
445 0782A 8B           GOYES  srh10
446 0782C 870          ?ST=1  point   On est arrivÅ È la fin du fichier.
447 0782F 00           RTNYES
448 07831 145          DAT1=C A      Pour l'ajout, on se positionne sur la
449 07834 D0           A=0    A      derniÈre qui n'est pas encore crÅÅe.
450 07836 E4           A=A+1  A
451 07838 732B         GOSUB  seekA   On saute la derniÈre ligne
452 0783C 1F00000      D1=(5) =STMTD0
453 07843 145          DAT1=C A
454 07846 01           RTN
455
```

```

456 07848 8D00000 atnclr GOVLNG =ATNCLR
457 0784F 8D00000 finda GOVLNG =FINDA
458 07856 8D00000 ldcset GOVLNG =LDCSET
459 0785D 8D00000 move*m GOVLNG =MOVE*M
460
461 07864 06          editfc RSTK=C          C=Nb de quartets (fiche);
462                *          D0=^ fiche (avec les 4 q.)
463 07866 8F00000          GOSBVL =OBCOLL AVMEMS=OUTBS (Utilise C(A) D1)
464 0786D 07          C=RSTK
465 0786F 8F00000          GOSBVL =MEMCKL Utilise (A,B,C) (A) D1. A(A)=(AVMEMS)
466 07876 590          GONC      edt005
467 07879 8D00000          GOVLNG =BSERR  Insufficient memory
468 07880 D6          edt005 C=A      A      C=Dest address
469                *          (AVMEMS puisque OBCOLL avant)
470 07882 132          AD0EX          A=source address
471 07885 74DF          GOSUB  move*m
472 07889 C9          C=C+B  A      Dest addr + length in nibs
473 0788B 1F00000          D1=(5) =AVMEMS
474 07892 145          DAT1=C  A
475 07895 D2          C=0      A      No du chp
476 07897 1B00000          D0=(5) =STMTR1
477 0789E 144          DAT0=C  A
478 078A1 1B00000  edt010 D0=(5) =STMTR1  On ne sait pas d'oË on vient
479 078A8 146          C=DAT0  A      C(A)=No du champ È afficher
480 078AB 7060          GOSUB  edt020
481 078AF E416D656          NIBASC 'Name : '
482                02A3
482 078BB 0586F6E6          NIBASC 'Phone: '
483                56A3
483 078C7 14464627          NIBASC 'Addr1: '
484                13A3
484 078D3 14464627          NIBASC 'Addr2: '
485                23A3
485 078DF 14464627          NIBASC 'Addr3: '
486                33A3
486 078EB 14464627          NIBASC 'Addr4: '
487                43A3
487 078F7 E4F64756          NIBASC 'Note : '
488                02A3
488 07903 34279647          NIBASC 'Crit.: '
489                E2A3
489 0790F C6          edt020 C=C+C  A
490 07911 DA          A=C      A
491 07913 C4          A=A+A  A
492 07915 C4          A=A+A  A
493 07917 CA          A=A+C  A
494 07919 CA          A=A+C  A
495 0791B 07          C=RSTK
496 0791D C2          C=C+A  A      Adresse du champ
497 0791F 135          D1=C
498 07922 3160          LC(2) 6
499 07926 7DAA          GOSUB  dspcnc
500 0792A 31E3          LC(2) '>'  Curseur on
501 0792E 8F00000          GOSBVL =ESCSEQ
502 07935 7AAA          edt030 GOSUB  chpedt  En sortie, D0 pointe sur le bon champ
503 07939 132          AD0EX
504 0793C 131          D1=A

```

```

505 0793F 749A          GOSUB  dspcnc  Affichage du champ pointÅ par FUNCD0.
506 07943 8F00000      GOSBVL  =CURSFL
507
508 0794A 1B199F2      edt040  D0=(5) RESJPC
509 07951 1520          A=DAT0 P
510 07955 301          LCHEX   1
511 07958 0E0E          A=A!C P
512 0795C 1500          DAT0=A P
513 07960 1A0000        D0=(4) =S-R1-0
514 07966 146          C=DAT0 A
515 07969 1A0000        D0=(4) =LDCSPC
516 0796F 144          DAT0=C A
517 07972 7A51          GOSUB   chedit
518 07976 D8            B=A     A
519 07978 1B199F2      D0=(5) RESJPC
520 0797F 1520          A=DAT0 P
521 07983 30E          LCHEX   E
522 07986 0E06          A=A&C P
523 0798A 1500          DAT0=A P
524 0798E D4            A=B A
525 07990 8A8          ?A=0   A
526 07993 7B            GOYES  edt040
527 07995 76BE          GOSUB   finda
528 07999 00            CON(2) =kcEOL  13
529 0799B 2A0          REL(3) edtEND
530 0799E 00            CON(2) =kcATTN 14
531 079A0 C80          REL(3) edtATN
532 079A3 00            CON(2) =kcRUN  15
533 079A5 0A0          REL(3) edtRUN
534 079A8 00            CON(2) =kcUP   18
535 079AA 740          REL(3) edtUP
536 079AD 00            CON(2) =kcDOWN 19
537 079AF C50          REL(3) edtDWN
538 079B2 00            CON(2) =kcTOP  20
539 079B4 910          REL(3) edtTOP
540 079B7 00            CON(2) =kcBOT  21
541 079B9 E20          REL(3) edtBOT
542 079BC 00            CON(2) =kcOFF  24
543 079BE 900          REL(3) edtOFF
544 079C1 00            NIBHEX 00
545 079C3 668F          GOTO    edt040
546
547 079C7 8C60A0      edtOFF  GOLONG  o838B
548
549 079CD 7E41          edtTOP  GOSUB   finlin
550 079D1 20            edttop  P=      0
551 079D3 1B00000      setchp  D0=(5) =STMTR1
552 079DA D2            C=0     A
553 079DC 80F0          CPEX    0          C(0)=P ; P=0
554 079E0 144          DAT0=C A
555 079E3 6DBE          GOTO    edt010
556 079E7 7431          edtBOT  GOSUB   finlin
557 079EB 27            edtbot  P=      CHPMAX
558 079ED 65EF          GOTO    setchp
559 079F1 7A21          edtUP   GOSUB   finlin
560 079F5 1B00000      D0=(5) =STMTR1
561 079FC 146          C=DAT0 A

```

```
562 079FF 80D0          P=C      0
563 07A03 0D           P=P-1
564 07A05 4BC          GOC      edttop
565 07A08 5AC          GONC     setchp   B.E.T.
566 07A0B 7011        edtDWN   GOSUB   finlin
567 07A0F 307        edtdwn   LC(1)    CHPMAX
568 07A12 DA           A=C      A
569 07A14 1B00000     D0=(5)  =STMTR1
570 07A1B 146          C=DAT0   A
571 07A1E 902          ?A=C     P           Ici, P vaut 0
572 07A21 AC           GOYES    edtbot
573 07A23 80D0        P=C      0
574 07A27 0C          P=P+1
575 07A29 59A          GONC     setchp   B.E.T.
576
577 07A2C 7EC0        edtATN   GOSUB   attn
578 07A30 560          GONC     edat10
579 07A33 661F        GOTO     edt040
580 07A37 07          edat10   C=RSTK    Pas trEs utile, mais fait plus propre !
581 07A39 6B63        GOTO     BS:
582
583 07A3D 7AB9        edtEND   GOSUB   o73B7
584 07A41 6DCF        GOTO     edtdwn
585
586 07A45 72B9        edtRUN   GOSUB   o73B7
587 07A49 D2           C=0      A
588 07A4B 308          LC(1)    8
589 07A4E DA           A=C     A
590 07A50 1B00000     D0=(5)  =S-R1-0
591 07A57 140          DAT0=A   A
592 07A5A 7589        GOSUB   chpedt
593 07A5E 181          D0=D0-  2
594 07A61 132          AD0EX
595 07A64 7EED        GOSUB   ldcset
596 07A68 EE           C=A-C    A
597 07A6A 15C3        DAT0=C   4
598 07A6E 1F00000     D1=(5)  =AVMEMS
599 07A75 141          DAT1=A   A
600 07A78 132          AD0EX
601 07A7B 130        o7A37   D0=A
602 07A7E 163          D0=D0+  4
603 07A81 D0           A=0      A
604 07A83 14A          A=DAT0   B
605 07A86 1F00000     D1=(5)  =FUNCD0
606 07A8D 141          DAT1=A   A
607 07A90 D8           B=A      A
608 07A92 8F00000     GOSBVL  =COLLAP
609 07A99 1F00000     D1=(5)  =FUNCD1
610 07AA0 147          C=DAT1   A
611 07AA3 135          D1=C
612 07AA6 8F00000     GOSBVL  =D=AVMS
613 07AAD 591          GONC     o7A83
614 07AB0 161        o7A6C   D0=D0+  2
615 07AB3 14A          A=DAT0   B
616 07AB6 8F00000     GOSBVL  =CONVUC
617 07ABD AE6          C=A      B
618 07AC0 8F00000     GOSBVL  =STKCHR
```

```

619 07AC7 CD          o7A83 B=B-1 A
620 07AC9 56E                GONC o7A6C
621 07ACC 08                CLRST
622 07ACE 01                RTN
623
624 07AD0 78CA          chedit GOSUB SD1sve
625 07AD4 24                P= 4
626 07AD6 8F00000        GOSBVL =R<RSTK
627 07ADD 8F00000        GOSBVL =CHEDIT
628 07AE4 440            GOC ched10
629 07AE7 D0              A=0 A
630 07AE9 24          ched10 P= 4
631 07AEB 8F00000        GOSBVL =RSTK<R
632 07AF2 D8              B=A A
633 07AF4 705D            GOSUB atnclr
634 07AF8 D4              A=B A
635 07AFA 63BA            GOTO SD1res
636
637 07AFE 8F00000        attn GOSBVL =D1=AVE Sortie:
638 *                          Cy=0: ATTN doit arreter la commande
639 07B05 850              ST=1 0 Cy=1: ATTN efface la ligne
640 07B08 8F00000        GOSBVL =DSP$00
641 07B0F 8F00000        GOSBVL =POP1S
642 07B16 CC          attn05 A=A-1 A
643 07B18 CC          A=A-1 A
644 07B1A 8AC          ?A#0 A
645 07B1D 90            GOYES attn10
646 07B1F 8D00000        finlin GOVLNG =FINLIN retour avec Cy=0, P=0
647 07B26 8F00000        attn10 GOSBVL =CURSFL
648 07B2D 8F00000        GOSBVL =-LINE
649 07B34 02            RTNSC Cy=1, on a efface la ligne.
650
651 07B36 1B00000        o7AF2 D0=(5) =KEYPTR
652 07B3D D2              C=0 A
653 07B3F 1562            C=DAT0 XS
654 07B43 92E            ?C#0 XS
655 07B46 00              RTNYES
656 07B48 1A064F          D0=(4) #F460
657 07B4E 14C            DAT0=C B
658 07B51 01              RTN
659
660
661
662 * *****
663 * KA entry
664 * *****
664 07B53 00000          REL(5) =KAd
665 07B58 00000          REL(5) =KAp
666 07B5D 14A          =KAe A=DAT0 B
667 07B60 8F00000        GOSBVL =EOLXCK
668 07B67 571          GONC KA20 KA <spÅc. de fichier>
669 07B6A 7C00          GOSUB KA10
670 07B6E 00            CON(2) =tLITRL
671 07B70 14442535        NIBASC 'ADRS'
672 07B78 00            CON(2) =tEOL
673 07B7A 07          KA10 C=RSTK KA tEOL
674 07B7C 134          D0=C
675 07B7F 8F00000        KA20 GOSBVL =FSPECx

```

```

676 07B86 590          GONC   KA30
677 07B89 8D00000    bserr  GOVLNG =BSERR
678
679 07B90 97C        KA30   ?A#0   W
680 07B93 B0          GOYES  KA35
681 07B95 D2          C=0    A
682 07B97 31A3       LC(2)  #3A
683 07B9B 5DE        GONC   bserr
684 07B9E AC2        KA35   C=0    S      Nom de fichier lAgal
685 07BA1 943        ?C=D   S
686 07BA4 85         GOYES  KA40      :MAIN
687 07BA6 B46        C=C+1  S
688 07BA9 943        ?C=D   S
689 07BAC 05         GOYES  KA40      :PORT
690 07BAE AC2        C=0    S
691 07BB1 A4E        C=C-1  S
692 07BB4 943        ?C=D   S
693 07BB7 54         GOYES  KA40
694 07BB9 101        R1=A
695 07BEC 3423000    LC(5)  50
696 07BC3 8F00000    GOSBVL =SALLOC
697 07BCA 111        A=R1
698 07BCD 843        ST=0   3      Source
699 07BD0 8F00000    GOSBVL =SVINF+
700 07BD7 AF0        A=0    W
701 07BDA D3        D=0    A
702 07BDC 853        ST=1   3
703 07BDF 8F00000    GOSBVL =SVINFO
704 07BE6 8F00000    GOSBVL =MGOSUB  Le fichier est copiA derriAre le dernier
705 07BED 00000      CON(5) =COPYu  fichier, mais une config. peut intervenir
706 07BF2 469        GOC    bserr
707 07BF5 119        C=R1
708 07BF8 6C70       GOTO   KA60
709 07BFC 1F00000    KA40   D1=(5) =STMTR0
710 07C03 AFB        C=D    W
711 07C06 1557      DAT1=C W
712 07C0A 17F       D1=D1+ 16
713 07C0D 1517      DAT1=A W
714 07C11 8F00000    GOSBVL =FINDF
715 07C18 137       CD1EX
716 07C1B 595       GONC   KA60
717 07C1E 1F00000    D1=(5) =STMTR0
718 07C25 1577      C=DAT1 W
719 07C29 AF7       D=C    W
720 07C2C D2        C=0    A
721 07C2E 31D3      LC(2)  37+16+2*4
722 07C32 8F00000    GOSBVL =CRETf+  ThAoriq., le Lex KA ne doit pas bouger
723 07C39 457       GOC    Bserr
724 07C3C 119       C=R1
725 07C3F 1F00000    D1=(5) =STMTR1
726 07C46 1537      A=DAT1 W
727 07C4A 135       D1=C
728 07C4D 1517      DAT1=A W      Nom du fichier
729 07C51 17F       D1=D1+ 16
730 07C54 33422E    LC(4)  fADRS
731 07C5A 15D3      DAT1=C 4
732 07C5E AF0       A=0    W

```



```

733 07C61 17F          D1=D1+ 16
734 07C64 174          D1=D1+ 5
735 07C67 1517         DAT1=A W
736 07C6B 17F          D1=D1+ 16
737 07C6E 1597         DAT1=A 8
738 07C72 119          C=R1
739
740 07C75 8F00000 KA60 GOSBVL =LOCADR
741 07C7C 94B          ?D=0 S
742 07C7F 41           GOYES KA65
743 07C81 A4F          D=D-1 S
744 07C84 94B          ?D=0 S
745 07C87 C0           GOYES KA65
746 07C89 D2           C=0 A
747 07C8B 31C3         LC(2) #3C
748 07C8F 6F10         GOTO Bserr
749 07C93 17F          KA65 D1=D1+ 16
750 07C96 D0           A=0 A
751 07C98 D2           C=0 A
752 07C9A 15B3         A=DAT1 4
753 07C9E 33422E       LC(4) fADRS
754 07CA4 8A2          ?A=C A
755 07CA7 C0           GOYES KA70
756 07CA9 33F300       LC(4) 63 Invalid file type
757 07CAF 69DE         Bserr GOTO bserr
758 07CB3 17F          KA70 D1=D1+ 16
759 07CB6 174          D1=D1+ 5
760 07CB9 1537         A=DAT1 W Password
761 07CBD 17F          D1=D1+ 16 D1 ^ Nb max de fiches
762 07CC0 1B00000      D0=(5) =STMTD1 Adresse du dÃbut du fichier.
763 07CC7 137          CD1EX
764 07CCA 144          DAT0=C A STMTD1=bonne valeur=^ Nb max de fiches
765 07CCD 135          D1=C
766 07CD0 17B          D1=D1+ 12
767 07CD3 137          CD1EX Pointe sur le dÃbut des donnÃes
768 *                   * de la 1Ãre ligne
769 07CD6 184          D0=D0- 5 STMTD0
770 07CD9 144          DAT0=C A dÃbut de la ligne pointÃe par D0
771 07CDC 185          D0=D0- 6 No de la ligne courante (S-R1-2)
772 07CDF D2           C=0 A
773 07CE1 144          DAT0=C A S-R1-2 = 0
774 07CE4 1FAF8F2      D1=(5) #2F8FA
775 07CEB D2           C=0 A
776 07CED 307          LC(1) 7
777 07CF0 145          DAT1=C A
778
779 * STMTD1: adresse du fichier
780 * STMTD0: adresse de la ligne courante
781 * S-R1-2: No de la fiche correspondante a la ligne STMTD0
782 * S-R1-3: No du champ (0 Ã 6)
783
784 07CF3 978          ?A=0 W
785 07CF6 81           GOYES BP:
786 07CF8 7509         GOSUB pass
787 07CFC 580          GONC KA80
788 07CFF 8C1A20       nxtstm GOLONG EXIT
789 07D05 1567         KA80 C=DAT0 W

```

```
790 07D09 977          ?C#D  W
791 07D0C 3F          GOYES  nxtstm
792                * Maintenant, le password est bon.
793
794                *****
795                * Boucle Principale
796                *****
797
798 07D0E 7713        BP:    GOSUB  NoFich
799 07D12 555          GONC   BP10   Fichier non vide
800 07D15 7410        FV10   GOSUB  FV50
801 07D19 54D60747  FV20   NIBASC  'Empty '
      9702
802 07D25 6696C656          NIBASC  'file'
803 07D2D 07          FV50   C=RSTK
804 07D2F 135          D1=C
805 07D32 31A0          LC(2)  ((FV50)-(FV20))/2
806 07D36 8EB96F        GOSUBL  dspcnc
807 07D3C 8EDDDF        GOSUBL  finlin
808 07D42 8E0C9F        FV70   GOSUBL  keywt
809 07D48 8E10BF        GOSUBL  finda
810 07D4E 00            CON(2) =k#ATTN #2B      EXIT
811 07D50 252          REL(3)  EXIT
812 07D53 04            CON(2)  #40
813 07D55 262          REL(3)  KEY
814 07D58 B4            CON(2)  #4B
815 07D5A 9F4          REL(3)  INPUT
816 07D5D 00            CON(2) =k#OFF #63
817 07D5F 076          REL(3)  o838B
818 07D62 00            NIBHEX  00
819 07D64 6DDF          GOTO    FV70
820 07D68 8BE          BP10   ?C<=A  A      Fiche courante <= Max ?
821 07D6B 61            GOYES  BP20   On ne change rien
822 07D6D 24            P=     4
823 07D6F B06            C=C+1  P
824 07D72 20            P=     0
825 07D74 460          GOC     BP18   saut si C[4:4]=0, c'èd il valait #F
826 07D77 D0            A=0     A      On ne comprend pas, on remet È 1...
827 07D79 E4            A=A+1   A
828 07D7B D6            BP18   C=A     A      Fiche courante = Nb max
829 07D7D 15D3          DAT1=C  4
830 07D81 06            BP20   RSTK=C
831 07D83 8EBA5F        GOSUBL  seekC
832 07D89 1F00000       D1=(5) =STMTD0
833 07D90 145          DAT1=C  A
834 07D93 07            C=RSTK
835 07D95 1C5          D1=D1- 6      S-R1-2
836 07D98 145          DAT1=C  A
837 07D9B 174          D1=D1+ 5      S-R1-3 = No du champ
838 07D9E AC0           A=0     S
839 07DA1 1514          DAT1=A  S      S-R1-3: chp a afficher
840
841                *****
842                * Boucle Secondaire
843                *****
844 07DA5 7082        BS:    GOSUB  NoFich
845 07DA9 560          GONC   BS05   Fichier non vide
```

```

846 07DAC 686F          GOTO    FV10
847 07DB0 8E706F      BS05    GOSUBL  affchp    Affichage du champ S-R1-3
848 07DB6 8E36DF          GOSUBL  finlin
849 07DBC 8F00000      BS07    GOSBVL  =RPTKY
850 07DC3 480          GOC     BS11
851 07DC6 8EC39F      BS10    GOSUBL  keywt
852 07DCC 8ED7AF      BS11    GOSUBL  finda
853 07DD2 00          CON(2) =k#UP    #32
854 07DD4 302          REL(3) HAUT
855 07DD7 00          CON(2) =k#DOWN #33
856 07DD9 122          REL(3) BAS
857 07DDC 42          CON(2) #24
858 07DDE 382          REL(3) ANCIEN
859 07DE1 52          CON(2) #25
860 07DE3 A82          REL(3) NOUVEL
861 07DE6 2A          CON(2) #A2
862 07DE8 002          REL(3) gHAUT
863 07DEB 3A          CON(2) #A3
864 07DED C22          REL(3) gBAS
865 07DF0 49          CON(2) #94
866 07DF2 162          REL(3) PREM
867 07DF5 59          CON(2) #95
868 07DF7 252          REL(3) DERN
869 07DFA 55          CON(2) #55
870 07DFC 3E4          REL(3) EDIT
871 07DFE B4          CON(2) #4B
872 07E01 254          REL(3) INPUT
873 07E04 B5          CON(2) #5B
874 07E06 572          REL(3) DELETE
875 07E09 65          CON(2) #56
876 07E0B 733          REL(3) CAT
877 07E0E 00          CON(2) =k#ATTN #2B
878 07E10 291          REL(3) EXIT
879 07E13 04          CON(2) #40
880 07E15 2A1          REL(3) KEY
881 07E18 00          CON(2) =k#OFF  #63
882 07E1A 5B5          REL(3) o838B
883 07E1D 00          NIBHEX 00
884
885 07E1F 731D          GOSUB  o7AF2
886 07E23 103          R3=A          Sauvegarde de A(B) (Code physique) en R3
887 07E26 8F00000      GOSBVL  =KEYNAM  Conversion en "Nom de Touche"
888 07E2D 881          ?P#    1          # 1 caractere ?
889 07E30 69          GOYES   BS10
890 07E32 20          P=      0
891          *      LC(4) 256*((CHPMAX)+'0')+'0'
892 07E34 330373        LC(4)  #3730
893 07E3A 8F00000      GOSBVL  =RANGE
894 07E41 460          GOC     BS20      Cy=0 si A(B) in ['0'..' ']
895 07E44 6ED1          GOTO    ChampA
896 07E48 3314A5      BS20    LC(4) 256*'Z'+ 'A'
897 07E4E 8F00000      GOSBVL  =RANGE
898 07E55 560          GONC   BS30      Cy=0 si A(B) in ['A'..'Z']
899 07E58 6D6F          GOTO    BS10
900 07E5C 8EE57F      BS30    GOSUBL  fkey      (Touche R3(B) sauvegardee plus haut)
901 07E62 7010          GOSUB   RECH20
902 07E66 35561627    RECH19  NIBASC  'Search: '

```

```
3686A302
903 07E76 07 RECH20 C=RSTK
904 07E78 135 D1=C
905 07E7B 3180 LC(2) ((RECH20)-(RECH19))/2
906 07E7F 8E255F GOSUBL dspcnc
907 07E85 8E54CF RECH25 GOSUBL chedit
908 07E8B 8A8 ?A=0 A
909 07E8E 7F GOYES RECH25
910 07E90 8E9B9F GOSUBL finda
911 07E96 00 CON(2) =kcATN 14
912 07E98 910 REL(3) RECATN
913 07E9B 00 CON(2) =kcEOL 13
914 07E9D 120 REL(3) RECEND RECherche [ENDline]
915 07EA0 00 CON(2) =kcOFF 24
916 07EA2 900 REL(3) RECOFF
917 07EA5 00 NIBHEX 00
918 07EA7 6DDF GOTO RECH25
919
920 07EAB 8C2250 RECOFF GOLONG o838B
921
922 07EB1 8E74CF RECATN GOSUBL attn
923 07EB7 4DC GOC RECH25
924 07EBA 6AEE bs GOTO BS:
925
926 07EBE 8EB46F RECEND GOSUBL prpstr
927 07EC4 8AA ?C=0 A
928 07EC7 3F GOYES bs
929 07EC9 8E586F GOSUBL uprc$
930 07ECF 08 CLRST
931
932 07ED1 850 *** changes:
933 07ED4 31E2 ST=1 point * added rev X
934 07ED8 966 LCASC '.'
935 07EDB A1 ?A#C B
936 GOYES RECE10
937 07EDD 840 * ST=1 point * removed rev X
938 07EE0 851 ST=0 point * added rev X
939 07EE3 1B00000 ST=1 slash * added rev X
940 07EEA 146 D0=(5) =FUNCD0 On limite la comparaison È la
941 07EED CE C=DAT0 A sous-chaÑne d'limitÅe par le
942 07EEF 144 C=C-1 A point.
943 07EF2 580 DAT0=C A
944 07EF5 8E376F GONC RECE20 B.E..T
945 RECE10 GOSUBL find/
946 * GOC RECE20 On a trouve le "/" => S1=0 rem rev X
947 * ST=1 slash On n'a pas trouvÅ le "/" rem rev X
948 07EFB 1B00000 *** end of changes
949 RECE20 D0=(5) =STMTD0 Adresse de la fiche courante.
950 07F02 146 C=DAT0 A
951 07F05 185 D0=D0- 6 S-R1-2
952 07F08 142 A=DAT0 A No de la fiche correspondante
953 07F0B 184 D0=D0- 5
954 07F0E 144 DAT0=C A S-R1-1 := STMTD0
955 07F11 184 D0=D0- 5
956 07F14 140 DAT0=A A S-R1-0 := S-R1-2
957 07F17 8EE98F GOSUBL search
958 07F1D 879 ?ST=1 trouve
959 07F20 06 GOYES RECE90 TrouvÅ !
```

```
959 07F22 870          ?ST=1  point
960 07F25 B5          GOYES  RECE90
961 07F27 1B00000    D0=(5) =S-R1-0
962 07F2E 142        A=DAT0 A
963 07F31 164        D0=D0+ 5
964 07F34 146        C=DAT0 A
965 07F37 164        D0=D0+ 5
966 07F3A 140        DAT0=A A
967 07F3D 165        D0=D0+ 6
968 07F40 144        DAT0=C A
969 07F43 7810       GOSUB  RECE30
970 07F47 E4F64702 RECE29 NIBASC 'Not foun'
      66F657E6
971 07F57 46E2E2E2    NIBASC 'd...'
972 07F5F 07          RECE30 C=RSTK
973 07F61 135        D1=C
974 07F64 31C0       LC(2)  ((RECE30)-(RECE29))/2
975 07F68 8E964F     GOSUBL dspcnc
976 07F6E 8F00000    GOSBVL =CHIRP
977 07F75 8F00000    GOSBVL =CRLFS
978 07F7C 682E       GOTO   BS:
979 07F80 1F00000    RECE90 D1=(5) =S-R1-2  No de la fiche trouvee
980 07F87 147        C=DAT1 A
981 07F8A 1F00000    D1=(5) =STMTD1
982 07F91 143        A=DAT1 A
983 07F94 131        D1=A
984 07F97 173        D1=D1+ 4          D1 ^ fiche courante
985 07F9A 15D3       DAT1=C 4
986 07F9E 6F6D       GOTO   BP:
987
988 07FA2 709B       EXIT   GOSUB  o7AF2
989 07FA6 8F00000    GOSBVL =NOSCRL
990 07FAD 84E        ST=0   14          NoCont
991 07FB0 8D00000    GOVLNG =NXTSTM
992
993 07FB7 7B7B       KEY   GOSUB  o7AF2
994 07FBB 8E046F     GOSUBL pass
995 07FC1 5A0        GONC   KEY10
996 07FC4 96F        ?D#0   B
997 07FC7 C0         GOYES  KEY20
998 07FC9 AF3        D=0     W
999 07FCC AFB       KEY10  C=D     W
1000 07FCF 1547      DAT0=C  W
1001 07FD3 6A3D     KEY20  GOTO   BP:
1002
1003 07FD7 1F00000    HAUT   D1=(5) =S-R1-3
1004 07FDE 1574      C=DAT1  S
1005 07FE2 A4E       C=C-1   S
1006 07FE5 550       GONC    Champ
1007 07FE8 AC2       gHAUT  C=0     S
1008 07FEB 1F00000    Champ  D1=(5) =S-R1-3
1009 07FF2 1554      DAT1=C  S
1010 07FF6 6EAD      GOTO   BS:
1011 07FFA 1F00000    BAS    D1=(5) =S-R1-3
1012 08001 1534      A=DAT1  S
1013 08005 307      LC(1)   CHPMAX
1014 08008 816      CSRC
```

```

1015 0800B 9CE          ?A>=C  S
1016 0800E DD          GOYES  Champ  Ecriture de CHPMAX
1017 08010 B44         A=A+1  S      => Cy=0
1018 08013 AC6         C=A    S
1019 08016 54D         GONC   Champ  B.E.T.
1020 08019 307         gBAS   LC(1)  CHPMAX
1021 0801C 816         gBAS10 CSRC
1022 0801F 6BCF        GOTO   Champ
1023 08023 D6          ChampA C=A    A
1024 08025 66FF        GOTO   gBAS10
1025
1026 08029 1F00000     NoFich D1=(5) =STMTD1 Adresse du fichier
1027 08030 147         C=DAT1 A
1028 08033 135         D1=C    D1 ^ Max
1029 08036 D0          A=0    A
1030 08038 15B3        A=DAT1 4  A=Max (1..Max)
1031 0803C 173         D1=D1+ 4  D1 ^ Fiche courante
1032 0803F CC          A=A-1  A    Pour adapter au No de la fiche courante
(0..Max-1)
1033 08041 D2          C=0    A
1034 08043 15F3        C=DAT1 4  C=Fiche courante
1035 08047 01          RTN     C=Fiche courante; A=Max (0.. Max-1);
1036                    *      Cy=1 Fichier vide
1037
1038 08049 7CDF         DERN   GOSUB  NoFich
1039 0804D D6          FicheA C=A    A
1040 0804F 6900        GOTO   FicheC
1041 08053 72DF        PREM   GOSUB  NoFich
1042 08057 D2          Fiche0 C=0    A
1043 08059 15D3        FicheC DAT1=C 4
1044 0805D 60BC        GOTO   BP:
1045 08061 74CF        ANCIEN GOSUB  NoFich
1046 08065 CE          C=C-1  A
1047 08067 51F         GONC   FicheC
1048 0806A 4CE         GOC    Fiche0  B.E.T.
1049 0806D 78BF        NOUVEL GOSUB  NoFich
1050 08071 8BA         ?C>=A  A
1051 08074 9D          GOYES  FicheA
1052 08076 E6          C=C+1  A
1053 08078 50E         GONC   FicheC  B.E.T.
1054
1055 0807B 77BA         DELETE GOSUB  o7AF2
1056 0807F 7A10        GOSUB  DEL10
1057 08083 4456C656     DEL05  NIBASC 'Delete: '
      4756A302
1058 08093 95F2E402    NIBASC 'Y/N ?'
      F3
1059 0809D 07          DEL10  C=RSTK
1060 0809F 135         D1=C
1061 080A2 31D0        LC(2)  ((DEL10)-(DEL05))/2
1062 080A6 8EB23F      GOSUBL dspnc
1063 080AC 8ED6AF      GOSUBL finlin
1064 080B2 8E056F      GOSUBL keywt
1065 080B8 3160        LC(2)  #06    Touche [Y] (Oui)
1066 080BC 962         ?C=A  B
1067 080BF 60          GOYES  DEL15
1068 080C1 63EC        GOTO   BS:
1069 080C5 7400        DEL15  GOSUB  delete

```

```

1070 080C9 679F          GOTO  ANCIEN
1071
1072 080CD 07          delete C=RSTK
1073 080CF DA          A=C    A
1074 080D1 8E1020      GOSUBL pshupd
1075 080D7 8F00000     GOSBVL =OBCOLL
1076 080DE 1B00000     D0=(5) =STMTD0
1077 080E5 146         C=DAT0 A
1078 080E8 134         D0=C
1079 080EB 183         D0=D0- 4
1080 080EE D2          C=0    A
1081 080F0 15E3        C=DAT0 4
1082 080F4 10B         R3=C
1083 080F7 132         AD0EX          A(A)=Adresse de la longueur de la ligne
1084 080FA CA          A=A+C  A      Adresse du dernier quartet + 1 de la ligne
courante
1085 080FC D8          B=A    A      Sauvegarde temporaire ds B(A)
1086 080FE 79B2        GOSUB  Entete  C=adresse de l'en-tete du fichier.
1087 08102 D4          A=B    A
1088 08104 8F00000     GOSBVL =MGOSUB
1089 0810B 00000       CON(5) =RPLLIN
1090 08110 AC2         C=0    S
1091 08113 550         GONC   o80D5
1092 08116 B46         C=C+1  S
1093 08119 108         o80D5  R0=C
1094 0811C 8EFA10      GOSUBL popupd
1095 08122 DB          C=D    A
1096 08124 06          RSTK=C
1097 08126 118         C=R0
1098 08129 94E         ?C#0   S
1099 0812C F0          GOYES  BserR
1100 0812E 77FE        GOSUB  NoFich
1101 08132 1C3         D1=D1- 4
1102 08135 1593        DAT1=A 4      Une fiche en moins.
1103 08139 01          RTN
1104 0813B 8D00000     BserR  GOVLNG =BSERR
1105
1106 08142 70F9        CAT    GOSUB  o7AF2
1107 08146 8E054F      GOSUBL SD1sve
1108 0814C 1F00000     D1=(5) (=FUNCR0)-2
1109 08153 D2          C=0    A
1110 08155 31C2        LC(2)  44
1111 08159 8F00000     GOSBVL =WIPOUT
1112 08160 1D00        D1=(2) (=FUNCR0)-2
1113 08164 1B00000     D0=(5) =S-R1-1
1114 0816B 146         C=DAT0 A
1115 0816E 134         D0=C
1116 08171 D0          A=0    A
1117 08173 15A3        A=DAT0 4      A(A)=Nb max de fiches, en Hexa.
1118 08177 8E545F      GOSUBL Num2D1
1119 0817D 39023616   LCASC  'drac '
      2746
1120 08189 15D9        DAT1=C 10
1121 0818D 179        D1=D1+ 10
1122 08190 D0          A=0    A
1123 08192 15A3        A=DAT0 4
1124 08196 CC          A=A-1  A
1125 08198 8A8        ?A=0   A

```

```

1126 0819B C0          GOYES  o8163
1127 0819D 3137       LCASC  's'
1128 081A1 14D        DAT1=C B
1129 081A4 171        D1=D1+ 2
1130 081A7 33C202    o8163 LCHEX  202C
1131 081AD 15D3       DAT1=C 4
1132 081B1 173        D1=D1+ 4
1133 081B4 132        AD0EX
1134 081B7 D2         C=0    A
1135 081B9 3151       LC(2)  #15
1136 081BD EE         C=A-C  A
1137 081BF 134        D0=C
1138 081C2 146        C=DAT0 A
1139 081C5 130        D0=A
1140 081C8 DE         CAEX   A
1141 081CA D2         C=0    A
1142 081CC 31D1       LC(2)  #1D
1143 081D0 EE         C=A-C  A
1144 081D2 D0         A=0    A
1145 081D4 15A3       A=DAT0 4
1146 081D8 DE         CAEX   A
1147 081DA 8F00000    GOSBVL =IDIVA
1148 081E1 20         P=     0
1149 081E3 100        R0=A
1150 081E6 137        CD1EX
1151 081E9 10B        R3=C
1152 081EC 1B00000    D0=(5) =STMTD0
1153 081F3 146        C=DAT0 A
1154 081F6 8F00000    GOSBVL =LOCADR
1155 081FD 8F00000    GOSBVL =FLADDR
1156 08204 E2         C=C-A  A
1157 08206 DA         A=C    A
1158 08208 11B        C=R3
1159 0820B 135        D1=C
1160 0820E 118        C=R0
1161 08211 8A8        ?A=0  A
1162 08214 21         GOYES  cat10
1163 08216 8F00000    GOSBVL =IDIVA
1164 0821D 20         P=     0
1165 0821F 8A8        ?A=0  A
1166 08222 40         GOYES  cat10
1167 08224 CC         A=A-1  A
1168 08226 8E694F    cat10 GOSUBL Num2D1
1169 0822C 3B021667  LCASC  'liava '
      1696C6
1170 0823A 15DB       DAT1=C 12
1171 0823E 1D00       D1=(2) (=FUNCR0)-2
1172 08242 8F00000    GOSBVL =VIEWD1
1173 08249 8EF53F     GOSUBL SD1res
1174 0824F 6C6B       GOTO   BS07      On ne reaffiche pas inutilement
1175
1176 08253 7FD8       INPUT  GOSUB  o7AF2
1177 08257 7610       GOSUB  INP10
1178 0825B           INP05  BSS   4      Longueur LIF de la ligne
1179 *                   *                   (sans signification ici)
1180 0825F 10         CON(2) 1
1181 08261 F2        CON(2) '/'

```



```

1182 08263          BSS      2*(CHPMAX)
1183 08271 07      INP10  C=RSTK
1184 08273 134          D0=C
1185 08276 D2          C=0    A
1186 08278 3161         LC(2)  (INP10)-(INP05) En quartets...
1187 0827C 8E2E5F      GOSUBL editfc
1188 08282 8E1ADF      INP20  GOSUBL NoFich  A=Nb max de fiches (0..max-1)
1189 08288 480          GOC    INP25  Cy=1 <=> Le fichier est vide
1190 0828B 8EA25F      GOSUBL search
1191 08291 D2          INP25  C=0    A
1192 08293 10B          R3=C          Taille de la ligne È remplacer
1193          *          (0 car insertion)
1194 08296 7121         GOSUB  Entete  C(A)=En-tÁte
1195 0829A 184          D0=D0- 5      D0=(5) STMTD0
1196 0829D 142          A=DAT0 A      Adresse de la ligne juste aprÉs...
1197 082A0 130          D0=A
1198 082A3 183          D0=D0- 4      D0 ^ dÁbut de la ligne juste aprÉs
1199 082A6 132          AD0EX        A      "      "
1200          *  R3=0 ; A(A)=^ ligne suivante ; C(A)=^ en-tÁte du fichier
1201 082A9 8F00000      GOSBVL =MGOSUB
1202 082B0 00000        CON(5) =RPLLIN
1203 082B5 560          GONC   INP30
1204 082B8 628E          GOTO   BserR
1205 082BC 8E76DF      INP30  GOSUBL NoFich
1206 082C2 1C3          D1=D1- 4
1207 082C5 E4          A=A+1  A
1208 082C7 E4          A=A+1  A
1209 082C9 1593        DAT1=A  4
1210 082CD 62BC        GOTO   RECE90  RÁactualisation du No de
1211          *          la fiche courante.
1212
1213 082D1 8D00000      popupd GOVLNG =POPUPD
1214 082D8 8D00000      pshupd  GOVLNG =PSHUPD
1215
1216 082DF 8E158F      EDIT   GOSUBL o7AF2
1217 082E5 1F00000      D1=(5) =STMTD0
1218 082EC 147          C=DAT1  A
1219 082EF 134          D0=C
1220 082F2 183          D0=D0- 4
1221 082F5 D2          C=0    A
1222 082F7 15E3        C=DAT0  4
1223 082FB 8E365F      GOSUBL editfc
1224 08301 1B00000      D0=(5) =STMTD0
1225 08308 146          C=DAT0  A
1226 0830B 134          D0=C
1227 0830E 8E524F      GOSUBL tststr
1228 08314 879          ?ST=1  trouve
1229 08317 36          GOYES  EDIT50
1230 08319 1B00000      D0=(5) =AVMEMS
1231 08320 1F00000      D1=(5) =STMTR1
1232 08327 146          C=DAT0  A
1233 0832A 184          D0=D0- 5
1234 0832D 142          A=DAT0  A
1235 08330 141          DAT1=A  A      STMTR1=OUTBS
1236 08333 E2          C=C-A   A
1237 08335 174          D1=D1+ 5
1238 08338 145          DAT1=C  A      S-R1-1=longueur de la fiche en quartets

```

```
1239 0833B 7E8D          GOSUB delete
1240 0833F 7AD0          GOSUB o83D9
1241 08343 8F00000      GOSBVL =OBCOLL
1242 0834A 1F00000      D1=(5) 5+(=STMTR1)
1243 08351 143          A=DAT1 A
1244 08354 D8           B=A A Taille en quartets
1245 08356 1C4          D1=D1- 5
1246 08359 143          A=DAT1 A A(A)=ancien OUTBS
1247 0835C 8EBF4F      GOSUBL move*m
1248 08362 DA           A=C A
1249 08364 C9           C=C+B A
1250 08366 1B00000      D0=(5) =AVMEMS
1251 0836D 144          DAT0=C A
1252 08370 8E507F      GOSUBL o7A37
1253 08376 6B0F        GOTO INP20
1254 0837A 1B00000      EDIT50 D0=(5) =STMTD0
1255 08381 146          C=DAT0 A
1256 08384 134          D0=C
1257 08387 183          D0=D0- 4
1258 0838A D2           C=0 A
1259 0838C 15E3         C=DAT0 4
1260 08390 10B          R3=C
1261 08393 132          AD0EX
1262 08396 C2           C=C+A A C(A)=adresse de la ligne suivante
1263 08398 135          D1=C Sauvegarde ds D1
1264 0839B 7C10         GOSUB Entete C(A)=Adresse de l'en-tÁte du fichier
1265 0839F 133          AD1EX
1266 * R3(A)=taille de la vieille ligne en quartets;
1267 * C(A)=adresse de l'en-tÁte du fichier
1268 * A(A)=adresse de la ligne suivante
1269 083A2 8F00000      GOSBVL =MGOSUB
1270 083A9 00000        CON(5) =RPLLIN
1271 083AE 560          GONC EDIT60
1272 083B1 698D        GOTO BserR
1273 083B5 8C759F      EDIT60 GOLONG BP:
1274
1275 083BB 1B00000      Entete D0=(5) =STMTD1
1276 083C2 142          A=DAT0 A
1277 083C5 D2           C=0 A
1278 083C7 3153         LC(2) 37+16
1279 083CB EE           C=A-C A
1280 083CD 01           RTN
1281
1282 083CF 1B00000      o838B D0=(5) =STMTD0
1283 083D6 142          A=DAT0 A
1284 083D9 7BFE          GOSUB pshupd
1285 083DD 1B00000      D0=(5) =STMTD1
1286 083E4 142          A=DAT0 A
1287 083E7 7DEE          GOSUB pshupd
1288 083EB 8F00000      GOSBVL =MGOSUB
1289 083F2 00000        CON(5) =DSLEEP
1290 083F7 76DE          GOSUB popupd
1291 083FB DB           C=D A
1292 083FD 1B00000      D0=(5) =STMTD1
1293 08404 144          DAT0=C A
1294 08407 76CE          GOSUB popupd
1295 0840B DB           C=D A
```

```
1296 0840D 1B00000          D0=(5) =STMTD0
1297 08414 144              DAT0=C A
1298 08417 8C5F8F          GOLONG BP:
1299
1300 0841D 1B00000  o83D9  D0=(5) =STMTD1
1301 08424 142              A=DAT0 A
1302 08427 D2               C=0 A
1303 08429 30C              LC(1) #C
1304 0842C CA               A=A+C A
1305 0842E 184              D0=D0- 5
1306 08431 140              DAT0=A A
1307 08434 19B8            D0=(2) #8B
1308 08438 D2               C=0 A
1309 0843A 144              DAT0=C A
1310 0843D 01              RTN
1311
1312 0843F                  END
```

-LINE	Ext		-	648						
ANCIEN	Abs	32865	#00008061	-	1045	858	1070			
ATNCLR	Ext			-	456					
AVMEMS	Ext			-	153	162	473	598	1230	1250
BAS	Abs	32762	#00007FFA	-	1011	856				
BP10	Abs	32104	#00007D68	-	820	799				
BP18	Abs	32123	#00007D7B	-	828	825				
BP20	Abs	32129	#00007D81	-	830	821				
BP:	Abs	32014	#00007D0E	-	798	785	986	1001	1044	1273 1298
BS05	Abs	32176	#00007DB0	-	847	845				
BS07	Abs	32188	#00007DBC	-	849	1174				
BS10	Abs	32198	#00007DC6	-	851	889	899			
BS11	Abs	32204	#00007DCC	-	852	850				
BS20	Abs	32328	#00007E48	-	896	894				
BS30	Abs	32348	#00007E5C	-	900	898				
BS:	Abs	32165	#00007DA5	-	844	581	924	978	1010	1068
BSERR	Ext			-	139	467	677	1104		
BserR	Abs	33083	#0000813B	-	1104	1099	1204	1272		
Bserr	Abs	31919	#00007CAF	-	757	723	748			
CAT	Abs	33090	#00008142	-	1106	876				
CHEDIT	Ext			-	627					
CHIRP	Ext			-	976					
CHPMAX	Abs	7	#00000007	-	26	557	567	1013	1020	1182
CKSREQ	Ext			-	352					
CNVUCR	Ext			-	212					
COLLAP	Ext			-	190	608				
CONVUC	Ext			-	382	616				
COPYu	Ext			-	705					
CRETf+	Ext			-	722					
CRLFSD	Ext			-	977					
CURSF	Ext			-	506	647				
Champ	Abs	32747	#00007FEB	-	1008	1006	1016	1019	1022	
ChampA	Abs	32803	#00008023	-	1023	895				
D1=AVE	Ext			-	637					
D=AVMS	Ext			-	612					
DEL05	Abs	32899	#00008083	-	1057	1061				
DEL10	Abs	32925	#0000809D	-	1059	1056	1061			
DEL15	Abs	32965	#000080C5	-	1069	1067				
DELETE	Abs	32891	#0000807B	-	1055	874				
DERN	Abs	32841	#00008049	-	1038	868				
DSLEEP	Ext			-	1289					
DSP\$00	Ext			-	192	640				
DSPCHX	Ext			-	267	289				
DSPCNA	Ext			-	93					
EDIT	Abs	33503	#000082DF	-	1216	870				
EDIT50	Abs	33658	#0000837A	-	1254	1229				
EDIT60	Abs	33717	#000083B5	-	1273	1271				
EOLXCK	Ext			-	667					
ESCSEQ	Ext			-	501					
EXIT	Abs	32674	#00007FA2	-	988	788	811	878		
Entete	Abs	33723	#000083BB	-	1275	1086	1194	1264		
FINDA	Ext			-	457					
FINDF	Ext			-	714					
FINLIN	Ext			-	646					
FLADDR	Ext			-	1155					
FSPECx	Ext			-	675					
FUNCDO	Ext			-	112	129	142	170	219	362 605

Symbol	Address	Value	Symbol	Address	Value	Symbol	Address	Value	Symbol	Address	Value
		939									
FUNCD1	Ext	-	197	609							
FUNCR0	Ext	-	1108	1112	1171						
FV10	Abs	32021 #00007D15	-	800	846						
FV20	Abs	32025 #00007D19	-	801	805						
FV50	Abs	32045 #00007D2D	-	803	800	805					
FV70	Abs	32066 #00007D42	-	808	819						
Fiche0	Abs	32855 #00008057	-	1042	1048						
FicheA	Abs	32845 #0000804D	-	1039	1051						
FicheC	Abs	32857 #00008059	-	1043	1040	1047	1053				
HAUT	Abs	32727 #00007FD7	-	1003	854						
HEXDEC	Ext	-	326								
IDIVA	Ext	-	1147	1163							
INP05	Abs	33371 #0000825B	-	1178	1186						
INP10	Abs	33393 #00008271	-	1183	1177	1186					
INP20	Abs	33410 #00008282	-	1188	1253						
INP25	Abs	33425 #00008291	-	1191	1189						
INP30	Abs	33468 #000082BC	-	1205	1203						
INPUT	Abs	33363 #00008253	-	1176	815	872					
KA10	Abs	31610 #00007B7A	-	673	669						
KA20	Abs	31615 #00007B7F	-	675	668						
KA30	Abs	31632 #00007B90	-	679	676						
KA35	Abs	31646 #00007B9E	-	684	680						
KA40	Abs	31740 #00007BFC	-	709	686	689	693				
KA60	Abs	31861 #00007C75	-	740	708	716					
KA65	Abs	31891 #00007C93	-	749	742	745					
KA70	Abs	31923 #00007CB3	-	758	755						
KA80	Abs	32005 #00007D05	-	789	787						
KAd	Ext	-	664								
=KAe	Abs	31581 #00007B5D	-	666							
KAp	Ext	-	665								
KEY	Abs	32695 #00007FB7	-	993	813	880					
KEY10	Abs	32716 #00007FCC	-	999	995						
KEY20	Abs	32723 #00007FD3	-	1001	997						
KEYBUF	Ext	-	250	257							
KEYNAM	Ext	-	887								
KEYPTR	Ext	-	651								
LDCSET	Ext	-	458								
LDCSPC	Ext	-	515								
LOCADR	Ext	-	740	1154							
MEMCKL	Ext	-	137	465							
MGOSUB	Ext	-	704	1088	1201	1269	1288				
MOVE*M	Ext	-	459								
NOSCRL	Ext	-	989								
NOUVEL	Abs	32877 #0000806D	-	1049	860						
NXTSTM	Ext	-	991								
NoFich	Abs	32809 #00008029	-	1026	798	844	1038	1041	1045	1049	
			-	1100	1188	1205					
Num2D1	Abs	30402 #000076C2	-	324	1118	1168					
OBCOLL	Ext	-	463	1075	1241						
POP1S	Ext	-	193	641							
POPBUF	Ext	-	354								
POPUPD	Ext	-	1213								
PREM	Abs	32851 #00008053	-	1041	866						
PSHUPD	Ext	-	1214								
R<RSTK	Ext	-	104	626							
RANGE	Ext	-	893	897							

o7A37	Abs	31355	#00007A7B	-	601	1252						
o7A6C	Abs	31408	#00007AB0	-	614	620						
o7A83	Abs	31431	#00007AC7	-	619	613						
o7AF2	Abs	31542	#00007B36	-	651	885	988	993	1055	1106	1176	
						1216						
o80D5	Abs	33049	#00008119	-	1093	1091						
o8163	Abs	33191	#000081A7	-	1130	1126						
o838B	Abs	33743	#000083CF	-	1282	360	547	817	882	920		
o83D9	Abs	33821	#0000841D	-	1300	1240						
pass	Abs	30209	#00007601	-	267	786	994					
pass05	Abs	30237	#0000761D	-	274	279						
pass10	Abs	30265	#00007639	-	277	273	279					
pass20	Abs	30291	#00007653	-	286	298						
pass21	Abs	30295	#00007657	-	287	303	321					
pass30	Abs	30331	#0000767B	-	296	285						
pass35	Abs	30366	#0000769E	-	310	307						
point	Abs	0	#00000000	-	22	446	932	937	959			
popupd	Abs	33489	#000082D1	-	1213	1094	1290	1294				
prpstr	Abs	29967	#0000750F	-	188	105	299	926				
pshupd	Abs	33496	#000082D8	-	1214	1074	1284	1287				
search	Abs	30651	#000077BB	-	412	956	1190					
seek10	Abs	29531	#0000735B	-	48	38						
seek20	Abs	29556	#00007374	-	56	62						
seek30	Abs	29570	#00007382	-	61	55						
seekA	Abs	29535	#0000735F	-	50	451						
seekC	Abs	29492	#00007334	-	35	425	831					
setchp	Abs	31187	#000079D3	-	551	558	565	575				
slash	Abs	1	#00000001	-	23	377	393	938				
srh10	Abs	30690	#000077E2	-	424	420	445					
tEOL	Ext			-	672							
tLITRL	Ext			-	670							
trouve	Abs	9	#00000009	-	24	372	405	416	432	957	1228	
tst10	Abs	30559	#0000775F	-	374	389						
tst15	Abs	30582	#00007776	-	382	378						
tst20	Abs	30597	#00007785	-	386	373						
tst30	Abs	30614	#00007796	-	393	387						
tst35	Abs	30631	#000077A7	-	403	394						
tst50	Abs	30641	#000077B1	-	407	385						
tststr	Abs	30521	#00007739	-	362	415	431	1227				
uprc\$	Abs	30036	#00007554	-	209	929						
uprc10	Abs	30042	#0000755A	-	212	216						
uprc20	Abs	30052	#00007564	-	214	211						

Input Parameters

Source file name is ka.a

Listing file name is ka.l

Object file name is ka.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE MAPLEX
2
3          * *****
4          * Character mapping module
5          *
6          * Keyword(s):
7          *   MAP$, MAP
8          * Author(s):
9          *   Tapani Tarvainen
10         * History:
11         *   See below
12         *   Published in JPC #46
13         *
14         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
15         * *****
16
17 0843F          ABS      #843F
18
19         * *****
20         *   HP71 char set converter, rev B
21         *
22         *   Tapani Tarvainen 86/09/07, 87/02/24
23         *
24         *   Function MAP$ maps given set of characters onto another
25         *   in a string; statement MAP does the same for
26         *   all or part of a TEXT (LIF1) file.
27         *
28         *   Syntax: MAP$(<str0>,<str1>,<str2>)
29         *           MAP <file>,<str1>,<str2>[,<num1>[,<num2>]]
30         *           MAP #<chnl>,<str1>,<str2>[,<num1>[,<num2>]]
31         *   where <str0>, <str1> and <str2> are string expressions,
32         *   <num1> and <num2> are numeric expressions,
33         *   <file> is a file specifier (literal or string expr),
34         *   <chnl> is channel number (numeric expression).
35         *   <num1> and <num2> specify beginning and end record;
36         *   defaults are beginning and end of file, respectively.
37         *
38         *   The file must reside in RAM and it mustn't be secured;
39         *   the strings <str1> and <str2> must be of equal length.
40         *   If some char occurs several times in <str1>,
41         *   it's the first one that counts.
42         *
43         *   For example, MAP$("calculator","ac","xz") returns
44         *   "zxlzulxtor", and
45         *   MAP F$, "abcdefghijklmnopqrstuvwxy",
46         *   "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
47         *   will convert entire file F$ to upper case.
48         *
49         * *****
50         *
51         * Errors:
52         * "Chnl# Not Found",
53         * "Invalid Filespec", "File Not Found" as appropriate;
54         * "Invalid File Type" if non-TEXT file;
55         * "Illegal Access" if not in RAM;
56         * "Protected" if the file is SECURED;
57         * "Invalid Arg" if LEN(<str1>)#LEN(<str2>), or
```

```
58      * <num1> or <num2> negative or >2^20-1
59      * "Insufficient Memory" results if there isn't enough
60      * room for the conversion table (256 bytes, see below);
61      * "End of File" means the file is bad (record length points
62      * beyond end of file).
63      *
64      * *****
65      *
66      * Algorithm: Create a 256-byte table, where Nth byte
67      * is the new value for char N, and replace every char
68      * in the file with corresponding element in the table.
69      *
70      * *****
71      *
72      * It is rather unusual to allow both channel # and filespec
73      * to be used. Channel is faster, once the file is opened,
74      * and so to be preferred if it is likely that the operation
75      * is done among other file processing, but if it is of
76      * "global" nature, i.e., if it is likely that a separate
77      * ASSIGN is needed each time, filespec is clearly better.
78      * MAP statement was originally conceived as TRANSFORM-like
79      * total operation, for sending files to different computers
80      * and printers with different character sets, so filespec was
81      * obvious choice; later different applications emerged, with
82      * need for record limits and situations where the file was
83      * likely to have been opened, suggesting the use of channel.
84      * An additional consideration is the need to ensure FIB info
85      * remains correct, even if the file is referred to by name:
86      * fortunately, the MAP operation requires no changes to the FIB.
87      * So why not let the user choose?
88      * The only cost (besides longer code) is an extra byte for
89      * the # token (it needn't be tokenized if it were mandatory).
90      * Even that could be avoided, at the cost of still more
91      * code (lots): if there is a numeric expression it must
92      * be channel#, else filespec (even that one comma now
93      * tokenized isn't really necessary, but omitting it
94      * would cost a ridiculous amount of code (or ingenuity)).
95      *
96      * *****
97
98
99      * local symbols
100     sMAP$ EQU 3          flag to distinguish MAP$ and MAP
101
102
103     * *****
104     * MAP$ entry
105     * *****
106     0843F 44433          NIBHEX 44433      3 string params
107     08444 853           =MAPse ST=1  sMAP$    indicates were in MAP$, not MAP
108     08447 136           CD0EX          save D0 (program counter)
109     0844A 108           R0=C           in R0
110     0844D 6C41          GOTO    bldtbl    go build conversion table
111
112
113
114     * *****
```

```
115          * MAP entry
116          * *****
117 08451 00000          REL(5) =MAPd
118 08456 00000          REL(5) =MAPp
119 0845B 14A          =MAPE  A=DAT0 B          read 1st token to determine
120 0845E 3132          LCASC  '#'          whether we have channel# or filespec
121 08462 966          ?A#C  B          # present?
122 08465 95          GOYES  xfsp          no, go check filespec
123
124          *****
125          * we have channel#
126          * find file address, type &c from FIB
127 08467 161          D0=D0+ 2          skip # token
128 0846A 8F00000      GOSBVL =GETCH# channel# to A(B)
129 08471 8F00000      GOSBVL =FIBADR find FIB entry, save in STMTD1
130
131          * locating FIB entry before EXPEXC below means
132          * trouble with UDFs monkeying with channel #,
133          * but anybody doing that deserves what he gets
133 08478 161          D0=D0+ 2          step over comma
134 0847B 8F00000      GOSBVL =EXPEXC- other arguments to mathstack
135 08482 137          CD1EX
136 08485 134          D0=C
137 08488 1F00000      D1=(5) =STMTD1 recover FIB addr
138 0848F 147          C=DAT1 A
139 08492 135          D1=C
140 08495 174          D1=D1+ 5
141 08498 D2          C=0 A
142 0849A 15F3          C=DAT1 4          file type
143 0849E 173          D1=D1+ 4
144 084A1 1534          A=DAT1 S          protection
145 084A5 172          D1=D1+ 3
146 084A8 1574          C=DAT1 S          device code
147 084AC AC7          D=C S
148 084AF 178          D1=D1+ 9
149 084B2 143          A=DAT1 A          data start addr
150 084B5 131          D1=A
151 084B8 1C4          D1=D1- 5          link field addr
152 084BB 5C6          GONC  chktyp
153
154          *****
155          * we have filespec
156          * get type &c from header
157 084BE 8F00000      xfsp  GOSBVL =FILXQ^ execute filespec
158 084C5 490          GOC    fspOK filespec OK
159 084C8 3100          LC(2) =eFSPEC "Invalid Filespec"
160 084CC 566          GONC  mferr B.E.T.
161
162          * We save FILXQ^ result (file name & port info) in
163          * statement scratch during EXPEXC. Note that we can't
164          * call FINDF first, as EXPEXC might change the file
165          * address, or SECURE or PURGE it (or CREATE!)
165 084CF 1F00000      fspOK D1=(5) =STMTR0 save FILXQ^ result (A & D)
166 084D6 1517          DAT1=A W          file name
167 084DA 1D00          D1=(2) =STMTR1 port info
168          * STMTR0 and STMTR1 have same 3 high order digits
169 084DE AFB          C=D W
170 084E1 1557          DAT1=C W
171 084E5 161          D0=D0+ 2          step over comma
```

```

172 084E8 8F00000      GOSBVL =EXPEX-  strings to mathstack
173 084EF 137          CD1EX          FINDF+ &c use D1 but not D0,
174 084F2 134          D0=C          so we save stack pt in D0
175 084F5 1F00000      D1=(5) =STMTR0 recover FILXQ^ result
176 084FC 1537         A=DAT1 W
177 08500 1D00         D1=(2) =STMTR1
178 08504 1577         C=DAT1 W
179 08508 AF7          D=C          W
180 0850B 8F00000      GOSBVL =FINDF+  find the file
181 08512 402          GOC mferr    not found, error
182 08515 17F          D1=D1+ 16
183 08518 D2           C=0          A
184 0851A 15F3         C=DAT1 4      file type
185 0851E 173          D1=D1+ 4
186 08521 1534         A=DAT1 S      protection
187 08525 17B          D1=D1+ 12     at link field
188
189
190                    *****
191                    * the file has been found, either by filespec or channel#
192                    * C(A)=file type, A(S)=protection nib, D(S)=device code
193                    * D1=link field addr, D0=math stack pt
193 08528 CE          chktyp C=C-1  A      test type:
194 0852A 8AA          ?C=0          A      text?
195 0852D D0          GOYES textf
196 0852F 3100         LC(2) =eFTYPE  "Invalid File Type"
197 08533 8D00000      mferr GOVLNG =MFERR
198 0853A 8F00000      textf GOSBVL =RAMROM in RAM?
199 08541 3100         LC(2) =eFACCS "Illegal Access"
200 08545 5DE          GONC mferr
201 08548 8F00000      GOSBVL =?PRFI+ check protection
202 0854F 43E          GOC mferr    secure, complain
203
204                    * Now we know it's unprotected text file in RAM
205 08552 136          CD0EX          mathstack pt
206 08555 137          CD1EX          back to D1,
207 08558 108          R0=C          link field addr to R0
208 0855B 843          ST=0          sMAP$ indicates were in MAP, not MAP$
209                    * clearing sMAP$ earlier isn't a good idea:
210                    * some sub we call (EXPEXC!) might change it
211
212                    *****
213                    * now see if record limits were given:
214                    * are there numbers in the mathstack?
215 0855E D2           C=0          A      default start record=0
216 08560 109          R1=C
217 08563 CE          C=C-1  A      default record count=inf
218 08565 10A          R2=C          (or 2^20-1)
219 08568 AC3          D=0          S      flag 1st time through
220                    * peel numbers (up to 2) from top of mathstack
221 0856B 309          num? LCHEX 9      P=0
222 0856E 1537         A=DAT1 W      read stack entry
223 08572 986          ?A>C  P      real number?
224 08575 52          GOYES bldtbl no, go process strings
225 08577 17F          D1=D1+ 16     pop it off
226 0857A 8F00000      GOSBVL =FLT DH convert to hex integer
227 08581 583          GONC argerr  error if negative or too big
228 08584 121          AR1EX          substitute for start record

```

```
229 08587 A4F          D=D-1  S      did we already have a number?
230 0858A 40E          GOC    num?   no, look for another
231 0858D 119          C=R1   S      yes, it was end record
232 08590 EA           A=A-C  A      <end record>--<start record>
233 08592 540          GONC   b<=e  was start record>end record?
234 08595 D0           A=0    A      yes, convert start record only
235 08597 102          b<=e   R2=A   number of records to convert -1
236
237
238 *****
239 * MAP$ function and MAP statement join here
240
241 * the strings are in mathstack: get addresses
241 0859A 8F00000      bldtbl  GOSBVL =POP1S  pop str2
242 085A1 137          CD1EX
243 085A4 134          D0=C    S      str2 addr to D0
244 085A7 D8           B=A    A      str2 length
245 085A9 CA           A=A+C  A      skip past str2
246 085AB 131          D1=A   S      point at str1 header
247 085AE 8F00000      GOSBVL =POP1S  pop str1
248 085B5 8A0          ?A=B   A      equal lengths?
249 085B8 90           GOYES  strsoK
250 085BA 8D00000      argerr GOVLNG =ARGERR no, Invalid Arg
251 085C1 DE           strsoK ACEX  A      str2 addr to A (for CHKmem)
252 085C3 AD3          D=0    M      we'd only need to clear nib 5
253 085C6 D7           D=C    A      length in nibs
254 085C8 81F          DSRB   S      in bytes (used as counter)
255 085CB 137          CD1EX
256 085CE D5           B=C    A      str1 addr to B(A)
257
258 * The conversion table will be located at AVMEMS
259 * First we check there is room for it (256 bytes)
260 085D0 3400200      LC(5)  512   # of nibs needed
261 085D7 8F00000      GOSBVL =CHKmem memory check
262 085DE 436          GOC    mfer1 Insufficient Memory
263 085E1 131          D1=A   S      available memory start
264 * Initialize the table to map each char to itself
265 085E4 AE2          C=0    B      start with CHR$(0)
266 085E7 14D          init   DAT1=C B   write char to table
267 085EA 171          D1=D1+ 2   next table element
268 085ED B66          C=C+1  B      next char
269 085F0 56F          GONC   init
270
271 * Initialization could be done faster as follows:
272 *      LCHEX 0101010101010101 assumes P=0
273 *      A=C    W
274 *      LCHEX 0706050403020100
275 * init  DAT1=C W
276 *      D1=D1+ 16
277 *      C=C+A  W
278 *      GONC  init
279 *
280 * This would cost 37 nibs and save ca 7000 cycles
281
282 * Then we change the table so that if Ith char in
283 * str1 is N, Nth entry is set to Ith char of str2
284 085F3 DC           ABEX   A      table start to B(A)
285 085F5 133          AD1EX  S      str1 addr to D1
```

```
286          * ready to set up the conversion table:
287          * D1=str1 addr, D0=str2 addr, B(A)=table addr, D(A)=str length
288 085F8 CF          D=D-1 A      adjust counter
289 085FA 432        GOC      nop    in case str1 & str2 null strings
290 085FD D0        setup  A=0     A
291 085FF 14B        A=DAT1 B      str1 char
292 08602 A34        A=A+A X      double to get nibble offset
293 08605 C0        A=A+B A      add table start
294 08607 133        AD1EX
295 0860A 14E        C=DAT0 B      corresp. str2 char
296 0860D 14D        DAT1=C B     put in table
297 08610 133        AD1EX      str1 adr back to D1
298 08613 171        D1=D1+ 2    next char
299 08616 161        D0=D0+ 2    in both strings
300 08619 CF        D=D-1 A      decrement counter
301 0861B 51E        GONC      setup
302          * table is ready, its addr in B(A)
303
304 0861E 863        nop      ?ST=0  sMAP$  in MAP statement?
305 08621 52         GOYES     stmt
306
307          *****
308          * set things up for converting str0 (in mathstack)
309 08623 137        CD1EX      str0 header addr; it is where
310 08626 135        D1=C        D1 (stack pt) must be on exit
311 08629 06        RSTK=C     we save it in RSTK
312 0862B AD0        A=0 M      need to clear nib 5
313 0862E 171        D1=D1+ 2    skip type tag (0F or 8F)
314 08631 143        A=DAT1 A    str0 length (nibs)
315 08634 81C        ASRB      convert to bytes
316 08637 17D        D1=D1+ 14   skip rest of str0 header
317 0863A 6F70      GOTO     iconv  join MAP
318
319          *****
320          * End of File -error (corrupted file)
321 0863E 3100       Eof     LC(2)  =eEOFIL
322 08642 60FE       mfer1   GOTO   mferr
323
324          *****
325          * Initialize conversion of file
326 08646 110       stmt    A=R0      recover file addr
327 08649 131        D1=A      link field addr
328 0864C 143        A=DAT1 A    link field contents
329 0864F 174        D1=D1+ 5    beginning of data
330          * we need the end-of-file address to check
331          * if a line length header points beyond it
332          * (which means the file is corrupted, but
333          * even that shouldn't cause total disaster)
334 08652 137        CD1EX
335 08655 135        D1=C
336 08658 C2        C=A+C A      end of file
337 0865A D7        D=C A      save in D(A)
338 0865C AC3       D=0 S      indicates initial skipping phase
339          *****
340          * File conversion loop
341          * D1 = next record length field addr
342          * B(A) = conversion table addr
```

```

343          * D(A) = end-of-file addr (used as a safeguard)
344          * D(S) = 0 when skipping records before starting
345          * R1(A) = # of records left to skip/convert
346          * R(2) = (max) # of records to convert
347          * P=0
348 0865F 119      nxtrec C=R1          records left
349 08662 CE          C=C-1  A
350 08664 541      GONC      cont
351 08667 94B          ?D=0   S      were we skipping initial recs?
352 0866A 90          GOYES   action yes, start the real work
353 0866C 8D00000  nxtstm GOVLNG =NXTSTM all done
354 08673 B47      action D=D+1 S      so we know we've been here
355 08676 11A          C=R2          record count
356 08679 109      cont   R1=C
357          * read record header (length field)
358 0867C AA0          A=0    XS
359 0867F 14B          A=DAT1 B      1st byte of rec length
360 08682 171          D1=D1+ 2
361 08685 F0          ASL    A      move left
362 08687 F0          ASL    A
363 08689 14B          A=DAT1 B      2nd byte - now in proper order
364 0868C 171          D1=D1+ 2
365          * last 7 lines could be replaced by the following:
366          *   A=0    A
367          *   A=DAT1 4
368          *   D1=D1+ 4
369          *   GOSBVL SWPBYT equ #17A24
370          * this would save 1 nib and cost 38 cycles / record
371 0868F 23          P=      3      so we can test carry below
372 08691 B14          A=A+1  WP      must add 1 if odd
373 08694 20          P=      0      if that carried, it's #FFFF
374 08696 45D          GOC    nxtstm i.e., eof, and we're done
375 08699 30E          LCHEX  E      P=0
376 0869C 0E06        A=A&C  P      clear lsb
377          * we'll convert the pad byte too, but who cares
378 086A0 137          CD1EX          next check for corrupt file
379 086A3 135          D1=C
380 086A6 C2          C=C+A  A      add rec len twice (it's bytes!)
381 086A8 C2          C=C+A  A
382 086AA 8B7          ?C>D  A      past eof?
383 086AD 19          GOYES  Eof   yes, error
384 086AF 94F          ?D#0   S      initial skipping phase?
385 086B2 80          GOYES  iconv  no, convert the record
386 086B4 135          D1=C          next rec addr
387 086B7 57A          GONC   nxtrec
388
389
390          *****
391          * Here MAP and MAP$ join again
392          * D1 points to first char of record/str0
393          * A(A)= # of chars to convert (str/rec length)
394          * B(A)= conversion table addr
395 086BA CC          iconv  A=A-1  A      adjust counter
396 086BC 4D1          GOC    nullst null string or zero-length record
397          * innermost loop: this is where the time is spent
398 086BF D2          conv   C=0    A
399 086C1 14F          C=DAT1 B      read char

```



```
400 086C4 A36          C=C+C X      double to get nibble offset
401 086C7 C9          C=C+B A      add table start
402 086C9 134         D0=C
403                   * it might be possible to set things up so
404                   * that D0=CS could be used above, and all
405                   * A fields replaced with WP (with P=3) or X,
406                   * but I don't think it's worth the trouble
407 086CC 14E         C=DAT0 B      read corresp. char in table
408 086CF 14D         DAT1=C B      write back
409 086D2 171         D1=D1+ 2      skip to next char
410 086D5 CC          A=A-1 A
411 086D7 57E         GONC conv
412 086DA 863        nullst ?ST=0 sMAP$ MAP statement?
413 086DD 28         GOYES nxtrec yes, next record
414
415                   *****
416
417                   * MAP$ done: restore pointers & return to EXPR
418 086DF 07          C=RSTK        recover D1 (mathstack pt)
419 086E1 135         D1=C
420 086E4 118         C=R0          and D0 (program counter)
421 086E7 134         D0=C
422 086EA 8D00000     GOVLNG =EXPR return
423                   *
424 086F1             END
```

?PRFI+	Ext		-	201						
ARGERR	Ext		-	250						
CHKmem	Ext		-	261						
EXPEX-	Ext		-	134	172					
EXPR	Ext		-	422						
Eof	Abs	34366	#0000863E	-	321	383				
FIBADR	Ext		-	129						
FILXQ^	Ext		-	157						
FINDF+	Ext		-	180						
FLTDH	Ext		-	226						
GETCH#	Ext		-	128						
MAPd	Ext		-	117						
=MAPe	Abs	33883	#0000845B	-	119					
MAPp	Ext		-	118						
=MAPse	Abs	33860	#00008444	-	107					
MFERR	Ext		-	197						
NXTSTM	Ext		-	353						
POP1S	Ext		-	241	247					
RAMROM	Ext		-	198						
STMTD1	Ext		-	137						
STMTR0	Ext		-	165	175					
STMTR1	Ext		-	167	177					
action	Abs	34419	#00008673	-	354	352				
argerr	Abs	34234	#000085BA	-	250	227				
b<=e	Abs	34199	#00008597	-	235	233				
bldtbl	Abs	34202	#0000859A	-	241	110	224			
chktyp	Abs	34088	#00008528	-	193	152				
cont	Abs	34425	#00008679	-	356	350				
conv	Abs	34495	#000086BF	-	398	411				
eEOFIL	Ext		-	321						
eFACCS	Ext		-	199						
eFSPEC	Ext		-	159						
eFTYPE	Ext		-	196						
fspOK	Abs	33999	#000084CF	-	165	158				
iconv	Abs	34490	#000086BA	-	395	317	385			
init	Abs	34279	#000085E7	-	266	269				
mfer1	Abs	34370	#00008642	-	322	262				
mferr	Abs	34099	#00008533	-	197	160	181	200	202	322
nop	Abs	34334	#0000861E	-	304	289				
nullst	Abs	34522	#000086DA	-	412	396				
num?	Abs	34155	#0000856B	-	221	230				
nxtrec	Abs	34399	#0000865F	-	348	387	413			
nxtstm	Abs	34412	#0000866C	-	353	374				
sMAP\$	Abs	3	#00000003	-	100	107	208	304	412	
setup	Abs	34301	#000085FD	-	290	301				
stmt	Abs	34374	#00008646	-	326	305				
strsOK	Abs	34241	#000085C1	-	251	249				
textf	Abs	34106	#0000853A	-	198	195				
xfsp	Abs	33982	#000084BE	-	157	122				

Input Parameters

Source file name is maplex.a
Listing file name is maplex.l
Object file name is maplex.o
Flags set on command line
None

Errors

None

```
1 TITLE JPC LEX: MODULE MARGE
2
3 * *****
4 * Margin command
5 *
6 * Keyword(s):
7 * MARGIN
8 * Author(s):
9 * Pierre David, Michel Martinet, Janick Taillandier
10 * History:
11 * Initially published in JPC #26
12 *
13 * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14 * *****
15
16 086F1 ABS #86F1
17
18 bMARGI EQU #0083D MARGIN buffer
19 RESJPC EQU #2F991 JPC reserved location
20
21 * *****
22 * MARGIN entry
23 * *****
24 086F1 00000 REL(5) =STdD
25 086F6 00000 REL(5) =MARGIp
26 086FB 14A =MARGIe A=DAT0 B
27 086FE 8F00000 GOSBVL =EOLXCK
28 08705 475 GOC MARG10
29 08708 8F00000 GOSBVL =EXPEX- "Collapse" la Math Stack
30 * * et y d'apose le paramÉtre.
31 0870F 8F00000 GOSBVL =RNDAHx "POP", teste et convertit
32 * * l'expression en un nombre
33 * * HEXA dans A(A).
34 08716 5F6 GONC argerr Si l'expression est < 0,err
35
36 08719 3106 LC(2) 96 On charge dans C(B): 96
37 0871D 9E6 ?C<A B L'expression > 96 ?
38 08720 66 GOYES argerr oui, erreur
39 08722 CC A=A-1 A
40 08724 483 GOC MARG10
41 08727 100 R0=A
42 0872A D2 C=0 A
43 0872C 302 LCHEX 2
44 0872F D5 B=C A
45 08731 32D38 LC(3) bMARGI
46 08736 8F00000 GOSBVL =I/OALL
47 0873D 110 A=R0
48 08740 149 DAT1=A B
49 08743 1B199F2 D0=(5) RESJPC
50 0874A 1520 A=DAT0 P
51 0874E 304 LCHEX 4
52 08751 0E0E A=A!C P
53 08755 1500 DAT0=A P
54 08759 6520 GOTO fin
55 0875D 1B199F2 MARG10 D0=(5) RESJPC
56 08764 1520 A=DAT0 P
57 08768 30B LCHEX B
```

```
58 0876B 0E06          A=A&C  P
59 0876F 1500          DAT0=A  P
60 08773 32D38         LC(3)  bMARGI
61 08778 8F00000       GOSBVL =I/ODAL
62 0877F 8D00000  fin   GOVLNG =NXTSTM  Fin d'exAcution ....
63 08786 8D00000  argerr GOVLNG =ARGERR
64
65 0878D              END
```

ARGERR	Ext		-	63		
EOLXCK	Ext		-	27		
EXPEX-	Ext		-	29		
I/OALL	Ext		-	46		
I/ODAL	Ext		-	61		
MARG10	Abs	34653	#0000875D	-	55	28 40
=MARGIe	Abs	34555	#000086FB	-	26	
MARGIp	Ext		-	25		
NXTSTM	Ext		-	62		
RESJPC	Abs	194961	#0002F991	-	19	49 55
RNDAHx	Ext		-	31		
STDd	Ext		-	24		
argerr	Abs	34694	#00008786	-	63	34 38
bMARGI	Abs	2109	#0000083D	-	18	45 60
fin	Abs	34687	#0000877F	-	62	54

Input Parameters

Source file name is marge.a

Listing file name is marge.l

Object file name is marge.o

Flags set on command line
None

Errors

None

```
1 TITLE JPC LEX: MODULE MENU
2
3 * *****
4 * Menu module
5 *
6 * Keyword(s):
7 * MENU
8 * Author(s):
9 * Jean-Jacques Dhenin
10 * History:
11 * Published in JPC #26
12 *
13 * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14 * *****
15
16 0878D ABS #878D
17
18 0878D 8F00000 PARM GOSBVL =RNDAHX Convertir Reel sur MATSTK
19 08794 5C0 GONC INVPAR en Hex A(A)
20 08797 D2 C=0 A
21 08799 A6E C=C-1 B C(B)= FF
22 0879C 8BA ?C>=A A Restriction È 255
23 0879F 00 RTNYES
24
25 087A1 8D00000 INVPAR GOVLNG =IVAERR
26
27 087A8 1B00000 Read D0=(5) =PRGMEN Limite de recherche
28 087AF 146 C=DAT0 A pour les DATA
29 087B2 D7 D=C A Copie PRGMEN dans D(A)
30 087B4 1A0000 D0=(4) =DATPTR
31 087BA 146 C=DAT0 A Lit DATPTR
32 087BD 8AE ?C#0 A DATPTR=0?
33 087C0 B0 GOYES R10 Non, alors OK. DATA
34 087C2 1A0000 D0=(4) =PRGMST Oui, alors utilise PRGMST
35 087C8 146 C=DAT0 A
36 087CB 134 R10 D0=C
37 087CE 14A R11 A=DAT0 B Lit le token
38 087D1 31C2 LCASC ', '
39 087D5 962 ?A=C B Est-ce une virgule?
40 087D8 A2 GOYES R20 Oui, alors les data suivent
41 087DA 183 D0=D0- 4
42 087DD 90C ?A#0 P
43 087E0 50 GOYES R15 Non, alors pas de ligne#
44 087E2 163 D0=D0+ 4
45 087E5 316C R15 LCHEX C6
46 087E9 8F00000 GOSBVL =TKSCN+ trouve le data statement
47 087F0 411 GOC R20
48 087F3 7F80 GOSUB popbuf
49 087F7 3102 LCHEX 20 =eNODAT Error: No data
50 087FB 8D00000 MFERRj GOVLNG =MFERR
51 08802 161 R20 D0=D0+ 2
52 08805 136 CD0EX
53 08808 134 D0=C
54 0880B 10A R2=C Sauve l'ad. du dÃbut de DATA
55 0880E 31C2 LCASC ', '
56 08812 AE5 B=C B
57 08815 31D0 LC(2) 13
```



```

58 08819          R60
59 08819 14A      A=DAT0  B
60 0881C 960      ?A=B    B
61 0881F A0       GOYES   R80     Oui. Fin de la DATA
62 08821 161      D0=D0+ 2
63 08824 966      ?A#C    B         Est-ce une fin de ligne?
64 08827 2F       GOYES   R60     Non. Cherche encore
65
66 08829 118      R80  C=R0       Rapel des pointeurs
67 0882C A6E      C=C-1  B       DÀcrÅmente le pointeur.
68 0882F 108      R0=C      Sauve le pointeur.
69 08832 96E      ?C#0    B       Est-ce la DATA recherchÅe?
70 08835 99       GOYES   R11     Non. Alors DATA suivante.
71
72 08837 101      R83  R1=A       Sauve le terminateur.
73 0883A 1F00000  D1=(5) =FUNCR0 D1 pointe la sauvegarde.
74 08841 1B00000  D0=(5) =DSPMSK
75 08848 7E10     GOSUB   dsp
76 0884C 8F00000  GOSBVL =CURSFL  Curseur È gauche.
77 08853 111      A=R1       Rappel le terminator.
78 08856 11A      C=R2       Rappel l'adresse de la DATA
79 08859 135      D1=C       D1 pointe la DATA
80 0885C 8F00000  GOSBVL =DSPBUF  Affiche la DATA
81 08863 8D00000  GOVLNG =-LINE   efface le fin de ligne
82 0886A 1577     dsp  C=DAT1  W
83 0886E 1547     DAT0=C  W
84 08872 17F      D1=D1+ 16
85 08875 16F      D0=D0+ 16
86 08878 27       P=        7
87 0887A 1571     C=DAT1  WP
88 0887E 1541     DAT0=C  WP
89 08882 20       P=        0
90 08884 01       RTN
91 08886 8F00000  popbuf GOSBVL =CRLFND
92 0888D 8D00000  popb   GOVLNG =POPBUF
93
94
95
96
97
98 08894 8812     NIBHEX 8812     2 chaÑnes num. 1 option.
99 08898          =MENUe
100 08898 71FE     GOSUB   PARM     Lit et convertit le 1er parm.
101 0889C 17F      D1=D1+ 16
102 0889F 103      R3=A           Sauve le paramÈtre
103 088A2 D0       A=0    A
104 088A4 E4       A=A+1  A
105 088A6 A4E      C=C-1  S
106 088A9 A4E      C=C-1  S
107 088AC 4C0      GOC     m10
108 088AF 7ADE     GOSUB   PARM     Lire le 2Ème paramÈtre.
109 088B3 17F      D1=D1+ 16
110 088B6 123      AR3EX
111
112 088B9 11B      m10  C=R3
113 088BC F2       CSL    A
114 088BE F2       CSL    A

```

```
115 088C0 AE6          C=A      B
116 088C3 132          AD0EX
117 088C6 8F00000     GOSBVL  =ASLW5
118 088CD 133          AD1EX
119 088D0 8F00000     GOSBVL  =ASLW5
120 088D7 104          R4=A
121 088DA 108          R0=C
122 088DD 10B          R3=C
123 088E0 7400         GOSUB   pop
124 088E4 02           NIBHEX  02
125 088E6 FF           NIBHEX  FF
126 088E8 07           pop     C=RSTK
127 088EA 135          D1=C
128 088ED 8F00000     GOSBVL  =BF2DSP
129 088F4 1B00000     D0=(5) =FUNCR0  D0 pointe la sauvegarde
130 088FB 1F00000     D1=(5) =DSPMSK
131 08902 746F         GOSUB   dsp
132 08906 87D          mAFF   ?ST=1 13
133 08909 12           GOYES   mREAD
134 0890B 311C         LCHEX   C1
135 0890F 8F00000     GOSBVL  =SFLAG?
136 08916 431          GOC     mREAD
137 08919 8F00000     GOSBVL  =GETSTC
138 08920 590          GONC    mREAD
139 08923 8D00000     GOVLNG  =MFERR
140
141 0892A 7A7E         mREAD  GOSUB   Read
142 0892E 7B5F         menuF  GOSUB   popb  Attend une touche.
143 08932 8F00000     GOSBVL  =SCRLLR
144
145 08939 8F00000     menuR  GOSBVL  =FINDA  CASE OF de la touche
146 08940 00           CON(2) =k#ATTN
147 08942 990          REL(3) quit
148 08945 00           CON(2) =k#DOWN
149 08947 A30          REL(3) menuD
150 0894A 00           CON(2) =k#UP
151 0894C 710          REL(3) menuU
152 0894F 00           CON(2) =k#TOP
153 08951 D10          REL(3) menuT
154 08954 00           CON(2) =k#BOT
155 08956 C30          REL(3) menuB
156 08959 00           CON(2) =k#EOL
157 0895B 640          REL(3) menuE
158 0895E 00           CON(2) 0
159 08960 5DC          GONC    menuF
160
161 08963 11B          menuU  C=R3
162 08966 A6E          C=C-1  B
163 08969 96E          ?C#0   B
164 0896C B0           GOYES   m+
165
166 0896E 11B          menuT  C=R3
167 08971 AE2          C=0    B
168 08974 B66          C=C+1  B
169 08977 108          m+     R0=C
170 0897A 10B          R3=C
171 0897D 688F         GOTO   mAFF
```

```
172
173 08981 11B      menuD  C=R3
174 08984 DA      A=C      A
175 08986 F4      ASR      A
176 08988 F4      ASR      A
177 0898A B66     C=C+1   B
178 0898D 9E6     ?C<A   B
179 08990 7E      GOYES   m+
180
181 08992 11B      menuB  C=R3
182 08995 DA      A=C      A
183 08997 F4      ASR      A
184 08999 F4      ASR      A
185 0899B AE6     C=A      B
186 0899E 58D     GONC    m+
187
188 089A1          menuE
189 089A1 71EE     GOSUB   popbuf
190 089A5 8F00000 GOSBVL  =CRLFND
191 089AC D0       A=0     A
192 089AE 11B     C=R3
193 089B1 AEA     A=C     B
194 089B4 8F00000 GOSBVL  =HDFLT  conversion flottant
195 089BB 04      SETHEX
196 089BD AF6     C=A     W
197 089C0 114     A=R4
198 089C3 8F00000 GOSBVL  =ASRW5
199 089CA 131     D1=A
200 089CD 8F00000 GOSBVL  =ASRW5
201 089D4 8D00000 GOVLNG  =FNRTN2
202
203 089DB          quit
204 089DB 1B00000 D0=(5) =PCADDR
205 089E2 142     A=DAT0  A
206 089E5 130     D0=A
207 089E8 8F00000 GOSBVL  =EOLSCN
208 089EF 8D00000 GOVLNG  =RUNRT1
209
210 089F6          END
```

-LINE	Ext		-	81			
ASLW5	Ext		-	117	119		
ASRW5	Ext		-	198	200		
BF2DSP	Ext		-	128			
CRLFND	Ext		-	91	190		
CURSFLL	Ext		-	76			
DATPTR	Ext		-	30			
DSPBUF	Ext		-	80			
DSPMSK	Ext		-	74	130		
EOLSCN	Ext		-	207			
FINDA	Ext		-	145			
FNRTN2	Ext		-	201			
FUNCR0	Ext		-	73	129		
GETSTC	Ext		-	137			
HDFLT	Ext		-	194			
INVPAR	Abs	34721	#000087A1	-	25	19	
IVAERR	Ext			-	25		
=MENUe	Abs	34968	#00008898	-	99		
MFERR	Ext			-	50	139	
MFERRj	Abs	34811	#000087FB	-	50		
PARM	Abs	34701	#0000878D	-	18	100	108
PCADDR	Ext			-	204		
POPBUF	Ext			-	92		
PRGMEN	Ext			-	27		
PRGMST	Ext			-	34		
R10	Abs	34763	#000087CB	-	36	33	
R11	Abs	34766	#000087CE	-	37	70	
R15	Abs	34789	#000087E5	-	45	43	
R20	Abs	34818	#00008802	-	51	40	47
R60	Abs	34841	#00008819	-	58	64	
R80	Abs	34857	#00008829	-	66	61	
R83	Abs	34871	#00008837	-	72		
RNDAHx	Ext			-	18		
RUNRT1	Ext			-	208		
Read	Abs	34728	#000087A8	-	27	141	
SCRLLR	Ext			-	143		
SFLAG?	Ext			-	135		
TKSCN+	Ext			-	46		
dsp	Abs	34922	#0000886A	-	82	75	131
k#ATTN	Ext			-	146		
k#BOT	Ext			-	154		
k#DOWN	Ext			-	148		
k#EOL	Ext			-	156		
k#TOP	Ext			-	152		
k#UP	Ext			-	150		
m+	Abs	35191	#00008977	-	169	164	179 186
m10	Abs	35001	#000088B9	-	112	107	
mAFF	Abs	35078	#00008906	-	132	171	
mREAD	Abs	35114	#0000892A	-	141	133	136 138
menuB	Abs	35218	#00008992	-	181	155	
menuD	Abs	35201	#00008981	-	173	149	
menuE	Abs	35233	#000089A1	-	188	157	
menuF	Abs	35118	#0000892E	-	142	159	
menuR	Abs	35129	#00008939	-	145		
menuT	Abs	35182	#0000896E	-	166	153	
menuU	Abs	35171	#00008963	-	161	151	
pop	Abs	35048	#000088E8	-	126	123	

popb	Abs	34957	#0000888D	-	92	142	
popbuf	Abs	34950	#00008886	-	91	48	189
quit	Abs	35291	#000089DB	-	203	147	

Input Parameters

Source file name is menu.a

Listing file name is menu.l

Object file name is menu.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE PRINT
2
3          * *****
4          * Printer support module
5          *
6          * Keyword(s):
7          *   BELL, BOLD, PCR, ESC$, PFF, PLF, MODE,
8          *   PERF, PAGELEN, UNDERLINE, WRAP
9          * Author(s):
10         *   Pierre David
11         * History:
12         *   Published in JPC #40
13         *   Changed in rev X (R. Rosenbaum)
14         *
15         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
16         * *****
17
18 089F6          ABS      #89F6
19
20         * *****
21         *
22         * U T I L I T A I R E      D ' I M P R E S S I O N *
23         *
24         * *****
25
26
27         * *****
28         * PRINT
29         *
30         * But: Cette routine imprime la chaine pointÃe par
31         * la premiÃre adresse de la pile de retours.
32         * Le retour peut s'effectuer de deux maniÃres:
33         * - PR-BAS: retour Ã Basic
34         * - PR-RTN: retour Ã la routine appelante.
35         *
36         * Principe:
37         * Le poll pPRTIS est envoyÃ, pour Ãtre interceptÃ
38         * par l'HPIL. Celui-ci renvoie l'adresse de la
39         * routine PRASC qui se chargera de l'impression.
40         *
41         * Schema de l'appel:
42         *   GOSUB  PR-BAS (ou PR-RTN)
43         *   CON(1) n
44         *   CON(2) C1
45         *   :
46         *   CON(2) Cn
47         * NB: utilise R0, R2, R3
48         * *****
49
50 089F6 844      PR-BAS ST=0  4      Le retour se fera selon
51 089F9 6600      GOTO   print  l'Ãtat du flag 4 (4 parce
52 089FD 854      PR-RTN ST=1  4      les autres Ãtaient deja
53 08A00 20       print P=    0      utilisÃs...)
54 08A02 04       SETHEX
55 08A04 8F00000  GOSBVL =POLL
56 08A0B 00       CON(2) =pPRTIS  Envoi du Poll
57 08A0D 831      ?XM=0      A-t-il ÃtÃ interceptÃ ?
```

```
58 08A10 D0          GOYES noerr   Oui: pas d'erreur
59 08A12 3100        LC(2) =eDVCNF Non: "Unit  non trouv e"
60 08A16 8D00000     GOVLNG =MFERR
61
62 08A1D             noerr
63                  *****
64                  * Modification du <861023.2238>
65                  *
66                  * nettoyage de la boucle apr s l'envoi des ordres
67                  * par appel   la routine PREND, dont l'adresse est
68                  * donn e par un REL(5) avant le code de PRASCI.
69                  *
70 08A1D 131          D1=A          D1 := ^ code de PRASCI
71 08A20 1C4          D1=D1- 5       D1 := ^ (REL(5) =PREND)
72 08A23 147          C=DAT1 A       C(A) := REL(5)
73 08A26 D5           B=C A          B(A) := REL(5)
74 08A28 137          CD1EX          C(A) := ^ #F107A
75 08A2B C9           C=C+B A       C(A) := ^ code de PREND
76 08A2D 108          R0=C
77                  *
78                  * Fin de la modification
79                  *****
80 08A30 07           C=RSTK          Adresse de la table
81 08A32 135          D1=C          dans D1
82
83
84                  ***** changes:
85                  **** rev D
86                  * C=C+1 A       C: ^ premier caract re
87                  * D=C A          D=ad. du premier caract re
88                  * GOSUB icil     Pour avoir la position du
89                  * icil C=RSTK     pointeur programme.
90                  * B=C A
91                  * LC(5) (ret1)-(icil)
92                  * C=C+B A       C=adresse de retour
93                  * RSTK=C        empil e
94                  *
95                  * Apr s le poll pPRTIS, si celui-ci est intercept 
96                  * par l'HPIL, nous avons l'adresse de la routine
97                  * qui se chargera de l'impression dans A(A).
98                  *
99                  * Le seul moyen,   ma connaissance, pour faire un
100                 * GOTO   une adresse variable est d'empiler cette
101                 * adresse dans la pile de retour, puis de faire
102                 * un RTN.
103                 *
104                 * C=A A
105                 * RSTK=C
106                 * A=0 A
107                 * A=DAT1 1       A(A)= nb de caract res.
108                 * RTN           <=> GOTO A(A)
109                 *ret1
110                 *****
111                 * Modification du <861023.2240>
112                 *
113                 * nettoyage effectif :
114                 *
```



```

115      *      GOSUB  ici2
116      *ici2  C=RSTK
117      *      A=C      A
118      *      LC(5)   (ret2)-(ici2)
119      *      C=C+A   A
120      *      RSTK=C      RSTK := ^ ret2
121      *      C=R0
122      *      RSTK=C      RSTK := ^ PREND
123      *      CD1EX
124      *      R0=C
125      *      RTN          <==> GOVLNG =PREND
126      *
127      *ret2  P=      0      # 0 en sortie de PREND
128      *      ?ST=0  4      PR-BAS ?
129      *      GOYES  nxtstm  GOYES retour È Basic
130      *      C=R0
131      *      RSTK=C
132      *      RTN          retour È l'appelant
133      *nxtstm GOVLNG =NXTSTM  sinon retour È Basic
134      *
135      * Fin de la modification
136      *****
137
138      **** rev X
139 08A35 D7      D=C      A
140 08A37 E7      D=D+1  A
141 08A39 D2      C=0      A
142 08A3B 15F0   C=DAT1  1
143 08A3F C6      C=C+C   A
144 08A41 CB      C=C+D   A
145 08A43 06      RSTK=C
146 08A45 D6      C=A      A
147 08A47 D0      A=0      A
148 08A49 15B0   A=DAT1  1
149 08A4D 7310   GOSUB  INDJ
150 08A51 118    C=R0
151 08A54 7C00   GOSUB  INDJ
152 08A58 874    ?ST=1  4
153 08A5B 00     RTNYES
154 08A5D 8D00000  nxtstm GOVLNG =NXTSTM
155 08A64 06     INDJ  RSTK=C
156 08A66 01     RTN
157      ***** end of changes
158
159
160      *****
161      * onoff?
162      * But: tester le token suivant:
163      * Sortie: Cy=1 si tON, Cy=0 sinon
164      *****
165
166 08A68 1524   onoff? A=DAT0 S
167 08A6C 948   ?A=0      S
168 08A6F 00    RTNYES      Retour avec Carry=1
169 08A71 01    RTN          Avec Cy=0
170
171

```

```
172
173      * *****
174      * BELL entry
175      * *****
176 08A73 00000      REL(5) =NOARGd
177 08A78 00000      REL(5) =NOARGp
178 08A7D 757F      =BELLe GOSUB PR-BAS
179 08A81 1          CON(1) 1      1 seul caractÈre
180 08A82 70        CON(2) 7      code 7
181
182      * *****
183      * BOLD entry
184      * *****
185 08A84 00000      REL(5) =ONOFFd
186 08A89 00000      REL(5) =ONOFFp
187 08A8E 76DF      =BOLDe GOSUB onoff?      ON ou OFF ?
188 08A92 411        GOC boldon      -> ON
189 08A95 7D5F      GOSUB PR-BAS      -> OFF
190 08A99 5          CON(1) 5
191 08A9A B1        CON(2) 27
192 08A9C 82370324  NIBASC '(s0B'
193 08AA4 7E4F      boldon GOSUB PR-BAS
194 08AA8 5          CON(1) 5
195 08AA9 B1        CON(2) 27
196 08AAB 82371324  NIBASC '(s1B'
197
198
199      * *****
200      * PCR entry
201      * *****
202 08AB3 00000      REL(5) =NOARGd
203 08AB8 00000      REL(5) =NOARGp
204 08ABD 7400      =PCRe GOSUB cr      CR est aussi utilisÅ dans
205 08AC1 6B9F      GOTO nxtstm      LF et FF
206
207      *****
208      * cr
209      *
210      * But: envoyer un retour-chariot, et remettre la
211      * position de la tÅte d'Åcriture Å 0.
212      * Note: la position de la tÅte d'Åcriture est
213      * gardÅe en mÅmoire par le HP71. Elle est utilisÅe
214      * par TAB.
215      *****
216
217 08AC5 1F00000    cr      D1=(5) =PPOS      Printer POSition
218 08ACC AE2        C=0      B
219 08ACF 14D        DAT1=C B      PPOS=0 (TAB=0)
220 08AD2 772F      GOSUB PR-RTN      Impression avec retour...
221 08AD6 1          CON(1) 1
222 08AD7 D0        CON(2) 13
223 08AD9 01        RTN
224
225
226      * *****
227      * ESC$ entry
228      * *****
```

```

229 08ADB 401          NIBHEX 401
230 08ADE 94E        =ESCe  ?C#0  S
231 08AE1 82         GOYES  ESC20
232 08AE3 137        CD1EX          D1=pointeur de pile math.
233 08AE6 109        R1=C          dans R1 pour ADHEAD
234 08AE9 135        D1=C
235 08AEC 1C1        D1=D1- 2
236 08AEF 20         P= 0
237 08AF1 31B1       LC(2) 27      ESC
238 08AF5 14D        DAT1=C B
239 08AF8 8F00000    ESC10  GOSBVL =D=AVMS  requis par Basic
240 08AFF 840        ST=0 0        Pas de retour
241 08B02 8D00000    GOVLNG =ADHEAD  Ajout de l'en-tête
242
243 08B09 136        ESC20  CD0EX
244 08B0C 06         RSTK=C
245 08B0E 8F00000    GOSBVL =POP1S
246 08B15 137        CD1EX          D1 := ^ fin de l'ère chaÑne
247
248 08B18 134        D0=C          Start of source
249
250 08B1B 135        D1=C
251 08B1E 1C1        D1=D1- 2      Start of dest
252
253 08B21 10A        R2=C          R2 := ^ fin de l'ère chaÑne
254 08B24 C2         C=C+A  A      C(A) := ^ d'Åbut de chaÑne
255 08B26 D8         B=A  A        B(A) := block length
256 08B28 109        R1=C          R1 := ^ d'Åbut de chaÑne
257 08B2B 8F00000    GOSBVL =MOVEU0
258
259 08B32 11A        C=R2
260 08B35 135        D1=C
261 08B38 1C1        D1=D1- 2      Top of Math-Stack
262 08B3B 119        C=R1
263 08B3E 134        D0=C
264 08B41 181        D0=D0- 2
265 08B44 31B1       LC(2) 27
266 08B48 14C        DAT0=C B
267
268 08B4B 07         C=RSTK
269 08B4D 134        D0=C
270 08B50 57A        GONC  ESC10   B.E.T.
271
272
273
274
275
276 08B53 00000      REL(5) =NOARGd
277 08B58 00000      REL(5) =NOARGp
278 08B5D 746F       =PFFe  GOSUB  cr      On remet d'abord PPOS È 0
279 08B61 719E       GOSUB  PR-BAS     et on envoie...
280 08B65 1          CON(1) 1         ...un caractÈre...
281 08B66 C0         CON(2) 12        ...de code 12
282
283 08B68 6EE0       Argerr GOTO  argerr
284
285

```

```
286
287      * *****
288      * PLF entry
289      * *****
290 08B6C 00000          REL(5) =STDd
291 08B71 00000          REL(5) =PLFp
292 08B76 14A          =PLFe  A=DAT0 B
293 08B79 8F00000      GOSBVL =EOLXCK
294      *****
295      * Modification du <861023.2000>
296      *
297      *
298 08B80 D0           A=0    A
299 08B82 570          GONC   lf05
300 08B85 E4           A=A+1  A
301 08B87 531          GONC   noexpr  B.E.T.
302
303 08B8A 8F00000      lf05   GOSBVL =EXPEX-
304 08B91 8F00000      GOSBVL =RNDAHX
305 08B98 5FC          GONC   Argerr   paramÉtre < 0
306
307 08B9B 101          noexpr R1=A
308 08B9E 732F         GOSUB   cr      une seule fois !
309 08BA2 6D00         GOTO    lf20
310
311      *
312      * Boucle d'envoi des LF.
313      * La boucle peut ne pas Átre exÁcutÁe (LF 0), ma
314      * ne doit pas gÅnÅrer d'erreur.
315      *
315 08BA6 101          lf10   R1=A
316 08BA9 705E         GOSUB   PR-RTN
317 08BAD 1            CON(1) 1
318 08BAE A0           CON(2) 10
319 08BB0 111          lf20   A=R1
320 08BB3 CC           A=A-1  A
321 08BB5 50F          GONC   lf10
322 08BB8 6881         GOTO    nxtstM
323      *
324      * Fin de la modification
325      *****
326
327 08BBC 6A90         ARgerr GOTO   argerr
328
329
330      * *****
331      * MODE entry
332      * *****
333 08BC0 00000          REL(5) =STDd
334 08BC5 00000          REL(5) =FIXp
335 08BCA 7760          =MODEe GOSUB  evalex   Ávaluation de ce paramÉtre
336 08BCE AF2           C=0    W
337 08BD1 A96           C=A    WP      Ici, on a P=0
338 08BD4 976           ?A#C   W      nb > 9 ?
339 08BD7 5E            GOYES  ARgerr
340 08BD9 04            SETHEX
341 08BDE 3103          LCASC  '0'      Conversion en Ascii
342 08BDF A6A          A=A+C  B
```

```

343 08BE2 8F00000          GOSBVL =ASLW4   DÅcalage È gauche de 7
344 08BE9 8F00000          GOSBVL =ASLW3   quartets
345 08BF0 3A5B162B        LCHEX  53006B261B5   ESC '&k_S'
        60035

346          *
347          * A ce niveau, il est bien important de voir ce
348          * que l'on a dans les registres:
349          *
350          * C(W)= x x x x x 5 3 0 0 6 B 2 6 1 B 5
351          * A(W)= 0 0 0 0 0 0 0 3 n 0 0 0 0 0 0 (MODE n)
352          *
353          * L'addition de A et C nous donne bien Åvidemment
354          * la bonne sÅquence d'Åchappement.
355          *
356          * C va alors Åtre transfÅrÅ en mÅmoire (AVMEMS),
357          * afin d'Åtre envoyÅ È PRASC.
358          *
359 08BFD A72              C=C+A   W
360 08C00 8F00000        GOSBVL =D0=AVS   D0 := AVMEMS
361 08C07 15CA           DAT0=C 11      La table est crÅÅe È cet
362 08C0B 136           CD0EX      endroit, pointÅ par D0.
363 08C0E 06           RSTK=C      On simule un GOSUB.
364 08C10 65ED          GOTO    PR-BAS

365
366
367          * *****
368          * PERF entry
369          * *****
370 08C14 00000          REL(5) =ONOFFd
371 08C19 00000          REL(5) =ONOFFp
372 08C1E 764E          =PERFe GOSUB onoff?
373 08C22 D0           A=0      A
374 08C24 31C4          LCASC  'L'      Fin de sÅquence
375 08C28 540           GONC    prfoff
376 08C2B E4           A=A+1   A      1=perf on
377 08C2D 7D20          prfoff GOSUB escseq
378 08C31 6B2E          GOTO    nxtstm   retour È Basic
379
380          * *****
381          * evalex, trad
382          *
383          * But:
384          * - evalex: Åvaluer la chaine de tokens pointÅe
385          * par D0, et...
386          * - trad: dÅpiler le sommet de la pile-maths,
387          * convertir le nombre en dÅcimal, et vÅrifier
388          * qu'il est infÅrieur È 999.
389          * *****
390
391 08C35 04             evalex SETHEX      requis par EXPEX-
392 08C37 8F00000        GOSBVL =EXPEX-   Evaluation...
393 08C3E 8F00000        trad  GOSBVL =RJUST Traduction en entier dÅc.
394 08C45 04             SETHEX      aprÅs RJUST
395 08C47 4F0           GOC     argerr   Inf
396 08C4A AF2           C=0      W
397 08C4D 32999         LCHEX  999      C=00...00999
398 08C52 9F2           ?A<C   W

```

```
399 08C55 00          RTNYES          Retour autoris  si < 999
400 08C57 8D00000    argerr GOVLNG =ARGERR  sinon: "invalid argument"
401
402          *****
403          * escseq
404          *
405          * But: Envoyer une s quence du type: ESC & l ### $
406          * ou ### est dans A(A), et $ dans C(B)
407          *
408          * Note:
409          * Beaucoup de s quences d' chappement sont de la
410          * forme ESC & l # $, ou:
411          * # est un nombre entier
412          * $ est un caract re (f/F, p/P, l/L ou d/D)
413          * Il est   noter que les z ros de t te dans # ne
414          * sont pas pris en compte. Cette caract ristique
415          * est tr s int ressante...
416          *
417          * Principe:
418          * - La table est construite   partir de AVMEMS:
419          *   Les caract res ESC, & et l sont d'abord
420          *   plac s, puis on calcule # sur 3 chiffres
421          *   (octets), et enfin, on place le caract re $
422          *     la fin. Il faut noter que cette s quence
423          *   d' chappement est de longueur fixe.
424          * - La table est suivie en m moire d'un code RTN
425          *   (#01). Le retour de PR-RTN se fera l , d'o 
426          *   retour final   la routine appelante...
427          *   Simple, non ?
428          *****
429
430 08C5E AE7          escseq D=C      B          Sauvegarde du caract re
431 08C61 D8           B=A        A          et du nombre
432 08C63 1B00000     D0=(5) =STMTR0
433 08C6A 367B162C   LCHEX  6C261B7  7 caract res, dont: ESC & l
434          6
434 08C73 15C6          DAT0=C 7
435 08C77 166          D0=D0+ 7
436 08C7A 815          BSRC
437 08C7D 815          BSRC          B(0-0)=premier nombre
438 08C80 3103         LCASC  '0'      C(B)=#30
439 08C84 22           P=        2      3 boucles au total
440 08C86 D4          boucle A=B      A      A(A)=0000x
441 08C88 D1           B=0        A      B=xx0000...0000
442 08C8A A6A         A=A+C      B      A(B)=3x
443 08C8D 148         DAT0=A B
444 08C90 161         D0=D0+ 2
445 08C93 811         BSLC          B=x0000...0000x
446 08C96 0D          P=P-1
447 08C98 5DE         GONC  boucle  Boucle si P>=0
448          *****
449          * Je m'avais gour  tout plein...
450          *
451 08C9B 20          P=        0
452          *
453          *****
454 08C9D AEB         C=D        B          caract re final
```

```

455 08CA0 14C          DAT0=C B
456 08CA3 161          D0=D0+ 2
457 08CA6 3101        LCHEX 10          code de RTN
458 08CAA 14C          DAT0=C B
459 08CAD 136          CD0EX           } C=D0
460 08CB0 134          D0=C           }
461 08CB3 06           RSTK=C           RSTK := ^ RTN dans la table
462
463 08CB5 18E          D0=D0- 15       D0=dÅbut de la table
464 08CB8 136          CD0EX
465 08CBB 06           RSTK=C           puis l'adresse de la table
466 08CBD 6F3D        GOTO PR-RTN     Envoi...
467
468
469
470
471
472 08CC1 00000        REL(5) =STDD
473 08CC6 00000        REL(5) =PAGElp
474 08CCB              =PAGEle
475 08CCB 14A          A=DAT0 B        A(B)=token suivant
476 08CCE 8F00000     GOSBVL =EOLXCK  Fin de ligne ?
477 08CD5 5D1          GONC  parametres
478 08CD8 7A1D        GOSUB PR-BAS     Oui: chaÑne par dÅfaut
479 08CDC B            CON(1) 11        11 caractÉres
480 08CDD B1          CON(2) 27
481
482
483 08CDF 62C66363     * change:
      07630366        * NIBASC '&l72p66f' * rev D
      * NIBASC '&l66p60f' * rev X
484 08CEF 13C4          NIBASC '1L'
485 08CF3              parametres
486
487
488
489
490
491
492
493
494
495 08CF3 7E3F          GOSUB evalex     Evaluation des paramÉtres
496
497 08CF7 101          * D1 = ^ 72 (66)
      R1=A           R1 := dernier paramÉtre
498 08CFA 17F          D1=D1+ 16        DÅpilement
499 08CFD 133          AD1EX
500
501
502
503
504
505 08D00 1B00000     *****
      * Modification du <861023.2257>
      *
506 08D07 146          * Inversion de l'ordre d'envoi des deux paramÉtres
      *
      D0=(5) =FORSTK
507 08D0A 859          C=DAT0 A
      ST=1 9
508 08D0D 8BE          ?A>=C A         Un seul paramÉtre ?
509 08D10 61           GOYES unique     Oui: longueur de la page
510 08D12 849          ST=0 9           pas qu'un seul paramÉtre

```

```
511 08D15 131          D1=A
512 08D18 8F0000      GOSBVL =POP1R
513 08D1F 7B1F        GOSUB trad      A = premier paramÉtre
514 08D23 133          AD1EX
515 08D26              unique
516 08D26 133          AD1EX
517 08D29 3105        LCASC 'P'
518 08D2D 7D2F        GOSUB escseq
519 08D31 879         ?ST=1 9
520 08D34 D0          GOYES  nxtstmM
521 08D36 111         A=R1
522 08D39 3164        LCASC 'F'
523 08D3D 7D1F        GOSUB  escseq
524
525 08D41 6B1D        nxtstm GOTO  nxtstm  Retour È Basic
526 *
527 * Fin de la modification
528 *****
529
530
531 * *****
532 * UNDERLINE entry
533 * *****
534 08D45 00000        REL(5) =UNDERd
535 08D4A 00000        REL(5) =UNDERp
536 08D4F              =UNDERe
537 08D4F 751D        GOSUB  onoff?
538 08D53 4F0         GOC    UNDon
539 08D56 7C9C        GOSUB  PR-BAS   SÅquence d'Åchappement
540 08D5A 4           CON(1) 4       difficile È paramÅtrer
541 08D5B B1          CON(2) 27
542 08D5D 624604      NIBASC '&d@'
543 08D63 7F8C        UNDon  GOSUB  PR-BAS
544 08D67 4           CON(1) 4
545 08D68 B1          CON(2) 27
546 08D6A 624644      NIBASC '&dD'
547
548
549 * *****
550 * WRAP entry
551 * *****
552 08D70 00000        REL(5) =ONOFFd
553 08D75 00000        REL(5) =ONOFFp
554 08D7A 7AEC        =WRAPe GOSUB  onoff?
555 08D7E 411         GOC    wrapon
556 08D81 717C        GOSUB  PR-BAS
557 08D85 5           CON(1) 5
558 08D86 B1          CON(2) 27
559 08D88 62371334    NIBASC '&s1C'
560 08D90 726C        wrapon GOSUB  PR-BAS
561 08D94 5           CON(1) 5
562 08D95 B1          CON(2) 27
563 08D97 62370334    NIBASC '&s0C'
564
565 08D9F              END
```


ADHEAD	Ext		-	241							
ARGERR	Ext		-	400							
ARgerr	Abs	35772	#00008BBC	-	327	339					
ASLW3	Ext		-	344							
ASLW4	Ext		-	343							
Argerr	Abs	35688	#00008B68	-	283	305					
=BELLe	Abs	35453	#00008A7D	-	178						
=BOLDe	Abs	35470	#00008A8E	-	187						
D0=AVS	Ext		-	360							
D=AVMS	Ext		-	239							
EOLXCK	Ext		-	293	476						
ESC10	Abs	35576	#00008AF8	-	239	270					
ESC20	Abs	35593	#00008B09	-	243	231					
=ESCe	Abs	35550	#00008ADE	-	230						
EXPEX-	Ext		-	303	392						
FIXp	Ext		-	334							
FORSTK	Ext		-	505							
INDJ	Abs	35428	#00008A64	-	155	149	151				
MFERR	Ext		-	60							
=MODEe	Abs	35786	#00008BCA	-	335						
MOVEU0	Ext		-	257							
NOARGd	Ext		-	176	202	276					
NOARGp	Ext		-	177	203	277					
NXTSTM	Ext		-	154							
ONOFFd	Ext		-	185	370	552					
ONOFFp	Ext		-	186	371	553					
=PAGELe	Abs	36043	#00008CCB	-	474						
PAGELp	Ext		-	473							
=PCRe	Abs	35517	#00008ABD	-	204						
=PERFe	Abs	35870	#00008C1E	-	372						
=PFfe	Abs	35677	#00008B5D	-	278						
=PLFe	Abs	35702	#00008B76	-	292						
PLFp	Ext		-	291							
POLL	Ext		-	55							
POP1R	Ext		-	512							
POP1S	Ext		-	245							
PPOS	Ext		-	217							
PR-BAS	Abs	35318	#000089F6	-	50	178	189	193	279	364	478
					539	543	556	560			
PR-RTN	Abs	35325	#000089FD	-	52	220	316	466			
RJUST	Ext		-	393							
RNDAHx	Ext		-	304							
STDD	Ext		-	290	333	472					
STMTR0	Ext		-	432							
UNDERd	Ext		-	534							
=UNDERe	Abs	36175	#00008D4F	-	536						
UNDERp	Ext		-	535							
UNDon	Abs	36195	#00008D63	-	543	538					
=WRAPe	Abs	36218	#00008D7A	-	554						
argerr	Abs	35927	#00008C57	-	400	283	327	395			
boldon	Abs	35492	#00008AA4	-	193	188					
boucle	Abs	35974	#00008C86	-	440	447					
cr	Abs	35525	#00008AC5	-	217	204	278	308			
eDVCNF	Ext		-	59							
escseq	Abs	35934	#00008C5E	-	430	377	518	523			
evalcx	Abs	35893	#00008C35	-	391	335	495				
lf05	Abs	35722	#00008B8A	-	303	299					

lf10	Abs	35750	#00008BA6	-	315	321			
lf20	Abs	35760	#00008BB0	-	319	309			
noerr	Abs	35357	#00008A1D	-	62	58			
noexpr	Abs	35739	#00008B9B	-	307	301			
nxtstm	Abs	36161	#00008D41	-	525	322	520		
nxtstm	Abs	35421	#00008A5D	-	154	205	378	525	
onoff?	Abs	35432	#00008A68	-	166	187	372	537	554
pPRTIS	Ext			-	56				
parametres	Abs	36083	#00008CF3	-	485	477			
prfoff	Abs	35885	#00008C2D	-	377	375			
print	Abs	35328	#00008A00	-	53	51			
trad	Abs	35902	#00008C3E	-	393	513			
unique	Abs	36134	#00008D26	-	515	509			
wrapon	Abs	36240	#00008D90	-	560	555			

Input Parameters

Source file name is print.a

Listing file name is print.l

Object file name is print.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE REP
2
3          * *****
4          * String replace module
5          *
6          * Keyword(s):
7          *   REPLACE$
8          * Author(s):
9          *   Michel Martinet
10         * History:
11         *   Originally published in JPC#23
12         *   Changed to integrate the RPLC$ form.
13         *
14         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
15         * *****
16
17 08D9F          ABS      #8D9F
18
19 08D9F 8A9      Egale   ?B=0  A
20 08DA2 00          RTNYES
21 08DA4 181          D0=D0- 2
22 08DA7 1C1          D1=D1- 2
23 08DAA 14B          A=DAT1 B
24 08DAD 118          C=R0
25 08DB0 92E          ?C#0  XS
26 08DB3 70          GOYES  NJok
27 08DB5 962          ?A=C  B
28 08DB8 A0          GOYES  Jok
29 08DEA 14E          NJok   C=DAT0 B
30 08DBD 966          ?A#C  B
31 08DC0 00          RTNYES
32 08DC2 CD          Jok    B=B-1 A
33 08DC4 CD          B=B-1 A
34 08DC6 58D          GONC   Egale
35
36 08DC9 11C          C=R495 C=R4
37 08DCC 8D00000     GOVLNG =CSRC5
38 08DD3 8D00000     Argerr GOVLNG =ARGERR
39
40
41         * *****
42         * REPLACE$ entry
43         * *****
44 08DDA C44434          NIBHEX C44434
45 08DE0 80DF          =REPLAe P=C 15
46 08DE4 894          ?P= 4
47 08DE7 A0          GOYES  REP
48 08DE9 20          RPLC   P= 0
49 08DEB 8C0000     GOLONG =RPLCe  call RPLC$ form
50 08DF1 20          REP    P= 0
51 08DF3 1534          A=DAT1 S
52 08DF7 B44          A=A+1 S
53 08DFA 5EE          GONC   RPLC
54
55 08DFD 80DF          P=C    15
56 08E01 884          ?P#    4
57 08E04 46          GOYES  NJ
```

```
58 08E06 8F00000      GOSBVL =POP1S
59 08E0D 8A8          ?A=0   A
60 08E10 85           GOYES  NJ
61 08E12 3420000     LC(5)  2
62 08E19 8B6          ?A>C   A
63 08E1C 7B           GOYES  Argerr
64 08E1E 80C2         C=P    2
65 08E22 14F          C=DAT1 B
66 08E25 108          R0=C
67 08E28 171          D1=D1+ 2
68 08E2B 554          GONC   REPC
69
70 08E2E 11B          Test   C=R3
71 08E31 8AE          ?C#0   A
72 08E34 B0           GOYES  Test2
73 08E36 8A8          ?A=0   A
74 08E39 D0           GOYES  Exp1
75 08E3B 6110         GOTO   Exp
76 08E3F 8A8          Test2  ?A=0   A
77 08E42 B0           GOYES  Exp
78 08E44 01           RTN
79
80 08E46 7F7F         Exp1   GOSUB  C=R495
81 08E4A 135          D1=C
82 08E4D 1CF         Exp    D1=D1- 16
83 08E50 07           C=RSTK
84 08E52 6AA1         GOTO   Expr
85
86 08E56 137          A=AVMS CD1EX
87 08E59 1F00000     D1=(5) =AVMEMS
88 08E60 143          A=DAT1 A
89 08E63 135          D1=C
90 08E66 01           RTN
91
92 08E68 0C          NJ     P=P+1
93 08E6A 80C2         C=P    2
94 08E6E 108          R0=C
95
96 08E71 8F00000     REPC   GOSBVL =POP1S
97 08E78 137          CD1EX
98 08E7B 135          D1=C
99 08E7E 8F00000     GOSBVL =CSLC5
100 08E85 10C          R4=C
101 08E88 137          CD1EX
102 08E8B C2           C=C+A  A
103 08E8D 135          D1=C
104
105 08E90 8F00000     GOSBVL =POP1S
106 08E97 137          CD1EX
107 08E9A C2           C=C+A  A
108 08E9C 135          D1=C
109 08E9F 103          R3=A
110
111 08EA2 8F00000     GOSBVL =POP1S
112 08EA9 718F         GOSUB  Test
113 08EAD 137          CD1EX
114 08EB0 10A          R2=C
```

```
115 08EB3 C2          C=C+A  A
116 08EB5 D7          D=C    A
117 08EB7 D2          C=0    A
118 08EB9 109         R1=C
119 08EBC 118         C=R0
120 08EBF ABA         A=C    X
121 08EC2 136         CD0EX
122 08EC5 BF2         CSL    W
123 08EC8 BF2         CSL    W
124 08ECB BF2         CSL    W
125 08ECE AB6         C=A    X
126 08ED1 108         R0=C
127 08ED4 1B00000     D0=(5) =F-R0-0
128 08EDB DB          C=D    A
129 08EDD 144         DAT0=C A
130
131 08EE0 113         TestAB A=R3
132 08EE3 11A         C=R2
133 08EE6 D5          B=C    A
134 08EE8 DB          C=D    A
135 08EEA E9          C=C-B  A
136 08EEC 8B6         ?C<A  A
137 08EEF 65          GOYES  Stk1
138 08EF1 DB          C=D    A
139 08EF3 134         D0=C
140 08EF6 11A         C=R2
141 08EF9 135         D1=C
142 08EFC 1CF         D1=D1- 16
143 08EFF 11B         C=R3
144 08F02 D5          B=C    A
145 08F04 779E        GOSUB  Egale
146 08F08 8AD         ?B#0  A
147 08F0B A3          GOYES  Stk1
148 08F0D 78BE        GOSUB  C=R495
149 08F11 135         D1=C
150 08F14 1CD         D1=D1- 14
151 08F17 147         C=DAT1 A
152 08F1A 1C1         D1=D1- 2
153 08F1D 137         CD1EX
154 08F20 111         A=R1
155 08F23 E2          C=C-A  A
156 08F25 137         CD1EX
157 08F28 DA          A=C    A
158 08F2A 7B9E        GOSUB  C=R495
159 08F2E C2          C=C+A  A
160 08F30 136         CD0EX
161 08F33 D8          B=A    A
162 08F35 7D1F        GOSUB  A=AVMS
163 08F39 7040        GOSUB  Stk
164 08F3D 11B         C=R3
165 08F40 E3          D=D-C  A
166 08F42 5E2         GONC   StkC
167
168 08F45 7D0F        Stk1   GOSUB  A=AVMS
169 08F49 DB          C=D    A
170 08F4B 134         D0=C
171 08F4E 119         C=R1
```

```
172 08F51 D5          B=C    A
173 08F53 727E        GOSUB  C=R495
174 08F57 E9          C=C-B  A
175 08F59 135         D1=C
176 08F5C 1CF         D1=D1- 16
177 08F5F 3420000     LC(5)  2
178 08F66 D5          B=C    A
179 08F68 7110        GOSUB  Stk
180 08F6C 136         CD0EX
181 08F6F D7          D=C    A
182 08F71 11A         StkC   C=R2
183 08F74 8A3         ?C=D   A
184 08F77 E3          GOYES  Fin
185 08F79 666F        GOTO   TestAB
186
187 08F7D 137         Stk    CD1EX
188 08F80 E9          C=C-B  A
189 08F82 8B6         ?A>C   A
190 08F85 92          GOYES  Memerr
191 08F87 C9          C=C+B  A
192 08F89 135         D1=C
193 08F8C 8A9         Boucle ?B=0   A
194 08F8F 00          RTNYES
195 08F91 181         D0=D0- 2
196 08F94 1C1         D1=D1- 2
197 08F97 14E         C=DAT0 B
198 08F9A 14D         DAT1=C B
199 08F9D 119         C=R1
200 08FA0 E6          C=C+1  A
201 08FA2 E6          C=C+1  A
202 08FA4 109         R1=C
203 08FA7 CD          B=B-1  A
204 08FA9 CD          B=B-1  A
205 08FAB 50E        GONC   Boucle
206
207 08FAE 8D00000     Memerr GOVLNG =MEMERR
208
209 08FB5 1F00000     Fin    D1=(5) =F-R0-0
210 08FBC 147         C=DAT1 A
211 08FBF 135         D1=C
212 08FC2 730E        GOSUB  C=R495
213 08FC6 DA          A=C    A
214 08FC8 3401000     LC(5)  16
215 08FCF EA          A=A-C  A
216 08FD1 119         C=R1
217 08FD4 8F00000     GOSBVL =MOVEDD
218 08FDB 1CF         D1=D1- 16
219 08FDE 118         C=R0
220 08FE1 BF6         CSR    W
221 08FE4 BF6         CSR    W
222 08FE7 BF6         CSR    W
223 08FEA 134         D0=C
224 08FED 119         C=R1
225 08FF0 BF2         CSL    W
226 08FF3 BF2         CSL    W
227 08FF6 30F         LCHEX  F
228 08FF9 15D6        DAT1=C 7
```

Saturn Assembler JPC LEX: MODULE REP
Ver. 1.56, 12/20/89

Tue May 23 09:21:18 2006
rep.a Page 5

229 08FFD 8D00000 Expr GOVLNG =EXPR
230
231 09004 END

A=AVMS	Abs	36438	#00008E56	-	86	162	168		
ARGERR	Ext			-	38				
AVMEMS	Ext			-	87				
Argerr	Abs	36307	#00008DD3	-	38	63			
Boucle	Abs	36748	#00008F8C	-	193	205			
C=R495	Abs	36297	#00008DC9	-	36	80	148	158	173 212
CSLC5	Ext			-	99				
CSRC5	Ext			-	37				
EXPR	Ext			-	229				
Egale	Abs	36255	#00008D9F	-	19	34	145		
Exp	Abs	36429	#00008E4D	-	82	75	77		
Exp1	Abs	36422	#00008E46	-	80	74			
Expr	Abs	36861	#00008FFD	-	229	84			
F-R0-0	Ext			-	127	209			
Fin	Abs	36789	#00008FB5	-	209	184			
Jok	Abs	36290	#00008DC2	-	32	28			
MEMERR	Ext			-	207				
MOVEDD	Ext			-	217				
Memerr	Abs	36782	#00008FAE	-	207	190			
NJ	Abs	36456	#00008E68	-	92	57	60		
NJok	Abs	36282	#00008DBA	-	29	26			
POP1S	Ext			-	58	96	105	111	
REP	Abs	36337	#00008DF1	-	50	47			
REPC	Abs	36465	#00008E71	-	96	68			
=REPLAe	Abs	36320	#00008DE0	-	45				
RPLC	Abs	36329	#00008DE9	-	48	53			
RPLCe	Ext			-	49				
Stk	Abs	36733	#00008F7D	-	187	163	179		
Stk1	Abs	36677	#00008F45	-	168	137	147		
StkC	Abs	36721	#00008F71	-	182	166			
Test	Abs	36398	#00008E2E	-	70	112			
Test2	Abs	36415	#00008E3F	-	76	72			
TestAB	Abs	36576	#00008EE0	-	131	185			

Input Parameters

Source file name is rep.a

Listing file name is rep.l

Object file name is rep.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE ROMAN
2
3          * *****
4          * Roman character set module
5          *
6          * Keyword(s):
7          *   ROMAN
8          * Author(s):
9          *   Pierre David, Janick Taillandier
10         * History:
11         *   No source available (never published)
12         *   Source file rebuilt from JPC LEX disassembly
13         *
14         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
15         * *****
16
17 09004          ABS #9004
18
19 09004 3181      o8FE5 LCHEX 18
20 09008 8D00000  o8FE9 GOVLNG =BSERR
21 0900F 79F0     o8FF0 GOSUB o90ED
22 09013 8F00000  GOSBVL =IOFND0
23 0901A 431      GOC o900F
24 0901D 7BE0     GOSUB o90ED
25 09021 8F00000  GOSBVL =I/OALL
26 09028 5FD      GONC o8FE9
27 0902B 492      GOC o9036
28 0902E 78E0     o900F GOSUB o90FB
29 09032 422      GOC o9036
30 09035 73D0     GOSUB o90ED
31 09039 8F00000  GOSBVL =I/OEXP
32 09040 53C      GONC o8FE5
33 09043 DF        DCEX A
34 09045 137      CD1EX
35 09048 176      D1=D1+ 7
36 0904B 8F00000  GOSBVL =MOVED2
37 09052 1C6      D1=D1- 7
38 09055 7DC0     o9036 GOSUB WRTBLA
39 09059 174      D1=D1+ 5
40 0905C 3110     LCHEX 01
41 09060 14D      DAT1=C B
42 09063 1CB      D1=D1- 12
43 09066 301      LCHEX 1
44 09069 15D0     DAT1=C 1
45 0906D 8D00000  o904E GOVLNG =NXTSTM
46
47
48
49         * *****
50         * ROMAN entry
51         * *****
52 09074 00000     REL(5) =ONOFFd
53 09079 00000     REL(5) =ONOFFp
54 0907E 14A       =ROMANe A=DAT0 B
55 09081 3100     LC(2) =tON
56 09085 962      ?C=A B
57 09088 78        GOYES o8FF0
```

```
58 0908A 7E70      GOSUB 090ED
59 0908E 8F00000   GOSEVL =IOFND0
60 09095 5B4       GONC 090C2
61 09098 7E70      GOSUB 090FB
62 0909C 8BA       ?C>=A A
63 0909F 03        GOYES 090B0
64 090A1 EA        A=A-C A
65 090A3 D8        B=A A
66 090A5 133       ADLEX
67 090A8 131       D1=A
68 090AB CA        A=A+C A
69 090AD 130       D0=A
70 090B0 8F00000   GOSEVL =MOVEU0
71 090B7 7150      GOSUB 090ED
72 090BB 8F00000   GOSEVL =I/OCON
73 090C2 180       D0=D0- 1
74 090C5 D2        C=0 A
75 090C7 15C0      DAT0=C 1
76 090CB 6510      GOTO 090C2
77 090CF 7520      090B0 GOSUB 090D9
78 090D3 4D0       GOC 090C2
79 090D6 7230      GOSUB 090ED
80 090DA 8F00000   GOSEVL =I/ODAL
81 090E1 6B8F      090C2 GOTO 0904E
82 090E5 02        090C6 RTNSC
83
84                 * entry point for pCONFIG poll
85                 * find and check buffer
86 090E7           =ACBFND
87 090E7 8F00000   GOSEVL =IOFND0
88 090EE 56F       GONC 090C6
89 090F1 7520      GOSUB 090FB
90 090F5 5FE       GONC 090C6
91 090F8 174       090D9 D1=D1+ 5
92 090FB 14B       A=DAT1 B
93 090FE 1C4       D1=D1- 5
94 09101 3110      LCHEX 01
95 09105 966       ?C#A B
96 09108 00        RTNYES
97 0910A 01        RTN
98
99 0910C D2        090ED C=0 A
100 0910E 307      LCHEX 7
101 09111 D5       B=C A
102 09113 32000   LC(3) =bALTCH
103 09118 01       RTN
104 0911A D2       090FB C=0 A
105 0911C 307      LCHEX 7
106 0911F 8A2      ?C=A A
107 09122 00       RTNYES
108 09124 01       RTN
109
110                 * entry point for pCONFIG poll
111                 * write ROMAN table address @D1
112 09126           =WRTBLA
113 09126 7306     GOSUB 0970E
114 0912A 006     NIBHEX 006 LENGTH
```

```
115 0912D F0204FF8 NIBHEX F0204FF80F00
      0F00
116 09139 3151DF02 NIBHEX 3151DF028F00
      8F00
117 09145 3151DD02 NIBHEX 3151DD028D00
      8D00
118 09151 F1515D02 NIBHEX F1515D028D00
      8D00
119 0915D F151110F NIBHEX F151110F0100
      0100
120 09169 F15F190D NIBHEX F15F190D0F00
      0F00
121 09175 E050EF02 NIBHEX E050EF020D00
      0D00
122 09181 07E43CE4 NIBHEX 07E43CE40700
      0700
123 0918D F151A98A NIBHEX F151A98A8E00
      8E00
124 09199 F140F10F NIBHEX F140F10F0100
      0100
125 091A5 F0808F05 NIBHEX F0808F050100
      0100
126 091B1 7080710F NIBHEX 7080710F0100
      0100
127 091BD F0501F05 NIBHEX F0501F050100
      0100
128 091C9 F0909F86 NIBHEX F0909F868B00
      8B00
129 091D5 3151DF88 NIBHEX 3151DF888F00
      8F00
130 091E1 3151D100 NIBHEX 3151D1008F00
      8F00
131 091ED F090600F NIBHEX F090600F0800
      0800
132 091F9 F0906001 NIBHEX F09060018F00
      8F00
133 09205 F090EC8A NIBHEX F090EC8A8B00
      8B00
134 09211 F090E88A NIBHEX F090E88A8F00
      8F00
135 0921D F0906605 NIBHEX F09066058F00
      8F00
136 09229 F020FF02 NIBHEX F020FF020D00
      0D00
137 09235 3151D30C NIBHEX 3151D30C0300
      0300
138 09241 F1519F8A NIBHEX F1519F8A0500
      0500
139 0924D F09F9204 NIBHEX F09F92040F00
      0F00
140 09259 F1511F02 NIBHEX F1511F020F00
      0F00
141 09265 3151DF8A NIBHEX 3151DF8A0500
      0500
142 09271 F1511F09 NIBHEX F1511F090900
      0900
143 0927D F150998A NIBHEX F150998A8E00
```

```
      8E00
144 09289 F090998A NIBHEX F090998A8E00
      8E00
145 09295 F1D0798A NIBHEX F1D0798A8E00
      8E00
146 092A1 F080F98A NIBHEX F080F98A8E00
      8E00
147 092AD 00000000 NIBHEX 000000000000
      0000
148 092B9 87516141 NIBHEX 875161418700
      8700
149 092C5 87615161 NIBHEX 876151618700
      8700
150 092D1 C7556545 NIBHEX C75565454500
      4500
151 092DD C7655565 NIBHEX C76555654500
      4500
152 092E9 C7554555 NIBHEX C75545554500
      4500
153 092F5 0064D764 NIBHEX 0064D7640000
      0000
154 09301 0054C754 NIBHEX 0054C7540000
      0000
155 0930D 00002010 NIBHEX 000020100000
      0000
156 09319 00102000 NIBHEX 001020000000
      0000
157 09325 00201020 NIBHEX 002010200000
      0000
158 09331 00100010 NIBHEX 001000100000
      0000
159 0933D 00201020 NIBHEX 002010201000
      1000
160 09349 C3142404 NIBHEX C3142404C300
      C300
161 09355 C3241424 NIBHEX C3241424C300
      C300
162 09361 45E75514 NIBHEX 45E755142400
      2400
163 0936D 10101010 NIBHEX 101010101000
      1000
164 09379 00000000 NIBHEX 000000000000
      0000
165 09385 00000000 NIBHEX 000000000000
      0000
166 09391 20502000 NIBHEX 205020000000
      0000
167 0939D E3141C14 NIBHEX E3141C142200
      2200
168 093A9 83444C44 NIBHEX 83444C4444400
      4400
169 093B5 C7A01122 NIBHEX C7A01122D700
      D700
170 093C1 87A090A0 NIBHEX 87A090A01700
      1700
171 093CD 0000D700 NIBHEX 0000D7000000
      0000
```

172	093D9	03845404 0200	NIBHEX	038454040200
173	093E5	D5222222 D500	NIBHEX	D5222222D500
174	093F1	84E79414 2400	NIBHEX	84E794142400
175	093FD	92A2C7A2 9200	NIBHEX	92A2C7A29200
176	09409	02E45593 2000	NIBHEX	02E455932000
177	09415	0284E390 2000	NIBHEX	0284E3902000
178	09421	8344EF44 4400	NIBHEX	8344EF444400
179	0942D	02655565 8700	NIBHEX	026555658700
180	09439	83655565 8100	NIBHEX	836555658100
181	09445	83645464 8300	NIBHEX	836454648300
182	09451	83241424 8700	NIBHEX	832414248700
183	0945D	02456555 8700	NIBHEX	024565558700
184	09469	83456555 8100	NIBHEX	834565558100
185	09475	83446454 8300	NIBHEX	834464548300
186	09481	C3042414 C700	NIBHEX	C3042414C700
187	0948D	02455565 8700	NIBHEX	024555658700
188	09499	83556545 8100	NIBHEX	835565458100
189	094A5	83546444 8300	NIBHEX	835464448300
190	094B1	C3142404 C700	NIBHEX	C3142404C700
191	094BD	02554555 8700	NIBHEX	025545558700
192	094C9	83554555 8100	NIBHEX	835545558100
193	094D5	83544454 8300	NIBHEX	835444548300
194	094E1	C3140414 C700	NIBHEX	C3140414C700
195	094ED	87715171 8700	NIBHEX	877151718700
196	094F9	00A49724 0000	NIBHEX	00A497240000
197	09505	CB26A564 D300	NIBHEX	CB26A564D300
198	09511	E790F794 9400	NIBHEX	E790F7949400
199	0951D	02755575 8700	NIBHEX	027555758700
200	09529	0084A714	NIBHEX	0084A7140000

```
0000
201 09535 8B4645C4 NIBHEX 8B4645C4A300
      A300
202 09541 42458745 NIBHEX 424587458400
      8400
203 0954D 87514151 NIBHEX 875141518700
      8700
204 09559 0094A704 NIBHEX 0094A7040000
      0000
205 09565 C3342434 NIBHEX C3342434C300
      C300
206 09571 E3140414 NIBHEX E3140414E300
      E300
207 0957D C7456555 NIBHEX C74565554500
      4500
208 09589 0054C714 NIBHEX 0054C7140000
      0000
209 09595 EF109494 NIBHEX EF1094946300
      6300
210 095A1 87685868 NIBHEX 876858688700
      8700
211 095AD 87416151 NIBHEX 874161518700
      8700
212 095B9 87615161 NIBHEX 876151619700
      9700
213 095C5 02655565 NIBHEX 026555659700
      9700
214 095D1 80F79414 NIBHEX 80F79414E300
      E300
215 095DD 0384A4F7 NIBHEX 0384A4F72000
      2000
216 095E9 0044E754 NIBHEX 0044E7540000
      0000
217 095F5 0054E744 NIBHEX 0054E7440000
      0000
218 09601 87486858 NIBHEX 874868588700
      8700
219 0960D 87586848 NIBHEX 875868488700
      8700
220 09619 83645464 NIBHEX 836454649300
      9300
221 09625 03A494A4 NIBHEX 03A494A41300
      1300
222 09631 85556555 NIBHEX 855565554300
      4300
223 0963D 84556555 NIBHEX 845565550200
      0200
224 09649 E3042414 NIBHEX E3042414E300
      E300
225 09655 60900790 NIBHEX 609007906000
      6000
226 09661 C11A0A1A NIBHEX C11A0A1AC700
      C700
227 0966D 18FF5A42 NIBHEX 18FF5A428100
      8100
228 09679 FF444444 NIBHEX FF4444448300
      8300
```



```
229 09685 00000000 NIBHEX 000000000000
      0000
230 09691 00000000 NIBHEX 000000000000
      0000
231 0969D 00000000 NIBHEX 000000000000
      0000
232 096A9 00000000 NIBHEX 000000000000
      0000
233 096B5 80808080 NIBHEX 808080808080
      8080
234 096C1 22F18645 NIBHEX 22F18645AF00
      AF00
235 096CD 22F18CCA NIBHEX 22F18CCAAB00
      AB00
236 096D9 84555555 NIBHEX 84555555E500
      E500
237 096E5 E4151515 NIBHEX E4151515E400
      E400
238 096F1 8041A241 NIBHEX 8041A2412200
      2200
239 096FD E7E7E7E7 NIBHEX E7E7E7E7E700
      E700
240 09709 2241A241 NIBHEX 2241A2418000
      8000
241 09715 4444F544 NIBHEX 4444F5444400
      4400
242 09721 00000000 NIBHEX 000000000000
      0000
243 0972D 07      o970E C=RSTK
244 0972F 145     DAT1=C A
245 09732 03     RTNCC
246
247 09734      END
```

=ACBFND	Abs	37095	#000090E7	-	86					
BSERR	Ext			-	20					
I/OALL	Ext			-	25					
I/OCON	Ext			-	72					
I/ODAL	Ext			-	80					
I/OEXP	Ext			-	31					
IOFND0	Ext			-	22	59	87			
MOVED2	Ext			-	36					
MOVEU0	Ext			-	70					
NXTSTM	Ext			-	45					
ONOFFd	Ext			-	52					
ONOFFp	Ext			-	53					
=ROMANe	Abs	36990	#0000907E	-	54					
=WRTBLA	Abs	37158	#00009126	-	112	38				
bALTCH	Ext			-	102					
o8FE5	Abs	36868	#00009004	-	19	32				
o8FE9	Abs	36872	#00009008	-	20	26				
o8FF0	Abs	36879	#0000900F	-	21	57				
o900F	Abs	36910	#0000902E	-	28	23				
o9036	Abs	36949	#00009055	-	38	27	29			
o904E	Abs	36973	#0000906D	-	45	81				
o90B0	Abs	37071	#000090CF	-	77	63				
o90C2	Abs	37089	#000090E1	-	81	60	76	78		
o90C6	Abs	37093	#000090E5	-	82	88	90			
o90D9	Abs	37112	#000090F8	-	91	77				
o90ED	Abs	37132	#0000910C	-	99	21	24	30	58	71
o90FB	Abs	37146	#0000911A	-	104	28	61	89		
o970E	Abs	38701	#0000972D	-	243	113				
tON	Ext			-	55					

Input Parameters

Source file name is roman.a

Listing file name is roman.l

Object file name is roman.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE RPLC
2
3          * *****
4          * String replace module
5          *   (called by the rep module)
6          *
7          * Keyword(s):
8          *   none, called by REPLACE$
9          * Author(s):
10         *   Jean-Jacques Moreau
11         * History:
12         *   RPLC$ function originally published in JPC#35
13         *   Now called from REPLACE$ keyword
14         *
15         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
16         * *****
17
18 09734          ABS      #9734
19
20         * local symbols:
21         RegExp EQU 0
22         TopLvl EQU 7
23         Match EQU 9
24         Anchor EQU 11
25         First EQU 5
26         Close EQU 6
27         Ins EQU 4
28         BackS EQU 7
29         BackSl EQU 92          Backslash character
30
31         * Remplace dans S$, O$ par R$
32         * R$=R1$[&'\'&' [&R2$]]
33
34         * OUTBS      : DÅbut de R1$
35         * FUNCR0+25 : Fin de R1$ - DÅbut de R2$
36         * FUNCR0    : Fin de R2$ - DÅbut de O$
37         * AVMEMS    : Fin de O$
38         * FUNCR0+5  : DÅbut de S$
39         * FUNCR0+30 : Pointeur actif
40         * FUNCR0+20 : Fin de S$
41
42
43         * *****
44         * former RPLC$ entry point
45         * *****
46 09734 136      =RPLCe CD0EX          Sauvegarde D0
47 09737 1B00000          D0=(5) =FUNCD1
48 0973E 144          DAT0=C A
49 09741 80DF          P=C 15          La valeur par dÅfaut du
50 09745 893          ?P= 3          4Ème paramÈtre est zÅro
51 09748 81          GOYES REP04
52 0974A 8F00000          GOSBVL =POP1R
53 09751 17F          D1=D1+ 16
54 09754 8F00000          GOSBVL =FLTDH
55 0975B CC          A=A-1 A          -> Option Base 0
56 0975D 540          GONC REP05
57 09760 D0          REP04 A=0 A          ConsidÈre 0 comme 1
```

```

58 09762 08      REP05  CLRST
59 09764 20          P=      0
60 09766 1B00000   D0=(5) (=FUNCR0)+15
61 0976D 140       DAT0=A  A
62 09770 CC        A=A-1  A
63 09772 450       GOC      REP10
64 09775 855       ST=1    First
65
66 09778 8F00000   REP10  GOSBVL  =REVPOP  Analyse R$
67 0977F 137       CD1EX
68 09782 135       D1=C          D1 @ dÅbut de R$
69 09785 C2        C=C+A  A
70 09787 D7        D=C      A      D(A) @ fin de R$
71 09789 8F00000   GOSBVL  =D0=AVS  A(A)=AVMEMS
72 09790 1B00000   REP15  D0=(5) (=FUNCR0)+25
73 09797 140       DAT0=A  A      R1$='' - R2$=R$
74 0979A 130       D0=A
75 0979D 137       REP20  CD1EX
76 097A0 135       D1=C
77 097A3 8BF       ?C>=D  A      Fin de R$?
78 097A6 65        GOYES  REP100  Oui; analyse terminÅe
79 097A8 14B       A=DAT1  B      Lit un caractÈre
80 097AB 171       D1=D1+ 2
81 097AE 31C5      LC(2)  BackS1  LCASC '\\'
82 097B2 962       ?A=C   B      Est-ce 1 '\\' ?
83 097B5 71        GOYES  REP30    Oui
84 097B7 3162      LCASC  '&'
85 097BB 962       ?A=C   B      Est-ce 1 '&' ?
86 097BE 72        GOYES  REP50    Oui
87 097C0 148       REP22  DAT0=A  B      Stocke ce caractÅre
88 097C3 161       D0=D0+ 2
89 097C6 846       ST=0    Close   Si l'on est arrivÅ ici,
90 *                *                c'est que 2 '\\' ne se
91 *                *                suivent pas
92 097C9 53D       GONC   REP20  (B.E.T.)
93
94
95 097CC 877       REP30  ?ST=1  BackS   DÅjÈ 1 '\\' de trouvÅ?
96 097CF B0        GOYES  REP40    Oui
97 097D1 857       ST=1   BackS   C'est le 1er
98 097D4 856       ST=1   Close   Peut-Åtre y-en-a-t-il 2
99 097D7 55C       REP35  GONC   REP20  (B.E.T.) cÅte-È-cÅte
100
101 097DA 847       REP40  ST=0    BackS   Alors on sort du mode '\\'
102 097DD 876       ?ST=1  Close   A-t-on trouvÅ '\\\'?
103 097E0 0E        GOYES  REP22    Oui; on en garde 1 seul
104 097E2 54F       GONC   REP35  (B.E.T.) Non
105
106 097E5 867       REP50  ?ST=0  BackS   Hors mode '\\'?
107 097E8 8D        GOYES  REP22    Oui; '&' n'a aucune action
108 097EA 874       ?ST=1  Ins     '&' dÅjÈ trouvÅ?
109 097ED 3D        GOYES  REP22    Oui; alors rien È faire
110 097EF 847       ST=0   BackS   Sort du mode '\\'
111 097F2 854       ST=1   Ins     Indique '\\&' trouvÅ
112 097F5 132       AD0EX
113 097F8 679F     GOTO   REP15  On a trouvÅ la fin de R1$
114

```

```
115 097FC 132      REP100 AD0EX          On a trouv  la fin de R2$
116 097FF 1B00000      D0=(5) =AVMEMS
117 09806 140      DAT0=A A
118 09809 874      ?ST=1 Ins
119 0980C C0        GOYES REP102
120 0980E 1B00000      D0=(5) (=FUNCR0)+25 Si POS(R$,'\&')=0 alors
121 09815 140      DAT0=A A          R1$=R$ - R2=''
122 09818 8F00000      REP102 GOSBVL =REVPOP Analyse O$
123 0981F D8        B=A A
124 09821 137      CD1EX
125 09824 CA        A=A+C A
126 09826 100      R0=A
127 09829 142      A=DAT0 A
128 0982C DE        ACEX A
129 0982E 8F00000      GOSBVL =MOVE*M D place O$ derri re R2$
130 09835 109      R1=C
131 09838 1B00000      D0=(5) =AVMEMS Actualise AVMEMS
132 0983F C9        C=C+B A
133 09841 144      DAT0=C A
134 09844 D9        C=B A
135 09846 10B      R3=C
136
137 09849 1B00000      D0=(5) =FUNCR0 D but de O$ ->
138 09850 119      C=R1
139 09853 144      DAT0=C A
140 09856 135      D1=C          Analyse O$ - On ne peut
141 09859 113      A=R3          avoir LEN(O$)=0 ou O$='\ '
142 0985C C2        C=C+A A      Cf IDS 1
143 0985E 10A      R2=C
144 09861 134      D0=C
145 09864 8A8      ?A=0 A
146 09867 22      GOYES REP102
147 09869 181      D0=D0- 2
148 0986C 14A      A=DAT0 B
149 0986F 31C5      LC(2) BackS1 LCASC '\ '
150 09873 966      ?A#C B
151 09876 64      GOYES REP120
152 09878 181      D0=D0- 2
153 0987B 133      AD1EX
154 0987E 131      D1=A
155 09881 136      CD0EX
156 09884 8B2      ?A<C A
157 09887 C1        GOYES REP110
158 09889 110      REP102 A=R0
159 0988C 131      D1=A
160 0988F 1B00000      D0=(5) =FUNCD1
161 09896 142      A=DAT0 A
162 09899 130      D0=A
163 0989C 8D00000      GOVLNG =EXPR
164
165 098A3 134      REP110 D0=C
166 098A6 14A      A=DAT0 B
167 098A9 31C5      LC(2) BackS1 LCASC '\ '
168 098AD 962      ?A=C B
169 098B0 C0        GOYES REP120
170 098B2 11A      C=R2
171 098B5 CE        C=C-1 A
```

```
172 098B7 CE          C=C-1  A
173 098B9 10A        R2=C
174
175 098BC 1B00000    REP120 D0=(5) =AVMEMS  Actualise si O$=01$&'\'
176 098C3 112        A=R2
177 098C6 140        DAT0=A  A
178 098C9 D2         C=0    A          Calcule (AVMEMS)+21 pr SCAN
179 098CB 3151       LC(2)  21
180 098CF CA         A=A+C  A
181 098D1 120        AR0EX
182 098D4 131        D1=A
183 098D7 8F00000    GOSBVL =REVPOP  Etudie enfin S$
184 098DE 1B00000    D0=(5) (=FUNCR0)+5 DÅbut de S$
185 098E5 137        CD1EX
186 098E8 135        D1=C
187 098EB 144        DAT0=C  A
188 098EE 16E        D0=D0+ 15      Fin de S$
189 098F1 CA         A=A+C  A
190 098F3 140        DAT0=A  A
191 098F6 184        D0=D0- 5
192 098F9 142        A=DAT0  A
193 098FC 84B        ST=0  Anchor
194 098FF 16E        D0=D0+ 15      DÅbut de la recherche
195 09902 C2         C=C+A  A
196 09904 4A0        GOC  REPIVA  (Ne devrait pas arriver)
197 09907 C2         C=C+A  A
198 09909 144        DAT0=C  A
199 0990C 590        GONC  REP500
200 0990F 8D00000    REPIVA GOVLNG =ARGERR
201
202          *
203          *
204          * Boucle principale
205          * Remplace O$ ds S$ par R$=R1$&R2$
206          *
207          *
208
209
210 09916 1B00000    REP500 D0=(5) =FUNCR0
211 0991D 142        A=DAT0  A
212 09920 D8         B=A    A          B(A)= dÅbut du motif (O$)
213 09922 1900       D0=(2) (=FUNCR0)+20
214 09926 142        A=DAT0  A
215 09929 103        R3=A    A          R3= fin de la cible (S$)
216 0992C 169        D0=D0+ 10
217 0992F 146        C=DAT0  A
218 09932 D7         D=C    A          D(A)= dÅbut de la cible
219 09934 8BA        ?C>=A  A          Remplacement terminÅ?
220 09937 52         GOYES  REP540  Oui
221 09939 87B        ?ST=1  Anchor  Un remplacement en
222 0993C 02         GOYES  REP540  debut de ligne ne se
223          *          *          fait qu'une fois
224 0993E 1900       D0=(2) (=FUNCR0)+5
225 09942 142        A=DAT0  A
226 09945 131        D1=A    A          D1 @ mÅmoire libre
227 09948 1B00000    D0=(5) =AVMEMS
228 0994F 142        A=DAT0  A
```

```
229 09952 102          R2=A          R2 = fin du motif
230 09955 7021         GOSUB SCAN    Recherche du motif
231 09959 564          GONC REP550   Trouv 
232 0995C 8F00000 REP540 GOSBVL =OBCOLL
233 09963 D7           D=C A          D(A)=AVMEMS
234 09965 1B00000     D0=(5) (=FUNCR0)+20
235 0996C 146          C=DAT0 A
236 0996F 109          R1=C          R1 @ fin de S$
237 09972 18E          D0=D0- 15
238 09975 146          C=DAT0 A
239 09978 135          D1=C          D1 @ d but de S$
240 0997B 850          ST=1 0        Revient ici apr s ADHEAD
241 0997E 8F00000     GOSBVL =ADHEAD Ajoute un en-t te   S$
242 09985 8F00000     GOSBVL =REV$  L'inverse
243 0998C 1B00000     D0=(5) =FUNCD1
244 09993 142          A=DAT0 A
245 09996 130          D0=A          Remets D0 en place
246 09999 8D00000     GOVLNG =EXPR  Retourne S$
247
248 099A0 874          REP550 ?ST=1  Ins    Si l'on ins re, il ne faut
249 099A3 E2           GOYES REP600   pas d truire le motif
250 099A5 DB           C=D A
251 099A7 D5           B=C A
252 099A9 1B00000     D0=(5) (=FUNCR0)+5
253 099B0 142          A=DAT0 A
254 099B3 E8           B=B-A A
255 099B5 119          C=R1
256 099B8 EB           C=C-D A
257 099BA CA           A=A+C A
258 099BC 140          DAT0=A A      Actualise le pt actif
259 099BF DB           C=D A
260 099C1 111          A=R1
261 099C4 DE           ACEX A
262 099C6 D7           D=C A
263 099C8 DE           ACEX A
264 099CA 8F00000     GOSBVL =MOVEDM Ecrase le motif
265
266 099D1 1B00000 REP600 D0=(5) =OUTBS
267 099D8 142          A=DAT0 A
268 099DB 1B00000     D0=(5) (=FUNCR0)+25
269 099E2 146          C=DAT0 A
270 099E5 7720         GOSUB RPEDIT  Ins re R1$
271 099E9 119          C=R1
272 099EC D7           D=C A
273 099EE 1B00000     D0=(5) (=FUNCR0)+30
274 099F5 144          DAT0=C A
275 099F8 184          D0=D0- 5
276 099FB 142          A=DAT0 A
277 099FE 1B00000     D0=(5) =FUNCR0
278 09A05 146          C=DAT0 A
279 09A08 7400         GOSUB RPEDIT  Ins re R2$
280 09A0C 690F         GOTO REP500   Boucle
281
282 *
283 * Ins re une chaine ds une autre
284 * A(A)=D but du texte   ins rer
285 * C(A)=Fin du texte   ins rer
```



```
286          * D(A)=Position de l'insertion
287          *
288
289 09A10 102      RPEDIT R2=A          Sauvegarde A et C
290 09A13 10B          R3=C
291 09A16 E2          C=C-A  A          Longueur de l'insertion
292 09A18 D5      Memckl B=C  A          Verifie qu'il y a assez
293 09A1A D9          C=B  A          de place
294 09A1C DA          A=C  A
295 09A1E D2          C=0  A
296 09A20 3100        LC(2) =LEEWAY
297 09A24 C2          C=A+C  A
298 09A26 461          GOC  Memerr
299 09A29 1F00000     D1=(5) (=FUNCR0)+5
300 09A30 143          A=DAT1 A
301 09A33 8F00000     GOSBVL =CHKmem
302 09A3A 560          GONC  RPE10
303 09A3D 6F41      Memerr GOTO  memerr  Il n'y a pas de place
304
305 09A41 D9          RPE10 C=B  A          RÅcupÉre la longueur
306 09A43 1B00000     D0=(5) (=FUNCR0)+5
307 09A4A 142          A=DAT0 A          B(A) @ dÅbut de la chaÑne
308 09A4D D8          B=A  A          oÈ a lieu l'insertion
309 09A4F EA          A=A-C  A          Actualise sa nvelle pst
310 09A51 140          DAT0=A A
311 09A54 D9          C=B  A
312 09A56 EB          C=C-D  A
313 09A58 FA          C=-C  A
314 09A5A DD          BCEX  A
315 09A5C 8F00000     GOSBVL =MOVEUM  Fait de la place
316 09A63 132          AD0EX
317 09A66 137          CD1EX
318 09A69 D8          B=A  A
319 09A6B E1          B=B-C  A
320 09A6D 11B          C=R3
321 09A70 8F00000     GOSBVL =MOVEDM  Puis insÉre la chaÑne
322 09A77 01          RTN
323
324          *
325          *
326          * Routine HP
327          * Ecrire (sans doute) par SW ou FH
328          * Consulter les IDS Vol. 1, 17-44
329          *
330          *
331
332 09A79 84B          SCAN  ST=0  Anchor
333 09A7C 840          ST=0  RegExp
334 09A7F 7571         GOSUB  PATCHR
335 09A83 31C5         LC(2)  BackSl          LCASC '\ '
336 09A87 966          ?C#A  B
337 09A8A B3          GOYES  L190j
338 09A8C 161      L125.1 D0=D0+ 2
339 09A8F 14A          A=DAT0 B
340 09A92 3142        LCASC  '$ '
341 09A96 966          ?C#A  B
342 09A99 13          GOYES  L140
```

```
343 09A9B 161          D0=D0+ 2
344 09A9E 132          AD0EX
345 09AA1 11A          C=R2
346 09AA4 966          ?C#A  B
347 09AA7 32           GOYES  L140
348 09AA9 11B          C=R3
349 09AAC 86B          ?ST=0  Anchor
350 09AAF 70           GOYES  L125.2
351 09AB1 8A7          ?D#C  A
352 09AB4 00           RTNYES
353 09AB6 D7           L125.2 D=C  A
354 09AB8 109          MATCH+ R1=C
355 09ABB 03           RTNCC
356
357 09ABD 7331          L130   GOSUB  PATCH+
358 09AC1 31C5          LC(2)  BackS1      LCASC '\ '
359 09AC5 966          L190j  ?C#A  B
360 09AC8 44           GOYES  L190
361 09ACA D4           L140   A=B  A
362 09ACC 130          D0=A
363 09ACF 161          D0=D0+ 2
364 09AD2 14A          A=DAT0 B
365 09AD5 31C5          LC(2)  BackS1      LCASC '\ '
366 09AD9 962          ?C=A  B
367 09ADC F4           GOYES  L240j
368 09ADE E5           L150   B=B+1 A
369 09AE0 E5           B=B+1 A
370 09AE2 7C11          L160   GOSUB  RETOGL
371 09AE6 860          L170   ?ST=0  RegExp
372 09AE9 74           GOYES  L190.2
373 09AEB 87B          ?ST=1  Anchor
374 09AEE E1           GOYES  L190
375 09AF0 7401          GOSUB  PATCHR
376 09AF4 31E5          LCASC  '^'
377 09AF8 966          ?C#A  B
378 09AFB A1           GOYES  L190.1
379 09AFD 875          ?ST=1  First
380 09B00 00           RTNYES
381 09B02 85B          ST=1   Anchor
382 09B05 E5           B=B+1 A
383 09B07 E5           B=B+1 A
384
385
386 09B09 528          ** change:
387
388
389 09B0C 860          L190   ?ST=0  RegExp
390 09B0F 12           GOYES  L190.2
391 09B11 73E0          GOSUB  PATCHR
392 09B15 31E2          L190.1 LCASC  '.'
393 09B19 962          ?C=A  B
394 09B1C 15           GOYES  L240
395 09B1E 3142          LCASC  '$'
396 09B22 962          ?C=A  B
397 09B25 84           GOYES  L240
398 09B27 3104          LCASC  '@'
399 09B2B 962          L240j  ?C=A  B
```

```

400 09B2E F3          GOYES  L240
401 09B30 74C0       L190.2 GOSUB  PATCHR
402 09B34 31C5          LC(2)  BackS1      LCASC  '\ '
403 09B38 962         ?C=A   B
404 09B3B 23          GOYES  L240
405 09B3D 11A        L200   C=R2
406 09B40 8B5         ?B<C  A
407 09B43 80          GOYES  L210
408 09B45 DB          C=D    A
409 09B47 607F        GOTO   MATCH+
410
411 09B4B 11B        L210   C=R3
412 09B4E 8BB         ?D>=C A
413 09B51 00          RTNYES
414 09B53 DB          C=D    A
415 09B55 134         D0=C
416 09B58 14E         C=DAT0 B
417 09B5B 962         ?C=A   B
418 09B5E F0          GOYES  L240
419 09B60 87B         ?ST=1  Anchor
420 09B63 00          RTNYES
421 09B65 E7          D=D+1  A
422 09B67 E7          D=D+1  A
423 09B69 61EF        GOTO   L210
424
425 09B6D 857        L240   ST=1   TopLvl
426 09B70 7020        GOSUB  SCANSB
427 09B74 461         GOC    RTNCC
428 09B77 87B         ?ST=1  Anchor
429 09B7A 00          RTNYES
430 09B7C E7          D=D+1  A
431 09B7E E7          D=D+1  A
432 09B80 11B        C=R3
433 09B83 8BB         ?D>=C A
434 09B86 00          RTNYES
435 09B88 538        GONC   L190      (B.E.T.)
436 09B8B 03          RTNCC  RTNCC
437
438
439 09B8D 8D00000    memerr GOVLNG =MEMERR
440 09B94 137        SCANSB CD1EX
441 09B97 135         D1=C
442 09B9A 110        A=R0
443 09B9D 8B6        ?C<A  A
444 09BA0 DE          GOYES  memerr
445 09BA2 1C4        D1=D1- 5
446 09BA5 119        C=R1
447 09BA8 145        DAT1=C A
448 09BAB 1C4        D1=D1- 5
449 09BAE D9         C=B    A
450 09BB0 145        DAT1=C A
451 09BB3 1C4        D1=D1- 5
452 09BB6 DB         C=D    A
453 09BB8 145        DAT1=C A
454 09BBB 877        ?ST=1  TopLvl
455 09BBE B0         GOYES  SCNSB1
456 09BC0 119        C=R1

```

```
457 09BC3 D7          D=C      A
458 09BC5 CF          D=D-1   A
459 09BC7 CF          D=D-1   A
460 09BC9 847        SCNSB1 ST=0   TopLvl
461 09BCC 1C4          D1=D1-  5
462 09BCF 07          C=RSTK
463 09BD1 145          DAT1=C  A
464 09BD4 1C0          D1=D1-  1
465 09BD7 09          C=ST
466 09BD9 1550        DAT1=C  P
467 09BDD 11B        L340    C=R3
468 09BE0 109          R1=C
469 09BE3             L350
470 09BE3 11A        L360    C=R2
471 09BE6 8B5          ?B<C   A
472 09BE9 82          GOYES  L370
473 09BEB DB          C=D     A
474 09BED 109          R1=C
475 09BF0 63B0        GOTO   L640
476
477 09BF4 E5          PATCH+ B=B+1  A
478 09BF6 E5          B=B+1  A
479 09BF8 D4          PATCHR A=B    A
480 09BFA 130          D0=A
481 09BFD 14A          A=DAT0 B
482 09C00 03          RTNCC
483
484 09C02 870        RETOGL ?ST=1  RegExp
485 09C05 70          GOYES  RETOG0
486 09C07 850        ST=1   RegExp
487 09C0A 01          RTN
488 09C0C 840        RETOG0 ST=0   RegExp
489 09C0F 03          RTNCC
490
491
492 09C11 73EF        L370    GOSUB  PATCHR
493 09C15 31C5        LC(2)  BackSl
494 09C19 966          ?C#A   B
495 09C1C 32          GOYES  L405
496 09C1E 72DF        L380    GOSUB  PATCH+
497 09C22 11A        L390    C=R2
498 09C25 8B5          ?B<C   A
499 09C28 A0          GOYES  L400
500 09C2A CF          D=D-1  A
501 09C2C CF          D=D-1  A
502 09C2E 6C60        GOTO   EXIT
503
504 09C32 31C5        L400    LC(2)  BackSl
505 09C36 962          ?C=A   B
506 09C39 60          GOYES  L405
507 09C3B 73CF        GOSUB  RETOGL
508 09C3F 860        L405    ?ST=0  RegExp
509 09C42 93          GOYES  L440
510 09C44 31E2        L410    LCASC  '.'
511 09C48 966          ?C#A   B
512 09C4B E0          GOYES  L420
513 09C4D 119        L475    C=R1
```

```
514 09C50 8B7          ?D<C  A
515 09C53 C3           GOYES  L450
516 09C55 6230        GOTO   L500
517
518 09C59 3104        L420   LCASC  '@'
519 09C5D 966         ?C#A  B
520 09C60 60          GOYES  L430
521 09C62 68A0        GOTO   L550.1
522 09C66 3142        L430   LCASC  '$'
523 09C6A 966         ?C#A  B
524 09C6D E0          GOYES  L440
525 09C6F 119         C=R1
526 09C72 8B7          ?D<C  A
527 09C75 31          GOYES  L500
528 09C77 6320        GOTO   EXIT
529
530 09C7B DB          L440   C=D    A
531 09C7D 134         D0=C
532 09C80 14E         C=DAT0 B
533 09C83 962         ?C=A  B
534 09C86 90          GOYES  L450
535 09C88 849        L500   ST=0  Match
536 09C8B 6EF0        GOTO   SCNRTN
537
538 09C8F E5          L450   B=B+1  A
539 09C91 E5          B=B+1  A
540 09C93 11A         C=R2
541 09C96 8B5         ?B<C  A
542 09C99 21          GOYES  L460
543 09C9B DB          EXIT   C=D    A
544 09C9D E6          C=C+1  A
545 09C9F E6          C=C+1  A
546 09CA1 109         R1=C
547 09CA4 859        L640   ST=1  Match
548 09CA7 62E0        GOTO   SCNRTN
549
550 09CAB E7          L460   D=D+1  A
551 09CAD E7          D=D+1  A
552 09CAF 119         C=R1
553 09CB2 8BB         ?D>=C  A
554 09CB5 60          GOYES  L480
555 09CB7 6B2F        GOTO   L350
556
557 09CBB 793F        L480   GOSUB  PATCHR
558 09CBF E5          B=B+1  A
559 09CC1 E5          B=B+1  A
560 09CC3 860         ?ST=0  RegExp
561 09CC6 A2          GOYES  L490NR
562 09CC8 3104        LCASC  '@'
563 09CCC 966         ?C#A  B
564 09CCF D0          GOYES  L480.2
565 09CD1 11A         C=R2
566 09CD4 8B9         ?B>=C  A
567 09CD7 DC          GOYES  L640
568 09CD9 51E         GONC   L480      (B.E.T.)
569
570 09CDC 3142        L480.2 LCASC  '$'
```

```

571 09CE0 966          ?C#A  B
572 09CE3 5A          GOYES  L500
573 09CE5 11A         C=R2
574 09CE8 8B9         ?B>=C  A
575 09CEB 9B          GOYES  L640
576 09CED 5A9         GONC   L500      (B.E.T.)
577
578 09CF0 31C5        L490NR LC(2)  BackS1  LCASC  '\ '
579 09CF4 966          ?C#A  B
580 09CF7 19          GOYES  L500
581 09CF9 850         ST=1   RegExp
582 09CFC 5EB         GONC   L480      (B.E.T.)
583
584 09CFF E7          L520   D=D+1  A
585 09D01 E7          D=D+1  A
586 09D03 119         C=R1
587 09D06 8BB         ?D>=C  A
588 09D09 27          GOYES  L500j
589 09D0B E5          L550.1 B=B+1  A
590 09D0D E5          B=B+1  A
591 09D0F 11A         L530   C=R2
592 09D12 8B9         ?B>=C  A
593 09D15 F8          GOYES  L640
594 09D17 860         L540   ?ST=0  RegExp
595 09D1A 43          GOYES  L580
596 09D1C 78DE        L550   GOSUB  PATCHR
597 09D20 3104        LCASC  '@ '
598 09D24 966          ?C#A  B
599 09D27 50          GOYES  L560
600 09D29 51E         GONC   L550.1  (B.E.T.)
601
602 09D2C 31E2        L560   LCASC  '.'
603 09D30 962          ?C=A  B
604 09D33 CC          GOYES  L520
605 09D35 3142        L570   LCASC  '$ '
606 09D39 966          ?C#A  B
607 09D3C 61          GOYES  L580.1
608 09D3E 11A         C=R2
609 09D41 CE          C=C-1  A
610 09D43 CE          C=C-1  A
611 09D45 8B5         ?B<C  A
612 09D48 A0          GOYES  L580.1
613 09D4A 695F        GOTO   L640
614
615 09D4E 76AE        L580   GOSUB  PATCHR
616 09D52 31C5        L580.1 LC(2)  BackS1  LCASC  '\ '
617 09D56 962          ?C=A  B
618 09D59 62          GOYES  L610
619 09D5B 119         L590   C=R1
620 09D5E 134         D0=C
621 09D61 181         D0=D0- 2
622 09D64 14E         C=DAT0 B
623 09D67 962          ?C=A  B
624 09D6A 51          GOYES  L610
625 09D6C 119         L600   C=R1
626 09D6F CE          C=C-1  A
627 09D71 CE          C=C-1  A

```

```
628 09D73 109          R1=C
629 09D76 8B7          ?D<C  A
630 09D79 5D           GOYES  L580
631 09D7B 6C0F        L500j  GOTO  L500
632
633 09D7F             L610
634 09D7F 711E        L620  GOSUB  SCANSB
635 09D83 58E         GONC  L600
636 09D86 6D1F        GOTO  L640
637
638 09D8A 09          SCNRTN C=ST
639 09D8C 1570        C=DAT1 P
640 09D90 0A          ST=C
641 09D92 170         D1=D1+ 1
642 09D95 147         C=DAT1 A
643 09D98 06          RSTK=C
644 09D9A 174         D1=D1+ 5
645 09D9D 147         C=DAT1 A
646 09DA0 D7          D=C  A
647 09DA2 174         D1=D1+ 5
648 09DA5 147         C=DAT1 A
649 09DA8 D5          B=C  A
650 09DAA 174         D1=D1+ 5
651 09DAD 147         C=DAT1 A
652 09DB0 174         D1=D1+ 5
653 09DB3 879         ?ST=1 Match
654 09DB6 00          RTNYES
655 09DB8 109         R1=C
656 09DBB 03          RTNCC
657
658 09DBD             END
```

ADHEAD	Ext		-	241						
ARGERR	Ext		-	200						
AVMEMS	Ext		-	116	131	175	227			
Anchor	Abs	11	#0000000B	-	24	193	221	332	349	373 381
					419	428				
BackS	Abs	7	#00000007	-	28	95	97	101	106	110
BackS1	Abs	92	#0000005C	-	29	81	149	167	335	358 365
					402	493	504	578	616	
CHKmem	Ext		-	301						
Close	Abs	6	#00000006	-	26	89	98	102		
D0=AVS	Ext		-	71						
EXIT	Abs	40091	#00009C9B	-	543	502	528			
EXPR	Ext		-	163	246					
FLTDH	Ext		-	54						
FUNCD1	Ext		-	47	160	243				
FUNCR0	Ext		-	60	72	120	137	184	210	213
					224	234	252	268	273	277 299
					306					
First	Abs	5	#00000005	-	25	64	379			
Ins	Abs	4	#00000004	-	27	108	111	118	248	
L125.1	Abs	39564	#00009A8C	-	338	386				
L125.2	Abs	39606	#00009AB6	-	353	350				
L130	Abs	39613	#00009ABD	-	357					
L140	Abs	39626	#00009ACA	-	361	342	347			
L150	Abs	39646	#00009ADE	-	368					
L160	Abs	39650	#00009AE2	-	370					
L170	Abs	39654	#00009AE6	-	371					
L190	Abs	39692	#00009B0C	-	389	360	374	435		
L190.1	Abs	39701	#00009B15	-	392	378				
L190.2	Abs	39728	#00009B30	-	401	372	390			
L190j	Abs	39621	#00009AC5	-	359	337				
L200	Abs	39741	#00009B3D	-	405					
L210	Abs	39755	#00009B4B	-	411	407	423			
L240	Abs	39789	#00009B6D	-	425	394	397	400	404	418
L240j	Abs	39723	#00009B2B	-	399	367				
L340	Abs	39901	#00009BDD	-	467					
L350	Abs	39907	#00009BE3	-	469	555				
L360	Abs	39907	#00009BE3	-	470					
L370	Abs	39953	#00009C11	-	492	472				
L380	Abs	39966	#00009C1E	-	496					
L390	Abs	39970	#00009C22	-	497					
L400	Abs	39986	#00009C32	-	504	499				
L405	Abs	39999	#00009C3F	-	508	495	506			
L410	Abs	40004	#00009C44	-	510					
L420	Abs	40025	#00009C59	-	518	512				
L430	Abs	40038	#00009C66	-	522	520				
L440	Abs	40059	#00009C7B	-	530	509	524			
L450	Abs	40079	#00009C8F	-	538	515	534			
L460	Abs	40107	#00009CAB	-	550	542				
L475	Abs	40013	#00009C4D	-	513					
L480	Abs	40123	#00009CBB	-	557	554	568	582		
L480.2	Abs	40156	#00009CDC	-	570	564				
L490NR	Abs	40176	#00009CF0	-	578	561				
L500	Abs	40072	#00009C88	-	535	516	527	572	576	580 631
L500j	Abs	40315	#00009D7B	-	631	588				
L520	Abs	40191	#00009CFF	-	584	604				
L530	Abs	40207	#00009D0F	-	591					

SCAN	Abs	39545	#00009A79 -	332	230		
SCANSB	Abs	39828	#00009B94 -	440	426	634	
SCNRTN	Abs	40330	#00009D8A -	638	536	548	
SCNSB1	Abs	39881	#00009BC9 -	460	455		
TopLvl	Abs	7	#00000007 -	22	425	454	460
memerr	Abs	39821	#00009B8D -	439	303	444	

Input Parameters

Source file name is rplc.a

Listing file name is rplc.l

Object file name is rplc.o

Flags set on command line
None

Errors

None

```

1          TITLE JPC LEX: MODULE SCAN
2
3          * *****
4          * Scan module
5          *
6          * Keyword(s):
7          *   TOKEN, ENTRY$
8          * Author(s):
9          *   Jean-Jacques Moreau
10         * History:
11         *   Published in JPC #31
12         *
13         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14         * *****
15
16 09DBD          ABS      #9DBD
17
18 09DBD 80DF     Find    P=C    15      Nb d'arguments de la
19         *                               fonction dans P
20 09DC1 D0      A=0     A      Mets a zero le paramÁtre n
21         *                               (l'occurrence d'un mot-clÁ)
22 09DC3 891     ?P=    1      N'y a-t-il qu'un paramÁtre?
23 09DC6 41      GOYES   Find1
24 09DC8 8F00000 GOSBVL =RNDAHX Non; second paramÁtre ->
25         *                               A(A);
26 09DCF 544     GONC   Argerr  2nd parm <0 IF Argerr
27 09DD2 17F     D1=D1+ 16     D1 @ en-tÁte de la chaÑne
28         *                               È tokenizer
29 09DD5 CC      A=A-1  A      A(A) -> n (l'occurrence)
30 09DD7 4C3     GOC    Argerr  2nd parm 0= IF Argerr
31 09DDA 101     Find1   R1=A    Sauvegarde n, qui servira
32         *                               de compteur, ds R1
33 09DDD 20      P=      0
34 09DDF 8F00000 GOSBVL =REVPOP Mets la chaÑne È tokenizÁ
35         *                               ds le bon sens
36 09DE6 137     CD1EX
37 09DE9 135     D1=C      D1@ 1er char.
38 09DEC C2      C=C+A   A
39 09DEE D7      D=C     A      D(A)=(FORSTK)
40 09DF0 8F00000 GOSBVL =CSLW5  Sauvegarde D0&D1 ds R3
41 09DF7 136     CD0EX
42 09DFA 134     D0=C
43 09DFD 10B     R3=C
44 09E00 850     ST=1   0      On arme ST0 car on dÁsire
45         *                               connaÑtre la type du mot-clÁ
46 09E03 8F00000 GOSBVL =NTOKEN Mets A(?-0)=TOKEN,
47         *                               B(A)=adr d'ÁxÁcution (si elle existe)
48 09E0A 87B     Findx  ?ST=1 11     Voulait-on tokenizer 1
49         *                               variable?
50 09E0D 70      GOYES   Argerr
51 09E0F 8AC     ?A#0   A      Ou bien GO?
52 09E12 90      GOYES   Find2   Ni l'un ni l'autre
53 09E14 8D00000 Argerr  GOVLNG =ARGERR (Remarquez que =IVAERR
54         *                               s'Ácrit: GOVLNG =ARGERR)
55
56 09E1B 121     Find2  AR1EX   n (notre compteur) -> A(A);
57 09E1E CC      A=A-1  A      n-1 -> A

```

```

58 09E20 121          AR1EX          Sauvegarde le nv compteur
59 09E23 400          RTNC           Retourne si l'on a fait
60                    *              n-boucles
61 09E26 8F00000      GOSBVL =RESPTR Sinon restaure divers
62                    *              pointeurs pr que l'on puisse de nv
63 09E2D 850          ST=1  0        Utiliser =NTOKEN
64 09E30 DA           A=C    A        Information nAccessaire È
65                    *              =NTOKEN -> C(A);
66 09E32 8F00000      GOSBVL =RESCAN A peu prEs Åquivalent È
67                    *              GOSBVL =NTOKEN
68 09E39 60DF         GOTO   Findx   Boucle
69
70
71
72
73                    *
74                    *  TOKEN("KEYWORD"[,n]):
75                    *    Renvoie le TOKEN de la n-iÅme occurrence de
76                    *    KEYWORD sous la forme dÅcimale:
77                    *      1000*ID + TK
78                    *
79
80                    * *****
81                    *  TOKEN entry
82                    * *****
83 09E3D 8412          NIBHEX 8412
84 09E41 787F         =TOKENe GOSUB Find   Tokenize
85 09E45 844          ST=0   4
86
87                    *
88                    *
89                    *  Tokens: Avant et aprEs transformation
90                    *  FN      :   Tk , 0Tk00      (Function)
91                    *  STMT   :   Tk , 0Tk00      (Statement)
92                    *  XFN    :  TkIdB3 , 0TkId    (XFunction)
93                    *  FFN    :  TkIdB4 , 0TkId    (Funny Function)
94                    *  XSTMT  :  TkIdEF , 0TkId    (XStatement)
95                    *  WORD   :  TkIdEF , 0TkId    (Word)
96                    *
97                    *
98
99 09E48 3100          LC(2)  =tXWORD
100 09E4C 962          ?A=C   B
101 09E4F 02           GOYES  Schi++   Go if XWORD/XSTMT
102 09E51 3100          LC(2)  =tXFN
103 09E55 962          ?A=C   B
104 09E58 71           GOYES  Schi++   Go if XFN
105 09E5A B66          C=C+1  B
106 09E5D 962          ?A=C   B
107 09E60 C0           GOYES  Schif+   Go if FFN
108 09E62 F0           ASL    A        Ici si FN ou STMT
109 09E64 AD0          A=0    M        Effectue la transformation
110                    *              dÅcrite + haut
111 09E67 F0           ASL    A
112 09E69 5A0          GONC   Schift   (B.E.T.)
113 09E6C 854          Schif+ ST=1   4    Indique que l'on tokenize
114                    *              une FFN

```

```

115 09E6F BF4      Schi++ ASR   W      Effectue la transformation
116                *                d'acrite + haut
117 09E72 F4              ASR   A
118 09E74 100      Schift R0=A      Sauvegarde le TOKEN
119 09E77 AD0              A=0   M
120 09E7A AA0              A=0   XS      A(A)=Id
121 09E7D D2              C=0   A
122 09E7F 328E3          LC(3) 1000
123 09E84 8F00000      GOSBVL =A-MULT  A(A)=Id*1000
124 09E8B 118          C=R0   C(A)=0TkId
125 09E8E F6              CSR   A
126 09E90 AD2          C=0   M
127 09E93 F6              CSR   A      C(A)=Tk
128 09E95 CA          A=A+C  A      A(A)=Id*1000 + Tk
129 09E97 8F00000      Fnrtfn GOSBVL =HDFLT  A(A)=Resultat en decimal
130 09E9E AF8          B=A    W      Sauvegardons-le ds B(W)
131 09EA1 8F00000      GOSBVL =D1C=R3  Mets: D1=Stack pointer,
132                *                C(A)=PC
133 09EA8 AF4          A=B    W      Resultat -> C(W);
134 09EAB AFE          ACEX   W      PC -> A(A);
135 09EAE 864          ?ST=0 4      A-t-on tokenizé une FFN?
136 09EB1 80          GOYES  Pos    Non
137 09EB3 29          P=     9      Oui; indiquons-le en
138                *                renvoyant 1 nb negatif
139 09EB5 80FF          CPEX   15
140 09EB9 8D00000      Pos    GOVLNG =FNRTN2
141
142
143
144                *
145                * ENTRY$( "KEYWORD" [,n] ):
146                *   Renvoie l'adr à laquelle commence la routine
147                *   d'afinissant la n-ième
148                *   occurrence du mot-clé KEYWORD.
149                *
150
151                * *****
152                * ENTRY$ entry
153                * *****
154 09EC0 8412          NIBHEX 8412
155 09EC4 75FE          =ENTRYe GOSUB Find    Calcule l'adr d'attention
156                *                de KEYWORD
157 09EC8 8F00000      GOSBVL =D1C=R3  Stack pointer -> D1;
158 09ECF 134          D0=C      PC -> D0;
159 09ED2 D4          A=B    A      Mets l'adr en question ds
160                *                A(A)
161
162                *
163                *
164                *   Ci-suit 1 routine HP; elle convertit 1 nb en
165                *   hexa ds A(A) en 1 chaîne
166                *   alphanumérique qu'elle place sur la pile.
167                *
168                *
169
170 09ED4 AF2          C=0   W
171 09ED7 24          P=     4

```

```
172 09ED9 80FF          CPEX   15      Indicate 4+1 nibs to
173                    *          convert in ASCII
174 09EDD AF7           D=C   W        Save counter in D(W)
175 09EE0 8F00000      GOSBVL =HEXASC
176 09EE7 B47          D=D+1 S
177 09EEA A47          D=D+D S
178 09EED 813          DSLC
179 09EF0 DB           C=D   A        Compute output string
180                    *          length
181 09EF2 8F00000      GOSBVL =STRHDR Prepare header for string
182                    *          in B(W)
183 09EF9 80D0          P=C   0
184 09EFD 0D           P=P-1          String length-1 to P
185 09EFF AF4          A=B   W
186 09F02 1511         DAT1=A WP      Write string
187 09F06 1CF          D1=D1- 16     Move to header
188 09F09 20           P=   0
189 09F0B 8F00000      GOSBVL =REV$   Reverse string
190 09F12 8D00000      GOVLNG =EXPR
191
192 09F19              END
```

A-MULT	Ext		-	123			
ARGERR	Ext		-	53			
Argerr	Abs	40468	#00009E14	- 53	26	30	50
CSLW5	Ext		-	40			
D1C=R3	Ext		-	131	157		
=ENTRYe	Abs	40644	#00009EC4	- 155			
EXPR	Ext		-	190			
FNRTN2	Ext		-	140			
Find	Abs	40381	#00009DBD	- 18	84	155	
Find1	Abs	40410	#00009DDA	- 31	23		
Find2	Abs	40475	#00009E1B	- 56	52		
Findx	Abs	40458	#00009E0A	- 48	68		
Fnrtn	Abs	40599	#00009E97	- 129			
HDFLT	Ext		-	129			
HEXASC	Ext		-	175			
NTOKEN	Ext		-	46			
Pos	Abs	40633	#00009EB9	- 140	136		
RESCAN	Ext		-	66			
RESPTR	Ext		-	61			
REV\$	Ext		-	189			
REVPOP	Ext		-	34			
RNDAHX	Ext		-	24			
STRHDR	Ext		-	181			
Schi++	Abs	40559	#00009E6F	- 115	101	104	
Schif+	Abs	40556	#00009E6C	- 113	107		
Schift	Abs	40564	#00009E74	- 118	112		
=TOKENe	Abs	40513	#00009E41	- 84			
tXFN	Ext		-	102			
tXWORD	Ext		-	99			

Input Parameters

Source file name is scan.a

Listing file name is scan.l

Object file name is scan.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE STK
2
3          * *****
4          * Stack module
5          *
6          * Keyword(s):
7          *   STACK
8          * Author(s):
9          *   Henri KUDELSKI
10         * History:
11         *   Initially created by Michel Martinet, JPC #25
12         *   This new version published in JPC#60, dec.88-Jan.89
13         *
14         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
15         * *****
16
17 09F19          ABS      #9F19
18
19         * *****
20         * STACK entry
21         * *****
22 09F19 00000          REL(5) =STDd
23 09F1E 00000          REL(5) =FIXp
24 09F23 8F00000 =STACKe GOSBVL =EXPEXC
25 09F2A 8F00000          GOSBVL =RNDAHX
26 09F31 490           GOC   ARGOK
27 09F34 8D00000   argerr GOVLNG =ARGERR
28 09F3B 331001   ARGOK  LCHEX  1001
29 09F41 8F00000          GOSBVL =RANGE
30 09F48 4BE           GOC   argerr
31 09F4B CC          A=A-1  A
32 09F4D 1B00000          D0=(5) =MAXCMD
33 09F54 1580         DAT0=A  1
34 09F58 1B00000          D0=(5) =IOBFEN
35 09F5F 146         C=DAT0  A
36 09F62 134         D0=C
37 09F65 35000300          LCHEX  003000
38 09F6D 15C5        CMD1  DAT0=C  6
39 09F71 165         D0=D0+  6
40 09F74 CC          A=A-1  A
41 09F76 56F        GONC   CMD1
42 09F79 132        AD0EX
43 09F7C 1B00000          D0=(5) =RFNBFR
44 09F83 3150        LC(2)  5
45 09F87 140        CMD2  DAT0=A  A
46 09F8A 164        D0=D0+  5
47 09F8D A6E        C=C-1  B
48 09F90 56F        GONC   CMD2
49 09F93 8D00000          GOVLNG =NXTSTM
50
51 09F9A          END
```

ARGERR	Ext		-	27	
ARGOK	Abs	40763	#00009F3B	-	28 26
CMD1	Abs	40813	#00009F6D	-	38 41
CMD2	Abs	40839	#00009F87	-	45 48
EXPEXC	Ext		-	24	
FIXp	Ext		-	23	
IOBFEN	Ext		-	34	
MAXCMD	Ext		-	32	
NXTSTM	Ext		-	49	
RANGE	Ext		-	29	
RFNBFR	Ext		-	43	
RNDAHx	Ext		-	25	
=STACKe	Abs	40739	#00009F23	-	24
STDD	Ext		-	22	
argerr	Abs	40756	#00009F34	-	27 30

Input Parameters

Source file name is stk.a

Listing file name is stk.l

Object file name is stk.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE STRUC2
2
3          * *****
4          * Programming structure module
5          *
6          * Keyword(s):
7          *   SELECT, CASE, LEAVE, END, WHILE, REPEAT,
8          *   LOOP, UNTIL, IF, ELSE
9          * Author(s):
10         *   Pierre David, Janick Taillandier
11         * History:
12         *   Published in JPC #52
13         *
14         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
15         * *****
16
17 09F9A          ABS      #9F9A
18
19         tEND2   EQU      66
20         tWHILE  EQU      67
21         tREPEAT EQU      68
22         tUNTIL  EQU      69
23         tLEAVE  EQU      70
24
25         tLOOP   EQU      96
26         tSELECT EQU      97
27         tCASE   EQU      98
28         tIF2    EQU      99
29         tELSE2  EQU     100
30
31         id      EQU      #E1
32
33         STRUCT EQU      9          Type de GOSUB stack
34         *
35         * quartets de reconnaissance de :
36         *
37         qENDL   EQU      0          END LOOP
38         qENDS   EQU      1          END SELECT
39         qENDI   EQU      2          END IF
40
41         SAUVD0  EQU      #2F891     =STMTD0
42         CSETYP  EQU      #2F890     =S-R1-3 Type de l'objet (SELECT)
43
44         chEW    EQU      0          cherche END WHILE
45         chEL    EQU      1          ch. END LOOP
46         chU     EQU      2          ch. UNTIL
47         chEI    EQU      3          ch. END IF
48         chES    EQU      4          ch. END SELECT
49         chE|EI  EQU      5          ch. ELSE ou END IF
50         chC|ES  EQU      6          ch. CASE ou END SELECT
51         chEI-E  EQU      7          ch. END IF, saute ELSE
52         chES-C  EQU      8          ch. END SELECT, saute CASE
53         chL/EW  EQU      9          LEAVE cherche END WHILE
54         chL/EL  EQU     10          LEAVE cherche END LOOP
55         chL/U   EQU     11          LEAVE cherche UNTIL
56
57         * Predicats de test
```

```

58      Pred< EQU    1      %0001
59      Pred= EQU    2      %0010
60      Pred> EQU    4      %0100
61      Pred? EQU    8      %1000
62
63      *****
64      * SELECT
65      *
66      * But: executer l'ordre SELECT
67      * Historique:
68      *   87/03/01: conception & codage
69      *   87/08/29: correction SELECT 1 !
70      *   87/08/29: correction SELECT 1 / CASE 0 ! / END
71      * *****
72
73      *
74      * tokenisation des structures SELECT / END SELECT
75      *
76      * tXWORD id tSELECT <exp. alpha ou num.>
77      *   tXWORD id tCASE tRELOP specifier <exp>
78      *   tXWORD id tCASE <exp>
79      *   tXWORD id tCASE <exp> tTO <exp>
80      *   tXWORD id tCASE                               (CASE ELSE)
81      * tXWORD id tEND2 qENDS
82      *
83
84      * *****
85      * SELECT entry
86      * *****
87 09F9A 00000          REL(5) =STDD
88 09F9F 00000          REL(5) =SELECP
89 09FA4                =SELECe
90 09FA4 187           D0=D0- 6+2
91 09FA7 7035          GOSUB stoadr   SAUVD0 := ^ Stlen
92 09FAB 167           D0=D0+ 6+2
93 09FAE 8F00000       GOSBVL =EXPEX- Evaluer le SELECT
94
95      * A(W) := top 16 nib. of MS
96      *
97 09FB5 309           LCHEX  9      Legal BCD digit
98 09FB8 98A           ?A<=C  P
99 09FBB 13            GOYES  SL020   Real number
100 09FBD 31E0         LCHEX  0E     Complex number signature
101 09FC1 962           ?A=C    B
102 09FC4 82            GOYES  SL020   Complex number
103
104      * Nous sommes dans le cas "string" ou autre...
105      *
106 09FC6 8F00000       SL010  GOSBVL =POP1S  Erreur si "autre..."
107 09FCD 7C15          GOSUB  stOMS    Protéger la chaîne
108 09FD1 1B00000       D0=(5) =STMTR0
109 09FD8 137           CD1EX                C(A) := ^ dernier caract.
110 09FDB C2            C=C+A  A           C(A) := ^ premier caract.
111 09FDD 144           DAT0=C  A          S-R0-0 := ^ chaîne
112 09FE0 164           D0=D0+ 5
113 09FE3 140           DAT0=A  A          S-R0-1 := len en quartets
114 09FE6 AC2           C=0     S          Type := alpha

```

```

115 09FE9 5C0          GONC   SL030   B.E.T.
116                  *
117                  * Cas reel ou complexe
118                  *
119 09FEC 7DF4        SL020  GOSUB  stoMS
120 09FF0 AC2         C=0    S
121 09FF3 B46        C=C+1  S
122                  *
123                  * C(S) = type (0 = alpha / 1 = numerique)
124                  * AVMEME = pointeur de M.S.
125                  *
126 09FF6 1B098F2    SL030  D0=(5) CSETYP
127 09FFD 1544      DAT0=C S      CSETYP := type (alpha/num)
128
129                  *
130                  * Debut de la boucle de recherche de CASE ou
131                  * END SELECT
132                  *
133 0A001 77C4        SL100  GOSUB  recadr
134 0A005 7015      GOSUB  eol
135
136 0A009 26         P=     chC|ES
137 0A00B 7D15      GOSUB  cherche
138 0A00F 78C4      GOSUB  stoadr  SAUVD0 := adresse trouvee
139 0A013 165       D0=D0+ 6      D0 := ^ tCASE ou tEND2
140 0A016 3124      LC(2)  tEND2
141 0A01A 14A       A=DAT0 B
142 0A01D 962       ?A=C   B
143 0A020 F0        GOYES  sl900   END SELECT seulement
144
145 0A022 161       D0=D0+ 2      D0 := ^ apres tCASE
146
147 0A025 14A       A=DAT0 B
148 0A028 7D43      GOSUB  eolxck
149 0A02C 560       GONC   SL150   Il y a quelque chose.
150 0A02F 6D90     sl900  GOTO   SL900   CASE ELSE
151                  *
152                  * D0 = ^ debut de ce qu'il y a a analyser
153                  *
154 0A033           SL150
155                  *
156                  * Il y a quelque chose a analyser
157                  *
158 0A033 3100      LC(2)  =tRELOP
159 0A037 966       ?A#C   B
160 0A03A A3        GOYES  SL200   <exp> ou <exp> TO <exp>
161                  *
162                  * tRELOP specifier <exp>
163                  *
164 0A03C 1F00000    D1=(5) =S-R0-3
165 0A043 161       D0=D0+ 2
166 0A046 1564      C=DAT0 S      C(S) :=specifier
167 0A04A 1554      DAT1=C S      S-R0-3 := predicat
168 0A04E 160       D0=D0+ 1
169 0A051 7AA4      GOSUB  evalMS
170 0A055 136       CD0EX
171 0A058 1B00000    D0=(5) =S-R0-3

```

```

172 0A05F 1564          C=DAT0 S
173 0A063 134          D0=C
174 0A066 80DF          P=C    15
175 0A06A 7760          GOSUB  compar
176 0A06E 544          GONC   SL300
177 0A071 4B5          GOC    SL900    B.E.T.
178
179 * <exp>
180 * <exp> TO <exp>
181 *
182 0A074 7784          SL200  GOSUB  evalMS
183 0A078 14A          A=DAT0 B
184 0A07B 3100          LC(2)  =tTO
185 0A07F 962          ?A=C   B
186 0A082 E0           GOYES  SL250
187
188 * <exp>
189 *
190 0A084 22           P=      Pred=
191 0A086 7B40          GOSUB  compar
192 0A08A 582          GONC   SL300
193 0A08D 4F3          GOC    SL900    B.E.T.
194
195 * <exp> TO <exp>
196 *
197 0A090 26           SL250  P=      (Pred>)+(Pred=)
198 0A092 7F30          GOSUB  compar
199 0A096 551          GONC   SL290
200 0A099 161          D0=D0+ 2      On passe tTO
201 0A09C 7F54          GOSUB  evalMS
202 0A0A0 23           P=      (Pred<)+(Pred=)
203 0A0A2 7F20          GOSUB  compar
204 0A0A6 5C0          GONC   SL300
205 0A0A9 432          GOC    SL900    B.E.T.
206
207 0A0AC 161          SL290  D0=D0+ 2      On passe tTO
208 0A0AF 7C44          GOSUB  evalMS    Evaluation de <exp2> sans
209 *                   *      prendre le resultat
210 *
211 * Fin de l'evaluation. Si on est arrive jusqu'ici,
212 * toutes les conditions etaient fausses,
213 *
214 0A0B3 14A          SL300  A=DAT0 B
215 0A0B6 3100          LC(2)  =tCOMMA
216 0A0BA 966          ?A#C   B
217 0A0BD C0           GOYES  s1100    c'est donc tEOL
218 0A0BF 161          D0=D0+ 2      on passe la tCOMMA, et
219 0A0C2 14A          A=DAT0 B      on recommence
220 0A0C5 6D6F          GOTO   SL150
221 0A0C9 673F          s1100  GOTO   SL100
222 *
223 * Sortie et branchement. On a trouve :
224 * - une clause vraie, ou
225 * - CASE ELSE, ou
226 * - END SELECT
227 * Dans tous les cas, SAUVD0 = adresse de debut de
228 * la ligne (Stlen).

```



```
229 *
230 0A0CD 7BF3 SL900 GOSUB recadr
231 0A0D1 65B2 GOTO goeol
232
233 *****
234 * compar
235 *
236 * But: comparer l'objet pointe par D1 et l'objet
237 * pointe par AVMEME. Traite les verification de
238 * type (numerique / alpha).
239 * Entree:
240 * - P = predicat de comparaison
241 * - D1 = ^ objet2
242 * - AVMEME = ^ objet1
243 * - si CSETYP = 1 : type(objet1) = num
244 * - si CSETYP = 0 : type(objet1) = alpha, et
245 * S-R0-0 = adresse debut de objet1
246 * S-R0-1 = longueur en quartets
247 * Sortie:
248 * - Cy = 1 : vrai
249 * - Cy = 0 : faux
250 * Abime: A-D, D1, R0-R3
251 * Appelle: uTEST, FPOLL, STRTST
252 * Niveaux: 4 (FPOLL + pCMLPX)
253 * Note: D0 est sauvegarde dans STMTD1
254 * Historique:
255 * 87/03/01: conception & codage
256 *****
257
258 s1=2 EQU 0
259 s1>2 EQU 1
260
261 0A0D5 compar
262 *
263 * Sauvegarde P dans C(S)
264 *
265 0A0D5 80CF C=P 15 C(S) := Predicat
266 0A0D9 20 P= 0 Propre et net...
267 *
268 * Sauvegarde de D0
269 *
270 0A0DB 136 CD0EX
271 0A0DE 1B0000 D0=(5) =STMTD1
272 0A0E5 144 DAT0=C A
273
274 0A0E8 1B098F2 D0=(5) CSETYP Type de l'objet
275 0A0EF 1524 A=DAT0 S A(S) := type de objet1
276 *
277 * CSETYP = 1 si objet = num
278 * CSETYP = 0 si objet = str
279 *
280 0A0F3 A4C A=A-1 S Cy := 1 si (obj1 = str)
281 0A0F6 413 GOC cmpstr
282 *
283 * Comparaison numerique
284 *
285 0A0F9 109 cmpnum R1=C R1(S) := predicat
```

```

286 0A0FC 8F00000          GOSBVL =MPOP2N
287                        *
288                        * R0  = I(objet1)
289                        * A(W) = R(objet1)
290                        * R2  = I(objet2)
291                        * C(W) = R(objet2)
292                        *
293 0A103 04                SETHEX
294 0A105 4B1              GOC   cmpcmp   compare complex
295 0A108 129              CR1EX
296 0A10B 80DF            P=C   15      P = predicat
297 0A10F 129              CR1EX
298 0A112 8F00000          GOSBVL =uTEST
299 0A119 04                SETHEX
300 0A11B 20                P=    0
301 0A11D 65B0            GOTO   cmpfin   et puis voila...
302
303                        *
304                        * Comparaison de complexes
305                        *
306 0A121                  cmpcmp
307                        *
308                        * Pour le moment... on ne supporte pas les
309                        * complexes.
310                        *
311 0A121 8D00000          GOVLNG =CNFLCT  Conflict !!!
312                        *
313                        * Comparaison alphanumerique
314                        *
315 0A128 AC7              cmpstr D=C   S      D(S) := predicat de comp.
316 0A12B 8F00000          GOSBVL =POP1S
317 0A132 137              CD1EX          C(A) := ^ fin objet2
318 0A135 C2                C=C+A  A      C(A) := ^ debut objet2
319 0A137 135              D1=C          D1 := ^ 2eme chaine
320 0A13A 1B00000          D0=(5) =S-R0-1  D0 := ^ longueur objet1
321 0A141 146              C=DAT0 A      C(A) := longueur objet1
322 0A144 841              ST=0  s1>2    len1 < len2
323 0A147 840              ST=0  s1=2    len1 # len2
324 0A14A AC0              A=0    S
325 0A14D 8AA              ?C=0  A      len1 = 0 ?
326 0A150 21              GOYES  cmps00
327 0A152 8A8              ?A=0  A      len2 = 0 ?
328 0A155 86              GOYES  add>   obj1 > ""
329 0A157 8A6              ?C#A  A
330 0A15A 01              GOYES  cmps10
331 0A15C 850              ST=1  s1=2    len1 = len2
332 0A15F 541              GONC   cmps20  B.E.T.
333                        *
334                        * Longueur de obj1 = 0
335                        *
336 0A162 8A8              cmps00 ?A=0  A
337 0A165 C2                GOYES  add=
338 0A167 5C3              GONC   add<   B.E.T. (len1 < len2)
339
340 0A16A 8B6              cmps10 ?C<A  A      ?len1 < len2
341 0A16D 70              GOYES  cmps20
342 0A16F 851              ST=1  s1>2    len1 > len2

```

```
343 0A172 D6          C=A    A
344                  *
345                  * C(A) = longueur min
346                  * ST(s1>2) = 1 si (len1 > len2)
347                  * ST(s1=2) = 1 si (len1 = len2)
348                  * D0 = ^ obj1
349                  * D1 = ^ obj2
350                  *
351 0A174 184          cmps20 D0=D0- 5          D0=(5) =S-R0-0
352 0A177 142          A=DAT0 A
353 0A17A 130          D0=A          D0 := ^ objet1
354 0A17D 8F00000     GOSBVL =STRTST
355 0A184 20          P=    0
356 0A186 AC0          A=0    S          A(S):= predicat de sortie
357 0A189 482          GOC    cmps50   Strings not equal
358                  *
359                  * Les chaines sont egales jusqu'au minimum des
360                  * longueurs.
361                  *
362 0A18C 860          ?ST=0   s1=2
363 0A18F 01          GOYES   cmps30
364 0A191 302          add=   LC(1)   Pred=
365 0A194 816          CSRC
366 0A197 0E4E        A=A!C   S
367 0A19B 6B20        GOTO    cmps90   goto Bilan
368
369 0A19F 871          cmps30 ?ST=1   s1>2
370 0A1A2 B1          GOYES   add>
371
372 0A1A4 301          add<   LC(1)   Pred<
373 0A1A7 816          CSRC
374 0A1AA 0E4E        A=A!C   S
375 0A1AE 6810        GOTO    cmps90
376                  *
377                  * Les chaines ne sont pas egales.
378                  *
379 0A1B2 14A          cmps50 A=DAT0 B          A(B) := premier de objet1
380 0A1B5 14F          C=DAT1 B          C(B) := premier de objet2
381 0A1B8 9E2          ?A<C   B          obj1 < obj2
382 0A1BB 9E          GOYES   add<
383
384 0A1BD 304          add>   LC(1)   Pred>
385 0A1C0 816          CSRC
386 0A1C3 0E4E        A=A!C   S
387
388                  *
389                  * Bilan :
390                  * A(S) = predicat en sortie
391                  * D(S) = predicat en entree
392                  *
393 0A1C7 ACB          cmps90 C=D    S
394 0A1CA 0E46          A=A&C   S
395 0A1CE 94C          ?A#0   S
396 0A1D1 20          GOYES   cmpfin   Cy := 1
397 0A1D3              cmpfin
398                  *
399                  * Restauration de D0 sans perturber la Cy
```

```
400 *
401 0A1D3 1B00000          D0=(5) =STMTD1
402 0A1DA 146             C=DAT0 A
403 0A1DD 134             D0=C
404 0A1E0 01              RTN
405
406
407 *****
408 * CASE
409 *
410 * But: executer l'ordre CASE
411 * Hstorique:
412 * 87/03/01: conception & codage
413 *****
414
415 * *****
416 * CASE entry
417 * *****
418 0A1E2 00000          REL(5) =CASEd
419 0A1E7 00000          REL(5) =CASEp
420 0A1EC                =CASEe
421 0A1EC 187            D0=D0- 8
422 0A1EF 7623           GOSUB eol
423 0A1F3 28             P= chES-C
424 0A1F5 7333           GOSUB cherche
425 0A1F9 6D81           GOTO goeol
426
427
428 *****
429 * LEAVE
430 *
431 * But: executer l'ordre LEAVE
432 * Hstorique:
433 * 87/03/01: conception & codage
434 *****
435
436 * *****
437 * LEAVE entry
438 * *****
439 0A1FD 00000          REL(5) =NOARGd
440 0A202 00000          REL(5) =NOARGp
441 0A207                =LEAVEe
442 0A207 70D2           GOSUB stoadr
443 0A20B 8F00000        GOSBVL =POPUPD
444 0A212 403            GOC struce Pile vide
445 0A215 309            LC(1) STRUCT
446 0A218 816            CSRC C(S) := type de retour
447 0A21B 947            ?D#C S
448 0A21E 52             GOYES struce
449 *
450 * Le type est reconnu. Quel est le token qu'il faut
451 * chercher ?
452 *
453 0A220 DB              C=D A C(A) := ^ apres le token
454 0A222 134            D0=C
455 0A225 163            D0=D0+ 4 D0 := ^ token
456 0A228 14A            A=DAT0 B
```

```
457 0A22B 8F00000          GOSBVL =FINDA
458 0A232 34              CON(2) tWHILE
459 0A234 710            REL(3) LV10
460 0A237 06             CON(2) tLOOP
461 0A239 010            REL(3) LV20
462 0A23C 44             CON(2) tREPEAT
463 0A23E 900            REL(3) LV30
464 0A241 00             NIBHEX 00
465 0A243 6621          struce GOTO  STRUCe   "Structure Mismatch"
466
467                      *
468                      * REPEAT / UNTIL
469                      *
470 0A247 0C             LV30   P=P+1
471                      *
472                      * LOOP / END LOOP
473                      *
474 0A249 0C             LV20   P=P+1
475                      *
476                      * WHILE / END WHILE
477                      *
478 0A24B 80C0          LV10   C=P   0          C(0) := chX
479 0A24F 20              P=     0
480 0A251 A87            D=C    P          D(0) := chX
481 0A254 309            LC(1) (chL/EW)
482 0A257 A0B            C=C+D  P          C(0) := chL/EW + P
483 0A25A 80D0          P=C    0          P := chL/EW, chL/EL, chL/U
484 0A25E 7A62          GOSUB  recadr
485 0A262 76C2          GOSUB  cherche Chercher la fin de struct.
486 0A266 6021          GOTO    goeol
487
488
489                      *****
490                      * END2e
491                      *
492                      * But: traiter les ordres END WHILE, END LOOP,
493                      *   END SELECT et END IF
494                      * Entree:
495                      *   - D0 = ^ quartet de reconnaissance
496                      * Sortie:
497                      *   - voir chaque branche
498                      * Utilise: voir chaque branche
499                      * Abime: voir chaque branche
500                      * Niveaux: voir chaque branche
501                      * Historique:
502                      *   87/02/27: reconception
503                      *****
504
505                      * *****
506                      * END entry
507                      * *****
508 0A26A 00000          REL(5) =END2d
509 0A26F 00000          REL(5) =END2p
510 0A274              =END2e
511 0A274 14A            A=DAT0 B          token suivant / quartet
512 0A277 7EF0          GOSUB  eolxck
513 0A27B 4D0            GOC    ENDWe   tEOL ==> END WHILE
```

```
514      *
515      * A(0) = quartet de reconnaissance
516      *
517 0A27E 300      LC(1)  qENDL
518 0A281 902      ?A=C   P
519 0A284 72       GOYES  ENDLe   END LOOP
520 0A286 5D1      GONC   nxtstm  B.E.T.  END SELECT/IF
521
522      *****
523      * ENDWe
524      *
525      * But: proceder a l'execution de END WHILE
526      * Historique:
527      *   87/02/27: reconception
528      *   87/08/02: correction de "10 WHILE 1 !"
529      *
530      *****
531 0A289 3134      ENDWe  LC(2)  tWHILE
532 0A28D 79E1      GOSUB  pop
533 0A291 7F91      GOSUB  eval   Cy := 0 si e=0
534 0A295 5E0      GONC   nxtstm
535
536 0A298 7032      GOSUB  recadr  D0 := adresse depilee
537 0A29C 7CB1      GOSUB  push
538 0A2A0 6FD0      GOTO   go
539
540 0A2A4 8D0000    nxtstm GOVLNG =NXTSTM
541
542      *****
543      * ENDLe
544      *
545      * But: proceder a l'execution de END LOOP
546      * Historique:
547      *   87/02/27: reconception
548      *   87/08/02: correction de "10 LOOP !"
549      *
550      *****
551 0A2AB 3106      ENDLe  LC(2)  tLOOP
552 0A2AF 77C1      GOSUB  pop
553 0A2B3 75A1      GOSUB  push
554 0A2B7 68C0      GOTO   go
555
556      *****
557      * WHILEe
558      *
559      * Executer le statement WHILE
560      * Entree:
561      *   - D0 = ^ expression a analyser
562      * Sortie:
563      *   par go ou NXTSTM
564      * Historique:
565      *   87/02/28: reconception & codage
566      *
567      *****
568      *
569      * Tokenisation d'une boucle WHILE / END WHILE :
570      *
```

```
571          * tXWORD id tWHILE <exp>
572          *   <corps de la boucle>
573          * tWORD id tEND2
574          *
575
576          * *****
577          * WHILE entry
578          * *****
579 0A2BB 00000          REL(5) =STDD
580 0A2C0 00000          REL(5) =FIXp
581 0A2C5              =WHILEe
582 0A2C5 7212          GOSUB  stoadr
583 0A2C9 7761          GOSUB  eval
584 0A2CD 5E0           GONC   WHL10
585          *
586          * expression # 0 (vrai) : on continue en sequence
587          *
588 0A2D0 78F1          GOSUB  recadr   D0 := ^ debut expression
589 0A2D4 7481          GOSUB  push
590 0A2D8 6BCF          GOTO   nxtstm
591          *
592          * expression = 0 (faux) : on cherche END WHILE
593          * (D0 = ^ apres la fin de l'expression)
594          *
595 0A2DC 20            WHL10  P=     chew     chercher END WHILE
596 0A2DE 7A42          GOSUB  cherche
597 0A2E2 64A0          GOTO   goeol
598
599
600          * *****
601          * LOOP / REPEAT
602          *
603          * But: executer les ordres REPEAT et LOOP
604          * Entree:
605          * Sortie: par NXTSTM
606          * Appelle: push
607          * Historique:
608          *   87/02/28: conception & codage
609          * *****
610
611          *
612          * Tokenisation d'une boucle LOOP / END LOOP
613          *
614          * tWORD id tLOOP
615          *   <corps de la boucle>
616          * tXWORD id tEND2 qENDL
617          *
618
619          *
620          * Tokenisation d'une boucle REPEAT / UNTIL
621          *
622          * tXWORD id tREPEAT
623          *   <corps de la boucle>
624          * tXWORD id tUNTIL <exp>
625          *
626
627          * *****
```

```
628          * LOOP entry
629          * REPEAT entry
630          * *****
631 0A2E6 00000          REL(5) =NOARGd
632 0A2EB 00000          REL(5) =NOARGp
633 0A2F0          =LOOPe
634 0A2F0          =REPEAe
635 0A2F0 7861          GOSUB  push
636 0A2F4 6FAF          GOTO   nxtstm
637
638
639          *****
640          * UNTILe
641          *
642          * But: executer l'ordre UNTIL
643          * Entree:
644          *   - D0 = ^ debut de l'expression
645          * Sortie:
646          *   - par go ou nxtstm
647          * Appelle: pop, push, eval, recadr, eval
648          * Historique:
649          *   87/02/28: conception & codage
650          *   87/08/02: correction de "10 REPEAT !"
651          * *****
652
653          * *****
654          * UNTIL entry
655          * *****
656 0A2F8 00000          REL(5) =STDD
657 0A2FD 00000          REL(5) =FIXp
658 0A302          =UNTILe
659          *
660          * Sauver D0 pour eval
661          *
662 0A302 136          CD0EX
663 0A305 1B00000        D0=(5) =STMTD1
664 0A30C 144          DAT0=C A      STMTD1 := ^ debut de l'exp
665          *
666          * Depiler l'adresse de retour et verifier REPEAT
667          *
668 0A30F 3144          LC(2)  tREPEAT
669 0A313 7361          GOSUB  pop
670          *
671          * Evaluer l'expression
672          *
673 0A317 1B00000        D0=(5) =STMTD1
674 0A31E 146          C=DAT0 A
675 0A321 134          D0=C
676 0A324 7C01          GOSUB  eval
677 0A328 4E0          GOC    Nxtstm
678          *
679          * On boucle :
680          *
681 0A32B 7D91          GOSUB  recadr   D0 := ^ fin du REPEAT
682 0A32F 7921          GOSUB  push
683 0A333 6C40          GOTO   go      GOTO "fin du REPEAT"
684
```



```
685 0A337 6C6F      Nxtstm GOTO   nxtstm
686
687
688                *****
689                * IF2
690                *
691                * But: executer l'ordre IF
692                * Hstorique:
693                *   87/03/01: conception & codage
694                *****
695
696                *
697                * Tokenisation d'une structure IF / END IF
698                *
699                * tXWORD id tIF2 <exp>
700                *   <alternant si vrai>
701                * tXWORD id tELSE2
702                *   <alternant si faux>
703                * tXWORD id tEND2 qENDI
704                *
705                * ou encore :
706                *
707                * tXWORD id tIF2 <exp>
708                *   <alternant si vrai>
709                * tXWORD id tEND2 qENDI
710                *
711
712                * *****
713                * IF entry
714                * *****
715 0A33B 00000          REL(5) =IF2d
716 0A340 00000          REL(5) =IF2p
717 0A345              =IF2e
718 0A345 7BE0          GOSUB  eval
719 0A349 4DE           GOC    Nxtstm
720 0A34C 25            P=    chE|EI
721 0A34E 7AD1          GOSUB  cherche
722 0A352 6430          GOTO   goeol
723
724                *****
725                * ELSE2
726                *
727                * But: executer l'ordre ELSE
728                * Hstorique:
729                *   87/03/01: conception & codage
730                *****
731
732                * *****
733                * ELSE entry
734                * *****
735 0A356 00000          REL(5) =NOARGd
736 0A35B 00000          REL(5) =ELSE2p
737 0A360              =ELSE2e
738 0A360 23            P=    chEI
739 0A362 76C1          GOSUB  cherche
740 0A366 6020          GOTO   goeol
741
```

```
742 *****
743 * STRUCe
744 *
745 * But: generer une erreur et retourner a Basic
746 * Entree: -
747 * Sortie: par BSERR
748 * Historique:
749 * 87/02/28: conception & codage
750 *****
751
752 0A36A 20 STRUCe P= 0
753 0A36C 33301E LC(4) #E103 (=id)~(=eSTRUC)
754 0A372 8D00000 GOVLNG =BSERR
755
756 0A379 8D00000 eolxck GOVLNG =EOLXCK
757
758 *****
759 * go, goeol
760 *
761 * But: revenir a Basic, en faisant un GOTO sur
762 * le statement suivant D0 (go), ou sur le token
763 * suivant la ligne dont la longueur est pointee
764 * par D0 (goeol).
765 * Entree:
766 * go :
767 * - SAUVD0 = ^ Stlen du debut de la ligne courante
768 * goeol:
769 * - D0 = ^ Stlen du debut de la ligne courante
770 * Sortie: par RUNRT1
771 * Algorithme:
772 * si execution au clavier
773 * alors
774 * si adresse de destination n'est pas dans
775 * dans bSTMT
776 * alors "Structure Mismatch"
777 * fin si
778 * fin si
779 * si trace
780 * alors trace
781 * fin si
782 * RUNRT1
783 * Historique:
784 * 87/02/27: conception & codage
785 * 87/03/02: integration de goeol
786 * 87/03/04: correction du bug "trace"
787 * 87/08/02: modification de go, bug "10 LOOP !"
788 *****
789
790 0A380 7841 go GOSUB recadr
791 0A384 187 D0=D0- 6+2 6 : token, 2 : Stlen
792
793 0A387 7E81 goeol GOSUB eol
794
795 *
796 * D(A) := D0
797 *
798 0A38B 136 CD0EX
```

```
799 0A38E 134          D0=C
800 0A391 D7          D=C   A
801                  *
802                  * si execution au clavier :
803                  *
804 0A393 870          ?ST=1  =PgmRun
805 0A396 D1          GOYES  go10
806 0A398 32000       LC(3)  =bSTMT   Statement buffer (clavier)
807 0A39D 8F00000     GOSBVL =IOFND0 D1 := ^ statement buffer
808                  *
809                  * Le buffer existe toujours. Du moins, on espere !
810                  *
811 0A3A4 137          CD1EX          C(A) := ^ buffer
812 0A3A7 8B7          ?D<C   A          dest < debut buffer ?
813 0A3AA 0C          GOYES  STRUCe
814 0A3AC C2          C=C+A  A          C(A) := ^ fin buffer
815 0A3AE 8B3          ?D>C   A          dest > fin buffer ?
816 0A3B1 9B          GOYES  STRUCe
817                  *
818                  * trace ?
819                  *
820 0A3B3 860          go10   ?ST=0  =Trace
821 0A3B6 77          GOYES  runrt1
822                  *
823                  * le mode TRACE est actif
824                  *
825                  *
826                  *
827                  * Ce qui suit est pompe dans trflck (#0FE18)
828                  * (avec modificatio)
829                  *
830 0A3B8 860          ?ST=0  =PgmRun
831 0A3BB 27          GOYES  runrt1
832 0A3BD 1F00000     D1=(5) =TRACEM
833 0A3C4 20          P=      0
834 0A3C6 302         LCHEX  2
835 0A3C9 14B         A=DAT1 B
836 0A3CC 0E06        A=A&C  P
837 0A3D0 A0C         A=A-1  P
838                  *
839                  * Jusqu'au 4 mars 1987, l'absence de cette
840                  * instruction declenchait abusivement le mode
841                  * TRACE FLOW lors de l'execution d'une fonction de
842                  * STRUCLEX
843                  *
844 0A3D3 495          GOC     runrt1
845                  *
846                  * Elle est toute petite, hein ?
847                  *
848 0A3D6 1F00000     D1=(5) =CURRST
849 0A3DD 147          C=DAT1 A
850 0A3E0 135          D1=C
851 0A3E3 170          D1=D1+ (=oFTYPh)-1
852 0A3E6 143          A=DAT1 A
853 0A3E9 F4          ASR    A
854 0A3EB 3400000     LC(5)  =fBASIC
855 0A3F2 8A6          ?A#C   A
```

```

856 0A3F5 83          GOYES  runrt1
857                  *
858                  * code Taillandier inspire par les dieux d'HP
859                  *
860 0A3F7 136         CD0EX
861 0A3FA 10A         R2=C
862 0A3FD 134         D0=C
863                  *
864                  * Est-on a la fin du fichier ?
865                  *
866 0A400 1F00000     D1=(5) =CURREN
867 0A407 143         A=DAT1 A      A(A) := CURREN
868 0A40A E6          C=C+1  A
869 0A40C E6          C=C+1  A      C(A) := ^ apres le tEOL
870 0A40E 8BA         ?C>=A  A
871 0A411 C1          GOYES  runrt1
872                  *
873                  * On n'est pas hors du fichier. On peut tracer en
874                  * toute quietude : le numero de ligne est valide.
875                  *
876 0A413 8F00000     GOSBVL =TRFROM  partie FROM de trace
877 0A41A 11A         C=R2
878 0A41D 134         D0=C
879 0A420 8F00000     GOSBVL =TRTO+  partie TO de trace
880 0A427 112         A=R2
881 0A42A 130         D0=A      restaure D0 et fin
882 0A42D 8D00000     runrt1 GOVLNG =RUNRT1
883
884                  *****
885                  * eval
886                  *
887                  * But: evaluer l'expression pointee par D0, et
888                  * renvoyer le resultat sous forme booleenne (Cy)
889                  * Entree:
890                  *   - D0 = ^ expression tokenisee
891                  * Sortie:
892                  *   - P = 0
893                  *   - D0 = ^ apres l'expression tokenisee
894                  *   - Cy = 0 : expression = 0
895                  *   - Cy # 0 : expression # 0
896                  * Appelle: EXPEX-, MPOP1N
897                  * Niveaux: 5 (EXPEX-)
898                  * Abime: A-D, D0, D1, R0-R4, FUNCtion scratch
899                  * Historique:
900                  *   87/02/27: ajout de documentation
901                  *   87/02/27: clarification du code
902                  *****
903
904 0A434 8F00000     eval  GOSBVL =EXPEX-  vide la M.S.
905 0A43B 8F00000     GOSBVL =MPOP1N  depile le nombre (C ou R)
906 0A442 04          SETHEX
907 0A444 AC0         A=0    S      ne prend pas le signe (-0)
908 0A447 5D0         GONC   eval10  nombre reel
909                  *
910                  * nombre complexe
911                  *
912 0A44A 97C         ?A#0  W

```

```
913 0A44D 00          RTNYES
914 0A44F 110        A=R0
915 0A452 AC0        A=0      S
916 0A455 97C      eval10 ?A#0  W
917 0A458 00          RTNYES
918 0A45A 01          RTN
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935 0A45C 309      push  LC(1)  STRUCt  C(0) := type de l'adresse
936 0A45F 816          CSRC      C(S) := return type
937 0A462 185          D0=D0- 6    D0 := ^ token
938 0A465 132          AD0EX     A(A) := adresse du token
939 0A468 8F00000     GOSBVL =PSHGSB
940 0A46F 816          CSRC      C(A) := adresse empilee
941 0A472 134          D0=C      D0 := ^ adresse du token
942 0A475 165          D0=D0+ 6   D0 := adresse en entree
943 0A478 01          RTN
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968 0A47A 108      pop    R0=C      R0(B) := token a verifier
969 0A47D 8F00000     GOSBVL =POPUPD
```

* push
*
* But: mettre D0 dans la pile des GOSUB.
* Entree:
* - D0 = valeur a empiler
* Sortie:
* - D0 = valeur en entree
* Appelle: PSHGSB
* Niveaux: 4 (PSHGSB)
* Abime: A-D, D1
* Historique:
* 87/02/27: conception & codage

* pop
*
* But: depile une adresse sur la GOSUB stack,
* verifie le type de l'adresse (tSTRUC), pointe
* D0 a cette adresse et verifie que le token
* pointe est le meme que C(B).
* Entree:
* - C(B) = token a verifier (2 q. seulement)
* Sortie:
* - D0 = ^ adresse depilee (passe le token)
* Appelle: POPUPD
* Niveaux: 2 (POPUPD)
* Utilise: A-D, D0, D1, R0
* Detail:
* si type de l'adresse # tSTRUC
* alors "Structure Mismatch"
* si token pointe n'est pas bon
* alors "Structure Mismatch"
* Historique:
* 87/02/27: conception & codage

```

970 0A484 434          GOC   STRUCe  pile vide
971 0A487 309          LC(1) STRUCt  type de retour
972 0A48A 816          CSRC          C(S) := STRUCt
973 0A48D 947          ?D#C   S      ce qui est depile ne nous
974 0A490 83           GOYES  STRUCe  appartient pas...
975                    *
976                    * le type est reconnu, c'est a nous. Est-ce qu'on
977                    * pointe sur la bonne structure ?
978                    *
979 0A492 DB           C=D    A
980 0A494 134          D0=C          D0 := ^ adresse de retour
981 0A497 AF0          A=0    W
982 0A49A 15A5         A=DAT0 6      A(W) := token lu
983 0A49E 165          D0=D0+ 6      D0 := ^ apres le token
984
985 0A4A1 7630          GOSUB  stoadr  SAUVD0 := D0 + 6
986
987 0A4A5 118          C=R0          C(B) := token
988 0A4A8 AE5          B=C    B      B(B) := token
989 0A4AB AF2          C=0    W
990 0A4AE AE9          C=B    B      C(W) := token
991 0A4B1 BF2          CSL   W
992 0A4B4 BF2          CSL   W
993 0A4B7 BF2          CSL   W
994 0A4BA BF2          CSL   W      C(5-4) := token
995 0A4BD 330000        LC(4)  (id)~(=tXWORD)
996 0A4C3 972          ?A=C   W
997 0A4C6 00           RTNYES          Tokens identiques : OK
998 0A4C8 61AE        STRUCe GOTO   STRUCe
999
1000                    *****
1001                    * recadr
1002                    *
1003                    * But: restaurer dans D0 l'adresse sauvee par pop.
1004                    * Entree: -
1005                    * Sortie:
1006                    *   - D0 = adresse sauvee par pop
1007                    * Niveaux: 0
1008                    * Abime: C(A), D0
1009                    * Historique:
1010                    *   87/02/28: conception & codage
1011                    *****
1012
1013 0A4CC 1B198F2      recadr D0=(5) SAUVD0
1014 0A4D3 146          C=DAT0 A
1015 0A4D6 134          D0=C
1016 0A4D9 01           RTN
1017
1018                    *****
1019                    * stoadr
1020                    *
1021                    * But: sauver D0 dans SAUVD0
1022                    * Entree:
1023                    *   - D0 = valeur a sauver
1024                    * Sortie:
1025                    *   - D0 inchange
1026                    * Niveaux: 0

```

```
1027      * Abime: C(A)
1028      * Historique:
1029      *   87/02/28: conception & codage
1030      *****
1031
1032 0A4DB 136      stoadr CD0EX
1033 0A4DE 1B198F2      D0=(5) SAUVDO
1034 0A4E5 144      DAT0=C A
1035 0A4E8 134      D0=C
1036 0A4EB 01      RTN
1037
1038      *****
1039      * stoMS
1040      *
1041      * But: sauver D1 dans LDCSPC
1042      * Entree:
1043      *   - D1 = valeur a sauver
1044      * Sortie:
1045      *   - D0 inchange
1046      * Niveaux: 0
1047      * Abime: C(A)
1048      * Historique:
1049      *   87/03/01: conception & pompage sur stoadr
1050      *****
1051
1052 0A4ED 137      stoMS CD1EX
1053 0A4F0 1F00000      D1=(5) =LDCSPC
1054 0A4F7 145      DAT1=C A
1055 0A4FA 135      D1=C
1056 0A4FD 01      RTN
1057
1058      *****
1059      * evalMS
1060      *
1061      * But: Evaluer l'expression pointee par D0, et la
1062      * ranger dans la pile a l'adresse contenue dans
1063      * LDCSPC.
1064      * Entree:
1065      *   - D0 = ^ debut de l'exp. tokenisee
1066      *   - LDCSPC = ^ M.S.
1067      * Sortie:
1068      *   - D0 = ^ apres l'expression (token non util.)
1069      *   - D1 = ^ M.S.
1070      *   - A(W) = 16 quartets au sommet de la M.S.
1071      * Niveaux: 5
1072      * Abime: A-D, R0-R4, ST, D0, D1, FUNC scratch
1073      * Historique:
1074      *   87/03/01: conception & codage
1075      *   87/03/02: modification pour integration EXPEXC
1076      *****
1077
1078 0A4FF 1F00000      evalMS D1=(5) =LDCSPC
1079 0A506 147      C=DAT1 A
1080 0A509 1E0000      D1=(4) =MTHSTK
1081 0A50F 145      DAT1=C A
1082 0A512 8D00000      GOVLNG =EXPEXC
1083
```

```
1084 *****
1085 * eol
1086 *
1087 * But: calculer l'adresse de la fin du statement
1088 * courant (adresse de tEOL ou t@) connaissant
1089 * l'adresse du debut de la ligne (^ Stlen).
1090 * Entree:
1091 * - D0 = ^ longueur de la ligne (Stlen)
1092 * Sortie:
1093 * - D0 = ^ t@ ou tEOL ou...
1094 * Abime: A(A), C(A), D0
1095 * Historique:
1096 * 87/03/01: extraction de "goeol"
1097 *****
1098
1099 0A519 D0      eol      A=0      A
1100 0A51B 14A    A=DAT0 B      A(B) := longueur
1101 0A51E 136    CD0EX      C(A) := ^ longueur
1102 0A521 C2    C=C+A  A      C(A) := ^ fin de la ligne
1103 0A523 134    D0=C
1104 0A526 01    RTN
1105
1106 0A528 614E  STRuce GOTO  STRUCe
1107
1108 *****
1109 * cherche
1110 *
1111 * But: chercher pour l'objet specifie par P
1112 * Entree:
1113 * - P = quartet de recherche
1114 * chEW : chercher END WHILE
1115 * chEL : END LOOP
1116 * chU  : UNTIL
1117 * chEI : END IF
1118 * chES : END SELECT
1119 * chE|EI : ELSE ou END IF
1120 * chC|ES : CASE ou END SELECT
1121 * chEI-E : END IF, autorise ELSE
1122 * chES-C : END SELECT, autorise CASE
1123 * chL/EW : END WHILE, autorise (*)
1124 * chL/EL : END LOOP, autorise (*)
1125 * chL/U  : UNTIL, autorise (*)
1126 * (*) = CASE, ELSE, END SELECT, END IF
1127 * - D0 = tEOL ou t@ du statement courant
1128 * Sortie:
1129 * si l'objet a ete trouve
1130 * - D0 = ^ Stlen du statement reconnu
1131 * Abime: A-D, D0, D1, P, SAUVD0
1132 * Niveaux:
1133 * Appelle: chpush, chpop, FINDA, TBLJMC, stoadr,
1134 * recadr, TKSCN7
1135 * Historique:
1136 * 87/03/01: conception & codage
1137 *****
1138
1139 *
1140 * Version recursive :
```



```
1141      *
1142      * SUB cherche (P)
1143      *   T1$ = premier token a chercher (selon P)
1144      *   T2$ = deuxieme token a chercher (si il y a)
1145      *   LOOP
1146      *     TKSCN7
1147      *     SELECT XWORD
1148      *       CASE "LOOP", "WHILE", "REPEAT", "SELECT", "IF"
1149      *         P= "ch" & (token final correspondant)
1150      *         CALL cherche (P)
1151      *         CASE T1$, T2$
1152      *           LEAVE
1153      *         CASE "END WHILE/LOOP/SELECT/IF", "UNTIL",
1154      *           "CASE", "ELSE"
1155      *           "Structure Mismatch"
1156      *     END SELECT
1157      *   END LOOP
1158      * END SUB
1159      *
1160
1161      *
1162      * Version derecursivee :
1163      *
1164      * cherche:
1165      *   initialiser la pile
1166      *   cher:
1167      *     T1$ = premier token a chercher (selon P)
1168      *     T2$ = deuxieme token a chercher (si il y a)
1169      *     TKNSCN7
1170      *     LOOP
1171      *       SELECT XWORD
1172      *         CASE "LOOP", "WHILE", "REPEAT", "SELECT", "IF"
1173      *           PUSH P
1174      *           Avancer sur la fin de la ligne
1175      *           P= "ch" & (token final correspondant)
1176      *           GOTO cher (Argghhhhhh !)
1177      *           CASE T1$, T2$
1178      *             IF pile_vide THEN GOTO sortie
1179      *             Avancer sur la fin de la ligne
1180      *             POP P
1181      *             GOTO cher (Argghhhhhh !)
1182      *           CASE "END WHILE/LOOP/SELECT/IF", "UNTIL",
1183      *             "CASE", "ELSE"
1184      *             "Structure Mismatch"
1185      *         END SELECT
1186      *     END LOOP
1187      *   Sortie:
1188      *
1189
1190      *
1191      * Zones memoires utilisees :
1192      *
1193      * AVMEME : pour voir si on ne depasse pas les
1194      * limites de la memoire du HP-71.
1195      * OUTBS : bas de la pile
1196      * AVMEMS : sommet de la pile
1197      *
```

```

1198          SP      EQU      #2F594    =AVMEMS
1199          BP      EQU      #2F58F    =OUTBS
1200          Pvar    EQU      #2F880    =S-R0-3
1201
1202 0A52C      cherche
1203 0A52C 80CF          C=P      15      C(S) := chX
1204 0A530 20          P=       0
1205 0A532 8F00000     GOSBVL =OBCOLL  initialiser la pile
1206          * OBCOLL n'a pas modifie D0
1207 0A539 80DF          P=C      15      P := chX
1208 0A53D 80C0      cher    C=P      0      C(0) := quartet de rech.
1209 0A541 20          P=       0      Ouf !
1210 0A543 1F088F2     D1=(5) Pvar
1211 0A54A 1550          DAT1=C P      Pvar := quartet de rech.
1212 0A54E      che010
1213          *
1214          * adresse de la fin d'analyse :
1215          *
1216 0A54E 1F00000     D1=(5) =PRGMEN
1217 0A555 147          C=DAT1 A      C(A) := PRGMEN par default
1218 0A558 870          ?ST=1 =PgmRun
1219 0A55B 31          GOYES  che020
1220          *
1221          * On cherche dans le buffer bSTMT
1222          *
1223 0A55D 32000     LC(3) =bSTMT
1224 0A562 8F00000     GOSBVL =IOFNDO
1225 0A569 137          CD1EX
1226 0A56C C2          C=C+A  A
1227 0A56E D7          che020 D=C  A
1228          *
1229          * On cherche le token tXWORD :
1230          *
1231 0A570 3100     LC(2) =tXWORD
1232 0A574 8F00000     GOSBVL =TKSCN7
1233          *
1234          * Cy = 0 : non trouve : "Structure Mismatch"
1235          * Cy = 1 : trouve. Les ennuis commencent...
1236          *
1237 0A57B 5CA          GONC   STRuce  "Structure Mismatch"
1238          *
1239          * Sauvegarde de l'adresse de tXWORD
1240          *
1241 0A57E 795F     GOSUB  stoadr  SAUVD0 := ^ tXWORD
1242          *
1243          * Dans SAUVD0, nous avons :
1244          *
1245          * Stlen tXWORD id tTOKEN
1246          * ^
1247          * SAUVD0
1248          *
1249 0A582 161     D0=D0+ 2      D0 := ^ id
1250          *
1251          * tXWORD id tTOKEN
1252          * ^
1253          * D0
1254          *

```

```
1255 0A585 14A          A=DAT0 B          A(B) := id trouve
1256 0A588 311E        LC(2) id          C(B) := NOTRE id
1257 0A58C 962         ?A=C B
1258 0A58F 60          GOYES che022
1259 0A591 6DA0        GOTO che900      EOL(D0-2) ; GOTO cher10
1260 0A595             che022
1261 *
1262 * C'est notre id
1263 *
1264 0A595 161          D0=D0+ 2         D0 := ^ TOKEN
1265 *
1266 * tXWORD id tTOKEN
1267 * ^
1268 * D0
1269 *
1270 0A598 14A          A=DAT0 B          A(B) := token
1271 *
1272 * Est-ce un token de structure ?
1273 * (WHILE, LOOP, REPEAT, SELECT, IF, END, UNTIL,
1274 * CASE et ELSE)
1275 *
1276 0A59B 3124         LC(2) tEND2
1277 0A59F 962         ?A=C B
1278 0A5A2 02          GOYES che025
1279 *
1280 * On pourrait aller directement au TBLJMC, mais ca
1281 * nuit a la lisibilite et a la maintenabilite.
1282 * Vive la programmation structuree !
1283 *
1284 0A5A4 3            NIBHEX 3         LC
1285 0A5A5 F           CON(1) 2*8-1     (15)
1286 0A5A6 34         CON(2) tWHILE
1287 0A5A8 06         CON(2) tLOOP
1288 0A5AA 44         CON(2) tREPEAT
1289 0A5AC 16         CON(2) tSELECT
1290 0A5AE 36         CON(2) tIF2
1291 0A5B0 54         CON(2) tUNTIL
1292 0A5B2 26         CON(2) tCASE
1293 0A5B4 46         CON(2) tELSE2
1294 0A5B6 2F         P= 2*8-1
1295 0A5B8 8F00000    GOSBVL =MEMBER
1296 0A5BF 4F7        GOC che900      Byte not in set
1297 0A5C2             che025
1298 *
1299 * Est-ce un debut de structure ?
1300 *
1301 0A5C2 8F00000    GOSBVL =FINDA
1302 0A5C9 34         CON(2) tWHILE
1303 0A5CB F50        REL(3) che200
1304 0A5CE 06         CON(2) tLOOP
1305 0A5D0 850        REL(3) che210
1306 0A5D3 44         CON(2) tREPEAT
1307 0A5D5 150        REL(3) che220
1308 0A5D8 36         CON(2) tIF2
1309 0A5DA 640        REL(3) che230
1310 0A5DD 16         CON(2) tSELECT
1311 0A5DF B30        REL(3) che240
```

```
1312 0A5E2 00          CON(2) 00
1313                  *
1314                  * Non. Va-t-on trouver ce qu'on cherche ?
1315                  *
1316 0A5E4 1F088F2      D1=(5) Pvar
1317 0A5EB 1570        C=DAT1 P
1318
1319 0A5EF 8F00000      che099 GOSBVL =TBLJMC
1320 0A5F6 850          REL(3) che100
1321 0A5F9 811          REL(3) che110
1322 0A5FC 760          REL(3) che120
1323 0A5FF E01          REL(3) che130
1324 0A602 D01          REL(3) che140
1325 0A605 D60          REL(3) che150
1326 0A608 F70          REL(3) che160
1327 0A60B 7F0          REL(3) che170
1328 0A60E 5D0          REL(3) che180
1329 0A611 A80          REL(3) che190
1330 0A614 580          REL(3) che1a0
1331 0A617 080          REL(3) che1b0
1332                  *
1333                  * "WHILE", "LOOP", "UNTIL", "IF", "SELECT"
1334                  *
1335 0A61A 28            che240 P=      chES-C   END SELECT, sans CASE
1336 0A61C 6D00         GOTO      che200
1337 0A620 27          che230 P=      cheI-E   END IF, sans ELSE
1338 0A622 6700         GOTO      che200
1339 0A626 0C          che220 P=P+1
1340 0A628 0C          che210 P=P+1
1341 0A62A              che200
1342                  *
1343                  * P = nouveau quartet de recherche
1344                  *
1345 0A62A 80CF          C=P      15      C(S) := nouveau quartet
1346 0A62E 20           P=      0
1347 0A630 7021         GOSUB    chpush   empiler le courant
1348
1349 0A634 1F088F2      D1=(5) Pvar
1350 0A63B 1554         DAT1=C S
1351
1352                  *
1353                  * ATTENTION : le code continue !!!
1354                  *
1355 0A63F 798E          che900 GOSUB    recadr
1356 0A643 181          D0=D0- 2
1357 0A646 7FCE          GOSUB    eol
1358 0A64A 630F          GOTO     che010
1359
1360                  *
1361                  * chercher END WHILE
1362                  *
1363 0A64E 7A7E          che100 GOSUB    recadr
1364 0A652 165          D0=D0+ 6      D0 := ^ q. de reconnais.
1365                  *
1366                  * tXWORD id tEND qENDx tEOL ou encore
1367                  * tXWORD id tEND tEOL
1368                  * ^
```

```

1369          *                D0
1370          *
1371 0A655 14A          A=DAT0 B
1372 0A658 7D1D        GOSUB eolxck
1373 0A65C 511        GONC  Struce  "Structure Mismatch"
1374 0A65F 60D0        GOTO  che300  B.E.T. a fin
1375          *
1376          * chercher UNTIL
1377          *
1378 0A663 3154        che120 LC(2)  tUNTIL
1379 0A667 7FD0        GOSUB  che400
1380 0A66B 471        GOC   che299  Trouve
1381 0A66E 6BFC        Struce GOTO  STRUCE  Non trouve
1382          *
1383          * chercher ELSE ou END IF
1384          *
1385 0A672 3146        che150 LC(2)  tELSE2
1386 0A676 70D0        GOSUB  che400
1387 0A67A 480        GOC   che299  Trouve
1388 0A67D 22          P=    qENDI  Non. Chercher END IF
1389 0A67F 6190        GOTO  che110  B.E.T.
1390          *
1391          * chercher CASE ou END SELECT
1392          *
1393 0A683 6CA0        che299 GOTO  che300
1394          *
1395 0A687 3126        che160 LC(2)  tCASE
1396 0A68B 7BB0        GOSUB  che400
1397 0A68F 43F        GOC   che299  Trouve
1398 0A692 21          P=    qENDS  Non. Chercher END SELECT
1399 0A694 5C7        GONC  che110  B.E.T.
1400          *
1401          * chercher END WHILE, mais pas CASE, ELSE,
1402          * END SELECT, END IF
1403          *
1404 0A697 0C          che1b0 P=P+1
1405 0A699 0C          che1a0 P=P+1
1406 0A69B 80CF        che190 C=P    15
1407 0A69F 20          P=    0
1408 0A6A1 AC7        D=C    S      D(S) := cheW, chEL ou chU
1409 0A6A4 14A        A=DAT0 B
1410 0A6A7 334626      LC(4) #6264  (=tCASE)~(=tELSE2)
1411 0A6AD 23          P=    2*2-1
1412 0A6AF 8F00000    GOSBVL =MEMBER
1413 0A6B6 5C3        GONC  che175  Trouve : on recommence
1414 0A6B9 3124        LC(2) tEND2
1415 0A6BD 966        ?A#C  B
1416 0A6C0 91         GOYES  che195
1417          *
1418          * tEND2 trouve : est-ce END SELECT ou END IF ?
1419          *
1420 0A6C2 161         D0=D0+ 2
1421 0A6C5 1520        A=DAT0 P
1422 0A6C9 301        LC(1) qENDS
1423 0A6CC 902        ?A=C  P
1424 0A6CF 42         GOYES  che175  Trouve, on recommence
1425 0A6D1 302        LC(1) qENDI

```

```

1426 0A6D4 902          ?A=C   P
1427 0A6D7 C1          GOYES  che175  Trouve, on recommence
1428 0A6D9 ACB        che195 C=D    S
1429 0A6DC 812          CSLC   C(0) := quartet de rech.
1430 0A6DF 6F0F        GOTO  che099  ch. END WHILE/LOOP/UNTIL
1431
1432 *
1433 * chercher END SELECT, mais pas CASE
1434
1434 0A6E3 3126        che180 LC(2)  tCASE
1435 0A6E7 7F50        GOSUB  che400
1436 0A6EB 470         GOC    che175  On repart pour une rech.
1437 0A6EE 21          P=     qENDS
1438 0A6F0 502         GONC  che110  B.E.T.
1439
1440 0A6F3 75DD        che175 GOSUB  recadr
1441 0A6F7 181         D0=D0- 2
1442 0A6FA 7B1E        GOSUB  eol
1443 0A6FE 6F4E        GOTO  che010  C'est reparti comme en 14
1444
1445 *
1446 * chercher END IF, mais pas ELSE
1447
1447 0A702 3146        che170 LC(2)  tELSE2
1448 0A706 7040        GOSUB  che400
1449 0A70A 48E         GOC    che175  On repart pour une rech.
1450
1451 *
1452 * ATTENTION : le code continue !!!
1453
1454 *
1455 *
1456 * chercher END IF
1457
1458 0A70D 0C          che130 P=P+1
1459
1460 *
1461 * chercher END SELECT
1462
1462 0A70F 0C          che140 P=P+1
1463
1464 *
1465 * chercher END LOOP
1466
1466 0A711            che110
1467
1468 * tXWORD id tEND qENDx tEOL ou encore
1469 * tXWORD id tEND tEOL
1470
1471 0A711 80CF        C=P    15      C(S) := qENDx
1472 0A715 20          P=     0
1473 0A717 71BD        GOSUB  recadr
1474 0A71B 165        D0=D0+ 6      D0 := ^ qENDx ou tEOL
1475 0A71E 14A        A=DAT0 B
1476 0A721 745C        GOSUB  eolxck
1477 0A725 402        GOC    STRuce  tEOL... ==> END WHILE
1478
1479 *
1480 * A(B) = token lu
1481
1481 0A728 812          CSLC   C(0) = qENDx
1482 0A72B 906          ?A#C  P

```

```
1483 0A72E 81          GOYES  STruce
1484                  *
1485                  * ATTENTION : le code continue !!!
1486                  *
1487
1488                  *
1489                  * Si la pile est vide
1490                  *   alors
1491                  *     sortir
1492                  *   sinon
1493                  *     eol
1494                  *   chpop
1495                  *   goto che010
1496                  * fin si
1497                  *
1498 0A730 789D          che300 GOSUB  recadr
1499 0A734 181          D0=D0- 2      D0 := ^ Stlen
1500 0A737 7250          GOSUB  chpop
1501 0A73B 400          RTNC      Sortie sur pile vide
1502 0A73E 77DD          GOSUB  eol
1503 0A742 6B0E          GOTO   che010
1504
1505 0A746 632C          STruce GOTO  STRUCe
1506
1507                  *
1508                  * Est-ce le token C(B) ?
1509                  *
1510 0A74A 14A          che400 A=DAT0 B
1511 0A74D 962          ?A=C   B
1512 0A750 00          RTNYES      Cy := 1
1513 0A752 01          RTN      Cy := 0
1514
1515                  *****
1516                  * chpush
1517                  *
1518                  * But: empiler les tokens courants (T1$, et T2$)
1519                  *   sur la pile de recherche.
1520                  * Entree:
1521                  *   - P = 0
1522                  *   - Pvar = recherche courante
1523                  *   - AVMEME = limite sup. de la memoire dispon.
1524                  *   - SP = sommet de la pile de recherche
1525                  * Sortie:
1526                  *   - si pas assez de place : memerr
1527                  *   - SP reactualise
1528                  * Abime: A(A), C(A), D1
1529                  * Appelle: D1=AVE
1530                  * Niveaux: 1
1531                  * Historique:
1532                  *   87/03/01: conception & codage
1533                  *****
1534
1535 0A754 1F495F2      chpush D1=(5) SP
1536 0A75B 143          A=DAT1 A      A(A) := SP
1537 0A75E E4           A=A+1  A      A(A) := SP + 1
1538 0A760 141          DAT1=A A      Sp := SP + 1
1539 0A763 8F00000      GOSBVL =D1=AVE C(A) := (AVMEME)
```

```
1540 0A76A 8BE          ?A>=C  A
1541 0A76D 91          GOYES  memerr
1542 0A76F 1F088F2     D1=(5) Pvar
1543 0A776 1570       C=DAT1 P          C(0) := Pvar
1544 0A77A 131        D1=A             D1 := SP + 1
1545 0A77D 1C0        D1=D1- 1
1546 0A780 1550       DAT1=C P          empiler Pvar
1547 0A784 01         RTN
1548
1549 0A786 8D0000     memerr GOVLNG =MEMERR
1550
1551 *****
1552 * chpop
1553 *
1554 * But: depiler les tokens au sommet de la pile de
1555 *       recherche, et les mettre dans les tokens
1556 *       courants.
1557 * Entree:
1558 *   - P = 0
1559 *   - BP = bas de la pile
1560 *   - SP = sommet de la pile de recherche
1561 * Sortie:
1562 *   - Cy = 0 : pile non vide
1563 *   - SP reactualise
1564 *   - Pvar, C(0) = recherche courante
1565 *   - Cy = 1 : la pile etait vide
1566 * Abime: A(A), C(A), D1
1567 * Niveaux: 0
1568 * Historique:
1569 *   87/03/01: conception & codage
1570 *****
1571
1572 0A78D 1FF85F2     chpop  D1=(5) BP
1573 0A794 147        C=DAT1 A          C(A) := bas de la pile
1574 0A797 174        D1=D1+ 5         D1=(5) SP
1575 0A79A 143        A=DAT1 A          A(A) := sommet de la pile
1576 0A79D 8A2        ?A=C  A           pile vide ?
1577 0A7A0 00         RTNYES          oui : Cy := 1
1578 0A7A2 CC         A=A-1  A
1579 0A7A4 141        DAT1=A A          nouveau SP
1580 0A7A7 131        D1=A             D1 = SP
1581 0A7AA 1570       C=DAT1 P          C(0) := valeur depilee
1582 0A7AE 1F088F2     D1=(5) Pvar
1583 0A7B5 1550       DAT1=C P          Pvar := valeur depilee
1584 0A7B9 01         RTN             Cy = 0 (A=A-1)
1585
1586 0A7BB           END
```


BP	Abs	193935	#0002F58F	-	1199	1572				
BSERR	Ext			-	754					
CASEd	Ext			-	418					
=CASEe	Abs	41452	#0000A1EC	-	420					
CASEp	Ext			-	419					
CNFLCT	Ext			-	311					
CSETYP	Abs	194704	#0002F890	-	42	126	274			
CURREN	Ext			-	866					
CURRST	Ext			-	848					
D1=AVE	Ext			-	1539					
=ELSE2e	Abs	41824	#0000A360	-	737					
ELSE2p	Ext			-	736					
END2d	Ext			-	508					
=END2e	Abs	41588	#0000A274	-	510					
END2p	Ext			-	509					
ENDLe	Abs	41643	#0000A2AB	-	551	519				
ENDWe	Abs	41609	#0000A289	-	531	513				
EOLXCK	Ext			-	756					
EXPEX-	Ext			-	93	904				
EXPEXC	Ext			-	1082					
FINDA	Ext			-	457	1301				
FIXp	Ext			-	580	657				
IF2d	Ext			-	715					
=IF2e	Abs	41797	#0000A345	-	717					
IF2p	Ext			-	716					
IOFNDO	Ext			-	807	1224				
LDCSPC	Ext			-	1053	1078				
=LEAVEe	Abs	41479	#0000A207	-	441					
=LOOPE	Abs	41712	#0000A2F0	-	633					
LV10	Abs	41547	#0000A24B	-	478	459				
LV20	Abs	41545	#0000A249	-	474	461				
LV30	Abs	41543	#0000A247	-	470	463				
MEMBER	Ext			-	1295	1412				
MEMERR	Ext			-	1549					
MPOP1N	Ext			-	905					
MPOP2N	Ext			-	286					
MTHSTK	Ext			-	1080					
NOARGd	Ext			-	439	631	735			
NOARGp	Ext			-	440	632				
NXTSTM	Ext			-	540					
Nxtstm	Abs	41783	#0000A337	-	685	677	719			
OBCOLL	Ext			-	1205					
POP1S	Ext			-	106	316				
POPUPD	Ext			-	443	969				
PRGMEN	Ext			-	1216					
PSHGSE	Ext			-	939					
PgmRun	Ext			-	804	830	1218			
Pred<	Abs	1	#00000001	-	58	202	372			
Pred=	Abs	2	#00000002	-	59	190	197	202	364	
Pred>	Abs	4	#00000004	-	60	197	384			
Pred?	Abs	8	#00000008	-	61					
Pvar	Abs	194688	#0002F880	-	1200	1210	1316	1349	1542	1582
=REPEAE	Abs	41712	#0000A2F0	-	634					
RUNRT1	Ext			-	882					
S-R0-1	Ext			-	320					
S-R0-3	Ext			-	164	171				
SAUVDO	Abs	194705	#0002F891	-	41	1013	1033			

=SELECe	Abs	40868	#00009FA4	-	89					
SELECP	Ext			-	88					
SL010	Abs	40902	#00009FC6	-	106					
SL020	Abs	40940	#00009FEC	-	119	99	102			
SL030	Abs	40950	#00009FF6	-	126	115				
SL100	Abs	40961	#0000A001	-	133	221				
SL150	Abs	41011	#0000A033	-	154	149	220			
SL200	Abs	41076	#0000A074	-	182	160				
SL250	Abs	41104	#0000A090	-	197	186				
SL290	Abs	41132	#0000A0AC	-	207	199				
SL300	Abs	41139	#0000A0B3	-	214	176	192	204		
SL900	Abs	41165	#0000A0CD	-	230	150	177	193	205	
SP	Abs	193940	#0002F594	-	1198	1535				
STDD	Ext			-	87	579	656			
STMTD1	Ext			-	271	401	663	673		
STMTR0	Ext			-	108					
STRTST	Ext			-	354					
STRUCe	Abs	41834	#0000A36A	-	752	465	813	816	998	1106 1381
					1505					
STRUCt	Abs	9	#00000009	-	33	445	935	971		
STRUce	Abs	42184	#0000A4C8	-	998	970	974			
STRuce	Abs	42280	#0000A528	-	1106	1237				
STruce	Abs	42822	#0000A746	-	1505	1477	1483			
Struce	Abs	42606	#0000A66E	-	1381	1373				
TBLJMC	Ext			-	1319					
TKSCN7	Ext			-	1232					
TRACEM	Ext			-	832					
TRFROM	Ext			-	876					
TRTO+	Ext			-	879					
Trace	Ext			-	820					
=UNTILe	Abs	41730	#0000A302	-	658					
=WHILEe	Abs	41669	#0000A2C5	-	581					
WHL10	Abs	41692	#0000A2DC	-	595	584				
add<	Abs	41380	#0000A1A4	-	372	338	382			
add=	Abs	41361	#0000A191	-	364	337				
add>	Abs	41405	#0000A1BD	-	384	328	370			
bSTMT	Ext			-	806	1223				
chC ES	Abs	6	#00000006	-	50	136				
chEI	Abs	3	#00000003	-	47	738				
chEI-E	Abs	7	#00000007	-	51	1337				
chEL	Abs	1	#00000001	-	45					
chES	Abs	4	#00000004	-	48					
chES-C	Abs	8	#00000008	-	52	423	1335			
chEW	Abs	0	#00000000	-	44	595				
chE EI	Abs	5	#00000005	-	49	720				
chL/EL	Abs	10	#0000000A	-	54					
chL/EW	Abs	9	#00000009	-	53	481				
chL/U	Abs	11	#0000000B	-	55					
chU	Abs	2	#00000002	-	46					
che010	Abs	42318	#0000A54E	-	1212	1358	1443	1503		
che020	Abs	42350	#0000A56E	-	1227	1219				
che022	Abs	42389	#0000A595	-	1260	1258				
che025	Abs	42434	#0000A5C2	-	1297	1278				
che099	Abs	42479	#0000A5EF	-	1319	1430				
che100	Abs	42574	#0000A64E	-	1363	1320				
che110	Abs	42769	#0000A711	-	1466	1321	1389	1399	1438	
che120	Abs	42595	#0000A663	-	1378	1322				

che130	Abs	42765	#0000A70D	-	1458	1323				
che140	Abs	42767	#0000A70F	-	1462	1324				
che150	Abs	42610	#0000A672	-	1385	1325				
che160	Abs	42631	#0000A687	-	1395	1326				
che170	Abs	42754	#0000A702	-	1447	1327				
che175	Abs	42739	#0000A6F3	-	1440	1413	1424	1427	1436	1449
che180	Abs	42723	#0000A6E3	-	1434	1328				
che190	Abs	42651	#0000A69B	-	1406	1329				
che195	Abs	42713	#0000A6D9	-	1428	1416				
che1a0	Abs	42649	#0000A699	-	1405	1330				
che1b0	Abs	42647	#0000A697	-	1404	1331				
che200	Abs	42538	#0000A62A	-	1341	1303	1336	1338		
che210	Abs	42536	#0000A628	-	1340	1305				
che220	Abs	42534	#0000A626	-	1339	1307				
che230	Abs	42528	#0000A620	-	1337	1309				
che240	Abs	42522	#0000A61A	-	1335	1311				
che299	Abs	42627	#0000A683	-	1393	1380	1387	1397		
che300	Abs	42800	#0000A730	-	1498	1374	1393			
che400	Abs	42826	#0000A74A	-	1510	1379	1386	1396	1435	1448
che900	Abs	42559	#0000A63F	-	1355	1259	1296			
cher	Abs	42301	#0000A53D	-	1208					
cherche	Abs	42284	#0000A52C	-	1202	137	424	485	596	721 739
chpop	Abs	42893	#0000A78D	-	1572	1500				
chpush	Abs	42836	#0000A754	-	1535	1347				
cmpcmp	Abs	41249	#0000A121	-	306	294				
cmpfin	Abs	41427	#0000A1D3	-	397	301	396			
cmpnum	Abs	41209	#0000A0F9	-	285					
cmps00	Abs	41314	#0000A162	-	336	326				
cmps10	Abs	41322	#0000A16A	-	340	330				
cmps20	Abs	41332	#0000A174	-	351	332	341			
cmps30	Abs	41375	#0000A19F	-	369	363				
cmps50	Abs	41394	#0000A1B2	-	379	357				
cmps90	Abs	41415	#0000A1C7	-	393	367	375			
cmpstr	Abs	41256	#0000A128	-	315	281				
compar	Abs	41173	#0000A0D5	-	261	175	191	198	203	
eol	Abs	42265	#0000A519	-	1099	134	422	793	1357	1442 1502
eolxck	Abs	41849	#0000A379	-	756	148	512	1372	1476	
eval	Abs	42036	#0000A434	-	904	533	583	676	718	
eval10	Abs	42069	#0000A455	-	916	908				
evalMS	Abs	42239	#0000A4FF	-	1078	169	182	201	208	
fBASIC	Ext			-	854					
go	Abs	41856	#0000A380	-	790	538	554	683		
go10	Abs	41907	#0000A3B3	-	820	805				
goeol	Abs	41863	#0000A387	-	793	231	425	486	597	722 740
id	Abs	225	#000000E1	-	31	995	1256			
memerr	Abs	42886	#0000A786	-	1549	1541				
nxtstm	Abs	41636	#0000A2A4	-	540	520	534	590	636	685
oFTYPh	Ext			-	851					
pop	Abs	42106	#0000A47A	-	968	532	552	669		
push	Abs	42076	#0000A45C	-	935	537	553	589	635	682
qENDI	Abs	2	#00000002	-	39	1388	1425			
qENDL	Abs	0	#00000000	-	37	517				
qENDS	Abs	1	#00000001	-	38	1398	1422	1437		
recadr	Abs	42188	#0000A4CC	-	1013	133	230	484	536	588 681
				-	790	1355	1363	1440	1473	1498
runrt1	Abs	42029	#0000A42D	-	882	821	831	844	856	871
s1=2	Abs	0	#00000000	-	258	323	331	362		

s1>2	Abs	1	#00000001	-	259	322	342	369				
s1100	Abs	41161	#0000A0C9	-	221	217						
s1900	Abs	41007	#0000A02F	-	150	143						
stoMS	Abs	42221	#0000A4ED	-	1052	107	119					
stoadr	Abs	42203	#0000A4DB	-	1032	91	138	442	582	985	1241	
struce	Abs	41539	#0000A243	-	465	444	448					
tCASE	Abs	98	#00000062	-	27	1292	1395	1434				
tCOMMA	Ext			-	215							
tELSE2	Abs	100	#00000064	-	29	1293	1385	1447				
tEND2	Abs	66	#00000042	-	19	140	1276	1414				
tIF2	Abs	99	#00000063	-	28	1290	1308					
tLEAVE	Abs	70	#00000046	-	23							
tLOOP	Abs	96	#00000060	-	25	460	551	1287	1304			
tRELOP	Ext			-	158							
tREPEAT	Abs	68	#00000044	-	21	462	668	1288	1306			
tSELECT	Abs	97	#00000061	-	26	1289	1310					
tTO	Ext			-	184							
tUNTIL	Abs	69	#00000045	-	22	1291	1378					
tWHILE	Abs	67	#00000043	-	20	458	531	1286	1302			
tXWORD	Ext			-	995	1231						
uTEST	Ext			-	298							

Input Parameters

Source file name is struc2.a
Listing file name is struc2.l
Object file name is struc2.o
Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE SWAP
2
3          * *****
4          * Variable swap module
5          *
6          * Keyword(s):
7          *   VARSWAP
8          * Author(s):
9          * History:
10         *   Originally created by J.J. Moreau, JPC #31, Feb.86.
11         *   Rewritten at some stage by PPC-Paris. Author(s) unknown.
12         *
13         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14         * *****
15
16 0A7BB          ABS      #A7BB
17
18         * *****
19         * VARSWAP entry
20         * *****
21 0A7BB 00000          REL(5) =STDD
22 0A7C0 00000          REL(5) =VARSWp
23 0A7C5 8F00000 =VARSWe GOSBVL =SVTRC
24 0A7CC 8F00000          GOSBVL =EXPEX-
25 0A7D3 3400000          LC(5) =LEXPTR
26 0A7DA 7550          GOSUB dest+
27 0A7DE 161          D0=D0+ 2
28 0A7E1 8F00000          GOSBVL =SVTRC
29 0A7E8 8F00000          GOSBVL =EXPEXC
30 0A7EF 3400000          LC(5) =LDCSPC
31 0A7F6 7930          GOSUB dest+
32 0A7FA 1F00000          D1=(5) =LDCSPC
33 0A801 7610          GOSUB store
34 0A805 7A40          GOSUB swap
35 0A809 1F00000          D1=(5) =LEXPTR
36 0A810 7700          GOSUB store
37 0A814 8D00000          GOVLNG =NXTSTM
38
39 0A81B 147          store C=DAT1 A
40 0A81E 135          D1=C
41 0A821 8F00000          GOSBVL =AVE=D1
42 0A828 1537          A=DAT1 W
43 0A82C 8D00000          GOVLNG =STORE
44
45 0A833 137          dest+ CD1EX
46 0A836 145          DAT1=C A
47 0A839 135          D1=C
48 0A83C 8F00000          GOSBVL =DEST
49 0A843 136          CD0EX
50 0A846 06          RSTK=C
51 0A848 7700          GOSUB swap
52 0A84C 07          C=RSTK
53 0A84E 134          D0=C
54 0A851 01          RTN
55
56 0A853 1B00000 swap D0=(5) =S-R0-0
57 0A85A 1F00000          D1=(5) =TRFMBF
```

```
58 0A861 1527      A=DAT0 W
59 0A865 1577      C=DAT1 W
60 0A869 1547      DAT0=C W
61 0A86D 1517      DAT1=A W
62 0A871 16F       D0=D0+ 16
63 0A874 17F       D1=D1+ 16
64 0A877 1527      A=DAT0 W
65 0A87B 1577      C=DAT1 W
66 0A87F 1547      DAT0=C W
67 0A883 1517      DAT1=A W
68 0A887 01        RTN
69
70 0A889           END
```

AVE=D1	Ext	-	41		
DEST	Ext	-	48		
EXPEX-	Ext	-	24		
EXPEXC	Ext	-	29		
LDCSPC	Ext	-	30	32	
LEXPTR	Ext	-	25	35	
NXTSTM	Ext	-	37		
S-R0-0	Ext	-	56		
STDD	Ext	-	21		
STORE	Ext	-	43		
SVTRC	Ext	-	23	28	
TRFMBF	Ext	-	57		
=VARSWe	Abs	42949 #0000A7C5	-	23	
VARSWp	Ext	-	22		
dest+	Abs	43059 #0000A833	-	45	26 31
store	Abs	43035 #0000A81B	-	39	33 36
swap	Abs	43091 #0000A853	-	56	34 51

Input Parameters

Source file name is swap.a

Listing file name is swap.l

Object file name is swap.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE SHRINK
2
3          * *****
4          * File shrink module
5          *
6          * Keyword(s):
7          *   SHRINK
8          * Author(s):
9          *   Jean-Jacques Moreau
10         * History:
11         *   Published in JPC #35
12         *
13         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14         * *****
15
16 0A889          ABS      #A889
17
18          sBADRC EQU    8
19
20         * *****
21         * SHRINK entry
22         * *****
23 0A889 00000          REL(5) =SHRIND
24 0A88E 00000          REL(5) =SHRINp
25 0A893          =SHRINe
26 0A893 8F00000      GOSBVL =FSPECx
27 0A89A 590          GONC   SHK00
28 0A89D 8D00000      Mferr  GOVLNG =MFERR
29
30 0A8A4 8F00000      SHK00  GOSBVL =FINDF+
31 0A8AB 41F          GOC    Mferr
32 0A8AE 137          CD1EX
33 0A8B1 10B          R3=C
34 0A8B4 135          D1=C
35 0A8B7 8F00000      GOSBVL =RDHDR1
36 0A8BE 112          A=R2
37 0A8C1 CC          A=A-1  A
38 0A8C3 CC          A=A-1  A
39 0A8C5 490          GOC    SHK10
40 0A8C8 3100         LC(2) =eFTYPE
41 0A8CC 50D          GONC   Mferr      (B.E.T.)
42
43 0A8CF 113          SHK10  A=R3
44 0A8D2 D2          C=0    A
45 0A8D4 CE          C=C-1  A
46 0A8D6 109          R1=C
47 0A8D9 7840        GOSUB  POSTXT
48 0A8DD 570          GONC   SHK20
49 0A8E0 8AE          ?C#0   A
50 0A8E3 AB          GOYES  Mferr
51
52 0A8E5 868          SHK20  ?ST=0  =sBADRC
53 0A8E8 90          GOYES  SHK30
54 0A8EA 3100         LC(2) =eEOFIL
55 0A8EE 5EA          GONC   Mferr      (B.E.T.)
56
57 0A8F1 DB          SHK30  C=D    A
```

```

58 0A8F3 DA          A=C    A
59 0A8F5 D2          C=0    A
60 0A8F7 CE          C=C-1  A
61 0A8F9 15F3       C=DAT1 4
62 0A8FD E6          C=C+1  A
63 0A8FF 550         GONC   SHK60
64 0A902 173         D1=D1+ 4
65 0A905 137         SHK60  CD1EX
66 0A908 D5          B=C    A
67 0A90A E8          B=B-A  A
68 0A90C 11B         C=R3
69 0A90F 8F00000    GOSBVL =MGOSUB
70 0A916 00000       CON(5) =MVMEM+
71 0A91B 418         GOC    Mferr
72 0A91E 8D00000    GOVLNG =NXTSTM
73
74 0A925 131         POSTXT D1=A
75 0A928 170         D1=D1+ =oFTYPh
76 0A92B 173         D1=D1+ 4      (=oFLAGh) - (=oFTYPh)
77 0A92E 8F00000    GOSBVL =FILSK+
78 0A935 D7          D=C    A
79 0A937 174         D1=D1+ 5
80 0A93A 137         CD1EX
81 0A93D 111         A=R1
82 0A940 7010       GOSUB  FRCRDr
83 0A944 500         RTNNC
84 0A947 D2          C=0    A
85 0A949 868         ?ST=0  =sBADRC
86 0A94C 00          RTNYES
87 0A94E 3100       LC(2)  =eEOFIL
88 0A952 02          RTNSC
89
90 0A954 101         FRCRDr R1=A
91 0A957 AF0         A=0    W
92 0A95A CC          A=A-1  A
93 0A95C 100         R0=A
94 0A95F 840         ST=0   =sEOF
95 0A962 848         ST=0   =sBADRC
96 0A965 7120       FRCR10 GOSUB  PRSREC
97 0A969 400         RTNC
98 0A96C 134         D0=C
99 0A96F 110         A=R0
100 0A972 E4         A=A+1  A
101 0A974 100         R0=A
102 0A977 119         C=R1
103 0A97A 8A2         ?A=C   A
104 0A97D 75         GOYES  rtncc
105 0A97F 878         ?ST=1  =sBADRC
106 0A982 00          RTNYES
107 0A984 136         CD0EX
108 0A987 5DD         GONC   FRCR10      (B.E.T.)
109
110 0A98A D1         PRSREC B=0    A
111 0A98C 135         D1=C
112 0A98F 8BF         ?C>=D  A
113 0A992 44         GOYES  PRSR10
114 0A994 173         D1=D1+ 4

```

```
115 0A997 137          CD1EX
116 0A99A 8B7          ?C>D  A
117 0A99D E3           GOYES  PRSR20
118 0A99F 15B3         A=DAT1 4
119 0A9A3 8F00000     GOSBVL =SWPBYT
120 0A9AA 23           P=      3
121 0A9AC A98          B=A     WP
122 0A9AF D9           C=B     A
123 0A9B1 B15          B=B+1  WP
124 0A9B4 20           P=      0
125 0A9B6 400          RTNC
126 0A9B9 DD           BCEX   A
127 0A9BB 81E          CSRB
128 0A9BE E6           C=C+1  A
129 0A9C0 C6           C=C+C  A
130 0A9C2 C6           C=C+C  A
131 0A9C4 133          AD1EX
132 0A9C7 131          D1=A
133 0A9CA C2           C=A+C  A
134 0A9CC 8BB          ?C<=D  A
135 0A9CF 50           GOYES  rtncc
136 0A9D1 858          ST=1   =sBADRC
137 0A9D4 03           rtncc  RTNCC
138
139 0A9D6 850          PRSR10 ST=1   =sEOF
140 0A9D9 02           RTNSC
141
142 0A9DB CD           PRSR20 B=B-1  A
143 0A9DD 858          ST=1   =sBADRC
144 0A9E0 02           RTNSC
145
146 0A9E2              END
```

FILSK+	Ext		-	77					
FINDF+	Ext		-	30					
FRCR10	Abs	43365	#0000A965	-	96	108			
FRCRDr	Abs	43348	#0000A954	-	90	82			
FSPECx	Ext		-	26					
MFERR	Ext		-	28					
MGOSUB	Ext		-	69					
MVMEM+	Ext		-	70					
Mferr	Abs	43165	#0000A89D	-	28	31	41	50	55 71
NXTSTM	Ext		-	72					
POSTXT	Abs	43301	#0000A925	-	74	47			
PRSR10	Abs	43478	#0000A9D6	-	139	113			
PRSR20	Abs	43483	#0000A9DB	-	142	117			
PRSREC	Abs	43402	#0000A98A	-	110	96			
RDHDR1	Ext		-	35					
SHK00	Abs	43172	#0000A8A4	-	30	27			
SHK10	Abs	43215	#0000A8CF	-	43	39			
SHK20	Abs	43237	#0000A8E5	-	52	48			
SHK30	Abs	43249	#0000A8F1	-	57	53			
SHK60	Abs	43269	#0000A905	-	65	63			
SHRINd	Ext		-	23					
=SHRINe	Abs	43155	#0000A893	-	25				
SHRINp	Ext		-	24					
SWPBYT	Ext		-	119					
eEOFIL	Ext		-	54	87				
eFTYPE	Ext		-	40					
oFPh	Ext		-	75					
rtnc	Abs	43476	#0000A9D4	-	137	104	135		
sBADRC	Abs	8	#00000008	-	18	52	85	95	105 136 143
sEOF	Ext		-	94	139				

Input Parameters

Source file name is shrink.a
Listing file name is shrink.l
Object file name is shrink.o
Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE SYSOP
2
3          * *****
4          * Opcode function module
5          *
6          * Keyword(s):
7          *   NEXTOP$, OPCODE$
8          * Author(s):
9          *   Jean-Jacques Dhenin, Pierre David, Janick Taillandier
10         * History:
11         *   Based on an unpublished source file (private
12         *     communication) and updated from actual JPC LEX
13         *     code disassembly
14         *   86/./...: JJD conception & codage
15         *   87/05/...: PD & JT documentation,
16         *                   add HP28C new opcodes (pc=(a), rsi)
17         *   88/??/???: PD & JT bug fixed in rev D (swap of the pc=(a)
18         *                   and rsi opcodes...)
19         *
20         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
21         * *****
22
23 0A9E2          ABS      #A9E2
24
25         *
26         * Voir OPCout pour les details sur ces EQU
27         *
28         scrтч EQU      #2F901          =SCRТCH
29         opcode EQU    (scrтч)+00      (=SCRТCH)+00    6 octets
30         modifi EQU    (scrтч)+12      (=SCRТCH)+12   16 octets
31         branch EQU   (scrтч)+44      (=SCRТCH)+44    6 octets
32         adress EQU   (scrтч)+56      (=SCRТCH)+56    5 octets
33
34         sysedt EQU   10              No de flag pour retour dans OPCODE$
35
36
37         *****
38         * INIT
39         *
40         * But: analyser la chaine au sommet de la M.S. et
41         *   en extraire une adresse valide.
42         * Entree:
43         *   - B(A) = adresse trouvee (sortie de ADDRCK)
44         * Sortie:
45         *   - R3(A) = adresse trouvee
46         *   - R2(W) = 16 quartets a l'adresse trouvee
47         *   - D(3-0) = 4 quartets a l'adresse trouvee + 16
48         *   - ST(1) = 0
49         *   - D1 actualise
50         * Utilise: A, B, C, D(A), R2, R3, ST(1), D1
51         * Niveaux: 0
52         * Historique:
53         *   86/./...: JJD conception & codage
54         *   87/05/13: PD ajout de documentation
55         *   87/05/20: PD & JT retrait de ADDRCK et renommage en INIT
56         * *****
57
```

```

58          * entry point for sysedit
59 0A9E2      =SYSINI
60 0A9E2 D4    INIT      A=B      A
61 0A9E4 103   R3=A          R3 = adresse trouvee
62 0A9E7 841   ST=0      1
63 0A9EA 132   AD0EX
64 0A9ED 1567  C=DAT0 W      C(W) := 16 q. pointes
65 0A9F1 10A   R2=C          R2 := 16 quartets pointes
66 0A9F4 16E   D0=D0+ 15    D0 = 15 quartets plus loin
67 0A9F7 146   C=DAT0 A
68 0A9FA D7    D=C      A      D(A) := 5 q. suivants
69 0A9FC 130   D0=A          D0 restaure
70 0A9FF 01    RTN
71
72
73          * *****
74          * NEXTOP$ entry
75          * *****
76 0AA01 411   NIBHEX 411
77 0AA04      =NEXTOe
78 0AA04 8F00000 GOSBVL =ADDRCK
79 0AA0B 73DF   GOSUB  INIT
80 0AA0F 7740   GOSUB  GETNXT
81 0AA13 DA     A=C      A
82 0AA15 AF2   C=0      W
83 0AA18 24    P=       4
84 0AA1A 80FF  CPEX    15
85 0AA1E AF7   D=C      W
86 0AA21 8F00000 GOSBVL =HEXASC
87 0AA28 B47   D=D+1   S
88 0AA2B A47   D=D+D   S
89 0AA2E 813   DSLC
90 0AA31 DB    C=D      A
91 0AA33 8F00000 GOSBVL =STRHDR
92 0AA3A 80D0  P=C      0
93 0AA3E 0D    P=P-1
94 0AA40 AF4   A=B      W
95 0AA43 1511  DAT1=A  WP
96 0AA47 1CF   D1=D1- 16
97 0AA4A 20    P=       0
98 0AA4C 8F00000 GOSBVL =REV$
99 0AA53 8D00000 GOVLNG =EXPR
100
101          *****
102          * GETNXT
103          *
104          * But: trouver l'adresse de l'instruction suivante
105          * Entree:
106          *   - R2 = 16 quartets a l'adresse courante
107          *   - B(A) = adresse courante
108          * Sortie:
109          *   - C(A) := nouvelle adresse
110          *   - P = ?
111          * Utilise: A, C, P
112          * Appelle: -
113          * Niveaux: 0
114          * Historique:

```



```
115      * 86/././.: JJD conception & codage
116      * 87/05/13: PD ajout de documentation
117      * 87/05/14: PD renonce a documenter
118      * 87/05/14: PD ajout de PC=(A)
119      *****
120
121      * entry point for sysedit
122 0AA5A =SYSGNX
123 0AA5A GETNXT
124      *
125      * Ajout de PC=(A)
126      * Que le concepteur de ce code incomprehensible
127      * veuillez bien me pardonner de ces quelques lignes
128      * aussi triviales. Je demande humblement pardon...
129      *
130 0AA5A 112      A=R2      A(W) := 16 quartets
131 0AA5D 3480810  LCHEX  01808  PC=(A)
132 0AA64 8A6      ?A#C  A
133 0AA67 C0      GOYES  GETN05  pas PC=(A)
134 0AA69 24      P=      4      5 quartets de long
135 0AA6B 80CF     C=P      15     C(S) := 4
136 0AA6F 6390     GOTO   GETN99  Mille excuses encore...
137 0AA73      GETN05
138      *
139      * Fin des lignes triviales. J'ai le sentiment
140      * d'avoir commis un sacrilege...
141      *
142 0AA73 11A      C=R2      C(W) := 16 quartets
143 0AA76 DA      A=C      A      A(A) := 5 quartets pointes
144 0AA78 80D0     P=C      0      P = 1er quartet de l'inst.
145 0AA7C 3F111122 LCHEX  9A1A2233A4221111
      4A3322A1
      A9
146      *
147      * 1er q.  C(S)
148      * 0      9      Instructions a 2 ou 4 quartets
149      * 1      A      Instructions a 3, 4, 6 ou 7 quartets
150      * 2      1      Instructions P=
151      * 3      A      Instructions LCHEX
152      * 4      2      Instructions a 3 quartets
153      * 5      2      Instructions a 3 quartets
154      * 6      3      Instructions a 4 ou 5 quartets
155      * 7      3      Instructions a 4 quartets
156      * 8      A      Instructions a 3, 4, 6 ou 7 quartets
157      * 9      4      Instructions a 3 quartets
158      * A      2      Instructions a 3 quartets
159      * B      2      Instructions a 3 quartets
160      * C      1      Instructions a 2 quartets
161      * D      1      Instructions a 2 quartets
162      * E      1      Instructions a 2 quartets
163      * F      1      Instructions a 2 quartets
164      *
165
166      *
167      * Si l'instruction necessite un traitement special
168      * le bit de poids fort est mis a 1.
169      * Sinon, ce bit est a 0, et les trois bits restant
```

```

170          * codent <len - 1>
171          *
172 0AA8E A46          C=C+C  S          Necessite trait. special ?
173 0AA91 81E          CSRB
174 0AA94 5E6          GONC  GETN99  Non : bit poids fort = 0
175          *
176          * L'instruction necessite un traitement special
177          *
178 0AA97 ACA          A=C    S          A(S) = q. reconnaissance
179 0AA9A AF2          C=0    W
180 0AA9D 888          ?P#    8          Instructions 8...
181 0AAA0 11           GOYES  GETN10  Non
182          *
183          * Instructions 8???
184          * 3, 4, 6 ou 7 quartets
185          *
186 0AAA2 80C0         C=P    0          C(0) = 8
187 0AAA6 24           P=     4
188 0AAA8 966          ?C#A   B          Instruction = 80.. ?
189 0AAAAB 60          GOYES  GETN10
190          *
191          * Instruction de type 80...
192          *
193 0AAAD F4           ASR    A          8 disparait
194          *
195          * reste A(A) = 0???0
196          *
197 0AAAF 22           P=     2
198          *
199          * A(S) = quartet de reconnaissance
200          * P = 8, 4 ou 2
201          *
202 0AAB1 3410428      GETN10 LCHEX  82401
203 0AAB8 DE           CAEX   A
204 0AABA 80D1         P=C    1
205 0AABE 3102         LCHEX  20
206 0AAC2 C4           A=A+A  A
207 0AAC4 4B3         GOC    GETN20
208 0AAC7 37101100    LCHEX  10001101
209          01
209 0AAD1 C4           A=A+A  A
210 0AAD3 4C2         GOC    GETN20
211 0AAD6 3A431043    LCHEX  10001340134
212          10001
212 0AAE3 C4           A=A+A  A
213 0AAE5 4A1         GOC    GETN20
214 0AAE8 3C434322    LCHEX  200222223434
215          2222002
215 0AAF7 C4           A=A+A  A
216 0AAF9 460         GOC    GETN20
217 0AAFC 80CF        C=P    15
218 0AB00 A42         GETN20 C=C+A  S
219 0AB03 80DF        GETN99 P=C    15
220 0AB07 112         A=R2
221 0AB0A D9          C=B    A
222 0AB0C 570         GONC  GETN30
223 0AB0F 809         C+P+1

```

```
224 0AB12 2F          P=      15
225 0AB14 809        GETN30 C+P+1
226 0AB17 01         RTN
227
228 *****
229 * rsi
230 *
231 * But: traitez l'instruction RSI (opcode 80810)
232 * Historique:
233 *   87/05/15: PD conception & codage
234 *****
235
236 0AB19             rsi
237 *                 P=      0      (removed, rev D)
238 0AB19 35253594    LCASC  'ISR'
239 0AB21 1B109F2     D0=(5) opcode
240 0AB28 15C5        DAT0=C 6
241 0AB2C D2          C=0    A
242 0AB2E 303         LC(1)  3
243 0AB31 6320        GOTO   special
244
245 *****
246 * pca
247 *
248 * But: traitez l'instruction PC=(A) (opcode 808C)
249 * Historique:
250 *   87/05/15: PD conception & codage
251 *****
252
253 0AB35             pca
254 0AB35 20           P=      0      (added, rev D)
255 0AB37 3B0534D3    LCASC  ')A(=CP'
      821492
256 0AB45 1B109F2     D0=(5) opcode
257 0AB4C 15CB        DAT0=C 12
258 0AB50 D2          C=0    A
259 0AB52 306         LC(1)  6
260
261 0AB55             special
262 0AB55 108          R0=C
263 0AB58 D2          C=0    A
264 0AB5A 109          R1=C
265 0AB5D 10A          R2=C
266 0AB60 10B          R3=C
267
268 0AB63 8CBCA0      GOLONG out500
269
270 *****
271 * OPCODE$
272 *
273 * But: Obtenir la mnemonique de l'instruction
274 *   pointee par l'adresse indiquee.
275 * Entree:
276 *   - OPCODE$ : l'adresse est sur la M.S.
277 *   - getopc  :
278 *     - D1 = emplacement ou sera depose le resultat
279 *     - B(A) = adresse de l'instruction
```

```
280          * Sortie:
281          *   - l'instruction est sur la M.S.
282          *   - OPCODE$ : sortie par ADHEAD
283          *   - getopc : sortie par RTN en out999
284          *     R1 = ^ 1er caractere
285          *     D1 = ^ dernier caractere (cond. entree ADHEAD)
286          * Historique:
287          *   86/./...: JJD conception & codage
288          *   87/05/14: PD ajout de documentation
289          *   87/05/20: PD & JT ajout du point d'entree getopc (ST10)
290          *   ??/??/??: PD & JT ?? rsi and pc=(a) opcode swap bug fix
291          *****
292
293
294          * entry point for sysedit
295 0AB69      =SYSGOP
296 0AB69 85A  getopc ST=1   sysedt
297 0AB6C 6410 GOTO   OPC00
298
299 0AB70 64CF   pcA   GOTO   pca
300
301
302
303          * *****
304          * OPCODE$ entry
305          * *****
306 0AB74 411          NIBHEX 411
307 0AB77          =OPCODE
308 0AB77 8F00000    GOSBVL =ADDRCK
309 0AB7E 84A          ST=0   sysedt
310 0AB81          OPC00
311
312 0AB81 7D5E          GOSUB  INIT
313
314          *
315          * B(A) = adresse
316          * R2 = 16 quartets pointes par B(A)
317          * D(3-0) = 4 quartets pointes par B(A)+16
318          *
319 0AB85 71DE          GOSUB  GETNXT
320          *
321          * D0 = PC
322          * D1 = ^ M.S. derriere l'adresse
323          *
324 0AB89 136          CD0EX          C(A) := PC
325 0AB8C 1B00000    D0=(5) =FUNCD0
326 0AB93 144          DAT0=C A          FUNCD0 := PC
327 0AB96 164          D0=D0+ 5          D0=(5) =FUNCD1
328 0AB99 137          CD1EX          C(A) := MS
329 0AB9C 144          DAT0=C A          FUNCD1 := MS
330 0AB9F 136          CD0EX          D0 := ^ MS
331 0ABA2 20          P= 0
332          *
333          * Traitement des deux instructions speciales
334          *
335          ** bug fixed in rev D (swap of the pc=(a) and rsi opcodes...):
336 0ABA4 112          A=R2
```

```
337 0ABA7 3480810          LCHEX  01808
338 0ABAE 8A2              ?A=C   A
339 0ABB1 31                GOYES  rsI   fixed
340 0ABB3 33808C           LCHEX  C808
341 0ABB9 23                P=     3
342 0ABBB 916              ?A#C   WP
343 0ABBE A0                GOYES  opc001
344 0ABC0 6FAF              GOTO   pca   fixed
345 0ABC4 645F              rsI    GOTO   rsi   added
346                          ** end of bug fix
347 0ABC8                    opc001
348 0ABC8 20                P=     0
349                          *
350                          * Fin du traitement special des deux instructions
351                          *
352 0ABCA 331400            LCHEX  0041
353 0ABD0 AFD              BCEX   W
354 0ABD3 8F00000          GOSBVL =CSLW5   C(9-5) := adresse
355 0ABDA 109              R1=C   R1 (9-5) := adresse
356 0ABDD 844              ST=0   4
357 0ABE0 847              ST=0   7
358 0ABE3 7C93            GOSUB  OPC10
359
360 0ABE7 01222222          NIBHEX 01222222
361 0ABEF 3458679A          NIBHEX 3458679A
362 0ABF7 D8000000          NIBHEX D8000000
363 0ABFF 00000011          NIBHEX 00000011
364 0AC07 2005210C          NIBHEX 2005210C
365 0AC0F 06211004          NIBHEX 06211004
366 0AC17 21200721          NIBHEX 21200721
367 0AC1F 30152148          NIBHEX 30152148
368 0AC27 20F05038          NIBHEX 20F05038
369 0AC2F E060D0A1          NIBHEX E060D0A1
370 0AC37 7080AD80          NIBHEX 7080AD80
371 0AC3F 00AB9000          NIBHEX 00AB9000
372 0AC47 ACA050A0          NIBHEX ACA050A0
373 0AC4F B820A8F2          NIBHEX B820A8F2
374 0AC57 0021E022          NIBHEX 0021E022
375 0AC5F C41680DC          NIBHEX C41680DC
376 0AC67 06809113          NIBHEX 06809113
377 0AC6F 43101384          NIBHEX 43101384
378 0AC77 01384092          NIBHEX 01384092
379 0AC7F 84092840          NIBHEX 84092840
380 0AC87 60000001          NIBHEX 60000001
381 0AC8F 11111111          NIBHEX 11111111
382 0AC97 16223000          NIBHEX 16223000
383 0AC9F 7421004C          NIBHEX 7421004C
384 0ACA7 320C4433          NIBHEX 320C4433
385 0ACAF 0E725400          NIBHEX 0E725400
386 0ACB7 FB5500FB          NIBHEX FB5500FB
387 0ACBF D622262D          NIBHEX D622262D
388 0ACC7 43072BC2          NIBHEX 43072BC2
389 0ACCF 08AC430B          NIBHEX 08AC430B
390 0ACD7 10430CAA          NIBHEX 10430CAA
391 0ACDF C20F10C2          NIBHEX C20F10C2
392 0ACE7 06222220          NIBHEX 06222220
393 0ACEF 48031000          NIBHEX 48031000
```

394	0ACF7	64318085	NIBHEX	64318085
395	0ACFF	310186A1	NIBHEX	310186A1
396	0AD07	8187000A	NIBHEX	8187000A
397	0AD0F	98F01222	NIBHEX	98F01222
398	0AD17	22222345	NIBHEX	22222345
399	0AD1F	55589000	NIBHEX	55589000
400	0AD27	00000123	NIBHEX	00000123
401	0AD2F	32232232	NIBHEX	32232232
402	0AD37	4000A331	NIBHEX	4000A331
403	0AD3F	0083320D	NIBHEX	0083320D
404	0AD47	10030D80	NIBHEX	10030D80
405	0AD4F	04918415	NIBHEX	04918415
406	0AD57	B1C0060E	NIBHEX	B1C0060E
407	0AD5F	80070302	NIBHEX	80070302
408	0AD67	D0200000	NIBHEX	D0200000
409	0AD6F	00000000	NIBHEX	00000000
410	0AD77	000E024E	NIBHEX	000E024E
411	0AD7F	014E0FC0	NIBHEX	014E0FC0
412	0AD87	4E0A1229	NIBHEX	4E0A1229
413	0AD8F	41083C20	NIBHEX	41083C20
414	0AD97	602D2058	NIBHEX	602D2058
415	0AD9F	0F200C24	NIBHEX	0F200C24
416	0ADA7	126A0835	NIBHEX	126A0835
417	0ADAF	0B0400FE	NIBHEX	0B0400FE
418	0ADB7	00C52813	NIBHEX	00C52813
419	0ADB7	01700614	NIBHEX	01700614
420	0ADC7	00A14000	NIBHEX	00A14000
421	0ADC7	0000A145	NIBHEX	0000A145
422	0ADD7	32520500	NIBHEX	32520500
423	0ADDF	832500C4	NIBHEX	832500C4
424	0ADE7	458805C4	NIBHEX	458805C4
425	0ADEF	88066588	NIBHEX	88066588
426	0ADF7	47E48848	NIBHEX	47E48848
427	0ADFF	62404964	NIBHEX	62404964
428	0AE07	40415181	NIBHEX	40415181
429	0AE0F	30131028	NIBHEX	30131028
430	0AE17	C1024C50	NIBHEX	C1024C50
431	0AE1F	08C5004C	NIBHEX	08C5004C
432	0AE27	81301310	NIBHEX	81301310
433	0AE2F	20D1020E	NIBHEX	20D1020E
434	0AE37	1028D102	NIBHEX	1028D102
435	0AE3F	8EA122C2	NIBHEX	8EA122C2
436	0AE47	A8A8D400	NIBHEX	A8A8D400
437	0AE4F	38E088BA	NIBHEX	38E088BA
438	0AE57	F4004810	NIBHEX	F4004810
439	0AE5F	18132250	NIBHEX	18132250
440	0AE67	52840528	NIBHEX	52840528
441	0AE6F	40528422	NIBHEX	40528422
442	0AE77	28481302	NIBHEX	28481302
443	0AE7F	44008400	NIBHEX	44008400
444	0AE87	28400284	NIBHEX	28400284
445	0AE8F	80204101	NIBHEX	80204101
446	0AE97	81326503	NIBHEX	81326503
447	0AE9F	28424284	NIBHEX	28424284
448	0AEA7	03284032	NIBHEX	03284032
449	0AEAF	84813011	NIBHEX	84813011
450	0AEB7	00414008	NIBHEX	00414008

```

451 0AEBF 14002A42      NIBHEX 14002A42
452 0AEC7 22A4         NIBHEX 22A4
453
454 0AECB 112          OPC20  A=R2
455 0AECE 32147        LCHEX  741
456 0AED3 986          ?C<A  P
457 0AED6 82           GOYES  OPC22
458 0AED8 908          ?A=0  P
459 0AEDB 71           GOYES  OPC21
460 0AEDD 0E32        C=A&C  X
461 0AEE1 10A         R2=C
462 0AEE4 A24         A=A+A  XS
463 0AEE7 BB0         ASL    X
464 0AEEA F4          ASR    A
465 0AEEC 456         GOC    OPC31
466 0AEEF 5E1         GONC   OPC23
467 0AEF2 B24         OPC21  A=A+1  XS
468 0AEF5 415         GOC    OPC26
469 0AEF8 A2C         A=A-1  XS
470 0AEFB 521         GONC   OPC23
471 0AEFE F0          OPC22  ASL    A
472 0AF00 171         D1=D1+ 2
473 0AF03 9AA         ?C>=A  XS
474 0AF06 80          GOYES  OPC23
475 0AF08 17F         D1=D1+ 16
476 0AF0B 179         D1=D1+ 10
477 0AF0E D6          OPC23  C=A    A
478 0AF10 80D2        P=C    2
479 0AF14 3F70BC56    LCHEX  0665CB070665CB07
      6070BC56
      60
480 0AF26 C6          C=C+C  A
481 0AF28 80D0        P=C    0
482 0AF2C 3B750585    LCASC  'BMSXPW'
      35D424
483 0AF3A 480         GOC    OPC25
484 0AF3D 22          P=     2
485 0AF3F 3100        LCHEX  00
486 0AF43 20          OPC25  P=     0
487 0AF45 D5          B=C    A
488 0AF47 F4          OPC26  ASR    A
489 0AF49 F4          ASR    A
490 0AF4B F4          ASR    A
491 0AF4D 03          RTNCC
492 0AF4F 112         OPC30  A=R2
493 0AF52 331313      OPC31  LCHEX  3131
494 0AF58 902         ?C=A  P
495 0AF5B 41          GOYES  OPC33
496 0AF5D A2E         C=C-1  XS
497 0AF60 A04         A=A+A  P
498 0AF63 440         GOC    OPC32
499 0AF66 F0          ASL    A
500 0AF68 96C         OPC32  ?A#0  B
501 0AF6B 40          GOYES  OPC33
502 0AF6D F4          ASR    A
503 0AF6F 05          OPC33  SETDEC
504 0AF71 A22         C=C+A  XS

```

```

505 0AF74 450          GOC    OPC34
506 0AF77 AE2          C=0    B
507 0AF7A 04          OPC34  SETHEX
508 0AF7C D5          B=C    A
509 0AF7E 854         ST=1   4
510 0AF81 03          RTNCC
511 0AF83 07          OPC10  C=RSTK
512 0AF85 135         D1=C
513 0AF88 14F         OPC11  C=DAT1 B
514 0AF8B 170         D1=D1+ 1
515 0AF8E 7400        GOSUB  TBLJMc
516 0AF92 65FF        GOTO   OPC11
517 0AF96 8F00000     TBLJMc GOSBVL =TBLJMC
518 0AF9D A20         REL(3) OPC40
519 0AFA0 B2F         REL(3) OPC20
520 0AFA3 683         REL(3) OPC140
521 0AFA6 361         REL(3) OPC100
522 0AFA9 150         REL(3) OPC50
523 0AFAC 3AF         REL(3) OPC30
524 0AFAF A50         REL(3) OPC60
525 0AFB2 2A0         REL(3) OPC72
526 0AFB5 D90         REL(3) OPC71
527 0AFB8 890         REL(3) OPC70
528 0AFBB AA0         REL(3) OPC80
529 0AFBE E11         REL(3) OPC90
530 0AFC1 9E1         REL(3) OPC110
531 0AFC4 052         REL(3) OPC120
532
533 0AFC7 D6          OPC40  C=A    A
534 0AFC9 A87         D=C    P
535 0AFCC 1577        C=DAT1 W
536 0AFD0 BF2         CSL    W
537 0AFD3 17E         D1=D1+ 15
538 0AFD6 816         OPC41  CSRC
539 0AFD9 A0C         A=A-1 P
540 0AFDC 59F         GONC   OPC41
541 0AFDF F4          ASR    A
542 0AFE1 D2          C=0    A
543 0AFE3 14F         OPC42  C=DAT1 B
544 0AFE6 171         D1=D1+ 2
545 0AFE9 A4E         C=C-1 S
546 0AFEC 400         RTNC
547 0AFEF 133         AD1EX
548 0AFF2 CA          A=A+C A
549 0AFF4 133         AD1EX
550 0AFF7 5BE         GONC   OPC42
551 0AFFA 338554      OPC50  LCHEX  4558
552 0B000 183         D0=D0- 4
553 0B003 15C3        DAT0=C 4
554 0B007 03         RTNCC
555
556 0B009 112         OPC60  A=R2
557 0B00C AC9         C=B    S
558 0B00F 877         ?ST=1 7
559 0B012 F2         GOYES  OPC62
560 0B014 7AA4        GOSUB  OPC170
561 0B018 A4D         B=B-1 S

```


562	0B01B	A4D		B=B-1	S
563	0B01E	AC9		C=B	S
564	0B021	12A		CR2EX	
565	0B024	F7		DSR	A
566	0B026	AEB		C=D	B
567	0B029	816		CSRC	
568	0B02C	816		CSRC	
569	0B02F	BF0	OPC61	ASL	W
570	0B032	A8A		A=C	P
571	0B035	BF6		CSR	W
572	0B038	A4D		B=B-1	S
573	0B03B	53F		GONC	OPC61
574	0B03E	11A		C=R2	
575	0B041	7984	OPC62	GOSUB	OPC172
576	0B045	BF4		ASR	W
577	0B048	A4E		C=C-1	S
578	0B04B	55F		GONC	OPC62
579	0B04E	03		RTNCC	
580	0B050	21	OPC70	P=	1
581	0B052	0C	OPC71	P=P+1	
582	0B054	DB	OPC72	C=D	A
583	0B056	BF6	OPC73	CSR	W
584	0B059	0D		P=P-1	
585	0B05B	5AF		GONC	OPC73
586	0B05E	DA		A=C	A
587	0B060	20		P=	0
588	0B062	4ED		GOC	OPC62
589	0B065	11A	OPC80	C=R2	
590	0B068	DA		A=C	A
591	0B06A	80D2		P=C	2
592	0B06E	3F410FEB		LCHEX	5236789CDABEF014
		ADC98763			
		25			
593	0B080	AAA		A=C	XS
594	0B083	A26		C=C+C	XS
595	0B086	80D2		P=C	2
596	0B08A	3F1A0C1C		LCHEX	A0C1C0A1A0C1C0A1
		0A1A0C1C			
		0A			
597	0B09C	D7		D=C	A
598	0B09E	20		P=	0
599	0B0A0	3224C		LCHEX	C42
600	0B0A5	550		GONC	OPC81
601	0B0A8	AE5		B=C	B
602	0B0AB	CE	OPC81	C=C-1	A
603	0B0AD	966		?C#A	B
604	0B0B0	50		GOYES	OPC82
605	0B0B2	854		ST=1	4
606	0B0B5	0E22	OPC82	C=A&C	XS
607	0B0B9	80D2		P=C	2
608	0B0BD	3F0CF081		LCHEX	6F300FC86F180FC0
		F68CF003			
		F6			
609	0B0CF	F2		CSL	A
610	0B0D1	111		A=R1	
611	0B0D4	0EF6		A=A&C	A
612	0B0D8	6421		GOTO	OPC116

```

613 0B0DC 11A      OPC90  C=R2
614 0B0DF 80D2      P=C    2
615 0B0E3 A26      C=C+C  XS
616 0B0E6 3F210321    LCHEX  3412301234123012
      43210321
      43
617 0B0F8 21      P=     1
618 0B0FA 30A      LCHEX  A
619 0B0FD 550      GONC   OPC91
620 0B100 30C      LCHEX  C
621 0B103 20      OPC91  P=     0
622 0B105 D7      D=C    A
623 0B107 03      RTNCC
624 0B109 854      OPC100 ST=1   4
625 0B10C 31C0     LCHEX  0C
626 0B110 0E02     C=A&C  P
627 0B114 B0A      A=A-C  P
628 0B117 814      ASRC
629 0B11A 143      A=DAT1 A
630 0B11D C6      C=C+C  A
631 0B11F C6      C=C+C  A
632 0B121 F2      CSL    A
633 0B123 27      P=     7
634 0B125 172     D1=D1+ 3
635 0B128 36804040 LCHEX  0040408
      0
636 0B131 A2E      C=C-1  XS
637 0B134 4A2      GOC    OPC101
638 0B137 174     D1=D1+ 5
639 0B13A 3480044   LCHEX  44008
640 0B141 A2E      C=C-1  XS
641 0B144 4A1      GOC    OPC101
642 0B147 179     D1=D1+ 10
643 0B14A 31C4     LCHEX  4C
644 0B14E A2E      C=C-1  XS
645 0B151 5D0      GONC   OPC101
646 0B154 1C4     D1=D1- 5
647 0B157 35C40044 LCHEX  44004C
648 0B15F ADA      OPC101 A=C    M
649 0B162 BF2      OPC102 CSL    W
650 0B165 A2C      A=A-1  XS
651 0B168 59F      GONC   OPC102
652 0B16B A42     C=C+A  S
653 0B16E 80DF     P=C    15
654 0B172 A46     C=C+C  S
655 0B175 550      GONC   OPC103
656 0B178 844     ST=0   4
657 0B17B 3FBABBBB OPC103 LCHEX  CDBCACCCCB BBBBAB
      CCCCCACB
      DC
658 0B18D F3      DSL    A
659 0B18F AE7     D=C    B
660 0B192 AF6     C=A    W
661 0B195 F0      ASL    A
662 0B197 92C     ?A#0  XS
663 0B19A 8C      GOYES  OPC102
664 0B19C 20     P=     0

```

```
665 0B19E 147          C=DAT1  A
666 0B1A1 8AA          ?C=0   A
667 0B1A4 26           GOYES  OPC118
668 0B1A6 6791        GOTO   OPC142
669 0B1AA 112         OPC110 A=R2
670 0B1AD 3400140     LCHEX  04100
671 0B1B4 922         OPC111 ?C=A   XS
672 0B1B7 F3          GOYES  OPC114
673 0B1B9 C6          C=C+C  A
674 0B1BB 92E        ?C#0   XS
675 0B1BE 6F          GOYES  OPC111
676 0B1C0 3483F24     LCHEX  42F38
677 0B1C7 962        ?C=A   B
678 0B1CA D1          GOYES  OPC113
679 0B1CC 926        ?C#A   XS
680 0B1CF 31          GOYES  OPC112
681 0B1D1 8F2EEC1     GOSBVL #1CEE2  D1+21B : non supporte
682 0B1D8 3400A87     LCHEX  78A00
683 0B1DF 5B1        GONC   OPC115
684 0B1E2 928         OPC112 ?A=0   XS
685 0B1E5 F1          GOYES  OPC117
686 0B1E7 AB2         OPC113 C=0    X
687 0B1EA 928        ?A=0   XS
688 0B1ED 90          GOYES  OPC114
689 0B1EF F2          CSL    A
690 0B1F1 C6          C=C+C  A
691 0B1F3 AA6         C=A    XS
692 0B1F6 111         OPC114 A=R1
693 0B1F9 C2          C=C+A  A
694 0B1FB DA          OPC115 A=C    A
695 0B1FD 101        OPC116 R1=A
696 0B200 20          P=     0
697 0B202 03          RTNCC
698
699 0B204 07          OPC117 C=RSTK
700 0B206 857         OPC118 ST=1   7
701 0B209 34C8080     LCHEX  0808C
702 0B210 6EE0        GOTO   OPC133
703 0B214 07          OPC120 C=RSTK
704 0B216 119         C=R1
705 0B219 AC9         C=B    S
706 0B21C 8AE        ?C#0   A
707 0B21F 51          GOYES  OPC121
708 0B221 7F72        GOSUB  OPC161
709 0B225 31F2        LCHEX  2F
710 0B229 7992        GOSUB  OPC171
711 0B22D 3420202     LCHEX  20202
712 0B234 A06         OPC121 C=C+C  P
713 0B237 447         GOC    OPC126
714 0B23A 112         A=R2
715 0B23D A4E         OPC122 C=C-1  S
716 0B240 BF4         ASR    W
717 0B243 A2E         C=C-1  XS
718 0B246 56F        GONC   OPC122
719 0B249 D1          B=0    A
720 0B24B 80DF        P=C    15
721 0B24F A98         B=A    WP
```

```
722 0B252 A04          A=A+A  P
723 0B255 570          GONC   OPC123
724 0B258 B99          B=-B   WP
725 0B25B F9           B=-B   A
726 0B25D 20           OPC123 P=    0
727 0B25F A06          C=C+C  P
728 0B262 5F0          GONC   OPC124
729 0B265 D0           A=0    A
730 0B267 AEA          A=C    B
731 0B26A BE4          ASR    B
732 0B26D 8A0          ?A=B  A
733 0B270 69           GOYES  OPC118
734 0B272 A06           OPC124 C=C+C  P
735 0B275 5A1          GONC   OPC125
736 0B278 8AD          ?B#0  A
737 0B27B 51           GOYES  OPC125
738 0B27D 7C50         GOSUB  OPC130
739 0B281 000F9        NIBHEX 000F9
740 0B286 02089        NIBHEX 02089
741 0B28B 06089        NIBHEX 06089
742 0B290 10A           OPC125 R2=C
743 0B293 80D4          P=C    4
744 0B297 8F00000      GOSBVL =CSRC5
745 0B29E 809          C+P+1
746 0B2A1 C9           C=C+B  A
747 0B2A3 BF2          CSL    W
748 0B2A6 BF2          CSL    W
749 0B2A9 12A          CR2EX
750 0B2AC 7D20         OPC126 GOSUB  OPC130
751 0B2B0 40076        NIBHEX 40076
752 0B2B5 42006        NIBHEX 42006
753 0B2BA 46006        NIBHEX 46006
754 0B2BF 48007        NIBHEX 48007
755 0B2C4 40C16        NIBHEX 40C16
756 0B2C9 4D106        NIBHEX 4D106
757 0B2CE 45306        NIBHEX 45306
758 0B2D3 40D16        NIBHEX 40D16
759 0B2D8 40716        NIBHEX 40716
760
761 0B2DD 80D3         OPC130 P=C    3
762 0B2E1 07           C=RSTK
763 0B2E3 135          D1=C
764 0B2E6 0D           OPC131 P=P-1
765 0B2E8 480          GOC    OPC132
766 0B2EB 174          D1=D1+ 5
767 0B2EE 57F          GONC   OPC131
768 0B2F1 306          OPC132 LCHEX  6
769 0B2F4 AC5          B=C    S
770 0B2F7 20           P=    0
771 0B2F9 112          A=R2
772 0B2FC 147          C=DAT1 A
773 0B2FF D0           OPC133 A=0    A
774 0B301 7941         GOSUB  OPC150
775 0B305 2574F445    NIBASC 'RGOTNYES'
      E4955435
776 0B315 552465C4    NIBASC 'UBVLONCG'
      F4E43474
```

```
777 0B325 0560          NIBHEX 0560
778
779 0B329 14B          OPC140 A=DAT1 B
780 0B32C 170          D1=D1+ 1
781 0B32F 15F5        OPC141 C=DAT1 6
782 0B333 175          D1=D1+ 6
783 0B336 983          ?D>C P
784 0B339 6F           GOYES OPC141
785 0B33B BF6         CSR W
786 0B33E 7C01        OPC142 GOSUB OPC150
787
788 0B342 F370C3E3     NIBHEX F370C3E3
789 0B34A D332D235     NIBHEX D332D235
790 0B352 25C48062     NIBHEX 25C48062
791 0B35A 12B2D290     NIBHEX 12B2D290
792 0B362 400313D0     NIBHEX 400313D0
793 0B36A 34D3C425     NIBHEX 34D3C425
794 0B372 35544584     NIBHEX 35544584
795 0B37A 44E43534     NIBHEX 44E43534
796 0B382 B4D35434     NIBHEX B4D35434
797 0B38A 85D49400     NIBHEX 85D49400
798 0B392 D0C434B2     NIBHEX D0C434B2
799 0B39A 05404403     NIBHEX 05404403
800 0B3A2 13D30534     NIBHEX 13D30534
801 0B3AA 440313B2     NIBHEX 440313B2
802 0B3B2 D2135060     NIBHEX D2135060
803 0B3BA 10A0B070     NIBHEX 10A0B070
804 0B3C2 D3254414     NIBHEX D3254414
805 0B3CA 4580D370     NIBHEX 4580D370
806 0B3D2 54853500     NIBHEX 54853500
807 0B3DA 00000000     NIBHEX 00000000
808 0B3E2 3584F455     NIBHEX 3584F455
809 0B3EA 94E445D3     NIBHEX 94E445D3
810 0B3F2 34F43514     NIBHEX 34F43514
811 0B3FA D39444E4     NIBHEX D39444E4
812 0B402 64649474     NIBHEX 64649474
813 0B40A C0F384D4     NIBHEX C0F384D4
814 0B412 35358503     NIBHEX 35358503
815 0B41A 45052524     NIBHEX 45052524
816 0B422 D4D33203     NIBHEX D4D33203
817 0B42A 1350D000     NIBHEX 1350D000
818 0B432 24553534     NIBHEX 24553534
819 0B43A C4258454     NIBHEX C4258454
820 0B442 15F33554     NIBHEX 15F33554
821 0B44A 4534         NIBHEX 4534
822
823 0B44E 121          OPC150 AR1EX
824 0B451 DA           A=C A
825 0B453 121          AR1EX
826 0B456 07           C=RSTK
827 0B458 135          D1=C
828 0B45B A0C          OPC151 A=A-1 P
829 0B45E 421          GOC OPC153
830 0B461 17F          D1=D1+ 16
831 0B464 17F          D1=D1+ 16
832 0B467 177          D1=D1+ 8
833 0B46A 50F          GONC OPC151
```

```

834 0B46D 7550      OPC152 GOSUB  OPC171
835 0B471 3101      OPC153 LCHEX  10
836 0B475 DA        A=C      A
837 0B477 121      AR1EX
838 0B47A 8A8      ?A=0    A
839 0B47D D1        GOYES   OPC160
840
841 0B47F 14F      OPC154 C=DAT1  B
842 0B482 171      D1=D1+  2
843 0B485 C4        A=A+A   A
844 0B487 57F      GONC    OPC154
845 0B48A 121      AR1EX
846 0B48D 9EA      ?C>=A  B
847 0B490 DD        GOYES   OPC152
848 0B492 700B     GOSUB   TBLJMc
849 0B496 6ADF     GOTO    OPC153
850
851 0B49A 07        OPC160 C=RSTK
852 0B49C 7400     GOSUB   OPC161
853 0B4A0 6340     GOTO    OPCout
854
855 0B4A4 864      OPC161 ?ST=0  4
856 0B4A7 00        RTNYES
857 0B4A9 7510     GOSUB   OPC170
858 0B4AD 844      ST=0    4
859 0B4B0 7400     GOSUB   OPC162
860 0B4B4 F5        BSR     A
861 0B4B6 F5        BSR     A
862 0B4B8 969      OPC162 ?B=0  B
863 0B4BB 00        RTNYES
864 0B4BD D9        C=B     A
865 0B4BF 560      GONC    OPC171  B.E.T.
866 0B4C2 3102     OPC170 LCHEX  20
867 0B4C6 181      OPC171 D0=D0-  2
868 0B4C9 14C      DAT0=C  B
869 0B4CC 03        RTNCC
870
871 0B4CE 32934    OPC172 LCHEX  439
872 0B4D3 98A      ?C>=A  P
873 0B4D6 70        GOYES   OPC173
874 0B4D8 B0A      A=A-C   P
875 0B4DB F6        CSR     A
876 0B4DD A86      OPC173 C=A     P
877 0B4E0 65EF     GOTO    OPC171
878
879 0B4E4 136      OPCout CD0EX      C(A) := ^ M.S.
880 0B4E7 137      CD1EX      D1 := ^ M.S.
881 0B4EA 1B00000  D0=(5) =FUNCD1
882 0B4F1 146      C=DAT0 A   C(A) := vieux pointeur M.S.
883 0B4F4 109      R1=C      R1 := bottom of M.S.
884
885
886
887
888
889
890
*
* A nouveau, je m'excuse de profaner un code aussi
* beau tellement il est illisible. Helas, les
* necessites m'obligent a souiller ces lieux. Je
* m'en vais decrire les exactions que j'envisage :
*
* Les opcodes peuvent etre decomposes en 6 classes

```

```
891      * Opcode seul :
892      *   ex: RTN
893      * Opcode + Modifier :
894      *   ex: A=0 A
895      * Opcode + Modifier + Branch :
896      *   ex: ?A=0 A/RTNYES
897      * Opcode + Branch :
898      *   ex: ?MP=0/RTNYES
899      * Opcode + Modifier + Branch + Adresse :
900      *   ex: ?A=0 A/GOYES 12345
901      * Opcode + Branch + Adresse :
902      *   ex: ?MP=0/GOYES 12345
903      *
904      * Taille maximum des champs :
905      *   opcode   : 6 octets
906      *   modifier : 16 octets (LCHEX)
907      *   branch   : 6 octets
908      *   adresse  : 5 octets, puis 6 avec un #
909      *
910      *
911      * La transformation correspond au programme Basic
912      * suivant :
913      *
914      * 1000 SUB TRANSF(A$,R$) ! A$=donnee, R$=resultat
915      *   - Decomposition en 4 champs
916      * 1010 E=POS(A$, " ") ! espace
917      * 1020 S=POS(A$, "/" ) ! slash : il y a un test
918      * 1030 IF E=0 AND S=0 THEN R$=A$ @ END !opc. seul
919      * 1040 IF S THEN X$=A$[S+1] @ A$=A$[1,S-1]
920      *   - X$ = test s'il y a lieu
921      *   - A$ = donnee sans test s'il y a lieu
922      * 1050 IF E THEN M$=A$[E+1] @ A$=A$[1,E-1]
923      *   - M$ = modifier s'iul y a lieu
924      *   - A$ = opcode seul (sans modifier)
925      * 1060 O$=A$
926      * 1070 E=POS(X$, " ") ! espace dans le test
927      * 1080 IF E THEN D$=X$[E+1] @ X$=X$[1,E-1]
928      *   - D$ = adresse s'il y a lieu
929      *   - X$ = RTNYES ou GOYES ou chaine vide
930      * 1090 B$=X$
931      *   - Fin de la separation
932      *   O$ = opcode
933      *   M$ = modifier
934      *   B$ = branch if test
935      *   D$ = adresse if GOYES
936      * 1100 IF O$="LC" THEN O$="LCHEX"
937      * 1110 IF O$="D0=" OR O$="D1=" THEN
938      *   O$=O$&"("&STR$(LEN(M$))&")" @ M$="#"&M$
939      *   - D0=(n) #?????
940      *   valeur en hexadecimal
941      * 1120 IF O$[1,2]="GO" THEN M$="#"&M$
942      *   - GONC, GOC, GOSUB, GOTO, GOSUBL, GOLONG,
943      *   GOSBVL ou GOVLNG toujours suivis d'une
944      *   adresse en hexadecimal.
945      * 1130 R$=O$
946      * 1140 IF LEN(M$) THEN R$[8]=M$
947      * 1150 IF LEN(B$) THEN R$=R$&"/"&B$
```

```

948          * 1160 IF LEN(D$) THEN R$=R$&" #"&D$
949          * 1170 END SUB
950          *
951
952          *
953          * Resumons nous :
954          *   D0 = FUNCD1
955          *   D1 = M.S.
956          *   C(A) = vieux pointeur de M.S.
957          *
958 0B4F7 AF3          D=0   W       D(W) := 0
959 0B4FA D7          D=C   A       D(A) := vieux pointeur
960 0B4FC 137        CD1EX          C(A) := pointeur actuel
961          *
962 0B4FF E3          D=D-C A       D(A) := taille en quartets
963 0B501 81F        DSRB          D(A) := taille en octets
964          *
965          * L'exploration qui va suivre a besoin de :
966          * D(A) = nb de caracteres restant
967          *
968
969          *
970          * Champ opcode :
971          *
972 0B504 1B109F2      D0=(5) opcode
973 0B50B 7FE1        GOSUB  getchp
974 0B50F D9          C=B   A
975 0B511 108         R0=C
976 0B514 D1          B=0   A       lg de modifier par defaut
977 0B516 31F2        LCASC  '/'
978 0B51A 962        ?A=C  B
979 0B51D D0          GOYES  out100
980          *
981          * Champ modifier
982          *
983 0B51F 1BD09F2      D0=(5) modifi
984 0B526 74D1        GOSUB  getchp
985 0B52A             out100
986 0B52A D9          C=B   A
987 0B52C 109        R1=C
988          *
989          * Champ branch
990          *
991 0B52F 1BD29F2      D0=(5) branch
992 0B536 74C1        GOSUB  getchp
993 0B53A D9          C=B   A
994 0B53C 10A        R2=C
995          *
996          * Champ adresse
997          *
998 0B53F 1B939F2      D0=(5) adress
999 0B546 74B1        GOSUB  getchp
1000 0B54A D9          C=B   A
1001 0B54C 10B        R3=C
1002          *
1003          * Le decoupage en champs est termine.
1004          * On a :
```



```

1005      * R0 = longueur en octets de l'adresse
1006      * R1 = longueur en octets du branch
1007      * R2 = longueur en octets du modifier
1008      * R3 = longueur en octets de l'opcode
1009      *
1010 0B54F 1F00000      D1=(5) =FUNCD1
1011 0B556 147        C=DAT1 A
1012 0B559 135        D1=C          D1 := ^ M.S.
1013      *
1014      * Debut de la phase d'analyse :
1015      *
1016 0B55C 1B109F2      D0=(5) opcode
1017 0B563 D0          A=0 A
1018 0B565 D2          C=0 A
1019 0B567 15A3        A=DAT0 4
1020 0B56B 33C434      LCASC 'CL' LC
1021 0B571 8A6        ?A#C A
1022 0B574 E1          GOYES out110
1023      *
1024      * LC -> LCHEX
1025      *
1026 0B576 39C43484    LCASC 'XEHCL' LCHEX
      5485
1027 0B582 15C9        DAT0=C 10
1028 0B586 D2          C=0 A
1029 0B588 305        LC(1) 5
1030 0B58B 108        R0=C
1031 0B58E 61A0        GOTO out500
1032
1033      *
1034      * D0= ou D1= ?
1035      *
1036 0B592      out110
1037 0B592 110        A=R0          A(A) := longueur de opcode
1038 0B595 303        LC(1) 3
1039 0B598 906        ?A#C P
1040 0B59B B7          GOYES out120
1041 0B59D AF0        A=0 W
1042 0B5A0 AF2        C=0 W
1043 0B5A3 15A5        A=DAT0 6
1044 0B5A7 354403D3    LCASC '=0D'
1045 0B5AF 972        ?A=C W
1046 0B5B2 F0          GOYES out115
1047 0B5B4 354413D3    LCASC '=1D'
1048 0B5BC 976        ?A#C W
1049 0B5BF 75          GOYES out120
1050 0B5C1      out115
1051      *
1052      * D0= ou D1= -> D0=(n) #...
1053      *
1054 0B5C1 1970        D0=(2) (opcode)+2*3
1055 0B5C5 3182        LCASC '('
1056 0B5C9 14C        DAT0=C B
1057 0B5CC 161        D0=D0+ 2
1058 0B5CF 111        A=R1          C(A) := longueur modifier
1059 0B5D2 3103        LCASC '0'
1060 0B5D6 A62        C=C+A B

```

```
1061 0B5D9 14C          DAT0=C B
1062 0B5DC 161          D0=D0+ 2
1063 0B5DF 3192         LCASC  ') '
1064 0B5E3 14C          DAT0=C B
1065
1066 0B5E6 110          A=R0
1067 0B5E9 E4           A=A+1  A          longueur opcode + 3
1068 0B5EB E4           A=A+1  A
1069 0B5ED E4           A=A+1  A
1070 0B5EF 100          R0=A
1071
1072 0B5F2              out117
1073                    *
1074                    * Insertion d'un # dans le modifier
1075                    *
1076 0B5F2 1BD09F2       D0=(5) modifi
1077 0B5F9 15A9          A=DAT0 2*5
1078 0B5FD 3132         LCASC  '# '
1079 0B601 14C          DAT0=C B
1080 0B604 161          D0=D0+ 2
1081 0B607 1589         DAT0=A 2*5
1082
1083 0B60B 111          A=R1
1084 0B60E E4           A=A+1  A          longueur modifier + 1
1085 0B610 101          R1=A
1086
1087 0B613 5C1          GONC   out500    B.E.T.
1088
1089 0B616              out120
1090 0B616 1B109F2       D0=(5) opcode
1091 0B61D D0           A=0    A
1092 0B61F D2           C=0    A
1093 0B621 15A3         A=DAT0 4
1094 0B625 3374F4       LCASC  'OG'
1095 0B62B 8A2          ?A=C   A
1096 0B62E 4C          GOYES  out117   inserer un #
1097
1098                    *
1099                    * Reconstitution de l'opcode a partir des 4 champs
1100                    *
1101 0B630              out500
1102                    *
1103                    * Sauvegarde de la longueur de l'opcode en B(S)
1104                    * et du modifier en R0.
1105                    *
1106 0B630 118          C=R0
1107 0B633 816          CSRC           C(S) := longueur opcode
1108 0B636 AC5          B=C    S          B(S) := longueur opcode
1109 0B639 119          C=R1
1110 0B63C 108          R0=C
1111
1112                    *
1113                    * Recuperation des pointeurs fondamentaux.
1114                    *
1114 0B63F 8F00000       GOSBVL =D=AVMS
1115 0B646 1F00000       D1=(5) =FUNCD1
1116 0B64D 147          C=DAT1 A
1117 0B650 135          D1=C
```

```

1118 0B653 109          R1=C          Pour ADHEAD
1119
1120          *
1121          * Champ opcode :
1122          *
1123 0B656 1B109F2      D0=(5) opcode
1124 0B65D AF0          A=0      W
1125 0B660 AC4          A=B      S
1126 0B663 810          ASLC          A(A) := longueur en octets
1127 0B666 70C0        GOSUB wrtchp
1128
1129 0B66A 118          C=R0          C(A) := longueur modifier
1130 0B66D 8AA          ?C=0      A
1131 0B670 A2           GOYES out600  branch / adresse
1132          *
1133          * Tabulation du modifier
1134          *
1135 0B672 811          BSLC          B(0) := longueur en octets
1136 0B675 307          LC(1) 7
1137 0B678 B0D          B=C-B P      C(0) := nb esp. a ajouter
1138 0B67B 3102        LCASC ' '
1139 0B67F 560          GONC out520
1140 0B682 78B0        out510 GOSUB stkchr
1141 0B686 A0D          out520 B=B-1 P
1142 0B689 58F          GONC out510
1143          *
1144          * Ajout du modifier
1145          *
1146 0B68C 1BD09F2      D0=(5) modifi
1147 0B693 110          A=R0          A(=A) := lg du modifier
1148 0B696 7090        GOSUB wrtchp
1149          *
1150          * Y a t-il un branch ?
1151          *
1152 0B69A          out600
1153 0B69A 11A          C=R2
1154 0B69D 8AA          ?C=0      A
1155 0B6A0 24           GOYES out999  Non : sortie
1156 0B6A2 31F2        LCASC '/'
1157 0B6A6 7490        GOSUB stkchr
1158          *
1159          * Branch
1160          *
1161 0B6AA 1BD29F2      D0=(5) branch
1162 0B6B1 112          A=R2
1163 0B6B4 7270        GOSUB wrtchp
1164          *
1165          * Y a-t-il une adresse ?
1166          *
1167 0B6B8 11B          C=R3
1168 0B6BB 8AA          ?C=0      A
1169 0B6BE 42           GOYES out999  Non : sortie
1170          *
1171          * Tabulation : 2 espaces et un "#"
1172          *
1173 0B6C0 3102        LCASC ' '
1174 0B6C4 7670        GOSUB stkchr

```

```
1175 0B6C8 7270          GOSUB  stkchr
1176 0B6CC 3132          LCASC  '#'
1177 0B6D0 7A60          GOSUB  stkchr
1178                      *
1179                      *  adresse :
1180                      *
1181 0B6D4 1B939F2        D0=(5)  adress
1182 0B6DB 113           A=R3
1183 0B6DE 7840          GOSUB  wrtchp
1184                      *
1185                      *  Fin !
1186                      *  Sortie de la fontion, la chaine est sur la MS
1187                      *
1188 0B6E2                out999
1189                      *
1190                      *  Sortie vers Basic, ou sortie vers l'appelant (SYSEDT) ?
1191                      *
1192 0B6E2 87A            ?ST=1  sysedt
1193 0B6E5 00            RTNYES      Retour a SYSEDT
1194                      *
1195                      *  Sortie vers Basic :
1196                      *
1197 0B6E7 1B00000        D0=(5)  =FUNCD0
1198 0B6EE 146           C=DAT0  A
1199 0B6F1 134           D0=C
1200 0B6F4 840          ST=0    0      No return desired
1201 0B6F7 8D00000        GOVLNG  =ADHEAD
1202
1203                      *****
1204                      *  getchp
1205                      *
1206                      *  But: placer a MEM(D0) le champ trouve sur la MS
1207                      *  et renvoyer sa longueur.
1208                      *  Entree:
1209                      *    - D0 = ^ sauvegarde du champ
1210                      *    - D1 = ^ MS
1211                      *    - D(A) = longueur en octets de ce qui reste
1212                      *  Sortie:
1213                      *    - B(A) = longueur trouvee
1214                      *    - MEM(D0) = champ trouve
1215                      *    - D(A) = longueur de ce qui reste
1216                      *    - A(B) = dernier caractere reconnu
1217                      *  Utilise: A(B), C(B), D(A), D0, D1
1218                      *  Appelle: -
1219                      *  Niveaux: 0
1220                      *  Detail:
1221                      *    Un champ est termine par le premier espace ou
1222                      *    slash rencontre.
1223                      *  Historique:
1224                      *    87/05/15: PD  conception & codage
1225                      *  *****
1226
1227 0B6FE                getchp
1228 0B6FE D1            B=0    A
1229 0B700                getc10
1230 0B700 8AB          ?D=0    A
1231 0B703 00            RTNYES
```

```
1232 0B705 CF          D=D-1  A
1233 0B707 1C1        D1=D1-  2
1234 0B70A 14B        A=DAT1  B
1235 0B70D 3102       LCASC  ' '
1236 0B711 962        ?A=C   B
1237 0B714 00         RTNYES
1238 0B716 31F2       LCASC  '/'
1239 0B71A 962        ?A=C   B
1240 0B71D 00         RTNYES
1241 0B71F 148        DAT0=A  B
1242 0B722 161        D0=D0+ 2
1243 0B725 E5         B=B+1  A
1244 0B727 58D        GONC   getc10  B.E.T.
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264 0B72A            wrtchp
1265 0B72A 8A8        ?A=0   A
1266 0B72D 00         RTNYES
1267 0B72F 14E        C=DAT0  B
1268 0B732 161        D0=D0+ 2
1269 0B735 7500       GOSUB  stkchr
1270 0B739 CC         A=A-1  A
1271 0B73B 5EE        GONC   wrtchp  B.E.T.
1272
1273 0B73E 8D00000     stkchr GOVLNG =STKCHR
1274
1275 0B745            END
```

* wrtchp
*
* But: envoie le champ pointe par D0 sur la MS
* Entree:
* - D0 = ^ champ
* - D1 = ^ MS
* - A(A) = longueur du champ
* - D(A) = AVMEMS
* Sortie:
* - D1 reactualise
* Utilise: A(A), C(B), D1
* Appelle: STKCHR
* Niveaux: 1
* Historique:
* 87/05/15: PD conception & codage

ADDRCK	Ext		-	78	308				
ADHEAD	Ext		-	1201					
CSLW5	Ext		-	354					
CSRC5	Ext		-	744					
D=AVMS	Ext		-	1114					
EXPR	Ext		-	99					
FUNCD0	Ext		-	324	1197				
FUNCD1	Ext		-	881	1010	1115			
GETN05	Abs	43635	#0000AA73	-	137	133			
GETN10	Abs	43697	#0000AAB1	-	202	181	189		
GETN20	Abs	43776	#0000AB00	-	218	207	210	213	216
GETN30	Abs	43796	#0000AB14	-	225	222			
GETN99	Abs	43779	#0000AB03	-	219	136	174		
GETNXT	Abs	43610	#0000AA5A	-	123	80	318		
HEXASC	Ext		-	86					
INIT	Abs	43490	#0000A9E2	-	60	79	312		
=NEXT0e	Abs	43524	#0000AA04	-	77				
OPC00	Abs	43905	#0000AB81	-	310	297			
OPC10	Abs	44931	#0000AF83	-	511	358			
OPC100	Abs	45321	#0000B109	-	624	521			
OPC101	Abs	45407	#0000B15F	-	648	637	641	645	
OPC102	Abs	45410	#0000B162	-	649	651	663		
OPC103	Abs	45435	#0000B17B	-	657	655			
OPC11	Abs	44936	#0000AF88	-	513	516			
OPC110	Abs	45482	#0000B1AA	-	669	530			
OPC111	Abs	45492	#0000B1B4	-	671	675			
OPC112	Abs	45538	#0000B1E2	-	684	680			
OPC113	Abs	45543	#0000B1E7	-	686	678			
OPC114	Abs	45558	#0000B1F6	-	692	672	688		
OPC115	Abs	45563	#0000B1FB	-	694	683			
OPC116	Abs	45565	#0000B1FD	-	695	612			
OPC117	Abs	45572	#0000B204	-	699	685			
OPC118	Abs	45574	#0000B206	-	700	667	733		
OPC120	Abs	45588	#0000B214	-	703	531			
OPC121	Abs	45620	#0000B234	-	712	707			
OPC122	Abs	45629	#0000B23D	-	715	718			
OPC123	Abs	45661	#0000B25D	-	726	723			
OPC124	Abs	45682	#0000B272	-	734	728			
OPC125	Abs	45712	#0000B290	-	742	735	737		
OPC126	Abs	45740	#0000B2AC	-	750	713			
OPC130	Abs	45789	#0000B2DD	-	761	738	750		
OPC131	Abs	45798	#0000B2E6	-	764	767			
OPC132	Abs	45809	#0000B2F1	-	768	765			
OPC133	Abs	45823	#0000B2FF	-	773	702			
OPC140	Abs	45865	#0000B329	-	779	520			
OPC141	Abs	45871	#0000B32F	-	781	784			
OPC142	Abs	45886	#0000B33E	-	786	668			
OPC150	Abs	46158	#0000B44E	-	823	774	786		
OPC151	Abs	46171	#0000B45B	-	828	833			
OPC152	Abs	46189	#0000B46D	-	834	847			
OPC153	Abs	46193	#0000B471	-	835	829	849		
OPC154	Abs	46207	#0000B47F	-	841	844			
OPC160	Abs	46234	#0000B49A	-	851	839			
OPC161	Abs	46244	#0000B4A4	-	855	708	852		
OPC162	Abs	46264	#0000B4B8	-	862	859			
OPC170	Abs	46274	#0000B4C2	-	866	560	857		
OPC171	Abs	46278	#0000B4C6	-	867	710	834	865	877

out999	Abs	46818	#0000B6E2	-	1188	1155	1169							
pcA	Abs	43888	#0000AB70	-	299	344								
pca	Abs	43829	#0000AB35	-	253	299								
rsI	Abs	43972	#0000ABC4	-	345	339								
rsi	Abs	43801	#0000AB19	-	236	345								
scrтч	Abs	194817	#0002F901	-	28	29	30	31	32					
special	Abs	43861	#0000AB55	-	261	243								
stkchr	Abs	46910	#0000B73E	-	1273	1140	1157	1174	1175	1177	1269			
sysedt	Abs	10	#0000000A	-	34	296	309	1192						
wrtchp	Abs	46890	#0000B72A	-	1264	1127	1148	1163	1183	1271				

Input Parameters

Source file name is sysop.a

Listing file name is sysop.l

Object file name is sysop.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE SYSED
2
3          * *****
4          * Memory editor/disassembler module
5          *   (uses the sysop module)
6          *
7          * Keyword(s):
8          *   SYSEEDIT
9          * Author(s):
10         *   Pierre David, Janick Taillandier
11         * History:
12         *   Based on an unpublished source file (private
13         *     communication) and updated from actual JPC LEX
14         *     code disassembly
15         *   Undocumented and undated changes
16         *   Updated with the changes from rev.X (R. Rosenbaum)
17         *
18         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
19         * *****
20
21 0B745          ABS          #B745
22
23         *
24         RESJPC EQU          #2F991      JPC reserved location
25         INSMK EQU          1           %0001
26         nINSMK EQU         (0-1-1)
27
28         EscSt0 EQU         0
29         EscSt1 EQU         1
30         *
31         * Definitions pour les routines dspxxx
32         *
33         sBLANC EQU         1           1 si un blanc rencontre
34         sGUIL EQU         2           1 si un guillemet rencontre
35         sCRLF EQU         3           1 si il faut afficher CR/LF
36         *
37         * Voir IDS 1, page 17-19
38         *
39         sBADRC EQU         8
40         *
41         * Pour [ENDLINE] et fichier
42         *
43         rplc# EQU          1
44         chrc/ EQU          2
45
46         *
47         * Les variables utilisees par SYSEEDIT :
48         *
49         * dADR      : adresse courante
50         *           5 quartets
51         * dKEY     : derniere touche prise en compte
52         *           2 quartets
53         *           code physique de la derniere touche lue
54         * dETAT    : etat de l'automate
55         *           1 quartet
56         *           0 a 6
57         * dREJEC   : rejet de la touche courante
```

```
58      *          1 quartet
59      *          1 si lecture clavier necessaire
60      * dLINE   : ligne courante
61      *          23*2 quartets + 2 (FF = marqueur fin ligne)
62      *          ligne couramment envoyee a l'affichage
63      * dSTACK  : pile
64      *          1+5*.....
65      *          la pile des adresses de retour + le pointeur
66      * dLEN    : longueur de l'instruction courante
67      *          2 quartets
68      * dFILE   : adresse dans le fichier de sortie
69      *          5 quartets
70      *          adresse ou inserer la prochaine ligne de texte
71      * dHEADR  : adresse du header du fichier de sortie
72      *          5 quartets
73      *          adresse du nom du fichier (1er quartet du header)
74      * dMODE   : mode desassemblage
75      *          1 quartet
76      *          1 si mode desassemblage, 0 sinon
77      *
78
79      Stsave EQU    #2F6BE
80      Ldcspc EQU    #2F6C1
81      Lexptr EQU    #2F6CF
82      Stmtr0 EQU    #2F871
83      Trfmbf EQU    #2F8C5
84
85      PCSAV  EQU    #2F9E0          rev X
86
87      dFILE  EQU    Ldcspc          5 quartets
88      dHEADR EQU    Lexptr          5 quartets
89      * dADR  EQU    (Trfmbf)+00    5 quartets rev D
90      dADR   EQU    PCSAV          rev X
91      dKEY   EQU    (Trfmbf)+05    2 quartets
92      dETAT  EQU    (Trfmbf)+07    1 quartet
93      dREJEC EQU    (Trfmbf)+08    1 quartet
94      dLEN   EQU    (Trfmbf)+09    2 quartets
95      dMODE  EQU    (Trfmbf)+11    1 quartet
96      dLINE  EQU    (Trfmbf)+12    48 quartets
97      dSTACK EQU    Stmtr0      7 niveaux dans la pile
98      MAXLVL EQU    7
99
100     dSC1   EQU    (Stmtr0)+1+7*5  4 quartets
101     dSC2   EQU    Stsave          3 quartets (actually not used)
102
103
104
105
106
107     *****
108     * clrrpt
109     *
110     * But: effacer toute trace de repetition de touche.
111     * Entree: -
112     * Sortie: -
113     * Abime: C(A), D(0)
114     * Niveaux: 0
```

```
115          * Detail: pompe dans le code de KA, lui meme pompe dans le
116          *   code de CHEDIT.
117          * Historique:
118          *   87/05/18: P.D. & J.T. reconception & repompage
119          *****
120
121 0B745 1B00000  clrrpt D0=(5) =KEYPTR
122 0B74C D2      C=0 A
123 0B74E 1562   C=DAT0 XS
124 0B752 92E   ?C#0 XS
125 0B755 00    RTNYES          retour si buffer vide
126 0B757 1A0000 D0=(4) (=KEYBUF)+2*14
127 0B75D 14C   DAT0=C B
128 0B760 01    RTN
129
130          *****
131          * dspmod
132          *
133          * But: afficher une ligne suivant le mode (desassemblage ou
134          *   hexadecimal) courant.
135          * Entree:
136          *   - dMODE = 1 si desassemKblage, 0 si hexadecimal
137          * Sortie:
138          *   - par stddsp ou desdsp
139          * Historique:
140          *   87/05/19: conception & codage sans desdsp
141          *****
142
143 0B762 6BA0    stddsP GOTO  stddsp
144
145 0B766 1B0D8F2 dspmod D0=(5) dMODE
146 0B76D 1564   C=DAT0 S
147 0B771 94A   ?C=0 S
148 0B774 EE    GOYES  stddsP
149          *
150          * Affichage en mode desassembliesage
151          *
152          *****
153          * desdsp
154          *
155          * But: afficher l'instruction courante.
156          * Entree:
157          *   - dADR : adresse de l'instruction
158          * Sortie:
159          *   - dLINE : instruction non reduite (avec les tabulations)
160          *   - dLEN : longueur de l'instruction
161          *   - dMODE = 1 (mode desassemblage)
162          * Abime: Function Scratch, CPU registers, ST, Math-Stack
163          * Appelle: getopc
164          * Niveaux: 5
165          * Historique:
166          *   87/05/20: P.D. & J.T. conception & codage
167          *****
168
169          *
170 0B776        desdsp
171          *
```

```

172          * Initialiser la M.S.
173          *
174 0B776 8F00000          GOSBVL =COLLAP
175 0B77D 135          D1=C          D1 := ^ M.S.
176          *
177          * Lecture de l'adresse
178          *
179 0B780 1B0E9F2          D0=(5) dADR
180 0B787 142          A=DAT0 A
181 0B78A D8          B=A A
182          *
183          * dLINE=OPCODE$(dADR)
184          *
185 0B78C 8E0000          GOSUBL =SYSGOP getopc
186          *
187 0B792 AF2          C=0 W
188 0B795 137          CD1EX          C(A) := ^ dernier caractere
189 0B798 111          A=R1
190 0B79B EE          C=A-C A          C(A) := nb de quartets de la chaine
191 0B79D 81E          CSRB          C(A) := nb de caracteres de la chaine
192          *
193 0B7A0 1F1D8F2          D1=(5) dLINE          D1 := ^ dLINE
194 0B7A7 111          A=R1
195 0B7AA 130          D0=A          D0 := ^ caractere dans la M.S.
196 0B7AD 6F00          GOTO desd20
197 0B7B1 181          desd10 D0=D0- 2
198 0B7B4 14A          A=DAT0 B
199 0B7B7 149          DAT1=A B
200 0B7BA 171          D1=D1+ 2
201 0B7BD CE          desd20 C=C-1 A
202 0B7BF 51F          GONC desd10
203          *
204          * FF
205          *
206 0B7C2 31FF          LC(2) 255 -1
207 0B7C6 14D          DAT1=C B
208          *
209          * dLEN := OPLEN (dADR)
210          *
211 0B7C9 1B0E9F2          D0=(5) dADR
212 0B7D0 142          A=DAT0 A
213 0B7D3 D8          B=A A          B(A) := adresse courante
214 0B7D5 8E0000          GOSUBL =SYSINI INIT
215 0B7DB 8E0000          GOSUBL =SYSGNX GETNXT
216 0B7E1 20          P= 0
217 0B7E3 1B0E9F2          D0=(5) dADR
218 0B7EA 142          A=DAT0 A
219          *** change
220          *          D0=(2) dLEN          * rev D
221 0B7ED 1AEC8F          D0=(4) dLEN          * rev X
222          *** end of change
223 0B7F3 E2          C=C-A A          C(A) := longueur de l'instruction
224 0B7F5 14C          DAT0=C B
225          *
226          * dMODE := 1
227          *
228 0B7F8 190D          D0=(2) dMODE

```

```
229 0B7FC 301          LC(1)  1
230 0B7FF 1540        DAT0=C  P
231 0B803 6B60        GOTO   dsp
232
233 *****
234 * stddsp, stdds-
235 *
236 * But: afficher une ligne hexadecimale.
237 * Entree:
238 *   - dADR = adresse des 16 quartets.
239 * Sortie:
240 *   - dLEN = 0
241 *   - dMODE = 0
242 * Abime: A-C, D(S)
243 * Appelle: HEXASC
244 * Niveaux: 1
245 * Historique:
246 *   87/05/18: P.D. & J.T. conception & codage
247 *****
248
249 0B807 843          stdds- ST=0   sCRLF
250 0B80A 6600        GOTO   stdd10
251 0B80E 853          stddsp ST=1   sCRLF
252 0B811 AC3          stdd10 D=0     S
253 0B814 A4F          D=D-1  S          D(S) := F
254 0B817 1F1D8F2     D1=(5) dLINE
255 0B81E 1B0E9F2     D0=(5) dADR
256 0B825 146          C=DAT0  A
257 0B828 134          D0=C
258 0B82B 15A0        std10  A=DAT0 1
259 0B82F AC2          C=0     S          C(S) := 1-1
260 0B832 8F00000     GOSBVL =HEXASC
261 0B839 149          DAT1=A  B
262 0B83C 160          D0=D0+ 1
263 0B83F 171          D1=D1+ 2
264 0B842 A4F          std20  D=D-1  S
265 0B845 55E          GONC   std10
266 *
267 * FF
268 *
269 0B848 31FF          LC(2)  255   -1
270 0B84C 14D          DAT1=C  B
271 *
272 * dLEN := dMODE := 0
273 *
274 0B84F AF2          C=0     W
275 0B852 1BEC8F2     D0=(5) dLEN
276 0B859 14C          DAT0=C  B
277 0B85C 190D        D0=(2) dMODE
278 0B860 1544        DAT0=C  S
279 0B864 6D00        GOTO   dsp10
280
281 *****
282 * dsp, dsp-
283 *
284 * But: afficher la ligne dLINE, en enlevant eventuellement
285 *   les espaces inutiles.
```

```

286          * Entree:
287          *   - dLINE = ^ ligne (23 caracteres maximum + 1)
288          *   - dADR = adresse courante
289          * Sortie:
290          * Utilise: A-D, D0, D1, ST(1,2)
291          * Appelle: DSPCHA
292          * Niveaux: 3
293          * Detail: dLINE est cense etre suivi d'un FF
294          *   dsp- n'affiche pas de CR/LF a la fin.
295          * Historique:
296          *   87/05/18: P.D. & J.T. conception & codage
297          *****
298
299 0B868 843      dsp-   ST=0   sCRLF
300 0B86B 6600      GOTO   dsp10
301
302 0B86F 853      dsp    ST=1   sCRLF
303 0B872          dsp10
304          *
305          * Afficher l'adresse suivie d'un ":"
306          *
307 0B872 1B0E9F2      D0=(5) dADR
308 0B879 142          A=DAT0 A
309 0B87C 24           P=      4
310 0B87E 80CF        C=P    15
311 0B882 20           P=      0
312 0B884 8F00000     GOSBVL =HEXASC
313 0B88B 1F00000     D1=(5) =FUNCR0
314 0B892 1599        DAT1=A 10
315 0B896 D2          C=0    A
316 0B898 305         LC(1) 5      5 caracteres
317 0B89B DA          A=C    A
318 0B89D 8F00000     GOSBVL =DSPCNA
319          *
320          * ":"
321          *
322 0B8A4 31A3         LCASC  ':'
323 0B8A8 8F00000     GOSBVL =DSPCHC
324          *
325          * afficher le reste de la ligne
326          *
327 0B8AF 341D8F2     LC(5)  dLINE
328 0B8B6 1B00000     D0=(5) =FUNCD0  5 quartets
329 0B8BD 144         DAT0=C A      adresse du caractere courant
330 0B8C0 841         ST=0   sBLANC
331 0B8C3 842         ST=0   sGUIL
332 0B8C6          dsp20
333 0B8C6 1B00000     D0=(5) =FUNCD0
334 0B8CD 142         A=DAT0 A
335 0B8D0 131        D1=A          D1 := adresse du caractere courant
336 0B8D3 E4         A=A+1 A
337 0B8D5 E4         A=A+1 A
338 0B8D7 140        DAT0=A A      =FUNCD0 := adresse du car. courant
339          *
340          * D1 = adresse du caractere courant
341          *
342 0B8DA 14B        A=DAT1 B

```

```
343 0B8DD B64          A=A+1  B
344 0B8E0 4A3          GOC    dsp70  sortie si FF
345                    *
346                    * REDUCE$ ?
347                    *
348 0B8E3 A6C          A=A-1  B
349 0B8E6 3102         LCASC  ' '
350
351 0B8EA 872          ?ST=1  sGUIL
352 0B8ED 32           GOYES  dsp65
353
354 0B8EF 966          ?A#C   B
355 0B8F2 70           GOYES  dsp50  on affiche le caractere en testant
356 0B8F4 871         ?ST=1  sBLANC
357 0B8F7 FC           GOYES  dsp20
358
359 0B8F9 841          dsp50  ST=0   sBLANC
360 0B8FC 966          ?A#C   B
361 0B8FF 50           GOYES  dsp60
362 0B901 851          ST=1   sBLANC
363 0B904 3172         dsp60  LCASC  \ '\
364 0B908 966          ?A#C   B
365 0B90B 50           GOYES  dsp65
366 0B90D 852          ST=1   sGUIL
367
368 0B910 8F00000      dsp65  GOSBVL =DSPCHA
369 0B917 6EAF         GOTO   dsp20  END LOOP
370                    *
371                    * CR/LF
372                    *
373 0B91B              dsp70
374 0B91B 863          ?ST=0  sCRLF
375 0B91E 00           RTNYES
376 0B920 8D00000      GOVLNG =CRLFND
377
378
379                    *****
380                    * SYSEDe
381                    *
382                    * But: Desassembleur interactif
383                    * Syntaxe: SYSEdit <adresse>
384                    * Historique:
385                    *   87/05/18: P.D. & J.T. conception & codage
386                    *****
387
388                    * *****
389                    * SYSEdit entry
390                    * *****
391
392                    *** changes
393                    * rev D
394                    *       REL(5) =STDD
395                    *       REL(5) =STRNGp
396                    *=SYSEDe
397                    * rev X
398 0B927 8D48730      SYSPAR  GOVLNG #3784 not supported!
399 0B92E 00000        REL(5) =STDD
```



```
400 0B933 4FFFF          REL(5) SYSPAR
401 0B938 14A           =SYSEDe A=DAT0 B
402 0B93B 1F0E9F2       D1=(5) PCSAV
403 0B942 147           C=DAT1 A
404 0B945 D5            B=C A
405 0B947 3100          LC(2) =tEOL
406 0B94B 962           ?C=A B
407 0B94E 01            GOYES oB91E
408                      *** end of change
409                      *
410                      * Lire l'adresse
411                      *
412 0B950 8F00000        GOSBVL =EXPEX-
413 0B957 8F00000        GOSBVL =ADDRCK
414                      *
415                      * B(A) = valid address
416                      *
417 0B95E D9             oB91E C=B A
418 0B960 1B0E9F2       D0=(5) dADR
419 0B967 144            DAT0=C A Adresse
420                      *
421                      * Initialisation des variables communes
422                      *
423 0B96A AF2            C=0 W
424                      ***
425                      * D0=(2) dETAT rev D
426 0B96D 1ACC8F        D0=(4) dETAT rev X
427                      ***
428 0B973 1544           DAT0=C S
429 0B977 190D           D0=(2) dMODE
430 0B97B 1544           DAT0=C S Mode desas := 0
431 0B97F 19DC           D0=(2) dREJEC
432 0B983 1544           DAT0=C S REJECT := 0
433                      ***
434                      * D0=(4) dSTACK rev D
435 0B987 1917           D0=(2) dSTACK rev X
436                      *** end changes
437 0B98B 1544           DAT0=C S clear stack
438 0B98F 1AFC6F         D0=(4) dHEADR
439 0B995 144            DAT0=C A Fichier := 0
440                      *
441                      * Display standard
442                      *
443 0B998 727E            GOSUB stddsp
444 0B99C                BOUCLE
445 0B99C 1BDC8F2       D0=(5) dREJEC
446 0B9A3 1564           C=DAT0 S
447 0B9A7 AC0            A=0 S
448 0B9AA 1504           DAT0=A S REJECT := 0 par default
449 0B9AE 94E            ?C#0 S
450 0B9B1 43             GOYES BP20
451                      *
452                      * Lecture clavier
453                      *
454 0B9B3 8F00000        GOSBVL =RPTKY
455 0B9BA 4A2            GOC BP20
456 0B9BD 8F00000        BP05 GOSBVL =SCRLLR
```

```
457 0B9C4 5D0          GONC   BP10
458 0B9C7 8F00000      GOSBVL =CKSREQ
459 0B9CE 6EEF         GOTO   BP05
460 0B9D2 8F00000      BP10   GOSBVL =POPBUF
461 0B9D9 D4           A=B    A      A(B) := keycode
462 0B9DB 1BAC8F2      D0=(5) dKEY
463 0B9E2 148          DAT0=A B      KEY := touche lue
464
465 0B9E5 1BCC8F2      BP20   D0=(5) dETAT
466 0B9EC 15E0         C=DAT0 1      C(0) := etat
467 0B9F0 8F00000      GOSBVL =TBLJMC
468 0B9F7 120          REL(3) etat0
469 0B9FA A16          REL(3) etat1
470 0B9FD 3C6          REL(3) etat2
471 0BA00 537          REL(3) etat3
472 0BA03 367          REL(3) etat4
473 0BA06 600          REL(3) Etat5
474 0BA09 900          REL(3) Etat6
475
476 0BA0C 8CD870      Etat5  GOLONG etat5
477 0BA12 8C6780      Etat6  GOLONG etat6
478
479 0BA18              etat0
480 0BA18 1BAC8F2      D0=(5) dKEY
481 0BA1F 14A          A=DAT0 B
482 0BA22 8F00000      GOSBVL =FINDA
483 0BA29 00           CON(2) =k#ATTN [ATTN]
484 0BA2B CA0          REL(3) vEXIT
485 0BA2E 83           CON(2) #38     [+ ]
486 0BA30 860          REL(3) v+1
487 0BA33 A2           CON(2) #2A     [- ]
488 0BA35 E60          REL(3) v-1
489 0BA38 C1           CON(2) #1C     [* ]
490 0BA3A 470          REL(3) v+16
491 0BA3D E0           CON(2) #0E     [/ ]
492 0BA3F A70          REL(3) v-16
493 0BA42 40           CON(2) #04     [R ]
494 0BA44 CA0          REL(3) vREL
495 0BA47 F0           CON(2) #0F     [A ]
496 0BA49 EA0          REL(3) vNIBA
497 0BA4C F1           CON(2) #1F     [C ]
498 0BA4E 0B0          REL(3) vCON
499 0BA51 22           CON(2) #22     [N ]
500 0BA53 2B0          REL(3) vNIBH
501 0BA56 32           CON(2) #23     [M ]
502 0BA58 3D3          REL(3) vMODIF
503 0BA5B 55           CON(2) #1D+56 [EDIT]
504 0BA5D EC3          REL(3) vMODIF
505 0BA60 41           CON(2) #14     [H ]
506 0BA62 5B0          REL(3) vHEXA
507 0BA65 D1           CON(2) #1D     [Z ]
508 0BA67 683          REL(3) vEDIT
509 0BA6A 81           CON(2) #18     [= ]
510 0BA6C 4F2          REL(3) vGOTO
511 0BA6F 42           CON(2) #24     [( ]
512 0BA71 A03          REL(3) vGOSUB
513 0BA74 52           CON(2) #25     [)]
```



```
571 *****
572
573 0BAD7          vEXIT
574 0BAD7 7A6C          GOSUB  clrrpt
575 0BADB 8F00000      GOSBVL =ATNCLR
576 0BAE2 8F00000      GOSBVL =NOSCRL
577 0BAE9 8D00000      GOVLNG =NXTSTM
578
579 *****
580 * vREL
581 *
582 * But: Traiter la touche [R] (REL(n)) a l'etat 0.
583 * Entree:
584 * Sortie: par BOUCLE
585 *   - dETAT = 1
586 * Historique:
587 *   87/05/19: conception & codage
588 *****
589
590 0BAF0 301          vREL  LC(1)  1
591 0BAF3 6410          GOTO  etat=C
592
593 *****
594 * vNIBA
595 *
596 * But: Traiter la touche [A] (NIBASC) a l'etat 0.
597 * Entree:
598 * Sortie: par BOUCLE
599 *   - dETAT = 2
600 * Historique:
601 *   87/05/19: conception & codage
602 *****
603
604 0BAF7 302          vNIBA  LC(1)  2
605 0BAFA 6D00          GOTO  etat=C
606
607 *****
608 * vCON
609 *
610 * But: Traiter la touche [C] (CON(n)) a l'etat 0.
611 * Entree:
612 * Sortie: par BOUCLE
613 *   - dETAT = 3
614 * Historique:
615 *   87/05/19: conception & codage
616 *****
617
618 0BAFE 303          vCON   LC(1)  3
619 0BB01 6600          GOTO  etat=C
620
621 *****
622 * vNIBH
623 *
624 * But: Traiter la touche [N] (NIBHEX) a l'etat 0.
625 * Entree:
626 * Sortie: par BOUCLE
627 *   - dETAT = 4
```

```
628          * Historique:
629          *   87/05/19: conception & codage
630          *****
631
632 0BB05 304      vNIBH LC(1)  4
633
634          *****
635          *   etat=C
636          *
637          *   But: changer l'etat, et revenir a la boucle.
638          *   Entree:
639          *     - C(0) = etat (0 .. 6)
640          *   Sortie: par BOUCLE
641          *     - dETAT modifie
642          *   Abime: D0
643          *   Niveaux: 0
644          *   Historique:
645          *     87/05/19: P.D. & J.T. conception & codage
646          *****
647
648 0BB08          etat=C
649 0BB08 1BCC8F2      D0=(5) dETAT
650 0BB0F 15C0        DAT0=C 1
651 0BB13 688E        GOTO   BOUCLE
652
653          *****
654          *   vHEXA
655          *
656          *   But: Traiter la touche [H] (hexa) a l'etat 0.
657          *   Entree:
658          *   Sortie: par BOUCLE
659          *   Historique:
660          *     87/05/19: conception & codage
661          *****
662
663 0BB17 73FC        vHEXA GOSUB  stddsp
664 0BB1B 608E        GOTO   BOUCLE
665
666          *****
667          *   vENDLN
668          *
669          *   But: Traiter la touche [ENDLINE] a l'etat 0.
670          *   Entree:
671          *   Sortie: par BOUCLE
672          *   Historique:
673          *     87/05/19: P.D. & J.T. conception & codage
674          *     87/05/28: P.D. & J.T. ajout de la sortie sur fichier
675          *****
676
677 0BB1F          vENDLN
678 0BB1F 1BFC6F2      D0=(5) dHEADR
679 0BB26 146         C=DAT0 A
680 0BB29 8AE         ?C#0  A
681 0BB2C A0         GOYES  vENDf1
682 0BB2E 6771        GOTO   vEND90
683
684 0BB32 6721        Bserr GOTO  bserr
```

```

685
686 0BB36          vENDf1
687              *
688              * Maximum de memoire necessite (attitude pessimiste)
689              * 8 + 23 caracteres pour une ligne seule
690              * 8 + 7 + 2 / 8 + 7 + 6 = 38 caracteres pour 2 lignes
691              * (soit 17 + 2 + 1 / 21 + 2 + 1 avec les longueurs LIF
692              * et le padding)
693              *
694 0BB36 D2          C=0      A
695 0BB38 3185        LC(2)   44*2      caracteres au maximum
696 0BB3C 20          P=       0        Oui, on veut le leeway
697 0BB3E 8F00000    GOSBVL  =MEMCKL
698 0BB45 4CE        GOC      Bserr    Not enough memory
699              *
700              * Impression de la ligne dLINE dans le fichier courant.
701              *
702 0BB48 8F00000    GOSBVL  =OBCOLL  C(A) = ^ buffer
703 0BB4F 134        D0=C      D0 = ^ buffer
704 0BB52 108        R0=C      R0 = ^ 1ere ligne (pour LIFlen)
705 0BB55 163        D0=D0+ 4    D0 = ^ apres la longueur
706              *
707              * <blanc>L<adr><blanc>
708              *
709 0BB58 3102        LCASC    ' '
710 0BB5C 8E0AA0    GOSUBL  putc
711 0BB62 31C4        LCASC    'L'
712 0BB66 8E69A0    GOSUBL  putc
713 0BB6C 1F0E9F2   D1=(5) dADR
714 0BB73 143        A=DAT1  A
715 0BB76 24         P=       4
716 0BB78 80CF      C=P      15      C(S) := 5-1
717 0BB7C 20         P=       0
718 0BB7E 8F00000   GOSBVL  =HEXASC
719 0BB85 1589       DAT0=A  10
720 0BB89 169        D0=D0+ 10
721 0BB8C 3102        LCASC    ' '
722 0BB90 8EC6A0    GOSUBL  putc
723              *
724              * si dLINE[1,1]="G" alors ST=1 rplc#
725              *
726 0BB96 1F1D8F2   D1=(5) dLINE
727              *
728              * D1 = ^ premier caractere de l'opcode dans dLINE
729              * D0 = ^ endroit ou va etre mis l'opcode
730              *
731 0BB9D 841        vEND00 ST=0   rplc#    Il ne faut pas remplacer le # en L
732
733              *** *****
734              *** changes (undated, undocumented):
735              **      A=DAT1 B
736              **      LCASC 'G'
737              **      ?A#C B
738              **      GOYES vEND05
739 0BBA0 15B5        A=DAT1  6
740 0BBA4 3174        LCASC    'G'
741 0BBA8 966        ?A#C    B

```

```

742 0BBAB 02          GOYES  vEND03
743 0BBAD 175        D1=D1+ 6
744 0BBB0 14B        A=DAT1 B
745 0BBB3 1C5        D1=D1- 6
746 0BBB6 3124       LCASC  'B'
747 0BBBA 962        ?C=A  B
748 0BBBD E1         GOYES  vEND05
749 0BBBF 31C4       LCASC  'L'
750 0BBC3 962        ?C=A  B
751 0BBC6 51         GOYES  vEND05
752 0BBC8 5F0        GONC   vEND04
753 0BBCB 352554C4  vEND03  LCHEX  4C4552  LER
754 0BBD3 8A6        ?C#A  A
755 0BBD6 50         GOYES  vEND05
756 0BBD8            vEND04
757                  *** end of changes
758                  *** *****
759
760 0BBD8 851          ST=1   rplc#
761 0BBDE 842          vEND05  ST=0   chrc/
762 0BBDE 31F3         LCASC  '?'
763 0BBE2 966          ?A#C  B
764 0BBE5 50           GOYES  vEND10
765 0BBE7 852          ST=1   chrc/
766
767                  *
768                  * recopie (intelligente) jusqu'au / ou FF
769                  *
769 0BBEA 14B          vEND10  A=DAT1 B
770 0BBED 171          D1=D1+ 2
771 0BBF0 3132        LCASC  '#'
772 0BBF4 966          ?A#C  B
773 0BBF7 E0          GOYES  vEND20  ce n'est pas un #
774 0BBF9 861          ?ST=0  rplc#
775 0BBFC 90           GOYES  vEND20  on ne remplace pas le # en L
776 0BBFE 31C4        LCASC  'L'
777 0BC02 AEA          A=C    B      remplacement effectif
778 0BC05 862          vEND20  ?ST=0  chrc/
779 0BC08 B0           GOYES  vEND25
780 0BC0A 31F2        LCASC  '/'
781 0BC0E 962          ?A=C  B
782 0BC11 41           GOYES  vEND30  il reste une ligne a traiter
783 0BC13 31FF        vEND25  LC(2)  255    -1
784 0BC17 962          ?A=C  B
785 0BC1A 74           GOYES  vEND50  on peut y aller
786 0BC1C 148          DAT0=A B      Ecriture dans le fichier
787 0BC1F 161          D0=D0+ 2
788 0BC22 57C          GONC   vEND10  B.E.T.
789
790                  *
791                  * Deuxieme ligne
792                  *
792 0BC25 137          vEND30  CD1EX    C(A) = D1
793 0BC28 109          R1=C        R1 n'est pas abime pas LIFlen
794 0BC2B 8E9D90       GOSUBL  LIFlen n'abime pas D0
795 0BC31 136          CD0EX
796 0BC34 134          D0=C
797 0BC37 108          R0=C        R0 = ^ liflen de la 2eme ligne
798 0BC3A 163          D0=D0+ 4

```

```

799 0BC3D 119          C=R1          C(A) = ^ R de RTNYES ou le G de GOYES
800 0BC40 135          D1=C
801                    *
802                    * <blanc><6 blancs><blanc>
803                    *
804 0BC43 3102          LCASC ' '
805 0BC47 28           P= 8
806 0BC49 8E3B90      vEND40 GOSUBL putc
807 0BC4F 0D          P=P-1
808 0BC51 880          ?P# 0
809 0BC54 5F          GOYES vEND40
810 0BC56 664F          GOTO vEND00 sur la premiere ligne
811
812 0BC5A 8D00000      bserr GOVLNG =BSERR
813                    *
814                    * Fin de la derniere ligne
815                    *
816 0BC61 8E3A90      vEND50 GOSUBL LIFlen
817 0BC67 136          CD0EX
818 0BC6A 1B00000      D0=(5) =AVMEMS
819 0BC71 144          DAT0=C A AVMEMS := ^ end of line
820
821 0BC74 D2           C=0 A
822 0BC76 10B          R3=C insertion dans le fichier
823
824 0BC79 1B1C6F2      D0=(5) dFILE
825 0BC80 142          A=DAT0 A A(A) := ^ old line
826 0BC83 1BFC6F2      D0=(5) dHEADR
827 0BC8A 146          C=DAT0 A
828
829 0BC8D 8F00000      GOSBVL =MGOSUB
830 0BC94 00000        CON(5) =RPLLIN
831                    *
832                    * R3 (3) = offset of move
833                    * Cy = 0 : output buffer collapsed
834                    * A(A) = end+1 of replaced line in file
835                    * B(A) = len of replacement line in nibbles
836                    * C(A) = (OUTBS)
837                    *
838 0BC99 40C          GOC bserr
839
840 0BC9C 1B1C6F2      D0=(5) dFILE
841 0BCA3 140          DAT0=A A
842                    *
843                    * Puis affichage de la nouvelle ligne
844                    *
845 0BCA6 1BEC8F2      vEND90 D0=(5) dLEN
846 0BCAD D2           C=0 A
847 0BCAF 14E          C=DAT0 B
848                    *** change
849                    * D0=(2) dADR * rev D
850 0BCB2 1A0E9F        D0=(4) PCSAV * rev X
851                    *** end of change
852 0BCB8 142          A=DAT0 A
853 0BCBB CA           A=A+C A
854 0BCBD 140          DAT0=A A
855                    *

```



```
856          * Affichage de la ligne suivante
857          *
858 0BCC0 72AA          GOSUB  dspmod
859          *
860          * Et retour...
861          *
862 0BCC4 67DC          GOTO   BOUCLE
863
864          *****
865          * vDESAS
866          *
867          * But: Traiter la touche [D] (desassemblage) a l'etat 0.
868          * Entree:
869          * Sortie: par BOUCLE
870          * Historique:
871          *   87/05/20: conception & codage
872          *****
873
874 0BCC8          vDESAS
875 0BCC8 7AAA          GOSUB  desdsp
876 0BCCC 6FCC          vDSout GOTO   BOUCLE
877
878          *****
879          * vLCASC
880          *
881          * But: Traiter la touche [L] (LCASC) a l'etat 0.
882          * Entree:
883          * Sortie: par BOUCLE
884          * Historique:
885          *   87/05/20: conception & codage
886          *****
887
888 0BCD0          vLCASC
889          *
890          * mode desassemblage ?
891          *
892 0BCD0 1B0D8F2          D0=(5) dMODE
893 0BCD7 1564          C=DAT0 S
894 0BCDB 94A          ?C=0 S
895 0BCDE EE          GOYES  vDSout
896          *
897          * LC ?
898          *
899 0BCE0 1B0E9F2          D0=(5) dADR
900 0BCE7 142          A=DAT0 A
901 0BCEA 130          D0=A          D0 := ^ instruction
902 0BCED 1520          A=DAT0 P
903 0BCF1 303          LC(1) 3
904 0BCF4 906          ?A#C P
905 0BCF7 5D          GOYES  vDSout
906          *
907          * LC(n), n pair ?          (3x, x = impair)
908          *
909 0BCF9 160          D0=D0+ 1
910 0BCFC AF2          C=0 W
911 0BCFF 1560          C=DAT0 P          x de 3x?????????
912          *
```

```

913          * Calcul de l'adresse du dernier caractere
914          *
915 0BD03 132          AD0EX
916 0BD06 CA          A=A+C  A          A(A) := ^ dernier car.
917 0BD08 130          D0=A          D0 := ^ dernier caractere
918          *
919          * Test de parite de x
920          *
921 0BD0B 822          SB=0
922 0BD0E 81E          CSRB
923 0BD11 832          ?SB=0          x pair
924 0BD14 8B          GOYES  vDSout  ==> sortie
925          *
926          * Ok. On y va.
927          *
928 0BD16 E6          C=C+1  A          C(A) := nb de caracteres
929 0BD18 D7          D=C  A          Sauvegarde en D(A)
930
931 0BD1A 1F1D8F2          D1=(5) dLINE
932          *
933          * LCASC  '
934          *
935 0BD21 3FC43414          LCASC  \'  CSACL\
          35340202
          72
936 0BD33 15DF          DAT1=C  8*2
937 0BD37 17F          D1=D1+  8*2
938          *
939          * ASC$(dADR + 2)
940          *
941 0BD3A 34EFFFF          LC(5)  (0-2)
942 0BD41 DF          CDEX  A          C(A) := nb de car. ; D(A) := -2
943
944 0BD43 7946          GOSUB  ASC$
945          *
946          *
947          *
948 0BD47 3172          LCASC  \' \
949 0BD4B 14D          DAT1=C  B
950 0BD4E 171          D1=D1+  2
951          *
952          * FF
953          *
954 0BD51 31FF          LC(2)  255          -1
955 0BD55 14D          DAT1=C  B
956
957 0BD58 731B          GOSUB  dsp
958
959 0BD5C 6F3C          vLCout GOTO  BOUCLE
960
961          *****
962          * vGOTO
963          *
964          * But: Traiter la touche [=] (GOTO) a l'etat 0.
965          * Entree:
966          * Sortie: par BOUCLE
967          * Historique:

```

```
968 * 87/05/21: P.D. conception & codage
969 *****
970
971 0BD60 vGOTO
972 0BD60 7366 GOSUB getadr
973 0BD64 421 GOC vGOout
974 0BD67 1B0E9F2 D0=(5) dADR
975 0BD6E D9 C=B A C(A) := adresse trouvee
976 0BD70 144 DAT0=C A
977 0BD73 7FE9 GOSUB dspmod
978 0BD77 642C vGOout GOTO BOUCLE
979
980 *****
981 * vGOSUB
982 *
983 * But: Traiter la touche [[] (GOSUB) a l'etat 0.
984 * Entree:
985 * Sortie: par BOUCLE
986 * Historique:
987 * 87/05/21: P.D. conception & codage
988 *****
989
990 0BD7B vGOSUB
991 0BD7B 7846 GOSUB getadr
992 0BD7F 47F GOC vGOout
993 0BD82 1B178F2 D0=(5) dSTACK
994 0BD89 1520 A=DAT0 P A(0) := pointeur de pile
995 0BD8D 307 LC(1) MAXLVL
996 0BD90 98E ?A>=C P
997 0BD93 4E GOYES vGOout
998
999 0BD95 A86 C=A P pour getstk
1000 0BD98 B04 A=A+1 P
1001 0BD9B 1500 DAT0=A P un niveau de plus
1002
1003 0BD9F 7276 GOSUB getstk D0 := ^ stack (C(0))
1004
1005 0BDA3 1F0E9F2 D1=(5) dADR
1006 0BDAA 147 C=DAT1 A C(A) := dADR
1007 0BDAD 144 DAT0=C A push dADR
1008
1009 0BDB0 D9 C=B A C(A) := adresse trouvee
1010 0BDB2 145 DAT1=C A dADR := adresse trouvee
1011
1012 0BDB5 7DA9 GOSUB dspmod
1013
1014 0BDB9 62EB GOTO BOUCLE
1015
1016 *****
1017 * vRTN
1018 *
1019 * But: Traiter la touche [)] (RTN) a l'etat 0.
1020 * Entree:
1021 * Sortie: par BOUCLE
1022 * Historique:
1023 * 87/05/21: P.D. conception & codage
1024 *****
```

```
1025
1026 0BDBD          vRTN
1027 0BDBD 1B178F2    D0=(5) dSTACK
1028 0BDC4 1560      C=DAT0 P
1029 0BDC8 90A       ?C=0 P
1030 0BDCB E1        GOYES vRTout
1031
1032 0BDCD A0E        C=C-1 P
1033 0BDD0 1540      DAT0=C P          un niveau de moins
1034
1035 0BDD4 7D36      GOSUB getstk
1036
1037 0BDD8 142        A=DAT0 A          A(A) := adresse depilee
1038 0BDDB 1F0E9F2    D1=(5) dADR
1039 0BDE2 141        DAT1=A A
1040
1041 0BDE5 7D79      GOSUB dspmod
1042
1043 0BDE9 62BB      vRTout GOTO BOUCLE
1044
1045 *****
1046 * vEDIT
1047 *
1048 * But: Traiter la touche [Z] (EDIT) a l'etat 0.
1049 * Entree:
1050 * Sortie: par BOUCLE
1051 * Historique:
1052 * 87/05/21: P.D. conception & codage
1053 *****
1054
1055 0BDED          vEDIT
1056 0BDED 777A      GOSUB dsp-
1057
1058 0BDF1 7C46      GOSUB finput format := 5UP
1059 0BDF5 FFFFFFFF  NIBHEX FFFFFFFF
1060 0BDFD FFFFFFFF  NIBHEX FFFFFFFF
1061 0BE05 FFFFFF    NIBHEX FFFFFF
1062 0BE0B 7         CON(1) 7          %0111
1063 0BE0C 0         CON(1) 0          %0000
1064 *
1065 * Traiter l'adresse entree :
1066 *
1067 0BE0D 451       GOC vEDout [ATTN] ou [OFF]
1068 0BE10 7357      GOSUB chkhex
1069 0BE14 48D       GOC vEDIT non valide
1070 0BE17 1B0E9F2  D0=(5) dADR
1071 0BE1E D4        A=B A          A(A) := sortie de chkhex
1072 0BE20 140       DAT0=A A
1073
1074 0BE23 7F39      vEDout GOSUB dspmod
1075 0BE27 647B      GOTO BOUCLE
1076
1077 *****
1078 * vMODIF
1079 *
1080 * But: Traiter la touche [M] ou [EDIT] (modif.) a l'etat 0.
1081 * Entree:
```

```
1082          * Sortie: par BOUCLE
1083          * Historique:
1084          *   87/05/21: P.D.          conception & codage
1085          *****
1086
1087 0BE2B          vMODIF
1088 0BE2B 78D9          GOSUB  stdds-
1089 0BE2F 7E06          GOSUB  finput  format := 6P16UP
1090 0BE33 FFFFFFFF          NIBHEX  FFFFFFFF
1091 0BE3B FFFFFFFF          NIBHEX  FFFFFFFF
1092 0BE43 FF          NIBHEX  FF
1093 0BE45 3          CON(1) 3          %0011
1094 0BE46 000          NIBHEX  000
1095 0BE49 C          CON(1) 12          %1100
1096 0BE4A F          NIBHEX  F
1097          *
1098          * Traiter la chaine entree :
1099          *
1100 0BE4B 4C2          GOC      vMDout  [ATTN] ou [OFF]
1101 0BE4E 7517          GOSUB  chkhex
1102 0BE52 48D          GOC      vMODIF  non valide
1103 0BE55 1B0E9F2          D0=(5) dADR
1104 0BE5C 142          A=DAT0 A
1105 0BE5F 130          D0=A          D0 := adresse courante
1106 0BE62 AF4          A=B      W          A(A) := sortie de chkhex
1107 0BE65 2F          P=      15
1108 0BE67 810          vMD10  ASLC
1109 0BE6A 1580          DAT0=A 1
1110 0BE6E 160          D0=D0+ 1
1111 0BE71 0D          vMD20  P=P-1
1112 0BE73 53F          GONC   vMD10
1113 0BE76 20          P=      0
1114
1115 0BE78 7AE8          vMDout GOSUB  dspmod
1116 0BE7C 6F1B          GOTO   BOUCLE
1117
1118          *****
1119          * vFILE
1120          *
1121          * But: Traiter la touche [F] (file) a l'etat 0.
1122          * Entree:
1123          * Sortie: par BOUCLE
1124          * Historique:
1125          *   87/05/26: P.D. & J.T. conception & codage
1126          *****
1127
1128 0BE80 D2          vFL00  C=0      A
1129 0BE82 1FFC6F2          D1=(5) dHEADR
1130 0BE89 145          DAT1=C  A
1131 0BE8C 135          D1=C          D1 := 0
1132 0BE8F 6561          GOTO   vFL90
1133
1134 0BE93          vFILE
1135 0BE93 7E00          GOSUB  vFL10
1136 0BE97 6496C656          NIBASC  'File: '
          A302
1137 0BEA3 FF          NIBHEX  FF
```

```
1138 0BEA5 07          vFL10  C=RSTK
1139 0BEA7 135         D1=C
1140 0BEAA 8F00000     GOSBVL =BF2DSP
1141
1142 0BEB1 1FFC6F2      D1=(5) dHEADR
1143 0BEB8 147         C=DAT1 A
1144 0BEBB 8AA         ?C=0  A
1145 0BEBE 60          GOYES  vFL15
1146
1147                 *
1148                 * Affichage du nom du fichier
1149                 *
1149 0BEC0 7A77         GOSUB  flname
1150 0BEC4 7975         vFL15  GOSUB  finput  format := 6P8UP
1151 0BEC8 FFFFFFFF     NIBHEX FFFFFFFF
1152 0BED0 FFFFFFFF     NIBHEX FFFFFFFF
1153 0BED8 FFFF         NIBHEX FFFF
1154 0BEDC 3           CON(1) 3           %0011
1155 0BEDD 0           CON(1) 0           %0000
1156 0BEDE C          CON(1) 12          %1100
1157 0BEDF F          CON(1) 15          %1111
1158
1159 0BEE0 479         GOC    vMDout    [ATTN] ou [OFF]
1160
1161                 *
1162                 * INPUT = NULL ?
1163                 *
1163 0BEE3 8A8         ?A=0  A
1164 0BEE6 A9         GOYES  vFL00
1165
1166                 *
1167                 * Reconstituer le string header pour FILXQ$
1168                 *
1168 0BEE8 137         CD1EX
1169 0BEEB 135         D1=C
1170 0BEEE C2         C=C+A  A          C(A) := ^ bas de la M.S.
1171 0BEF0 109         R1=C
1172 0BEF3 8F00000     GOSBVL =D=AVMS
1173 0BEFA 850         ST=1  0          return desired (un peu mon n'veu)
1174 0BEFD 8F00000     GOSBVL =ADHEAD
1175
1176 0BF04 8F00000     GOSBVL =FILXQ$
1177 0BF0B 578         GONC   vFILE     Invalid File Specification
1178
1179                 *
1180                 * Cy = 1 : mainframe recognisable file specifier found
1181                 * A(W) = file name
1182                 * D(S) = #F if device not specified
1183                 *
1183 0BF0E B47         D=D+1  S
1184 0BF11 518         GONC   vFILE     There is a device specifier
1185 0BF14 A4F         D=D-1  S          On remet tout en ordre (propre, net)
1186
1187 0BF17 1F00000     D1=(5) =FUNCR0  sauvegardes
1188 0BF1E AFB         C=D    W          Sauvegarde de D(W) (pour CRETF+)
1189 0BF21 1557        DAT1=C  W
1190 0BF25 17F         D1=D1+ 16
1191 0BF28 1517        DAT1=A  W          Sauvegarde du nom du fichier
1192
1193 0BF2C 8F00000     GOSBVL =FINDF+  Find file
1194                 *
```

```
1195          * A(W) = B(W) = file name
1196          * Cy = 0 : file found
1197          *   D1 = ^ file header
1198          *   D(S) = device type
1199          * Cy = 1 : file not found
1200          *
1201 0BF33 566          GONC   vFL30   File found   (D1 = ^ file header)
1202          *
1203          * Creation du fichier
1204          *
1205 0BF36 1F00000          D1=(5) =FUNCR0
1206 0BF3D 1577          C=DAT1 W
1207 0BF41 AF7          D=C   W           Restaurer pour CRETF+
1208 0BF44 D2          C=0   A
1209 0BF46 3192          LC(2) 37+4   header + EOF-mark
1210 0BF4A 8F00000          GOSBVL =CRETF+ Le Lex ne doit pas bouger
1211 0BF51 590          GONC   vFL20   Ok
1212 0BF54 8D00000          GOVLNG =BSERR
1213 0BF5B 119          vFL20 C=R1           C(A) := adresse du header
1214 0BF5E 1FFC6F2          D1=(5) dHEADR
1215 0BF65 145          DAT1=C A
1216 0BF68 1F00000          D1=(5) =FUNCR1
1217 0BF6F 1537          A=DAT1 W
1218 0BF73 135          D1=C
1219 0BF76 1517          DAT1=A W           Nom du fichier
1220 0BF7A 17F          D1=D1+ 16         D1 = ^ file type
1221
1222          *** *****
1223          *** changes (undated, undocumented)
1224          **           C=0   A
1225          **           C=C+1 A
1226          **           DAT1=C 4           file type := TEXT
1227 0BF7D 35100004          LCHEX 400001
1228 0BF85 15D5          DAT1=C 6
1229          *** end of changes
1230          *** *****
1231
1232 0BF89 17F          D1=D1+ 16
1233 0BF8C 174          D1=D1+ 5
1234          * EOF-Mark
1235 0BF8F D2          C=0   A
1236 0BF91 CE          C=C-1 A
1237 0BF93 15D3          DAT1=C 4           D1 := ^ fin du fichier
1238 0BF97 4D5          GOC   vFL90   B.E.T.
1239 0BF9A          vFL30
1240          *
1241          * Stockage de l'adresse du header
1242          *
1243 0BF9A 137          CD1EX
1244 0BF9D 1FFC6F2          D1=(5) dHEADR
1245 0BFA4 145          DAT1=C A
1246 0BFA7 135          D1=C
1247          *
1248          * Chercher le EOF
1249          *
1250 0BFAA 17F          D1=D1+ 16
1251          *
```

```

1252          * Tester le type
1253          *
1254 0BFAD D0          A=0      A
1255 0BFAF 15B3       A=DAT1  4
1256 0BFB3 CC        A=A-1   A
1257 0BFB5 8A8       ?A=0    A
1258 0BFB8 D0        GOYES   vFL35
1259 0BFBA 3100      LC(2)  =eFTYPE
1260 0BFBE 8D00000   GOVLNG  =MFERR
1261 0BFC5 17F       vFL35  D1=D1+ 16
1262 0BFC8 137       CD1EX
1263 0BFCB 135       D1=C          D1 := ^ REL(5) FiLeNd
1264 0BFCE 143       A=DAT1  A
1265 0BFD1 C2        C=C+A   A
1266 0BFD3 D7        D=C      A      D := ^ EOF according to file-chain
1267 0BFD5 174       D1=D1+ 5
1268 0BFD8 137       CD1EX          C := ^ start of data
1269 0BFDB 848       ST=0     sBADRC
1270 0BFDE 72D5     vFL40  GOSUB   PRSREC
1271 0BFE2 5BF       GONC    vFL40   Bouclage si Cy = 0
1272          * sortie de la boucle
1273 0BFE5 868       ?ST=0   sBADRC
1274 0BFE8 D0        GOYES   vFL90   Boucle si Ok
1275 0BFEA 3100      LC(2)  =eEOFIL  End Of File
1276 0BFEE 8D00000   GOVLNG  =MFERR
1277
1278 0BFF5 137       vFL90  CD1EX
1279 0BFF8 1F1C6F2   D1=(5) dFILE
1280 0BFFF 145       DAT1=C  A
1281 0C002 7968      GOSUB   dsp
1282 0C006 6599      GOTO    BOUCLE
1283
1284          *****
1285          * etat1
1286          *
1287          * But: Traiter le cas de la touche [R] suivie d'un nombre.
1288          * Entree:
1289          *   - dKEY = touche appuyee, censee etre un nombre 1..5
1290          * Sortie:
1291          *   - dETAT = 0
1292          *   - si nombre = 1..5
1293          *     dLINE = REL(n) #aaaaa
1294          *     dLEN = n
1295          *     dDESAS = 0
1296          *     sinon
1297          *     dREJEC = 1
1298          *     sortie par BOUCLE
1299          * Abime:
1300          * Appelle: numkey, dsp
1301          * Niveaux: 2
1302          * Historique:
1303          *   87/05/19: P.D. & J.T. conception & codage
1304          *****
1305
1306 0C00A 7273       etlrjt GOSUB  reject
1307
1308          ***** start of misc changes:

```



```

1309 0C00E          etlout
1310              *          D0=(5) dMODE      * rev D
1311 0C00E D2          C=0      A
1312              *          DAT0=C 1          * rev D
1313 0C010 67FA          GOTO    etat=C
1314
1315 0C014          etat1
1316 0C014 7B03          GOSUB  numkey
1317 0C018 51F          GONC   etlrjt
1318 0C01B 8A9          ?B=0   A
1319 0C01E CE           GOYES  etlrjt
1320 0C020 305          LC(1)  5
1321 0C023 981          ?B>C  P
1322 0C026 4E           GOYES  etlrjt
1323              *
1324              * La touche est comprise entre 1 et 5
1325              *
1326 0C028 1BEC8F2          D0=(5) dLEN
1327 0C02F D9          C=B     A
1328 0C031 14C          DAT0=C B          dLEN := valeur de la touche
1329
1330 0C034 1F1D8F2          D1=(5) dLINE
1331 0C03B 372554C4          LCASC  '(LER'
1332 0C045 15D7          DAT1=C 4*2
1333 0C049 177          D1=D1+ 4*2
1334 0C04C 3103          LCASC  '0'
1335 0C050 A69          C=C+B  B
1336 0C053 14D          DAT1=C B
1337 0C056 171          D1=D1+ 2
1338 0C059 35920232          LCASC  '# )'
1339 0C061 15D5          DAT1=C 3*2
1340 0C065 175          D1=D1+ 3*2
1341
1342              * Calcul de l'adresse
1343              *
1344              ***
1345              *          D0=(2) dADR      * rev D
1346 0C068 1A0E9F          D0=(4) PCSAV      * rev X
1347              ***
1348 0C06E 142          A=DAT0 A          A(A) := adresse courante
1349 0C071 130          D0=A
1350 0C074 D9          C=B     A          C(4-1) := 0
1351 0C076 80D0          P=C     0
1352 0C07A 0D          P=P-1
1353 0C07C 1561          C=DAT0 WP          les quartets de poids fort = 0
1354              *** change: added ver X:
1355 0C080 D5          B=C     A
1356 0C082 A06          C=C+C  P
1357 0C085 D2          C=0     A
1358 0C087 570          GONC   PLUSOF
1359 0C08A A1E          C=C-1  WP
1360 0C08D FE          C=-C-1 A
1361 0C08F 0EFD          PLUSOF C=C!B A
1362              *** end
1363 0C093 CA          A=A+C  A
1364 0C095 24          P=     4

```

```
1365 0C097 80CF          C=P    15
1366 0C09B 20           P=     0
1367 0C09D 8F00000      GOSBVL =HEXASC
1368 0C0A4 1599         DAT1=A 10
1369 0C0A8 179         D1=D1+ 10
1370 0C0AB 31FF        LC(2) 255    -1
1371 0C0AF 14D         DAT1=C B
1372
1373 0C0B2 8E7B7F      GOSUBL dsp
1374
1375 0C0B8 655F        GOTO   et1out
1376
1377
1378 * etat2
1379 *
1380 * But: Traiter le cas de la touche [A] suivie d'un nombre.
1381 * Entree:
1382 *   - dKey = touche appuyee, censee etre un nombre 1..8
1383 * Sortie:
1384 *   - dETAT = 0
1385 *   - si nombre = 1..8
1386 *     dLINE = NIBASC '....'
1387 *     dLEN = n*2
1388 *     dDESAS = 0
1389 *     sinon
1390 *     dREJEC = 1
1391 *     sortie par BOUCLE
1392 * Abime:
1393 * Appelle: numkey, dsp
1394 * Niveaux: 2
1395 * Historique:
1396 *   87/05/19: P.D. & J.T. conception & codage
1397 *****
1398
1399 0C0BC 6D4F          et2rjt GOTO   et1rjt
1400
1401 0C0C0              etat2
1402 0C0C0 7F52        GOSUB   numkey
1403 0C0C4 57F         GONC    et2rjt
1404 0C0C7 8A9         ?B=0   A
1405 0C0CA 2F         GOYES   et2rjt
1406 0C0CC 308        LC(1)   8
1407 0C0CF 981        ?B>C   P
1408 0C0D2 AE         GOYES   et2rjt
1409 *
1410 * La touche est comprise entre 1 et 8
1411 *
1412 0C0D4 1BEC8F2      D0=(5) dLEN
1413 0C0DB D9          C=B    A
1414 0C0DD C6          C=C+C  A
1415 0C0DF 14C        DAT0=C B      dLEN := n*2
1416
1417 0C0E2 1F1D8F2      D1=(5) dLINE
1418 0C0E9 3FE49424    LCASC   \' CSABIN\
      14353402
      72
1419 0C0FB 1557        DAT1=C W
```

```

1420 0C0FF 17F          D1=D1+ 16
1421
1422          ***
1423          *          D0=(2) dADR      * rev D
1424 0C102 1A0E9F      D0=(4) PCSAV      * rev X
1425          ***
1426 0C108 146          C=DAT0 A
1427 0C10B 134          D0=C              D0 := ^ caracteres
1428
1429 0C10E D3           D=0      A
1430 0C110 E7           D=D+1    A
1431 0C112 E7           D=D+1    A          D(A) := 2
1432 0C114 D9           C=B      A          C(A) := nb de car
1433 0C116 7672        GOSUB   ASC$
1434
1435 0C11A 3172          LCASC   '\ '
1436 0C11E 14D          DAT1=C B
1437 0C121 171          D1=D1+ 2
1438          *
1439          * FF
1440          *
1441 0C124 31FF          LC(2)   255      -1
1442 0C128 14D          DAT1=C B
1443
1444 0C12B 8EE37F        GOSUBL  dsp
1445
1446 0C131 6CDE          GOTO    et1out
1447
1448          *****
1449          * etat3
1450          *
1451          * But: traiter (pour CON(n)) le "n" ou le [H].
1452          * Entree:
1453          *   - dKEY = touche pressee
1454          * Sortie:
1455          *   - si touche = [H]
1456          *     dSC1 = 1
1457          *     sinon
1458          *     dSC1 = 0
1459          *     dREJEC = 1
1460          *   - dETAT = 5
1461          * Abime: C(A), D0
1462          * Appelle: reject
1463          * Niveaux: 1
1464          * Historique:
1465          *   87/05/19: P.D. & J.T. conception & codage
1466          * *****
1467
1468 0C135          etat3
1469 0C135 1F598F2      D1=(5) dSC1
1470 0C13C 1BAC8F2      D0=(5) dKEY
1471 0C143 14A          A=DAT0 B
1472 0C146 3141        LC(2)   #14      Touche [H]
1473 0C14A 966          ?A#C    B
1474 0C14D 80           GOYES   et3rjt
1475 0C14F 301          LC(1)   1          dSC1 := 1
1476 0C152 580          GONC    et3out    B.E.T.

```

```
1477 0C155 7722      et3rjt GOSUB reject
1478 0C159 D2         C=0   A       dSC1 := 0
1479 0C15B 1550      et3out DAT1=C P
1480 0C15F 305       LC(1) 5
1481 0C162 65A9      GOTO   etat=C
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503 0C166            etat4
1504 0C166 1F598F2   D1=(5) dSC1
1505 0C16D 1BAC8F2   D0=(5) dKEY
1506 0C174 14A       A=DAT0 B
1507 0C177 3163     LC(2) #36     Touche [.]
1508 0C17B 966      ?A#C   B
1509 0C17E 80        GOYES  et4rjt
1510 0C180 30A       LC(1) 10     dSC1 := 10
1511 0C183 580       GONC   et4out B.E.T.
1512 0C186 76F1     et4rjt GOSUB reject
1513 0C18A D2        C=0   A       dSC1 := 0
1514 0C18C 1550     et4out DAT1=C P
1515 0C190 306       LC(1) 6
1516 0C193 6479     GOTO   etat=C
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532 0C197 627E      et5rjt GOTO   et1rjt
1533
```

```
*****
* etat4
*
* But: traiter (pour NIBHEX) le "n" ou le [.]
* Entree:
*   - dKEY = touche pressee
* Sortie:
*   - si touche = [.]
*     dSC1 = 10
*     sinon
*     dSC1 = 0
*     dREJEC = 1
*   - dETAT = 6
* Abime: C(A), D0
* Appelle: reject
* Niveaux: 1
* Historique:
*   87/05/19: P.D. & J.T. conception & codage
*****
```

```
*****
* etat5
*
* But: traiter le cas general de CON().
* Entree:
*   - dKEY = touche pressee.
*   - dSC1 = 1 si mode [C][H]n, 0 si [C]n
* Sortie: -
* Abime:
* Appelle: numkey, dsp
* Historique:
*   87/05/19: P.D. & J.T. conception & codage
*****
```

```

1534 0C19B          etat5
1535 0C19B 7481      GOSUB  numkey
1536 0C19F 57F       GONC   et5rjt
1537 0C1A2 8A9       ?B=0   A
1538 0C1A5 2F        GOYES  et5rjt
1539 0C1A7 306       LC(1)  6
1540 0C1AA 981       ?B>C   P
1541 0C1AD AE        GOYES  et5rjt
1542 0C1AF A0E       C=C-1  P      C(0) := 5
1543 0C1B2 1B598F2   D0=(5) dSC1
1544 0C1B9 985       ?B<C   P
1545 0C1BC 60        GOYES  et5-10
1546 0C1BE 1540      DAT0=C  P      mode hexa := vrai (vrai = 5)
1547 0C1C2          et5-10
1548                *
1549                * dLEN := "n"
1550                *
1551 0C1C2 19EC       D0=(2) dLEN
1552 0C1C6 D9         C=B     A
1553 0C1C8 14C       DAT0=C  B      dLEN := n
1554                *
1555                * dLINE := CON(n) #
1556                *
1557                * CON(
1558 0C1CB 1F1D8F2     D1=(5) dLINE
1559 0C1D2 3734F4E4   LCASC   '(NOC'
      82
1560 0C1DC 15D7       DAT1=C  4*2
1561 0C1E0 177       D1=D1+ 4*2
1562                *      n
1563 0C1E3 3103       LCASC   '0'
1564 0C1E7 A69        C=C+B   B
1565 0C1EA 14D        DAT1=C  B
1566 0C1ED 171       D1=D1+  2
1567                *      ) #
1568 0C1F0 35920232   LCASC   '# )'
1569 0C1F8 15D5       DAT1=C  3*2
1570 0C1FC 175       D1=D1+  3*2
1571                *
1572                * # ou pas # ?
1573                *
1574 0C1FF 1959       D0=(2) dSC1
1575 0C203 1564      C=DAT0  S
1576
1577                ***
1578                *      D0=(2) dADR   * rev D
1579 0C207 1A0E9F     D0=(4) PCSAV * rev X
1580                ***
1581 0C20D 146        C=DAT0  A
1582 0C210 134        D0=C          D0 := adresse courante
1583
1584 0C213 AF0         A=0         W
1585 0C216 D9         C=B         A      C(0) := 1..6
1586 0C218 D7         D=C         A      D(A) := sauvegarde de la longueur
1587 0C21A 80D0       P=C         0
1588 0C21E 0D         P=P-1       P = nb quartets a charger - 1
1589 0C220 AF0        A=0         W

```

```

1590 0C223 1521          A=DAT0 WP
1591
1592 0C227 94E          ?C#0  S      Mode hexa ?
1593 0C22A 62           GOYES  et5-50
1594                    *
1595                    * non : on revient sur le #,
1596                    *
1597 0C22C 1C1          D1=D1- 2
1598                    *
1599                    * Et on convertit en decimal
1600                    *
1601 0C22F 8F00000      GOSBVL =HEXDEC  A(W) := nombre en decimal
1602 0C236 04           SETHEX
1603 0C238 2F           P=      15
1604 0C23A 90C         et5-20 ?A#0  P
1605 0C23D 90           GOYES  et5-40
1606 0C23F 0D           P=P-1
1607 0C241 880          ?P#    0
1608 0C244 6F           GOYES  et5-20
1609 0C246 D2          et5-40 C=0    A
1610 0C248 80C0         C=P    0
1611 0C24C E6           C=C+1  A
1612 0C24E D7           D=C    A
1613 0C250 80CF         et5-50 C=P    15
1614 0C254 20           P=      0
1615 0C256 8F00000      GOSBVL =HEXASC
1616 0C25D DB           C=D    A      C(A) := 1..6
1617 0C25F C6           C=C+C  A      C(A) := 2..12
1618 0C261 80D0         P=C    0
1619 0C265 0D           P=P-1
1620 0C267 1511         DAT1=A WP
1621 0C26B 20           P=      0
1622                    *
1623                    * Calcul de l'adresse suivante
1624                    *
1625 0C26D 133          AD1EX
1626 0C270 CA           A=A+C  A
1627 0C272 131         D1=A
1628                    *
1629                    * FF
1630                    *
1631 0C275 31FF         LC(2)  255      -1
1632 0C279 14D         DAT1=C  B
1633                    *
1634 0C27C 8EDE5F       GOSUBL  dsp
1635
1636 0C282 6B8D         GOTO   et1out
1637
1638                    *****
1639                    * etat6
1640                    *
1641                    * But: traiter le cas general de NIBHEX.
1642                    * Entree:
1643                    *   - dKEY = touche pressee.
1644                    *   - dSC1 = 10 si mode [N][.]n, 0 si [N]n
1645                    * Sortie: -
1646                    * Abime:

```

```
1647      * Appelle: numkey, dsp
1648      * Historique:
1649      *   87/05/19: P.D. & J.T. conception & codage
1650      *****
1651
1652 0C286 638D      et6rjt GOTO   et1rjt
1653
1654 0C28A          etat6
1655 0C28A 7590          GOSUB  numkey
1656 0C28E 57F          GONC   et6rjt
1657 0C291 1B598F2      D0=(5) dSC1
1658      *
1659      * [.] ?
1660      *
1661 0C298 AF2          C=0    W
1662 0C29B 1560         C=DAT0 P
1663 0C29F 90A         ?C=0   P
1664 0C2A2 41          GOYES  et6-10  pas de [.]
1665      *
1666      * Il y a un [.]
1667      * Valide entre 0 et 6
1668      *
1669 0C2A4 306          LC(1)  6
1670 0C2A7 981          ?B>C   P
1671 0C2AA CD           GOYES  et6rjt
1672 0C2AC D2          C=0    A
1673 0C2AE 30A         LC(1)  10
1674 0C2B1 C1          B=B+C  A      B(A) := n + 10
1675 0C2B3 570         GONC   et6-20  B.E.T.
1676      *
1677      * pas de [.]
1678      * Valide entre 1 et 9
1679      *
1680 0C2B6 8A9         et6-10 ?B=0   A
1681 0C2B9 DC          GOYES  et6rjt
1682      *
1683      * Ok. Longueur entre 1 et 16 dans B(A).
1684      *
1685 0C2BB          et6-20
1686      *
1687      * dLEN := longueur calculee
1688      *
1689 0C2BB 1BEC8F2      D0=(5) dLEN
1690 0C2C2 D9          C=B    A
1691 0C2C4 D7          D=C    A      D(A) := longueur calculee
1692 0C2C6 14C         DAT0=C B
1693      *
1694      * D0 := adresse courante
1695      *
1696      *   D0=(2) dADR   rev D
1697 0C2C9 1A0E9F      D0=(4) PCSAV  rev X
1698      ***** end of misc changes
1699
1700 0C2CF 146          C=DAT0 A
1701 0C2D2 134         D0=C
1702      *
1703      * NIBHEX
```

```

1704      *
1705 0C2D5 1F1D8F2          D1=(5) dLINE
1706 0C2DC 3DE49424      LCASC  ' XEHBIN'
          84548502
1707 0C2EC 15DD          DAT1=C 7*2
1708 0C2F0 17D          D1=D1+ 7*2
1709      *
1710      * Les B(A) nibhex.
1711      *
1712 0C2F3 591          GONC  et6-40  B.E.T.
1713 0C2F6 15A0      et6-30 A=DAT0 1
1714 0C2FA AC2          C=0  S
1715 0C2FD 8F00000      GOSBVL =HEXASC
1716 0C304 149          DAT1=A B
1717 0C307 160          D0=D0+ 1
1718 0C30A 171          D1=D1+ 2
1719 0C30D CF      et6-40 D=D-1  A
1720 0C30F 56E          GONC  et6-30
1721      *
1722      * FF
1723      *
1724 0C312 31FF          LC(2)  255  -1
1725 0C316 14D          DAT1=C B
1726      *
1727      * et affichage...
1728      *
1729 0C319 8E055F      GOSUBL dsp
1730
1731 0C31F 6EEC          GOTO  etlout
1732
1733      *****
1734      * numkey
1735      *
1736      * But: tester si la touche dKEY est une touche numerique, et
1737      * renvoyer sa valeur.
1738      * Entree:
1739      * - dKEY = code physique de la touche
1740      * Sortie:
1741      * - Cy = 1 : touche numerique
1742      *   B(A) = valeur (0000d, d=0..9)
1743      * - Cy = 0 : touche non numerique
1744      * Abime: A(B), B(A), C(A), D0
1745      * Appelle: FINDA
1746      * Niveaux: 1
1747      * Historique:
1748      * 87/05/19: P.D. & J.T. conception & codage
1749      *****
1750
1751 0C323      numkey
1752 0C323 D1          B=0  A
1753 0C325 1BAC8F2      D0=(5) dKEY
1754 0C32C 14A          A=DAT0 B
1755 0C32F 8F00000      GOSBVL =FINDA
1756 0C336 D0          CON(2) #0D  [9]
1757 0C338 430          REL(3) num9
1758 0C33B C0          CON(2) #0C  [8]
1759 0C33D 130          REL(3) num8

```



```
1760 0C340 B0          CON(2) #0B      [7]
1761 0C342 E20        REL(3) num7
1762 0C345 B1          CON(2) #1B      [6]
1763 0C347 B20        REL(3) num6
1764 0C34A A1          CON(2) #1A      [5]
1765 0C34C 820        REL(3) num5
1766 0C34F 91          CON(2) #19      [4]
1767 0C351 520        REL(3) num4
1768 0C354 92          CON(2) #29      [3]
1769 0C356 220        REL(3) num3
1770 0C359 82          CON(2) #28      [2]
1771 0C35B F10        REL(3) num2
1772 0C35E 72          CON(2) #27      [1]
1773 0C360 C10        REL(3) num1
1774 0C363 53          CON(2) #35      [0]
1775 0C365 910        REL(3) num0
1776 0C368 00          NIBHEX 00
1777 0C36A 03          RTNCC          Cy := 0 : non trouvee
1778
1779 0C36C E5          num9  B=B+1  A
1780 0C36E E5          num8  B=B+1  A
1781 0C370 E5          num7  B=B+1  A
1782 0C372 E5          num6  B=B+1  A
1783 0C374 E5          num5  B=B+1  A
1784 0C376 E5          num4  B=B+1  A
1785 0C378 E5          num3  B=B+1  A
1786 0C37A E5          num2  B=B+1  A
1787 0C37C E5          num1  B=B+1  A
1788 0C37E 02          num0  RTNSC          Cy := 1 : trouvee
1789
1790 *****
1791 * reject
1792 *
1793 * But: mettre a 1 la variable REJECT
1794 * Entree: -
1795 * Sortie:
1796 * - dREJEC = 1
1797 * Abime: C(0), D0
1798 * Niveaux: 0
1799 * Historique:
1800 * 87/05/19: P.D. & J.T. conception & codage
1801 *****
1802
1803 0C380          reject
1804 0C380 1BDC8F2      D0=(5) dREJEC
1805 0C387 301          LC(1) 1
1806 0C38A 1540        DAT0=C P
1807 0C38E 01          RTN
1808
1809 *****
1810 * ASC$
1811 *
1812 * But: afficher C(A) caracteres ASCII (entre 32 et 126)
1813 * pointes par D0.
1814 * Entree:
1815 * - D0 = ^ caracteres
1816 * - D1 = ^ destination
```

```

1817      *   - C(A) = compte de caracteres (et non de quartets)
1818      *   - D(A) = increment de D0 (+2 ou -2)
1819      *   Sortie:
1820      *   - D1 = ^ passe le dernier caractere
1821      *   Abime: A(A), B(A), C(A)
1822      *   Appelle: RANGE
1823      *   Niveaux: 1
1824      *   Historique:
1825      *   87/05/19: P.D. & J.T. conception & codage
1826      *****
1827
1828 0C390 AE5      ASC$   B=C   B           compteur en B(A)
1829 0C393 6B20      GOTO  ASC40
1830 0C397 3302E7   ASC10  LC(4)  (126)~(32)
1831 0C39D 14A      A=DAT0 B
1832 0C3A0 149      DAT1=A B           par default, on l'ecrit
1833 0C3A3 8F00000  GOSBVL =RANGE
1834 0C3AA 590      GONC   ASC30
1835 0C3AD 31E2      LCASC  '.'
1836 0C3B1 14D      DAT1=C B
1837 0C3B4 171      ASC30  D1=D1+ 2
1838 0C3B7 136      CD0EX           Reactualisation de D0
1839 0C3BA CB       C=C+D  A
1840 0C3BC 134      D0=C
1841 0C3BF A6D      ASC40  B=B-1 B
1842 0C3C2 54D      GONC   ASC10
1843 0C3C5 01      RTN
1844
1845      *****
1846      *   getadr
1847      *
1848      *   But: extraire une adresse a partir de la ligne contenue
1849      *   dans dLINE.
1850      *   Entree:
1851      *   - dLINE = ligne courante
1852      *   Sortie:
1853      *   - Cy = 1 : adresse non trouvee.
1854      *   - Cy = 0 : adresse trouvee :
1855      *   B(A) = adresse
1856      *   Abime: A(A), D1
1857      *   Niveaux: 0
1858      *   Historique:
1859      *   87/05/21: P.D.           conception & codage
1860      *****
1861
1862 0C3C7      getadr
1863 0C3C7 1F1D8F2   D1=(5) dLINE
1864 0C3CE 3142      LC(2)  '#' +1
1865 0C3D2 14B      geta10 A=DAT1 B
1866 0C3D5 171      D1=D1+ 2           D1 := ^ caractere suivant
1867 0C3D8 B64      A=A+1  B
1868 0C3DB 400      RTNC           Sortie si #FF (fin de dLINE)
1869 0C3DE 966      ?A#C  B
1870 0C3E1 1F      GOYES  geta10     boucle si pas de #
1871      *
1872 0C3E3 D1      B=0  A
1873 0C3E5 D0      A=0  A

```

```

1874 0C3E7 14B      geta20 A=DAT1 B
1875 0C3EA 171          D1=D1+ 2
1876 0C3ED B64          A=A+1 B
1877 0C3F0 422          GOC   geta90   Sortie si #FF
1878 0C3F3 A6C          A=A-1 B
1879 0C3F6 8F00000      GOSBVL =DRANGE Cy := 0 si 0..9
1880 0C3FD 3103          LC(2) '0'
1881 0C401 560          GONC   geta30
1882 0C404 3173          LC(2) 'A'-10
1883 0C408 B6A      geta30 A=A-C B      A(B) := 0..15
1884 0C40B F1          BSL   A
1885 0C40D C8          B=B+A A
1886 0C40F 67DF          GOTO  geta20
1887
1888 0C413 03      geta90 RTNCC
1889
1890
1891 *****
1891 * getstk
1892 *
1893 * But:
1894 * Entree:
1895 * - C(0) = niveau (0 pour le 1er niveau)
1896 * Sortie:
1897 * - D0 = ^ element dans la pile
1898 * Abime: C, P, D0
1899 * Niveaux: 0
1900 * Historique:
1901 * 87/05/21: P.D.      conception & codage
1902 *****
1903
1904 0C415      getstk
1905 0C415 80D0      P=C   0      P = 0..6
1906 0C419 1B178F2      D0=(5) dSTACK
1907 0C420 160      D0=D0+ 1
1908 0C423 136      CD0EX      C(A) := ^ stack (0)
1909 0C426 0D      P=P-1
1910 0C428 411      GOC   gets10   C(A) := ^ stack (0)
1911 0C42B 809      C+P+1
1912 0C42E 809      C+P+1
1913 0C431 809      C+P+1
1914 0C434 809      C+P+1
1915 0C437 809      C+P+1      C(A) := ^ stack (P)
1916 0C43A 20      gets10 P=   0
1917 0C43C 134      D0=C
1918 0C43F 01      RTN
1919
1920 *****
1921 * finput
1922 *
1923 * But: poker la bit-map, avertir le poll de FINPUT, CHEDIT,
1924 * avertir le poll de FINPUT, mettre la chaine sur la M.S.
1925 * Entree:
1926 * - RSTK = ^ bit map
1927 * - affichage ok
1928 * Sortie:
1929 * - Cy = 1 :Sortie par [ATTN] ou [OFF]
1930 * - Cy = 0 :Sortie par [ENDLINE]

```

```

1931      *      D1 = ^ Chaine au sommet de la M.S. (dernier caractere)
1932      *      A(A) = longueur en quartets
1933      *      C(A) = longueur en octets
1934      * Abime: Tout
1935      * Niveaux: 7 (CHEDIT)
1936      * Detail: La bit-map est codee comme suit :
1937      *      GOSUB finput
1938      *      CON(1) %xxxx      93-94-95-96 caracteres
1939      *      :
1940      *      CON(1) %xxxx      4-3-2-1 caracteres
1941      * Historique:
1942      *      87/05/22: P.D. & J.T. conception & codage
1943      *      *****
1944
1945 0C441      finput
1946 0C441 07      C=RSTK
1947 0C443 135      D1=C      D1 := ^ Bit-map
1948 0C446 1B00000      D0=(5) =DSPMSK
1949 0C44D 1577      C=DAT1 W
1950 0C451 1547      DAT0=C W
1951 0C455 17F      D1=D1+ 16
1952 0C458 16F      D0=D0+ 16
1953 0C45B 15F7      C=DAT1 8
1954 0C45F 15C7      DAT0=C 8
1955
1956 0C463 177      D1=D1+ 8      D1 := adresse de retour
1957 0C466 137      CD1EX
1958 0C469 06      RSTK=C
1959
1960 0C46B 8F00000      GOSBVL =CURSFL
1961 0C472      edit
1962      *
1963      * Previent le poll handler que c'est nous qu'on travaille
1964      *
1965      *** change. (undated, undocumented)
1966 0C472 1BFB6F2      D0=(5) #2F6BF
1967 0C479 AC0      A=0 S
1968 0C47C 1504      DAT0=A S
1969      *** end of change
1970 0C480 1B199F2      D0=(5) RESJPC
1971 0C487 15A0      A=DAT0 1
1972 0C48B 301      LC(1) INSMK
1973
1974 0C48E 0E0E      A=A!C P
1975 0C492 1580      DAT0=A 1
1976
1977 0C496 22      P= 2
1978 0C498 8F00000      GOSBVL =R<RSTK
1979
1980 0C49F      edit10
1981 0C49F 8F00000      GOSBVL =CHEDIT
1982 0C4A6 58F      GONC edit10 Si DEF KEY "...", "...":
1983      *
1984      * Previent le poll handler que c'est nous qu'on travaille plus
1985      *
1986 0C4A9 30E      LC(1) nINSMK 'INSMK
1987 0C4AC 816      CSRC

```

```

1988 0C4AF 1B199F2          D0=(5) RESJPC
1989 0C4B6 1524            A=DAT0 S
1990 0C4BA 0E46            A=A&C S
1991 0C4BE 1504            DAT0=A S
1992                        *
1993 0C4C2 22              P=      2
1994 0C4C4 8F00000         GOSBVL =RSTK<R  Ne modifie pas A(B)
1995
1996 0C4CB 8F00000         GOSBVL =FINDA
1997 0C4D2 D0              CON(2) 13      [ENDLINE]
1998 0C4D4 020            REL(3) fnpEND
1999 0C4D7 E0              CON(2) 14      [ATTN]
2000 0C4D9 E00            REL(3) fnpATN
2001 0C4DC 81             CON(2) 24      [OFF]
2002 0C4DE 900            REL(3) fnpATN
2003 0C4E1 00             NIBHEX 00
2004 0C4E3 6E8F          GOTO   edit
2005
2006 0C4E7 7630           fnpATN GOSUB   finlgn
2007 0C4EB 8F00000         GOSBVL =ATNCLR
2008 0C4F2 02             RTNSC          Cy := 1 : [ATTN]
2009
2010 0C4F4 7920           fnpEND GOSUB   finlgn
2011 0C4F8 8F00000         GOSBVL =D1MSTK
2012 0C4FF 850            ST=1    0      retour apres DSP$00
2013 0C502 8F00000         GOSBVL =DSP$00
2014 0C509 8F00000         GOSBVL =POP1S
2015 0C510 171            D1=D1+ 2      POPpe le CR de la fin
2016 0C513 CC             A=A-1  A
2017 0C515 CC             A=A-1  A      A(A) := longueur en quartets
2018 0C517 AD2            C=0    M
2019 0C51A D6             C=A    A
2020 0C51C 81E            CSRB          C(A) := longueur en octets
2021 0C51F 01             RTN          Cy := 0 : [ENDLINE]
2022
2023 0C521 1F00000         finlgn D1=(5) =DSPCHX
2024 0C528 143            A=DAT1 A
2025 0C52B 8A8            ?A=0  A
2026 0C52E E2             GOYES  crlfof
2027 0C530 1B00000         D0=(5) =ESCSTA
2028 0C537 301            LC(1)  EscSt1
2029 0C53A 15C0           DAT0=C 1
2030 0C53E 3130           LC(2)  3
2031 0C542 AEA            A=C    B
2032 0C545 147            C=DAT1 A
2033 0C548 7710           GOSUB  pc=c
2034 0C54C 20             P=     0
2035 0C54E 1B00000         D0=(5) =ESCSTA
2036 0C555 300            LC(1)  EscSt0
2037 0C558 15C0           DAT0=C 1
2038 0C55C 8D00000         crlfof GOVLNG =CRLFOF
2039
2040 0C563 06             pc=c   RSTK=C
2041 0C565 01             RTN
2042
2043                        *****
2044                        * chkhex

```

```
2045      *
2046      * But: verifier que la chaine au sommet de la M.S. ne
2047      *   ne contient que des chiffres hexadecimaux.
2048      * Entree:
2049      *   - D1 = ^ dernier caractere sur la M.S.
2050      *   - A(A) = longueur en quartets
2051      *   - C(A) = longueur en octets
2052      * Sortie:
2053      *   - Cy = 1 : adresse non valide
2054      *   - Cy = 0 : adresse non valide
2055      *     B(W) = valeur lue (16 quartets au maximum)
2056      * Abime:
2057      * Niveaux: 0
2058      * Historique:
2059      *   87/05/22: P.D. & J.T. conception & codage
2060      *****
2061
2062 0C567 D8      chkhex B=A      A
2063 0C569 133      AD1EX          A(A) := ^ dernier caractere
2064 0C56C C0      A=A+B A      A(A) := ^ premier caractere
2065 0C56E 131      D1=A A      D1 := ^ premier caractere
2066 0C571 D7      D=C A      D(A) := compteur de boucle
2067 0C573 AF1      B=0 W
2068 0C576 6630      GOTO chkh30
2069 0C57A 1C1      chkh10 D1=D1- 2
2070 0C57D 14B      A=DAT1 B      A(B) := caractere
2071 0C580 BF1      BSL W
2072 0C583 8F00000 GOSBVL =DRANGE
2073 0C58A 5F1      GONC chkh20      Cy = 0 : byte in range
2074 0C58D 8F00000 GOSBVL =CONVUC
2075 0C594 331464      LCASC 'FA'
2076 0C59A 8F00000 GOSBVL =RANGE
2077 0C5A1 400      RTNC          Cy = 1 : non entre 0..9 ou A..F
2078 0C5A4 309      LC(1) 9
2079 0C5A7 A0A      A=A+C P
2080 0C5AA A88      chkh20 B=A P
2081 0C5AD CF      chkh30 D=D-1 A
2082 0C5AF 5AC      GONC chkh10
2083 0C5B2 03      RTNCC          Tout s'est bien passe...
2084
2085      *****
2086      * PRSREC
2087      *
2088      * Voir IDS I, page 17-35.
2089      * Sortie:
2090      *   - Cy = 0 : il faut continuer la boucle
2091      *   - Cy = 1 : il faut sortir
2092      *     sBADRC = 1 : bad record
2093      *     sBADRC = 0 : EOF reached
2094      *           D1 = ^ debut de la lere ligne inexistante
2095      * Historique:
2096      *   87/05/28: P.D. & J.T. modification de l'interface
2097      *****
2098
2099 0C5B4 D1      PRSREC B=0      A
2100 0C5B6 135      D1=C
2101 0C5B9 8BF      ?C>=D A
```

```

2102 0C5BC 00          RTNYES          Return if EOF instead of line length
2103 0C5BE 173        D1=D1+ 4
2104 0C5C1 137        CD1EX
2105 0C5C4 8B7        ?C>D  A
2106 0C5C7 43         GOYES  PRSR20   Bad record (len a cheval sur la fin)
2107 0C5C9 15B3       A=DAT1 4
2108 0C5CD 8F00000    GOSBVL =SWPBYT
2109 0C5D4 23         P=      3
2110 0C5D6 A98        B=A     WP
2111 0C5D9 D9         C=B     A
2112 0C5DB B15        B=B+1  WP
2113 0C5DE 20         P=      0
2114 0C5E0 400        RTNC          Cy = 1 si FFFF
2115 0C5E3 DD         BCEX     A
2116 0C5E5 81E        CSRB
2117 0C5E8 E6         C=C+1  A
2118 0C5EA C6         C=C+C  A
2119 0C5EC C6         C=C+C  A
2120 0C5EE 133        AD1EX
2121 0C5F1 131        D1=A
2122 0C5F4 C2         C=A+C  A
2123 0C5F6 8BB        ?C<=D  A
2124 0C5F9 70         GOYES  rtncc
2125 0C5FB 858        PRSR20 ST=1   sBADRC
2126 0C5FE 02         RTNSC          On ne peut sortir
2127 0C600 03        rtncc  RTNCC
2128
2129 *****
2130 *  putc
2131 *
2132 *  But: mettre un caractere dans l'Output Buffer
2133 *  Entree:
2134 *    - D0 = ^ output buffer
2135 *    - C(B) = caractere
2136 *  Sortie:
2137 *    - C(B) = caractere
2138 *    - D0 = ^ caractere suivant
2139 *  Abime: D0
2140 *  Niveaux: 0
2141 *  Historique:
2142 *    87/05/28: P.D. & J.T. conception & codage
2143 *****
2144
2145 0C602 14C        putc  DAT0=C B
2146 0C605 161        D0=D0+ 2
2147 0C608 01        RTN
2148
2149 *****
2150 *  LIFlen
2151 *
2152 *  But: calculer et stocker la longueur LIF de la ligne
2153 *    introduite dans l'Output Buffer.
2154 *  Entree:
2155 *    - R0 = ^ debut de la ligne (sur la longueur LIF)
2156 *    - D0 = ^ passe le dernier caractere de la ligne
2157 *  Sortie:
2158 *    - Longueur calculee stockee en (R0)

```

```

2159      * - D0 avance le cas echeant pour compter le padding
2160      * Abime: A-C, D0, D1
2161      * Appelle: SWPBYT
2162      * Niveaux: 1
2163      * Historique:
2164      * 87/05/28: P.D. & J.T. conception & codage
2165      *****
2166
2167 0C60A 118      LIFlen C=R0          C(A) := ^ debut ligne (longueur)
2168 0C60D 135          D1=C
2169 0C610 173          D1=D1+ 4
2170 0C613 137          CD1EX          C(A) := ^ debut donnees ; D1 = ^ len
2171 0C616 132          AD0EX          A(A) := fin de la ligne
2172 0C619 130          D0=A
2173 0C61C EA          A=A-C A          A(A) := longueur ligne en quartets
2174 0C61E 81C          ASRB          A(3-0) := long. en octets
2175 0C621 D8          B=A A          B(A) := longueur en octets
2176 0C623 8F00000    GOSBVL =SWPBYT
2177 0C62A 1593      DAT1=A 4          La longueur exacte en octets
2178 0C62E 822          SB=0
2179 0C631 81D          BSRB          SB = 1 si longueur impaire
2180 0C634 832          ?SB=0
2181 0C637 00          RTNYES
2182 0C639 161          D0=D0+ 2          Padding si longueur impaire
2183 0C63C 01          RTN
2184
2185      *****
2186      * filename
2187      *
2188      * But: afficher un nom de fichier.
2189      * Entree:
2190      * - C(A) = adresse du file-header
2191      * Sortie: par CURSFL
2192      * Historique:
2193      * 87/05/28: P.D. & J.T. extraction de vFILE
2194      *****
2195
2196 0C63E 1B00000    filename D0=(5) =FUNCD0
2197 0C645 144          DAT0=C A          FUNCD0 = ^ header
2198 0C648 164          D0=D0+ 5
2199 0C64B 308          LC(1) 8
2200
2201 0C64E 15C0      flnm10 DAT0=C 1          FUNCD1 := reste a envoyer
2202 0C652 184          D0=D0- 5
2203 0C655 146          C=DAT0 A
2204 0C658 135          D1=C          D1 := ^ dans le file-name
2205 0C65B 14B          A=DAT1 B
2206
2207 0C65E 171          D1=D1+ 2
2208 0C661 137          CD1EX
2209 0C664 144          DAT0=C A          FUNCD0 += 2 ;
2210
2211 0C667 3102      LCASC ' '
2212 0C66B 962          ?A=C B
2213 0C66E C1          GOYES flnm20
2214 0C670 8F00000    GOSBVL =DSPCHA
2215 0C677 1B00000    D0=(5) =FUNCD1

```



```
2216 0C67E 15E0          C=DAT0 1
2217 0C682 A0E          C=C-1  P
2218 0C685 90E          ?C#0   P
2219 0C688 6C           GOYES  flnm10
2220
2221 0C68A 8D00000      flnm20 GOVLNG =CURSFL
2222
2223 0C691              END
```

ADDRCK	Ext		-	413							
ADHEAD	Ext		-	1174							
ASC\$	Abs	50064	#0000C390	-	1828	944	1433				
ASC10	Abs	50071	#0000C397	-	1830	1842					
ASC30	Abs	50100	#0000C3B4	-	1837	1834					
ASC40	Abs	50111	#0000C3BF	-	1841	1829					
ATNCLR	Ext		-	575	2007						
AVMEMS	Ext		-	818							
BF2DSP	Ext		-	1140							
BOUCLE	Abs	47516	#0000B99C	-	444	526	561	651	664	862	876
					959	978	1014	1043	1075	1116	1282
BP05	Abs	47549	#0000B9BD	-	456	459					
BP10	Abs	47570	#0000B9D2	-	460	457					
BP20	Abs	47589	#0000B9E5	-	465	450	455				
BSERR	Ext		-	812	1212						
Bserr	Abs	47922	#0000BB32	-	684	698					
CHEDIT	Ext		-	1981							
CKSREQ	Ext		-	458							
COLLAP	Ext		-	174							
CONVUC	Ext		-	2074							
CRETf+	Ext		-	1210							
CRLFND	Ext		-	376							
CRLFOf	Ext		-	2038							
CURSF1	Ext		-	1960	2221						
D1MSTK	Ext		-	2011							
D=AVMS	Ext		-	1172							
DRANGE	Ext		-	1879	2072						
DSP\$00	Ext		-	2013							
DSPCHA	Ext		-	368	2214						
DSPCHC	Ext		-	323							
DSPCHX	Ext		-	2023							
DSPCNA	Ext		-	318							
DSPMSK	Ext		-	1948							
ESCSTA	Ext		-	2027	2035						
EXPEX-	Ext		-	412							
EscSt0	Abs	0	#00000000	-	28	2036					
EscSt1	Abs	1	#00000001	-	29	2028					
Etat5	Abs	47628	#0000BA0C	-	476	473					
Etat6	Abs	47634	#0000BA12	-	477	474					
FILXQ\$	Ext		-	1176							
FINDA	Ext		-	482	1755	1996					
FINDF+	Ext		-	1193							
FUNCD0	Ext		-	328	333	2196					
FUNCD1	Ext		-	2215							
FUNCR0	Ext		-	313	1187	1205					
FUNCR1	Ext		-	1216							
HEXASC	Ext		-	260	312	718	1367	1615	1715		
HEXDEC	Ext		-	1601							
INSMsk	Abs	1	#00000001	-	25	1972					
KEYBUF	Ext		-	126							
KEYPTR	Ext		-	121							
LIFlen	Abs	50698	#0000C60A	-	2167	794	816				
Ldcspc	Abs	194241	#0002F6C1	-	80	87					
Lexptr	Abs	194255	#0002F6CF	-	81	88					
MAXLVL	Abs	7	#00000007	-	98	995					
MEMCKL	Ext		-	697							
MFERR	Ext		-	1260	1276						

desdsp	Abs	46966	#0000B776	-	170	875						
dsp	Abs	47215	#0000B86F	-	302	231	957	1281	1373	1444	1634	
					1729							
dsp-	Abs	47208	#0000B868	-	299	1056						
dsp10	Abs	47218	#0000B872	-	303	279	300					
dsp20	Abs	47302	#0000B8C6	-	332	357	369					
dsp50	Abs	47353	#0000B8F9	-	359	355						
dsp60	Abs	47364	#0000B904	-	363	361						
dsp65	Abs	47376	#0000B910	-	368	352	365					
dsp70	Abs	47387	#0000B91B	-	373	344						
dspmod	Abs	46950	#0000B766	-	145	559	858	977	1012	1041	1074	
					1115							
eEOFIL	Ext			-	1275							
eFTYPE	Ext			-	1259							
edit	Abs	50290	#0000C472	-	1961	2004						
edit10	Abs	50335	#0000C49F	-	1980	1982						
etlout	Abs	49166	#0000C00E	-	1309	1375	1446	1636	1731			
et1rjt	Abs	49162	#0000C00A	-	1306	1317	1319	1322	1399	1532	1652	
et2rjt	Abs	49340	#0000C0BC	-	1399	1403	1405	1408				
et3out	Abs	49499	#0000C15B	-	1479	1476						
et3rjt	Abs	49493	#0000C155	-	1477	1474						
et4out	Abs	49548	#0000C18C	-	1514	1511						
et4rjt	Abs	49542	#0000C186	-	1512	1509						
et5-10	Abs	49602	#0000C1C2	-	1547	1545						
et5-20	Abs	49722	#0000C23A	-	1604	1608						
et5-40	Abs	49734	#0000C246	-	1609	1605						
et5-50	Abs	49744	#0000C250	-	1613	1593						
et5rjt	Abs	49559	#0000C197	-	1532	1536	1538	1541				
et6-10	Abs	49846	#0000C2B6	-	1680	1664						
et6-20	Abs	49851	#0000C2BB	-	1685	1675						
et6-30	Abs	49910	#0000C2F6	-	1713	1720						
et6-40	Abs	49933	#0000C30D	-	1719	1712						
et6rjt	Abs	49798	#0000C286	-	1652	1656	1671	1681				
etat0	Abs	47640	#0000BA18	-	479	468						
etat1	Abs	49172	#0000C014	-	1315	469						
etat2	Abs	49344	#0000C0C0	-	1401	470						
etat3	Abs	49461	#0000C135	-	1468	471						
etat4	Abs	49510	#0000C166	-	1503	472						
etat5	Abs	49563	#0000C19B	-	1534	476						
etat6	Abs	49802	#0000C28A	-	1654	477						
etat=C	Abs	47880	#0000BB08	-	648	591	605	619	1313	1481	1516	
finlgn	Abs	50465	#0000C521	-	2023	2006	2010					
finput	Abs	50241	#0000C441	-	1945	1058	1089	1150				
flname	Abs	50750	#0000C63E	-	2196	1149						
flnm10	Abs	50766	#0000C64E	-	2201	2219						
flnm20	Abs	50826	#0000C68A	-	2221	2213						
fnpATN	Abs	50407	#0000C4E7	-	2006	2000	2002					
fnpEND	Abs	50420	#0000C4F4	-	2010	1998						
geta10	Abs	50130	#0000C3D2	-	1865	1870						
geta20	Abs	50151	#0000C3E7	-	1874	1886						
geta30	Abs	50184	#0000C408	-	1883	1881						
geta90	Abs	50195	#0000C413	-	1888	1877						
getadr	Abs	50119	#0000C3C7	-	1862	972	991					
gets10	Abs	50234	#0000C43A	-	1916	1910						
getstk	Abs	50197	#0000C415	-	1904	1003	1035					
k#ATTN	Ext			-	483							
k#OFF	Ext			-	523							

nINSMK	Abs	-2	#FFFFFFFE	-	26	1986			
num0	Abs	50046	#0000C37E	-	1788	1775			
num1	Abs	50044	#0000C37C	-	1787	1773			
num2	Abs	50042	#0000C37A	-	1786	1771			
num3	Abs	50040	#0000C378	-	1785	1769			
num4	Abs	50038	#0000C376	-	1784	1767			
num5	Abs	50036	#0000C374	-	1783	1765			
num6	Abs	50034	#0000C372	-	1782	1763			
num7	Abs	50032	#0000C370	-	1781	1761			
num8	Abs	50030	#0000C36E	-	1780	1759			
num9	Abs	50028	#0000C36C	-	1779	1757			
numkey	Abs	49955	#0000C323	-	1751	1316	1402	1535	1655
oB91E	Abs	47454	#0000B95E	-	417	407			
pc=c	Abs	50531	#0000C563	-	2040	2033			
putc	Abs	50690	#0000C602	-	2145	710	712	722	806
reject	Abs	50048	#0000C380	-	1803	1306	1477	1512	
rplc#	Abs	1	#00000001	-	43	731	760	774	
rtnc	Abs	50688	#0000C600	-	2127	2124			
sBADRC	Abs	8	#00000008	-	39	1269	1273	2125	
sBLANC	Abs	1	#00000001	-	33	330	356	359	362
sCRLF	Abs	3	#00000003	-	35	249	251	299	302 374
sGUIL	Abs	2	#00000002	-	34	331	351	366	
std10	Abs	47147	#0000B82B	-	258	265			
std20	Abs	47170	#0000B842	-	264				
std10	Abs	47121	#0000B811	-	252	250			
std10	Abs	47111	#0000B807	-	249	1088			
std10	Abs	46946	#0000B762	-	143	148			
std10	Abs	47118	#0000B80E	-	251	143	443	663	
tEOL	Ext			-	405				
v+1	Abs	47768	#0000BA98	-	545	486			
v+16	Abs	47790	#0000BAAE	-	549	490			
v-1	Abs	47779	#0000BAA3	-	547	488			
v-16	Abs	47801	#0000BAB9	-	551	492			
vCON	Abs	47870	#0000BAFE	-	618	498			
vDESAS	Abs	48328	#0000BCC8	-	874	516			
vDSout	Abs	48332	#0000BCCC	-	876	895	905	924	
vEDIT	Abs	48621	#0000BDED	-	1055	508	1069		
vEDout	Abs	48675	#0000BE23	-	1074	1067			
vEND00	Abs	48029	#0000BB9D	-	731	810			
vEND03	Abs	48075	#0000BBCB	-	753	742			
vEND04	Abs	48088	#0000BBDB	-	756	752			
vEND05	Abs	48091	#0000BBDB	-	761	748	751	755	
vEND10	Abs	48106	#0000BBEA	-	769	764	788		
vEND20	Abs	48133	#0000BC05	-	778	773	775		
vEND25	Abs	48147	#0000BC13	-	783	779			
vEND30	Abs	48165	#0000BC25	-	792	782			
vEND40	Abs	48201	#0000BC49	-	806	809			
vEND50	Abs	48225	#0000BC61	-	816	785			
vEND90	Abs	48294	#0000BCA6	-	845	682			
vENDLN	Abs	47903	#0000BB1F	-	677	522			
vENDf1	Abs	47926	#0000BB36	-	686	681			
vEXIT	Abs	47831	#0000BAD7	-	573	484	524		
vFILE	Abs	48787	#0000BE93	-	1134	520	1177	1184	
vFL00	Abs	48768	#0000BE80	-	1128	1164			
vFL10	Abs	48805	#0000BEA5	-	1138	1135			
vFL15	Abs	48836	#0000BEC4	-	1150	1145			
vFL20	Abs	48987	#0000BF5B	-	1213	1211			

vFL30	Abs	49050	#0000BF9A	-	1239	1201		
vFL35	Abs	49093	#0000BFC5	-	1261	1258		
vFL40	Abs	49118	#0000BFDE	-	1270	1271		
vFL90	Abs	49141	#0000BFF5	-	1278	1132	1238	1274
vGOSUB	Abs	48507	#0000BD7B	-	990	512		
vGOTO	Abs	48480	#0000BD60	-	971	510		
vGOout	Abs	48503	#0000BD77	-	978	973	992	997
vHEXA	Abs	47895	#0000BB17	-	663	506		
vLCASC	Abs	48336	#0000BCD0	-	888	518		
vLCout	Abs	48476	#0000BD5C	-	959			
vMD10	Abs	48743	#0000BE67	-	1108	1112		
vMD20	Abs	48753	#0000BE71	-	1111			
vMDout	Abs	48760	#0000BE78	-	1115	1100	1159	
vMODIF	Abs	48683	#0000BE2B	-	1087	502	504	1102
vNIBA	Abs	47863	#0000BAF7	-	604	496		
vNIBH	Abs	47877	#0000BB05	-	632	500		
vREL	Abs	47856	#0000BAF0	-	590	494		
vRTN	Abs	48573	#0000BDBD	-	1026	514		
vRTout	Abs	48617	#0000BDE9	-	1043	1030		

Input Parameters

Source file name is sysed.a

Listing file name is sysed.l

Object file name is sysed.o

Flags set on command line
None

Errors

None

```

1          TITLE JPC LEX: MODULE POSI
2
3          * *****
4          * Posi module
5          *
6          * Keyword(s):
7          *   POSI
8          * Author(s):
9          *   Pierre David, Jean-Pierre Bondu, Janick Taillandier
10         * History:
11         *   Published in JPC #37
12         *   Changed in rev X (R. Rosenbaum)
13         *
14         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
15         * *****
16
17 0C691          ABS      #C691
18
19         * *****
20         * POSI entry
21         * *****
22 0C691 CC423          NIBHEX CC423
23 0C696 D0          =POSIE A=0      A          le 3Éme vaudra 255 par dÅfaut
24 0C698 A6C          A=A-1    B          A[A]=000FF
25 0C69B 80DF          P=C      15          C[S]= nombre d'arguments
26 0C69F 892          ?P= 2          2 arguments?
27 0C6A2 60          GOYES    PRM2
28 0C6A4 77A0          GOSUB   POPARG
29 0C6A8 100          PRM2     R0=A
30 0C6AB 70A0          GOSUB   POPARG   A[A]= 2Éme argument
31 0C6AF 118          C=R0      B          C[A]= 1er      "
32 0C6B2 9E6          ?A>C     B
33 0C6B5 50          GOYES    FD1
34 0C6B7 AEE          ACEX     B
35 0C6BA F0          FD1      ASL     A          en entrÅe :
36         *          *          A[B]=MAX(A,C) ; C[B]=MIN
37 0C6BC F0          ASL      A          A[3-2]= borne sup
38 0C6BE AEA          A=C      B          A[B] = borne inf
39 0C6C1 D8          B=A      A          B[3-0]= intervalle
40 0C6C3 8F00000      GOSBVL =POP1S   A[A]= long, D1 @ dernier caractÉre
41
42         *** change
43         * rev D
44         *   ?A=0      A
45         *   GOYES    ENDF
46         * rev X
47 0C6CA 8AC          ?A#0    A
48 0C6CD 60          GOYES    FD2
49 0C6CF 60A0          GOTO    ivaerr
50         *** end of change
51
52 0C6D3 1C1          FD2      D1=D1- 2
53 0C6D6 137          CD1EX
54 0C6D9 135          D1=C
55 0C6DC C2          C=C+A   A          C[A] @ 1er caractÉre
56 0C6DE 8F00000      GOSBVL =CSLC5
57 0C6E5 137          CD1EX   C[A] @ dernier caractÉre

```



```

58 0C6E8 AF7          D=C    W    D[9-0]= nb de contr le :
59                    *    *    adr sup [9-5], adr inf [4-0]
60 0C6EB C2          C=C+A  A    on commence la recherche
61                    *    *      partir du 1er caract re (adr sup)
62 0C6ED 137         CD1EX    D1= pointeur de cha ne
63                    *    (adr sup en entr e)
64
65 0C6F0 14B         SEARCH A=DAT1 B    A[B]= octet de la cha ne
66 0C6F3 D9          C=B    A    C[A]= intervalle de recherche
67 0C6F5 8F00000    GOSBVL =RANGE caract re dans l'intervalle ?
68 0C6FC 522         GONC   FND   oui
69 0C6FF 1C1         D1=D1- 2    octet suivant
70 0C702 137         CD1EX
71 0C705 135         D1=C
72 0C708 8B7         ?C>D  A    pointeur > adr inf ?
73 0C70B 5E          GOYES  SEARCH oui: on continue la recherche
74 0C70D D0          NOTFND A=0    A    pas trouv : on retourne 0
75 0C70F AFB         C=D    W
76 0C712 8F00000    GOSBVL =CSRC5
77 0C719 135         D1=C    D1= adr inf= adr de base de la MStk
78 0C71C 5E1         GONC   ENDF  fin
79
80 0C71F AF0         FND    A=0    W    trouv : on va calculer la position
81                    *    *    du pointeur dans la cha ne
82 0C722 133         AD1EX    A[A]= pointeur
83 0C725 AFB         C=D    W
84 0C728 8F00000    GOSBVL =CSRC5 C[A]= adr du 1er caract re
85 0C72F 135         D1=C
86 0C732 EA          A=A-C  A
87 0C734 F8          A=-A   A    A[A]= adr sup - pointeur
88 0C736 81C         ASRB    A[A]= A[A]/2
89                    *    *    (nb quartets -> nb octets)
90 0C739 E4          A=A+1  A    position [0...] ->
91                    *    *    position [1...]
92
93 0C73B 8F00000    ENDF   GOSBVL =HDFLT position : A[A] hex ->
94                    *    *    d cimal flottant en A[W]
95 0C742 AF6         C=A    W
96 0C745 171         D1=D1+ 2
97 0C748 8D00000    GOVLNG =FNRTN1 renvoie C[W] au BASIC
98
99 0C74F 1534        POPARG A=DAT1 S
100 0C753 B44        A=A+1  S
101 0C756 402        GOC    POPAR2
102 0C759 8F00000    GOSBVL =RNDAHX A[A]= nombre hexa
103 0C760 5F0        GONC   ivaerr erreur si n gatif
104 0C763 17F        D1=D1+ 16    pointe l'argument suivant
105                    *    *    de la Math Stack
106 0C766 D2          C=0    A
107 0C768 AE6        C=A    B
108 0C76B 8A2        ?C=A  A    argument <= 255 ?
109 0C76E 00         RTNYES oui: retour
110 0C770 8D00000    ivaerr GOVLNG =IVAERR
111 0C777 8F00000    POPAR2 GOSBVL =POP1S
112 0C77E 137        CD1EX
113 0C781 C2          C=C+A  A
114 0C783 135        D1=C

```

```
115 0C786 8A8      ?A=0  A
116 0C789 00      RTNYES
117 0C78B 1C1      D1=D1- 2
118 0C78E 14B      A=DAT1 B
119 0C791 171      D1=D1+ 2
120 0C794 01      RTN
121
122 0C796          END
```

CSLC5	Ext		-	56		
CSRC5	Ext		-	76	84	
ENDF	Abs	51003	#0000C73B	-	93	78
FD1	Abs	50874	#0000C6BA	-	35	33
FD2	Abs	50899	#0000C6D3	-	52	48
FND	Abs	50975	#0000C71F	-	80	68
FNRTN1	Ext		-	97		
HDFLT	Ext		-	93		
IVAERR	Ext		-	110		
NOTFND	Abs	50957	#0000C70D	-	74	
POP1S	Ext		-	40	111	
POPAR2	Abs	51063	#0000C777	-	111	101
POPARG	Abs	51023	#0000C74F	-	99	28
=POSIE	Abs	50838	#0000C696	-	23	30
PRM2	Abs	50856	#0000C6A8	-	29	27
RANGE	Ext		-	67		
RNDAHx	Ext		-	102		
SEARCH	Abs	50928	#0000C6F0	-	65	73
ivaerr	Abs	51056	#0000C770	-	110	49
						103

Input Parameters

Source file name is posi.a

Listing file name is posi.l

Object file name is posi.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE EDIT
2
3          * *****
4          * Extended edit module
5          *
6          * Keyword(s):  EDIT
7          * Author(s):
8          *   Jean-Pierre Bondu, Pierre David, Michel Martinet
9          * History:
10         *   Initially published in JPC #37
11         *   Undated and undocumented changes
12         *
13         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14         * *****
15
16 0C796          ABS      #C796
17
18         * *****
19         * EDIT entry
20         * *****
21 0C796 00000          REL(5) =EDITp
22 0C79B 3400000 =EDITE LC(5) =lFILSV RÅserve 50 quartets
23 0C7A2 21          P=      1          pour la SAVSTK
24 0C7A4 8F00000          GOSBVL =SALLOC (= zone de sauvegarde)
25 0C7AB 486          GOC      jp          INSUFFICIENT MEMORY
26 0C7AE AF3          CPY010 D=0      W          PÅriphÅrique initialisÅ
27 0C7B1 330202          LCASC ' '          CaractÉres 9 et 10
28 0C7B7 108          R0=C          du nom du fichier ÅffacÅs
29 0C7BA 14A          A=DAT0 B          Lit token suivant
30
31         * *****
32         * COPY TO / COPY <spec> TO ...
33         * *****
34 0C7BD 3100          LC(2) =tTO
35 0C7C1 966          ?A#C      B          Token <> TO ?
36 0C7C4 11          GOYES      CPY030
37 0C7C6 161          D0=D0+ 2          Saute TO token
38 0C7C9 870          ?ST=1     =sDEST   Destination ?
39 0C7CC E1          GOYES      CPY040   Analyse N2:P2
40
41         * *****
42         *
43         * COPY TO ...
44         * Pas de fichier source spÅcifiÅ
45         * Prend le fichier et spÅcificateur courant
46         *
47         * *****
48 0C7CE 7A61          CPY020 GOSUB deffil N1:P1 <-- courant
49 0C7D2 544          GONC      CPY065   B.E.T.
50
51         * *****
52         *
53         * EDIT avec/sans paramÉtres / EDIT <spec> EOL
54         *
55         *   Si Source
56         *     EDIT workfile
57         *   Sinon      (CPY045)
```

```
58      *      N2:P2 <-- 0
59      *      Si N1 pas en mÃmoire
60      *      Recopier N1 (CPY080)
61      *      EDIT N1 (edit80)
62      *
63      *
64      *
65 0C7D5 310F  CPY030  LCHEX  F0      Token <> EOL ?
66 0C7D9 9E2      ?A<C  B
67 0C7DC E0      GOYES  CPY040  oui
68 0C7DE 870      ?ST=1  =sDEST  Destination ?
69 0C7E1 F7      GOYES  CPY045
70
71      *      *
72      *      * EDIT workfile
73      *      *
74 0C7E3 8D00000  GOVLNG =EDITWF
75
76      *
77      * COPY ... CARD
78      *
79 0C7EA 3100  CPY040  LC(2)  =tCARD  CARD token ?
80 0C7EE 966      ?A#C  B
81 0C7F1 51      GOYES  CPY060  non
82 0C7F3 300      LC(1)  =dCARD
83 0C7F6 A87      D=C    P      PÃriphÃrique= :CARD
84 0C7F9 161  CPY050  D0=D0+ 2      Saute token
85 0C7FC AF0      A=0    W      Pas de nom
86 0C7FF 594      GONC   CPY070  B.E.T.
87
88 0C802 62A0  edit79  GOTO   EDIT79  ** added
89
90 0C806 7C51  CPY060  GOSUB  tkysck  KEYS token ?
91 0C80A 5C0      GONC   CPY065  oui
92 0C80D 8F00000  GOSBVL =FSPECx  Analyse spÃcificateur
93 0C814 4DE      jp     GOC    edit79  ILLEGAL FILESPEC
94 0C817      CPY065
95      ** change: these lines added. (undated and undocumented):
96 0C817 1F00000  D1=(5) =TRFMBF
97 0C81E 978      ?A=0   W
98 0C821 60      GOYES  CPY066
99 0C823 1517     DAT1=A  W
100 0C827 17F     CPY066  D1=D1+ 16
101 0C82A 3200F   LCHEX  F00
102 0C82F 1553   DAT1=C  X
103 0C833 AFB     C=D    W
104 0C836 812     CSLC
105 0C839 ACB     C=D    S
106 0C83C A46     C=C+C  S
107 0C83F 460     GOC    CPY069
108 0C842 1553   DAT1=C  X
109 0C846     CPY069
110      ** end of change
111 0C846 813     DSLC      SpÃcificateur en D[A] et
112 0C849 8F00000  CPY070  GOSBVL =SVINFO  nom en A[W]&R0 sauvÃs.
113 0C850 870     ?ST=1  =sDEST  Destination ?
114 0C853 90      GOYES  CPY071  alors analyse terminÃe
```

```
115 0C855 850          ST=1  =sDEST  Sinon il reste
116 0C858 655F        GOTO  CPY010  ... TO N2:P2 È examiner
117 0C85C 63C0        CPY071 GOTO  CPY80+
118
119 *****
120 *
121 * EDIT  N1:P1
122 *
123 * Entrée:  sDEST=1 ; D[W]=0
124 *****
125 *
126 0C860 161          CPY045  D0=D0+ 2
127 0C863 AF0          A=0    W      Pas de fichier objet précisé
128 0C866 8F00000      GOSBVL =SVINFO N2:P2 <-- 0
129 0C86D 840          ST=0    =sDEST
130 0C870 8F00000      GOSBVL =RDINFO A[W] <-- N1 ; D[A] <-- P1
131 0C877 817          DSRC      D[S]= spécifiqueur
132 0C87A AFB          C=D     W
133 0C87D 10B          R3=C
134 0C880 8F00000      GOSBVL =FINDF+ N1 en mémoire ?
135 0C887 545          GONC    edit80  oui
136 0C88A 866          ?ST=0  6      N1= 0 ou P1= externe ?
137 0C88D 16          GOYES  CPY080
138 0C88F 11B          C=R3      Crée le fichier N1
139 0C892 AF7          D=C     W      en respectant P1
140 0C895 102          R2=A
141 0C898 D2           C=0     A
142 0C89A 3113         LC(2)   (#20)+( #11)  (=oFLENh)+(=oBSsod)
143 0C89E 8F00000      GOSBVL =CRETf+
144 0C8A5 444          EDIT79  GOC    cpyerx
145 0C8A8 111          A=R1
146 0C8AB 131          D1=A
147 0C8AE 11A          C=R2
148 0C8B1 15D0         DAT1=C  =lFNAMh
149 0C8B5 170          D1=D1+  =lFNAMh
150 0C8B8 330000      LC(4)   =fBASIC
151 0C8BE 15D0         DAT1=C  =lFTYPh
152 0C8C2 17F          D1=D1+  4+4+6+2  (=lFTYPh)+(=lTIMEh)+(=lDATEh)+(=lFLAGh)
153 0C8C5 170          D1=D1+  =lFLENh
154 0C8C8 AF2          C=0     W
155 0C8CB 15D9         DAT1=C  10
156 0C8CF 179          D1=D1+  10
157 0C8D2 3100        LC(2)   =tEOL
158 0C8D6 14D          DAT1=C  B
159 0C8D9 131          D1=A
160
161 *          *
162 *          * Edite le fichier pointé par D1
163 *          *
164 0C8DC 8F00000      edit80  GOSBVL =EDIT80
165 0C8E3 8D00000      GOVLNG =NXTSTM  FIN
166
167 0C8EA 6A40          cpyerx  GOTO  CPYERX  ** added
168
169 *****
170 *
171 * COPY N1:P1 TO N2:P2  et  EDITE N2:P2
```

```
172          *
173          * 1) Recopie N1:P1 dans N2:P2
174          * 2) Puis Ådite N2
175          *
176          * EntrÅe: suppose P2= pÅriph. interne
177          *****
178          *
179 0C8EE 8F00000 CPY080  GOSBVL =COPYu
180 0C8F5 4F3      GOC  CPYERX  Erreur
181          ** changes (undated and undocumented):
182          **      C=R1          R1 ^ fichier copiÅ
183          **      D1=C          D1 <-- R1
184          **      GONC  edit80  B.E.T.
185 0C8F8 1F00000      D1=(5) =TRFMBF
186 0C8FF 1537        A=DAT1 W
187 0C903 17F        D1=D1+ 16
188 0C906 1573        C=DAT1 X
189 0C90A D7         D=C  A
190 0C90C 817        DSRC
191 0C90F 8F00000    GOSBVL =FINDF
192 0C916 55C        GONC  edit80
193 0C919 8D00000    GOVLNG =CORUPT
194          ** end of changes
195
196          *      * Teste si P2= PÅriphÅrique interne
197          *      *      oui -> CPY080
198          *      *      non -> ILLEGAL FILESPEC
199          *      * EntrÅe: D[A] = :P2
200          *      *
201 0C920 817        CPY80+ DSRC          D[S]= type de P2
202 0C923 B47        D=D+1  S          P2 indÅfini ?
203 0C926 47C        GOC  CPY080  MAIN par dÅfaut
204 0C929 A47        D=D+D  S          externe ?
205 0C92C 51C        GONC  CPY080  non
206 0C92F 330000    LC(4)  =eFSPEC  ILLEGAL FILE SPECIFIER
207 0C935 8D00000  CPYERX  GOVLNG =BSERR  Basic Error
208
209
210 0C93C 8F00000  deffil  GOSBVL =CURDVC  Trouve pÅriphÅrique
211 0C943 8F00000  GOSBVL =FLDEVX  courant
212 0C94A 1F00000  D1=(5) =CURRST
213 0C951 147      C=DAT1 A
214 0C954 135      D1=C          D1 @ sommet fichier courant
215 0C957 1537      A=DAT1 W          A[W] <-- nom du fichier
216 0C95B 330202    LCASC  '  '
217 0C961 108      R0=C          cf. CPY010
218 0C964 03      RTNCC
219
220 0C966 14A      tkysck  A=DAT0 B
221 0C969 3100     LC(2)  =tKEYS
222 0C96D 966     ?A#C  B
223 0C970 00      RTNYES
224 0C972 161     D0=D0+ 2          Saute token
225 0C975 3FB65697 LCASC  '  syek'
      37020202
      02
226 0C987 AC3     D=0  S
```



```
227 0C98A AFA          A=C      W
228 0C98D 01          RTN
229
230 0C98F             END
```

BSERR	Ext		-	207				
COPYu	Ext		-	179				
CORUPT	Ext		-	193				
CPY010	Abs	51118	#0000C7AE	-	26	116		
CPY020	Abs	51150	#0000C7CE	-	48			
CPY030	Abs	51157	#0000C7D5	-	65	36		
CPY040	Abs	51178	#0000C7EA	-	79	39	67	
CPY045	Abs	51296	#0000C860	-	126	69		
CPY050	Abs	51193	#0000C7F9	-	84			
CPY060	Abs	51206	#0000C806	-	90	81		
CPY065	Abs	51223	#0000C817	-	94	49	91	
CPY066	Abs	51239	#0000C827	-	100	98		
CPY069	Abs	51270	#0000C846	-	109	107		
CPY070	Abs	51273	#0000C849	-	112	86		
CPY071	Abs	51292	#0000C85C	-	117	114		
CPY080	Abs	51438	#0000C8EE	-	179	137	203	205
CPY80+	Abs	51488	#0000C920	-	201	117		
CPYERX	Abs	51509	#0000C935	-	207	167	180	
CRETf+	Ext		-	143				
CURDVC	Ext		-	210				
CURRST	Ext		-	212				
EDIT79	Abs	51365	#0000C8A5	-	144	88		
EDIT80	Ext		-	164				
EDITWF	Ext		-	74				
=EDITe	Abs	51099	#0000C79B	-	22			
EDITp	Ext		-	21				
FINDF	Ext		-	191				
FINDF+	Ext		-	134				
FLDEVX	Ext		-	211				
FSPECx	Ext		-	92				
NXTSTM	Ext		-	165				
RDINFO	Ext		-	130				
SALLOC	Ext		-	24				
SVINFO	Ext		-	112	128			
TRFMBF	Ext		-	96	185			
cpyerx	Abs	51434	#0000C8EA	-	167	144		
dCARD	Ext		-	82				
deffil	Abs	51516	#0000C93C	-	210	48		
eFSPEC	Ext		-	206				
edit79	Abs	51202	#0000C802	-	88	93		
edit80	Abs	51420	#0000C8DC	-	164	135	192	
fBASIC	Ext		-	150				
jp	Abs	51220	#0000C814	-	93	25		
lFILSV	Ext		-	22				
lFLENh	Ext		-	153				
lFNAMh	Ext		-	148	149			
lFTYPh	Ext		-	151				
sDEST	Ext		-	38	68	113	115	129
tCARD	Ext		-	79				
tEOL	Ext		-	157				
tKEYS	Ext		-	221				
tTO	Ext		-	34				
tkysck	Abs	51558	#0000C966	-	220	90		

Input Parameters

Source file name is edit.a

Listing file name is edit.l

Object file name is edit.o

Flags set on command line
None

Errors

None

```
1          TITLE JPC LEX: MODULE PK
2
3          * *****
4          * Unrestricted peek/poke module
5          *
6          * Keyword(s):
7          *   PEEK$, POKE
8          * Author(s):
9          *   Pierre David, Laurent Istria, Michel Marinet
10         * History:
11         *   Originally published in JPC#23
12         *
13         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
14         * *****
15
16         * --LEX header--
17 00000 10      =PKLEX  CON(2) #01
18 00002 F3      CON(2) #3F
19 00004 04      CON(2) #40
20 00006 00000  REL(5) =ILMSG  next LEX
21 0000B F       NIBHEX F
22 0000C 0200   REL(4) (TxTSt2)+1
23 00010 0000   CON(4) 0      no msg
24 00014 00000  CON(5) 0      no poll handler
25
26         * --- main table---
27 00019 000    CON(3) (PEEKt)-(TxTSt2)
28 0001C DE000  REL(5) PEEKe
29 00021 F      NIBHEX F
30 00022 D00    CON(3) (POKEt)-(TxTSt2)
31 00025 B2000  REL(5) POKEe
32 0002A D      NIBHEX D
33
34         * --- text table---
35 0002B        TxTSt2
36 0002B 9      PEEKt  CON(1) #9
37 0002C 055454B4  NIBASC 'PEEK$'
38           42
39 00036 F3     CON(2) #3F
40 00038 7      POKEt  CON(1) #7
41 00039 05F4B454  NIBASC 'POKE'
42 00041 04     CON(2) #40
43 00043 1FF    TxTEn2 NIBHEX 1FF
44
45         * *****
46         * POKE entry
47         * *****
48 00046 00000  REL(5) =STdd
49 0004B 00000  REL(5) =POKEp
50 00050 8F00000  POKEe  GOSBVL =EXPEX-
51 00057 8F00000  GOSBVL =ADDRCK
52 0005E 1F00000  D1=(5) =S-R0-0
53 00065 D4      A=B    A
54 00067 141    DAT1=A  A
55 0006A 161    D0=D0+  2
56 0006D 8F00000  GOSBVL =EXPEX-
```

```

57 00074 8F00000      GOSBVL =POP1S
58 0007B AD2          C=0    M
59 0007E D6           C=A    A
60 00080 81E          CSRB
61 00083 D7           D=C    A
62 00085 1B00000     D0=(5) =S-R0-0
63 0008C 142          A=DAT0 A
64 0008F 131          D1=A
65 00092 1B00000     D0=(5) =TFORN
66 00099 142          A=DAT0 A
67 0009C 130          D0=A
68 0009F 6910         GOTO   POKE2
69 000A3 181          POKE1  D0=D0- 2
70 000A6 AC2          C=0    S
71 000A9 14A          A=DAT0 B
72 000AC 7C10         GOSUB  Aschex
73 000B0 D4           A=B    A
74 000B2 1590         DAT1=A 1
75 000B6 170          D1=D1+ 1
76 000B9 CF           POKE2  D=D-1  A
77 000BB 57E          GONC   POKE1
78 000BE 8D00000     GOVLNG =NXTSTM
79
80 000C5 8D00000     Err    GOVLNG =ARGERR
81
82 000CC 8F00000     Aschex GOSBVL =DRANGE *
83 000D3 401          GOC    ATH *
84 000D6 BF1          ATH1   BSL    W * * Routine Acshex, non support e
85 000D9 A88          B=A    P * A * par Hewlett-Packard.
86 000DC BF4          ASR    W * * Adresse 1C5E3 dans la version
87 000DF BF4          ASR    W * S * :1BBBB
88 000E2 01           RTN *
89 * * * * * C
90 000E4 8F00000     ATH    GOSBVL =CONVUC *
91 000EB 331464       LCASC  'FA' * H
92 000F1 8F00000     GOSBVL =RANGE *
93 000F8 4CC          GOC    Err * E
94 000FB 3173         LCHEX  37 *
95 000FF B6A          A=A-C  B * X
96 00102 53D          GONC   ATH1 *
97
98
99 * * * * *
100 * PEEK$ entry
101 * * * * *
102 00105 8422         NIBHEX 8422
103 00109 8F00000     PEEKe  GOSBVL =POP1R
104 00110 17F          D1=D1+ 16
105 00113 136          CD0EX
106 00116 1B00000     D0=(5) =F-R0-0
107 0011D 144          DAT0=C A
108 00120 8F00000     GOSBVL =FLTDH
109 00127 5D9          GONC   Err
110 0012A D6           C=A    A
111 0012C C6           C=C+C  A
112 0012E 469         GOC    Err
113 00131 100         R0=A

```

```
114 00134 8F00000      GOSBVL =ADDRCK
115 0013B 118          C=R0
116 0013E D7           D=C    A
117 00140 DA           A=C    A
118 00142 C9           C=C+B  A
119 00144 109          R1=C
120 00147 D9           C=B    A
121 00149 108          R0=C
122 0014C C4           A=A+A  A
123 0014E D6           C=A    A
124 00150 8F00000      GOSBVL =STRHDR
125 00157 110          A=R0
126 0015A 130          D0=A
127 0015D CF          PEEK1  D=D-1  A
128 0015F 4C1          GOC    PEEK2
129 00162 AC2          C=0    S
130 00165 15A0         A=DAT0 1
131 00169 8F00000      GOSBVL =HEXASC
132 00170 149          DAT1=A  B
133 00173 160          D0=D0+ 1
134 00176 171          D1=D1+ 2
135 00179 53E          GONC   PEEK1  (BET)
136 0017C 119          PEEK2  C=R1
137 0017F 135          D1=C
138 00182 20           P=     0
139 00184 8F00000      GOSBVL =REV$
140 0018B 1B00000      D0=(5) =F-R0-0
141 00192 146          C=DAT0  A
142 00195 134          D0=C
143 00198 8D00000      GOVLNG =EXPR
144
145 0019F              END
```

ADDRCK	Ext		-	51	114		
ARGERR	Ext		-	80			
ATH	Rel	228 #000000E4	-	90	83		
ATH1	Rel	214 #000000D6	-	84	96		
Aschex	Rel	204 #000000CC	-	82	72		
CONVUC	Ext		-	90			
DRANGE	Ext		-	82			
EXPEX-	Ext		-	50	56		
EXPR	Ext		-	143			
Err	Rel	197 #000000C5	-	80	93	109	112
F-R0-0	Ext		-	106	140		
FLTDH	Ext		-	108			
HEXASC	Ext		-	131			
ILMSG	Ext		-	20			
NXTSTM	Ext		-	78			
PEEK1	Rel	349 #0000015D	-	127	135		
PEEK2	Rel	380 #0000017C	-	136	128		
PEEKe	Rel	265 #00000109	-	103	28		
PEEKt	Rel	43 #0000002B	-	36	27		
=PKLEX	Rel	0 #00000000	-	17			
POKE1	Rel	163 #000000A3	-	69	77		
POKE2	Rel	185 #000000B9	-	76	68		
POKEe	Rel	80 #00000050	-	50	31		
POKEp	Ext		-	49			
POKEt	Rel	56 #00000038	-	39	30		
POP1R	Ext		-	103			
POP1S	Ext		-	57			
RANGE	Ext		-	92			
REV\$	Ext		-	139			
S-R0-0	Ext		-	52	62		
STDD	Ext		-	48			
STRHDR	Ext		-	124			
TFORN	Ext		-	65			
TxTEn2	Rel	67 #00000043	-	42			
TxTSt2	Rel	43 #0000002B	-	35	22	27	30

Input Parameters

Source file name is pk.a

Listing file name is pk.l

Object file name is pk.o

Flags set on command line
None

Errors

None


```
1          TITLE JPC LEX: MODULE ILMMSG
2
3          * *****
4          * HPIL message module
5          *
6          * Keyword(s):
7          *   none
8          * Author(s):
9          *   Michel Martinet
10         * History:
11         *   Originally published in JPC#37
12         *   Config poll handler added
13         *   JFG jan06 config poll handler documentation
14         *
15         * (c) Copyright PPC Paris 1986, 1987, 1988 and the Author(s)
16         * *****
17
18         * --LEX header--
19 00000 FF      =ILMSG CON(2) #FF
20 00002 00          CON(2) #00
21 00004 00          CON(2) #00
22 00006 00000     CON(5) #00000 last LEX
23 0000B F          NIBHEX F
24 0000C E000     REL(4) (TXTB3)+1
25 00010 8A00     REL(4) MSGTB3
26 00014 80000     REL(5) POLL3
27
28         * ---NO main table---
29 00019 1FF      TXTB3 NIBHEX 1FF
30
31         * *****
32         * Poll Handler
33         * *****
34
35 0001C 3100     POLL3 LC(2) =pCONFIG
36 00020 961      ?B=C B
37 00023 40      GOYES Config
38 00025 00      RTNSXM
39
40 00027          Config
41         *
42         * The purpose of this routine is to put
43         * the ILMMSG LEX *before* the HPIL ROM LEX
44         * in the LEX table to take over the messages
45         *
46 00027 8F00000   GOSBVL =LXFND look for LEX table
47 0002E 490      GOC conf
48 00031 8D00000   GOVLNG =CORUPT
49 00038 DB      conf C=D A
50 0003A 108      R0=C save D
51 0003D 137      CD1EX
52 00040 CA      A=A+C A
53 00042 135      D1=C
54 00045 D8      B=A A B:end of LEX table
55 00047 35FF0000 LCHEX 0000FF
56 0004F AF7      D=C W D:ILMSG LEX signature
57 00052 35FF1062 LCHEX 2601FF
```

```

58 0005A AFA          A=C    W      A:HPIL LEX signature
59 0005D 840          ST=0   0
60 00060 25           P=     5
61 00062 15F5        loop   C=DAT1 6
62 00066 917          ?D#C  WP      ILMMSG LEX ?
63 00069 A0           GOYES  test    no, test for HPIL one
64                               *      found ILMMSG LEX:
65 0006B 860          ?ST=0  0      already before HPIL LEX?
66 0006E 34           GOYES  exit    yes, exit
67 00070 5B2          GONC   swap   no, exchange entry
68 00073 870          test   ?ST=1  0      HPIL already found?
69 00076 70           GOYES  next    yes, go on
70 00078 912          ?C=A   WP      no. is this the HPIL LEX ?
71 0007B 31           GOYES  found   yes.
72 0007D 17A          next   D1=D1+ 11    try next LEX entry
73 00080 137          CD1EX
74 00083 8A1          ?B=C   A      end of table?
75 00086 B2           GOYES  exit    yes, exit
76 00088 135          D1=C
77 0008B 56D          GONC   loop   (BET)
78 0008E 137          found  CD1EX
79 00091 06           RSTK=C          save HPIL entry addr on RSTK
80 00093 850          ST=1   0      set flag
81 00096 135          D1=C
82 00099 43E          GOC    next   (BET)
83
84 0009C 15BA         swap   A=DAT1 11    read HPIL LEX entry ->A
85 000A0 07           C=RSTK
86 000A2 134          D0=C
87 000A5 15EA         C=DAT0 11    recover ILMMSG LEX entry ->C
88 000A9 15DA         DAT1=C 11    swap the 2 entries
89 000AD 158A         DAT0=A 11
90 000B1 118          exit   C=R0
91 000B4 D7           D=C A      restore D
92 000B6 00           RTNSXM
93
94                               * *****
95                               * ---Message table---
96                               * *****
97 000B8 00           MSGTB3 CON(2) 00
98 000BA 64           CON(2) 70
99
100 000BC 01           M000  CON(2) (M001)-(M000)
101 000BE 00           CON(2) 0      'HPIL '
102 000C0 4            CON(1) 4
103 000C1 840594C4    NIBASC 'HPIL '
      02
104 000CB C            CON(1) 12
105
106 000CC 51           M001  CON(2) (M003)-(M001)
107 000CE 10           CON(2) 1      'ASSIGN IO Needed'
108 000D0 5            CON(1) 5
109 000D1 14353594    NIBASC 'ASSIGN'
      74E4
110 000DD D            CON(1) 13
111 000DE 44           CON(2) 68
112 000E0 C            CON(1) 12

```

```
113
114 000E1 80      M003  CON(2) (M004)-(M003)
115 000E3 30      CON(2) 3      'Excess Chars'
116 000E5 E       CON(1) 14
117 000E6 E4      CON(2) 78
118 000E8 C       CON(1) 12
119
120 000E9 80      M004  CON(2) (M005)-(M004)
121 000EB 40      CON(2) 4      'Missing Parm'
122 000ED E       CON(1) 14
123 000EE 25      CON(2) 82
124 000F0 C       CON(1) 12
125
126 000F1 80      M005  CON(2) (M006)-(M005)
127 000F3 50      CON(2) 5      'Invalid Parm'
128 000F5 E       CON(1) 14
129 000F6 15      CON(2) 81
130 000F8 C       CON(1) 12
131
132 000F9 80      M006  CON(2) (M007)-(M006)
133 000FB 60      CON(2) 6      'Invalid Expr'
134 000FD E       CON(1) 14
135 000FE 05      CON(2) 80
136 00100 C       CON(1) 12
137
138 00101 80      M007  CON(2) (M016)-(M007)
139 00103 70      CON(2) 7      'Syntax'
140 00105 E       CON(1) 14
141 00106 B4      CON(2) 75
142 00108 C       CON(1) 12
143
144 00109 80      M016  CON(2) (M017)-(M016)
145 0010B 01      CON(2) 16     'File Protect'
146 0010D E       CON(1) 14
147 0010E D3      CON(2) 61
148 00110 C       CON(1) 12
149
150 00111 51      M017  CON(2) (M018)-(M017)
151 00113 11      CON(2) 17     'End of Medium'
152 00115 5       CON(1) 5
153 00116 54E64602 NIBASC 'End of'
      F666
154 00122 D       CON(1) 13
155 00123 34      CON(2) 67
156 00125 C       CON(1) 12
157
158 00126 32      M018  CON(2) (M019)-(M018)
159 00128 21      CON(2) 18     'Not LIF Format'
160 0012A 7       CON(1) 7
161 0012B E4F64702 NIBASC 'Not LIF '
      C4946402
162 0013B 5       CON(1) 5
163 0013C 64F627D6 NIBASC 'Format'
      1647
164 00148 C       CON(1) 12
165
166 00149 72      M019  CON(2) (M020)-(M019)
```

```
167 0014B 31          CON(2) 19      'Disk Drive Error'
168 0014D 7           CON(1) 7
169 0014E 449637B6   NIBASC 'Disk Dri'
      02442796
170 0015E 7           CON(1) 7
171 0015F 67560254   NIBASC 've Error'
      2727F627
172 0016F C           CON(1) 12
173
174 00170 D0         M020  CON(2) (M021)-(M020)
175 00172 41         CON(2) 20      'No Medium'
176 00174 1           CON(1) 1
177 00175 E4F6       NIBASC 'No'
178 00179 D           CON(1) 13
179 0017A 34         CON(2) 67
180 0017C C           CON(1) 12
181
182 0017D 80         M021  CON(2) (M022)-(M021)
183 0017F 51         CON(2) 21      'Low Battery'
184 00181 E           CON(1) 14
185 00182 61         CON(2) 22
186 00184 C           CON(1) 12
187
188 00185 80         M022  CON(2) (M023)-(M022)
189 00187 61         CON(2) 22      'File Not Found'
190 00189 E           CON(1) 14
191 0018A 93         CON(2) 57
192 0018C C           CON(1) 12
193
194 0018D F0         M023  CON(2) (M024)-(M023)
195 0018F 71         CON(2) 23      'New Medium'
196 00191 2           CON(1) 2
197 00192 E45677     NIBASC 'New'
198 00198 D           CON(1) 13
199 00199 34         CON(2) 67
200 0019B C           CON(1) 12
201
202 0019C 31         M024  CON(2) (M025)-(M024)
203 0019E 81         CON(2) 24      'Blank Medium'
204 001A0 4           CON(1) 4
205 001A1 24C616E6   NIBASC 'Blank'
      B6
206 001AB D           CON(1) 13
207 001AC 34         CON(2) 67
208 001AE C           CON(1) 12
209
210 001AF E2         M025  CON(2) (M026)-(M025)
211 001B1 91         CON(2) 25      'Wrong Dir # records'
212 001B3 7           CON(1) 7
213 001B4 7527F6E6   NIBASC 'Wrong di'
      76024696
214 001C4 7           CON(1) 7
215 001C5 27023202   NIBASC 'r # reco'
      275636F6
216 001D5 2           CON(1) 2
217 001D6 274637     NIBASC 'rds'
218 001DC C           CON(1) 12
```

```
219
220 001DD 61          M026  CON(2) (M028)-(M026)
221 001DF A1          CON(2) 26      'Checksum'
222 001E1 7           CON(1) 7
223 001E2 34865636   NIBASC 'Checksum'
      B63757D6
224 001F2 C           CON(1) 12
225
226 001F3 91          M028  CON(2) (M029)-(M028)
227 001F5 C1          CON(2) 28      'Size of File'
228 001F7 7           CON(1) 7
229 001F8 3596A756   NIBASC 'Size of '
      02F66602
230 00208 E           CON(1) 14
231 00209 AE          CON(2) 234
232 0020B C           CON(1) 12
233
234 0020C 80          M029  CON(2) (M030)-(M029)
235 0020E D1          CON(2) 29      'Write protected'
236 00210 E           CON(1) 14
237 00211 24          CON(2) 66
238 00213 C           CON(1) 12
239
240 00214 80          M030  CON(2) (M031)-(M030)
241 00216 E1          CON(2) 30      'File Exists'
242 00218 E           CON(1) 14
243 00219 B3          CON(2) 59
244 0021B C           CON(1) 12
245
246 0021C 32          M031  CON(2) (M032)-(M031)
247 0021E F1          CON(2) 31      'Directory Full'
248 00220 7           CON(1) 7
249 00221 44962756   NIBASC 'Director'
      3647F627
250 00231 5           CON(1) 5
251 00232 97026457   NIBASC 'y Full'
      C6C6
252 0023E C           CON(1) 12
253
254 0023F 80          M032  CON(2) (M034)-(M032)
255 00241 02          CON(2) 32      'Device Not Found'
256 00243 E           CON(1) 14
257 00244 04          CON(2) 64
258 00246 C           CON(1) 12
259
260 00247 C1          M034  CON(2) (M035)-(M034)
261 00249 22          CON(2) 34      'Device Not Ready'
262 0024B D           CON(1) 13
263 0024C 24          CON(2) 66
264 0024E 7           CON(1) 7
265 0024F E4F64702   NIBASC 'Not Read'
      25561646
266 0025F 0           CON(1) 0
267 00260 97          NIBASC 'y'
268 00262 C           CON(1) 12
269
270 00263 51          M035  CON(2) (M036)-(M035)
```

```
271 00265 32          CON(2) 35      'Loop Broken'
272 00267 D          CON(1) 13
273 00268 54          CON(2) 69
274 0026A 5          CON(1) 5
275 0026B 2427F6B6   NIBASC 'Broken'
      56E6
276 00277 C          CON(1) 12
277
278 00278 61          M036 CON(2) (M037)-(M036)
279 0027A 42          CON(2) 36      'Too Many Frames'
280 0027C E          CON(1) 14
281 0027D FE          CON(2) 239
282 0027F 4          CON(1) 4
283 00280 D416E697   NIBASC 'Many '
      02
284 0028A D          CON(1) 13
285 0028B 64          CON(2) 70
286 0028D C          CON(1) 12
287
288 0028E 31          M037 CON(2) (M038)-(M037)
289 00290 52          CON(2) 37      'Frames Lost'
290 00292 D          CON(1) 13
291 00293 64          CON(2) 70
292 00295 4          CON(1) 4
293 00296 02C4F637   NIBASC ' Lost '
      47
294 002A0 C          CON(1) 12
295
296 002A1 91          M038 CON(2) (M039)-(M038)
297 002A3 62          CON(2) 38      'Frames Altered'
298 002A5 D          CON(1) 13
299 002A6 64          CON(2) 70
300 002A8 7          CON(1) 7
301 002A9 0214C647   NIBASC ' Altered'
      56275646
302 002B9 C          CON(1) 12
303
304 002BA 02          M039 CON(2) (M040)-(M039)
305 002BC 72          CON(2) 39      'Unexpected Message'
306 002BE 7          CON(1) 7
307 002BF 55E65687   NIBASC 'Unexpect'
      07563647
308 002CF 2          CON(1) 2
309 002D0 564602     NIBASC 'ed '
310 002D6 D          CON(1) 13
311 002D7 14          CON(2) 65
312 002D9 C          CON(1) 12
313
314 002DA 80          M040 CON(2) (M041)-(M040)
315 002DC 82          CON(2) 40      'Too Many Frames'
316 002DE D          CON(1) 13
317 002DF 42          CON(2) 36
318 002E1 C          CON(1) 12
319
320 002E2 11          M041 CON(2) (M042)-(M041)
321 002E4 92          CON(2) 41      'Invalid Mode'
322 002E6 E          CON(1) 14
```

```
323 002E7 CE          CON(2) 236
324 002E9 3          CON(1) 3
325 002EA D4F64656   NIBASC 'Mode'
326 002F2 C          CON(1) 12
327
328 002F3 71        M042  CON(2) (M043) - (M042)
329 002F5 A2        CON(2) 42      'Message Altered'
330 002F7 D          CON(1) 13
331 002F8 14        CON(2) 65
332 002FA 6          CON(1) 6
333 002FB 14C64756   NIBASC 'Altered'
    275646
334 00309 C          CON(1) 12
335
336 0030A 71        M043  CON(2) (M044) - (M043)
337 0030C B2        CON(2) 43      'Loop Timeout'
338 0030E D          CON(1) 13
339 0030F 54        CON(2) 69
340 00311 6          CON(1) 6
341 00312 4596D656   NIBASC 'Timeout'
    F65747
342 00320 C          CON(1) 12
343
344 00321 12        M044  CON(2) (M045) - (M044)
345 00323 C2        CON(2) 44      'Bad Addresses'
346 00325 7          CON(1) 7
347 00326 24164602   NIBASC 'Bad Addr'
    14464627
348 00336 4          CON(1) 4
349 00337 56373756   NIBASC 'esses'
    37
350 00341 C          CON(1) 12
351
352 00342 72        M045  CON(2) (M047) - (M045)
353 00344 D2        CON(2) 45      'Self Test Failed'
354 00346 7          CON(1) 7
355 00347 3556C666   NIBASC 'Self Tes'
    02455637
356 00357 7          CON(1) 7
357 00358 47026416   NIBASC 't Failed'
    96C65646
358 00368 C          CON(1) 12
359
360 00369 11        M047  CON(2) (M052) - (M047)
361 0036B F2        CON(2) 47      'Device Type'
362 0036D D          CON(1) 13
363 0036E 24        CON(2) 66
364 00370 3          CON(1) 3
365 00371 45970756   NIBASC 'Type'
366 00379 C          CON(1) 12
367
368 0037A 41        M052  CON(2) (M053) - (M052)
369 0037C 43        CON(2) 52      'Aborted'
370 0037E 6          CON(1) 6
371 0037F 1426F627   NIBASC 'Aborted'
    475646
372 0038D C          CON(1) 12
```

```
373
374 0038E 41      M053  CON(2) (M054)-(M053)
375 00390 53      CON(2) 53      'Invalid Device Spec'
376 00392 E       CON(1) 14
377 00393 CE     CON(2) 236
378 00395 D       CON(1) 13
379 00396 24     CON(2) 66
380 00398 3       CON(1) 3
381 00399 35075636 NIBASC 'Spec'
382 003A1 C       CON(1) 12
383
384 003A2 80      M054  CON(2) (M056)-(M054)
385 003A4 63      CON(2) 54      'Data Type'
386 003A6 E       CON(1) 14
387 003A7 F1     CON(2) 31
388 003A9 C       CON(1) 12
389
390 003AA 80      M056  CON(2) (M057)-(M056)
391 003AC 83      CON(2) 56      'Invalid Arg'
392 003AE E       CON(1) 14
393 003AF B0     CON(2) 11
394 003B1 C       CON(1) 12
395
396 003B2 F0     M057  CON(2) (M059)-(M057)
397 003B4 93     CON(2) 57      'No Loop'
398 003B6 2       CON(1) 2
399 003B7 E4F602 NIBASC 'No '
400 003BD D       CON(1) 13
401 003BE 54     CON(2) 69
402 003C0 C       CON(1) 12
403
404 003C1 80      M059  CON(2) (M060)-(M059)
405 003C3 B3     CON(2) 59      'Insufficient Memory'
406 003C5 E       CON(1) 14
407 003C6 81     CON(2) 24
408 003C8 C       CON(1) 12
409
410 003C9 71      M060  CON(2) (M065)-(M060)
411 003CB C3     CON(2) 60      'RESTORE IO Needed'
412 003CD 6       CON(1) 6
413 003CE 25543545 NIBASC 'RESTORE'
    F42554
414 003DC D       CON(1) 13
415 003DD 44     CON(2) 68
416 003DF C       CON(1) 12
417
418 003E0 61      M065  CON(2) (M066)-(M065)
419 003E2 14     CON(2) 65      'Message '
420 003E4 7       CON(1) 7
421 003E5 D4563737 NIBASC 'Message '
    16765602
422 003F5 C       CON(1) 12
423
424 003F6 41      M066  CON(2) (M067)-(M066)
425 003F8 24     CON(2) 66      'Device '
426 003FA 6       CON(1) 6
427 003FB 44566796 NIBASC 'Device '
```



```

      365602
428 00409 C          CON(1) 12
429
430 0040A 41        M067 CON(2) (M068)-(M067)
431 0040C 34          CON(2) 67      ' Medium'
432 0040E 6          CON(1) 6
433 0040F 02D45646  NIBASC ' Medium'
      9657D6
434 0041D C          CON(1) 12
435
436 0041E B1        M068 CON(2) (M069)-(M068)
437 00420 44          CON(2) 68      ' IO Needed'
438 00422 7          CON(1) 7
439 00423 0294F402  NIBASC ' IO Need'
      E4565646
440 00433 1          CON(1) 1
441 00434 5646       NIBASC 'ed'
442 00438 C          CON(1) 12
443
444 00439 01        M069 CON(2) (M070)-(M069)
445 0043B 54          CON(2) 69      'Loop '
446 0043D 4          CON(1) 4
447 0043E C4F6F607  NIBASC 'Loop '
      02
448 00448 C          CON(1) 12
449
450 00449 21        M070 CON(2) (MFIN)-(M070)
451 0044B 64          CON(2) 70      ' Frames'
452 0044D 5          CON(1) 5
453 0044E 642716D6  NIBASC 'Frames'
      5637
454 0045A C          CON(1) 12
455
456 0045B FF        MFIN  NIBHEX FF
457
458
459                * end of file (for sasm only)
460 0045D          =FiLeNd
461
462 0045D          END
```

CORUPT	Ext		-	48		
Cfg	Rel	39	#00000027	-	40	37
=FileNd	Rel	1117	#0000045D	-	460	
=ILMSG	Rel	0	#00000000	-	19	
LXFND	Ext		-	46		
M000	Rel	188	#000000BC	-	100	100
M001	Rel	204	#000000CC	-	106	100 106
M003	Rel	225	#000000E1	-	114	106 114
M004	Rel	233	#000000E9	-	120	114 120
M005	Rel	241	#000000F1	-	126	120 126
M006	Rel	249	#000000F9	-	132	126 132
M007	Rel	257	#00000101	-	138	132 138
M016	Rel	265	#00000109	-	144	138 144
M017	Rel	273	#00000111	-	150	144 150
M018	Rel	294	#00000126	-	158	150 158
M019	Rel	329	#00000149	-	166	158 166
M020	Rel	368	#00000170	-	174	166 174
M021	Rel	381	#0000017D	-	182	174 182
M022	Rel	389	#00000185	-	188	182 188
M023	Rel	397	#0000018D	-	194	188 194
M024	Rel	412	#0000019C	-	202	194 202
M025	Rel	431	#000001AF	-	210	202 210
M026	Rel	477	#000001DD	-	220	210 220
M028	Rel	499	#000001F3	-	226	220 226
M029	Rel	524	#0000020C	-	234	226 234
M030	Rel	532	#00000214	-	240	234 240
M031	Rel	540	#0000021C	-	246	240 246
M032	Rel	575	#0000023F	-	254	246 254
M034	Rel	583	#00000247	-	260	254 260
M035	Rel	611	#00000263	-	270	260 270
M036	Rel	632	#00000278	-	278	270 278
M037	Rel	654	#0000028E	-	288	278 288
M038	Rel	673	#000002A1	-	296	288 296
M039	Rel	698	#000002BA	-	304	296 304
M040	Rel	730	#000002DA	-	314	304 314
M041	Rel	738	#000002E2	-	320	314 320
M042	Rel	755	#000002F3	-	328	320 328
M043	Rel	778	#0000030A	-	336	328 336
M044	Rel	801	#00000321	-	344	336 344
M045	Rel	834	#00000342	-	352	344 352
M047	Rel	873	#00000369	-	360	352 360
M052	Rel	890	#0000037A	-	368	360 368
M053	Rel	910	#0000038E	-	374	368 374
M054	Rel	930	#000003A2	-	384	374 384
M056	Rel	938	#000003AA	-	390	384 390
M057	Rel	946	#000003B2	-	396	390 396
M059	Rel	961	#000003C1	-	404	396 404
M060	Rel	969	#000003C9	-	410	404 410
M065	Rel	992	#000003E0	-	418	410 418
M066	Rel	1014	#000003F6	-	424	418 424
M067	Rel	1034	#0000040A	-	430	424 430
M068	Rel	1054	#0000041E	-	436	430 436
M069	Rel	1081	#00000439	-	444	436 444
M070	Rel	1097	#00000449	-	450	444 450
MFIN	Rel	1115	#0000045B	-	456	450
MSGTB3	Rel	184	#000000B8	-	97	25
POLL3	Rel	28	#0000001C	-	35	26

TXTB3	Rel	25	#00000019	-	29	24	
conf	Rel	56	#00000038	-	49	47	
exit	Rel	177	#000000B1	-	90	66	75
found	Rel	142	#0000008E	-	78	71	
loop	Rel	98	#00000062	-	61	77	
next	Rel	125	#0000007D	-	72	69	82
pCONFIG	Ext			-	35		
swap	Rel	156	#0000009C	-	84	67	
test	Rel	115	#00000073	-	68	63	

Input Parameters

Source file name is ilmsg.a

Listing file name is ilmsg.l

Object file name is ilmsg.o

Flags set on command line
None

Errors

None