```
Page 001          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0001 00000                TITLE   Editeur, chaines génériques <xgen.as>
0002 00000
0003 00000          *
0004 00000          * Ce module n'a pas l'air de consommer de mémoire (sauf pour
0005 00000          * stocker le buffer bien sur).
0006 00000          * En revanche, la consommation de registres est
0007 00000          * impressionnante.
0008 00000          *
0009 00000
0010 00000          *
0011 00000          * Définitions pour les tokens compilés
0012 00000          *
0013 00000           CHAR   EQU    0       a character (valeur cablee)
0014 00000           EOL    EQU    1       end of line
0015 00000           ANY    EQU    2       any character
0016 00000           CCL    EQU    3       character class
0017 00000           NCCL   EQU    4       not in character class
0018 00000           END    EQU    5       end of compiled stream
0019 00000           BOL    EQU    6       begin of line
0020 00000           CLOS   EQU    7       closure
0021 00000          *
0022 00000          * et maintenant leur taille
0023 00000          *
0024 00000           BOLS   EQU    1       BOL
0025 00000           EOLS   EQU    1       EOL
0026 00000           ANYS   EQU    1       ANY
0027 00000           CCLS   EQU    1+256   CCL c0 c1 ... c255
0028 00000           NCCLS  EQU    1+256   NCCL c0 c1 ... c255
0029 00000           CHARS  EQU    1+2     CHAR c
0030 00000           CLOSS  EQU    1       CLOS 'token'
0031 00000
0032 00000          *
0033 00000          * Définitions pour les caractères HP-UX
0034 00000          *
0035 00000           XBOL   EQU    '^'     begin of line
0036 00000           XEOL   EQU    '$'     end of line
0037 00000           XANY   EQU    '.'     any character
0038 00000           XBCCL  EQU    '['     begin of character class
0039 00000           XECCL  EQU    ']'     end of character class
0040 00000           XNCCL  EQU    '^'     not in character class
0041 00000           XICCL  EQU    '-'     interval in character class
0042 00000           XCLOS  EQU    '*'     closure
0043 00000           XREP   EQU    '%'     remplacement
0044 00000           XESC   EQU    '\'     escape
0045 00000          *
0046 00000          * Définitions pour les caractères HP-71
0047 00000          *
0048 00000           SBOL   EQU    '^'     begin of line
0049 00000           SEOL   EQU    '$'     end of line
0050 00000           SANY   EQU    '.'     any character
```

```
Page 002          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0051 00000           SREPT  EQU    '@'     repetition (.*)
0052 00000           SREP   EQU    '&'     remplacement
0053 00000           STOGL  EQU    '\'     toggle
0054 00000
0055 00000          *
0056 00000          * Notes sur les closures et leur utilisation :
0057 00000          * - Les closures sont codées sous forme d'un tableau de 256
0058 00000          *   quartets (on pourrait le compresser à 256 bits).
0059 00000          * - il y a donc un nombre maximum de 14 closures.
0060 00000          * - pour les utiliser, il faut une pile, contenant à chaque
0061 00000          *   fois 20 quartets.
0062 00000          * - on place cette pile en début de buffer, ce qui laisse
0063 00000          *   toujours 14 closures
0064 00000          *
0065 00000           STKSIZ EQU    14*20   14 closures * 20 quartets
0066 00000
0067 00000          ********************************************************
0068 00000          * COMPUX
0069 00000          *
0070 00000          * But: compiler une chaine générique
0071 00000          * Entree:
0072 00000          *   - D1 = " M.S. sur la chaine à compiler
0073 00000          *   - A(A) = longueur en octets
0074 00000          * Sortie:
0075 00000          *   - Cy = 1 : erreur
0076 00000          *        C(4-0) = numéro d'erreur pour BSERR
0077 00000          *   - Cy = 0 : pas d'erreur
0078 00000          *        R0(7-3) = D1 = " début du buffer (sur les donnees)
0079 00000          *        R0(2-0) = B(X) = buffer ID
0080 00000          * Abime: A-D, D0, D1, R0-R2
0081 00000          * Appelle: getbuf, adddchr, marque, lookah, getchr,
0082 00000          *   FINDA, I/OCON, WIPOUT, gettab, MOVED2
0083 00000          * Niveaux: 4 (getbuf)
0084 00000          * Detail:
0085 00000          *   créer le buffer
0086 00000          *   tant que non fin de chaine
0087 00000          *     faire
0088 00000          *       mémoriser le début du pattern précédent
0089 00000          *       selon caractère lu
0090 00000          *         '\' : ajouter_normal (caractere suivant)
0091 00000          *         '.' : ajouter (ANY)
0092 00000          *         '^' : si début de ligne
0093 00000          *                 alors: ajouter (BOL)
0094 00000          *                 sinon ajouter_normal ('^')
0095 00000          *               fin si
0096 00000          *         '$' : si fin de ligne
0097 00000          *                 alors ajouter (EOL)
0098 00000          *                 sinon ajouter_normal ('$')
0099 00000          *               fin si
0100 00000          *         '[' : ajouter (CCL ou NCCL selon caractère suivant
```

```
Page 003          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0101 00000          *             analyser ce qui suit et créer tableau
0102 00000          *         '*' : décaler le pattern précédent
0103 00000          *               insérer la closure avant
0104 00000          *           sinon : ajouter_normal (caractère lu)
0105 00000          *       fin selon
0106 00000          *   fin tant que
0107 00000          * Historique:
0108 00000          *   88/06/05: PD/JT conception & codage
0109 00000          *   88/10/29: PD/JT suppression du buffer en cas d'erreur
0110 00000          ********************************************************
0111 00000
0112 00000 7B24     =COMPUX GOSUB   getbuf
0113 00004 400            RTNC            si erreur
0114 00007          *
0115 00007          * B(A) = taille du buffer
0116 00007          * D(A) = taille de la chaine
0117 00007          * D1 = " chaine
0118 00007          * D0 = " buffer
0119 00007          * R0(2-0) = bufid
0120 00007          * R0(7-3) = " buffer
0121 00007          *
0122 00007 D2             C=0     A
0123 00009 109            R1=C            marque := 0
0124 0000C          *
0125 0000C          * D(A) = nb de caractères restant dans la chaine
0126 0000C          * B(A) = taille du buffer
0127 0000C          * D1 = " chaine
0128 0000C          * D0 = " buffer
0129 0000C          * R1 = " token précédent
0130 0000C          *
0131 0000C 8E0000    cux-10 GOSUBL  =getchr
0132 00012 404            GOC     cmpend  Fin de chaine
0133 00015 8F00000        GOSBVL  =FINDA
0134 0001C C5             CON(2)  XESC
0135 0001E 750            REL(3)  cuESC
0136 00021 E2             CON(2)  XANY
0137 00023 E60            REL(3)  cuANY
0138 00026 E5             CON(2)  XBOL
0139 00028 680            REL(3)  cuBOL
0140 0002B 42             CON(2)  XEOL
0141 0002D CB0            REL(3)  cuEOL
0142 00030 B5             CON(2)  XBCCL
0143 00032 3F0            REL(3)  cuCCL
0144 00035 A2             CON(2)  XCLOS
0145 00037 A02            REL(3)  cuCLOS
0146 0003A 00             NIBHEX  00
0147 0003C
0148 0003C 7CA4           GOSUB   marque  marque lasttoken
0149 00040 73B4           GOSUB   adddchr
0150 00044 400            RTNC            if error
```

```
Page 004          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0151 00047 64CF           GOTO    cux-10
0152 0004B
0153 0004B          ********************************************************
0154 0004B          * Sortie de la compilation
0155 0004B          ********************************************************
0156 0004B 6BD3      System GOTO    system
0157 0004F 64C4      norooM GOTO    norooM
0158 00053
0159 00053           cmpend
0160 00053          *
0161 00053          * Marqueur de fin
0162 00053          *
0163 00053 A3D            B=B-1   X
0164 00056 48F            GOC     norooM
0165 00059 305            LC(1)   END
0166 0005C 1540           DAT0=C  P
0167 00060          *
0168 00060          * Réduire la taille du buffer
0169 00060          *
0170 00060 118            C=R0            C(X) = buffer id
0171 00063 8F00000        GOSBVL  =I/OCON
0172 0006A 50E            GONC    System  buffer not found !
0173 0006D          *
0174 0006D          * Maintenant, il faut renvoyer les parametres du buffer
0175 0006D          *
0176 0006D 118            C=R0
0177 00070 AB5            B=C     X       B(A) := buffer id
0178 00073          * Cy = 0 : pas d'erreur
0179 00073          * D1 = " début du buffer (sur les donnees) (après I/OCON)
0180 00073          * B(X) = buffer ID
0181 00073 03             RTNCC
0182 00075
0183 00075 8E0000    cuESC  GOSUBL  =getchr
0184 00078 411            GOC     InvPat  "\" et fin de ligne
0185 0007E 7A64           GOSUB   marque
0186 00082 7174           GOSUB   adddchr
0187 00086 400            RTNC            if error
0188 00089 628F           GOTO    cux-10  et retour à la boucle
0189 0008D
0190 0008D 6112      InvPat GOTO    invpat
0191 00091
0192 00091 7754      cuANY  GOSUB   marque
0193 00095 32100           LC(3)   ANYS
0194 0009A B31             B=B-C   X
0195 0009D 474             GOC     Noroom
0196 000A0 302             LC(1)   ANY
0197 000A3 1540            DAT0=C  P
0198 000A7 160             D0=D0+  1
0199 000AA 616F            GOTO    cux-10  et retour à la boucle
0200 000AE
```

```
Page 005          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0201 000AE 111      cuBOL  A=R1
0202 000B1 7734            GOSUB   marque
0203 000B5 8A8             ?A=0    A       Début de ligne ?
0204 000B8 41              GOYES   cuBL10  oui
0205 000BA 31E5            LCASC   '^'
0206 000BE AEA             A=C     B
0207 000C1 7234            GOSUB   addchr  '^' normal
0208 000C5 400             RTNC            if error
0209 000C8 634F            GOTO    cux-10  et retour à la boucle
0210 000CC 32100    cuBL10 LC(3)   BOLS    '^' générique
0211 000D1 B31             B=B-C   X
0212 000D4 401             GOC     Noroom
0213 000D7 306             LC(1)   BOL
0214 000DA 1540            DAT0=C  P
0215 000DE 160             D0=D0+  1
0216 000E1 6A2F            GOTO    cux-10  et retour à la boucle
0217 000E5
0218 000E5 6E24     Noroom GOTO    noroom
0219 000E9
0220 000E9 7FF3     cuEOL  GOSUB   marque
0221 000ED 8E0000          GOSUBL  =lookah
0222 000F3 441             GOC     cuEL10  fin de ligne
0223 000F6 3142            LCASC   '$'
0224 000FA AEA             A=C     B
0225 000FD 76F3            GOSUB   addchr  '$' normal
0226 00101 400             RTNC            if error
0227 00104 670F            GOTO    cux-10  et retour à la boucle
0228 00108 32100    cuEL10 LC(3)   EOLS    '$' générique
0229 0010D B31             B=B-C   X
0230 00110 44D             GOC     Noroom
0231 00113 301             LC(1)   EOL
0232 00116 1540            DAT0=C  P
0233 0011A 160             D0=D0+  1
0234 0011D 6EEE            GOTO    cux-10  et retour à la boucle
0235 00121
0236 00121 6D71     Invpat GOTO    invpat
0237 00125
0238 00125 73C3     cuCCL  GOSUB   marque
0239 00129 32101           LC(3)   CCLS
0240 0012E B31             B=B-C   X
0241 00131 43B             GOC     Noroom
0242 00134 8E0000          GOSUBL  =lookah
0243 0013A 46E             GOC     Invpat  "[" et fin de ligne
0244 0013D 303             LC(1)   CCL
0245 00140 816             CSRC            C(S) := CCL
0246 00143 31E5            LCASC   '^'
0247 00147 966             ?A#C    B
0248 0014A E0              GOYES   cuCL10
0249 0014C 304             LC(1)   NCCL
0250 0014F 816             CSRC
```

```
Page 006          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0251 00152 8E0000          GOSUBL  =getchr
0252 00158 1544     cuCL10 DAT0=C  S
0253 0015C 160             D0=D0+  1
0254 0015F          •
0255 0015F          • tab [0..255] := 0
0256 0015F          •
0257 0015F 136             CD0EX
0258 00162 134             D0=C
0259 00165 137             CD1EX           D1 := D0
0260 00169 10A             R2=C            R2 := D1
0261 0016B 3400100         LC(5)   (CCLS)-1
0262 00172 8F00000         GOSBVL  =WIPOUT
0263 00179 11A             C=R2
0264 0017C 135             D1=C
0265 0017F
0266 0017F 8E0000   cuCL50 GOSUBL  =getchr
0267 00185 4B9             GOC     Invpat  "]" manquant
0268 00188 31D5            LC(2)   XECCL   ']'
0269 0018C 962             ?A=C    B
0270 0018F 24              GOYES   cuCL90
0271 00191 31C5            LC(2)   XESC    '\'
0272 00195 966             ?A#C    B
0273 00198 B0              GOYES   cuCL60
0274 0019A 8E0000          GOSUBL  =getchr
0275 001A0 408             GOC     Invpat  "[...\" et fin de ligne
0276 001A3 F0       cuCL60 ASL     A
0277 001A5 F0              ASL     A       A(3-2) = caractère
0278 001A7 8E0000          GOSUBL  =lookah
0279 001AD 4B0             GOC     cuCL70  fin de ligne (anormal)
0280 001B0 3102            LC(2)   XICCL   '-'
0281 001B4 962             ?A=C    B
0282 001B7 13              GOYES   cuCL80
0283 001B9          • caractère simple
0284 001B9 F4       cuCL70 ASR     A
0285 001BB F4              ASR     A       A(B) := caractère
0286 001BD 7F53            GOSUB   gettab  D0 := ^ caractère dans le tableau
0287 001C1 DA              A=C     A       A(A) := ancienne adresse
0288 001C3 301             LC(1)   1
0289 001C6 1540            DAT0=C  P       tab [caractère] := true
0290 001CA 130             D0=A            D0 := ^ tab [0]
0291 001CD 61BF            GOTO    cuCL50  et un tour de boucle de plus !
0292 001D1
0293 001D1 132      cuCL90 AD0EX
0294 001D4 3400100         LC(5)   (CCLS)-1
0295 001DB CA              A=A+C   A
0296 001DD 130             D0=A
0297 001E0 6B2E            GOTO    cux-10  et retour a la boucle
0298 001E4
0299 001E4 6AB0     invPat GOTO    invpat
0300 001E8          • ..---
```

```
Page 007          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0301 001E8 8E0000   cuCL80 GOSUBL  =getchr passer le '-'
0302 001EE 8E0000          GOSUBL  =getchr
0303 001F4 4FE             GOC     invPat  "[...-" et fin de ligne
0304 001F7 31C5            LC(2)   XESC
0305 001FB 966             ?A#C    B
0306 001FE B0              GOYES   cuCL82
0307 00200 8E0000          GOSUBL  =getchr
0308 00206 4DD             GOC     invPat  "[...-\" et fin de ligne
0309 00209 AE6      cuCL82 C=A     B       C(B) := upper
0310 0020C F4              ASR     A
0311 0020E F4              ASR     A       A(B) := lower
0312 00210 B62             C=C-A   B       range
0313 00213 10A             R2=C            R2 := range
0314 00216 4DC             GOC     invPat  lower > upper
0315 00219 7303            GOSUB   gettab  D0 := ^ tab [lower]
0316 0021D 12A             CR2EX           R2 := ^ tab [0] ; C(B) := range
0317 00220 AEA             A=C     B       compteur dans A(B)
0318 00223 301             LC(1)   1
0319 00226 1540     cuCL84 DAT0=C  P
0320 0022A 160             D0=D0+  1
0321 0022D A6C             A=A-1   B
0322 00230 55F             GONC    cuCL84
0323 00233 11A             C=R2            C(A) := ^ tab [0]
0324 00236 134             D0=C
0325 00239 654F            GOTO    cuCL50
0326 0023D
0327 0023D 66D2     noRoom GOTO    noroom
0328 00241
0329 00241 A3D      cuCLOS B=B-1   X
0330 00244 48F             GOC     noRoom
0331 00247 111             A=R1            last token
0332 0024A 8A8             ?A=0    A
0333 0024D 79              GOYES   invPat  "•"
0334 0024F 132             AD0EX           A(A) := courant; D0 := ^ last
0335 00252 102             R2=A            R2 := courant
0336 00255 1520            A=DAT0  P       A(0) := last token
0337 00259 306             LC(1)   BOL
0338 0025C 902             ?A=C    P
0339 0025F 04              GOYES   invpat  "^•"
0340 00261 307             LC(1)   CLOS
0341 00264 902             ?A=C    P
0342 00267 83              GOYES   invpat  "...••"
0343 00269
0344 00269 133             AD1EX           A(A) := D1
0345 0026C 122             AR2EX           A(A) := end of dest; R2 := D1
0346 0026F 131             D1=A
0347 00272 101             R1=A            on remettra last token après
0348 00275 170             D1=D1+  1       D1 := ^ end of dest
0349 00278 136             CD0EX
0350 0027B          •
```

```
Page 008          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0351 0027B          • C(A) = start of source
0352 0027B          • A(A) = end of source
0353 0027B          • D1 = end of destination
0354 0027B          • R1 = end of source
0355 0027B          • R2 = ancien D1
0356 0027B          •
0357 0027B 8F00000         GOSBVL  =MOVED2
0358 00282
0359 00282 112             A=R2            A(A) := ancien D1
0360 00285 131             D1=A            D1 est restauré
0361 00288 307             LC(1)   CLOS
0362 0028B 1540            DAT0=C  P
0363 0028F
0364 0028F 132             AD0EX           A(A) := ^ last token
0365 00292 121             AR1EX
0366 00295          • R1 := ^ last token ; A(A) := end of source
0367 00295 130             D0=A
0368 00298 160             D0=D0+  1
0369 0029B
0370 0029B 607D            GOTO    cux-10
0371 0029F
0372 0029F 7776     invpat GOSUB   =ENDPOS
0373 002A3 330000          LC(4)   (=id)^(=teIPAT)
0374 002A9 02              RTNSC
0375 002AB
0376 002AB          ********************************************************
0377 002AB          • COMP71
0378 002AB          •
0379 002AB          • But: compiler une chaine générique
0380 002AB          • Entrée:
0381 002AB          •   - D1 = ^ M.S. sur la chaine à compiler
0382 002AB          •   - A(A) = longueur en octets
0383 002AB          • Sortie:
0384 002AB          •   - Cy = 1 : erreur
0385 002AB          •       C(4-0) = numéro d'erreur pour BSERR
0386 002AB          •   - Cy = 0 : pas d'erreur
0387 002AB          •       R0(7-3) = D1 = ^ début du buffer (sur les données)
0388 002AB          •       R0(2-0) = B(X) = buffer ID
0389 002AB          • Abime: A-D, D0, D1, R0-R2
0390 002AB          • Appelle: getbuf, addchr, lookah, getchr, FINDA, I/OCON
0391 002AB          • Niveaux: 4 (getbuf)
0392 002AB          • Detail:
0393 002AB          •   créer le buffer
0394 002AB          •   état := 0
0395 002AB          •   tant que non fin de chaine
0396 002AB          •     faire
0397 002AB          •       lire le caractère
0398 002AB          •       si caractère lu = '\'
0399 002AB          •         alors
0400 002AB          •           selon état
```

```
Page 009          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0401 002AB          •              0 : état = 1
0402 002AB          •              1 : ajouter_normal ('\')
0403 002AB          •                  état = 0
0404 002AB          •              2 : état = 3
0405 002AB          •              3 : ajouter_normal ('\')
0406 002AB          •                  état = 2
0407 002AB          •            fin selon
0408 002AB          •          sinon
0409 002AB          •            selon état
0410 002AB          •              0 : ajouter_normal (caractère lu)
0411 002AB          •              1 : état = 2
0412 002AB          •                  traiter_spécial (caractère lu)
0413 002AE          •              2 : ajouter_normal (caractère lu)
0414 002AB          •              3 : état = 0
0415 002AB          •                  ajouter__normal (caractère lu)
0416 002AB          •        fin si
0417 002AB          •    fin tant que
0418 002AB          •
0419 002AB          • traiter_spécial (caractère)
0420 002AB          •   selon caractère lu
0421 002AB          •     '.' : ajouter (ANY)
0422 002AB          •     '^' : si début de ligne
0423 002AB          •               alors ajouter (EOL)
0424 002AB          •               sinon ajouter_normal ('^')
0425 002AB          •           fin si
0426 002AB          •     '$' : si fin de ligne
0427 002AB          •               alors ajouter (EOL)
0428 002AB          •               sinon ajouter_normal ('$')
0429 002AB          •           fin si
0430 002AB          •     '@' : ajouter (CLOSURE)
0431 002AB          •           ajouter (ANY)
0432 002AB          •   fin selon
0433 002AB          • Historique:
0434 002AB          •   88/06/26: PD/JT conception & codage d'après LOMFUX
0435 002AB          •   89/10/29: PD/JT suppression du buffer en cas d'erreur
0436 002AB          ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
0437 002AB
0438 002AB 7051     =COMP71 GOSUB   getbuf
0439 002AF 400              RTNC                si erreur
0440 002B0          •
0441 002B0          • B(A) = taille du buffer
0442 002B0          • D(A) = taille de la chaine
0443 002B0          • D1 =   chaine
0444 002B0          • D0 =   buffer
0445 002B0          • R0(2-0) = buffid
0446 002B0          • R0(7-3) =   buffer
0447 002B0          •
0448 002B0 300              LC(1)   0
0449 002B2 109              R1=C                état := 0
0450 002B5          •
```

```
Page 010          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0451 002B5          • D(A) = nb de caractères restant dans la chaine
0452 002B5          • B(A) = taille du buffer
0453 002B5          • D1 =   chaine
0454 002B5          • D0 =   buffer
0455 002B5          • P1(0) =   état de l'automate cablé
0456 002B5          •
0457 002B5 8E0000   c71-10 GOSUBL  =getchr
0458 002B8 412             GOC     Cmpend  Fin de chaine
0459 002B9 3105            LC(2)   STOGL
0460 002BB 9EE             D1=C    R       A(B) = caractère lu
0461 002BC C1              GOYES   c71nor  caractère normal (ou générique)
0462 002BD 119             C=R1
0463 002BD 8F0000          GOSBVL  =TBLJMP
0464 002C4 620             REL(3)  c7b0    back-slash
0465 002C7 D20             REL(3)  c7b1
0466 002CA 130             REL(3)  c7b2
0467 002CD 830             REL(3)  c7b3
0468 002E0 627D     Cmpend GOTO    cmpend  rallonge
0469 002E4 119      c71nor C=R1
0470 002E7 8F0000          GOSBVL  =TBLJMP
0471 002EE 0C0             REL(3)  c7n0    normal
0472 002F1 550             REL(3)  c7n1
0473 002F4 830             REL(3)  c7n2
0474 002F7 D50             REL(3)  c7n3
0475 002FA
0476 002FA 301      c7b0   LC(1)   1
0477 002FD 109             R1=C                état := 1
0478 00300 67BF            GOTO    c71-10
0479 00304 300      c7b1   LC(1)   0
0480 00307 6810            GOTO    c7b30
0481 0030B 503      c7b2   LC(1)   3
0482 0030E 109             R1=C                état := 3
0483 00311 66AF            GOTO    c71-10
0484 00315 302      c7b3   LC(1)   2
0485 00318 109      c7b30  R1=C                état := 0
0486 0031B          •
0487 0031B          • Attention ! le code continue !!!
0488 0031B          •
0489 0031B 7501     c7n0   GOSUB   addchr  ajouter_normal (caractere lu)
0490 0031F 400             RTNC
0491 00320 699F            GOTO    c71-10
0492 00324 0AC      c7n1   LC(4)   2
0493 00329 109             R1=C
0494 0032C 8Fa000   c7n2   GOSBVL  =FINDA  routine "traite_spéial"
0495 00330 E4              CON(2)  SANY
0496 00333 030             REL(3)  c7ANY
0497 00336 E6              CON(2)  '@'
0498 00339 440             REL(3)  c7B0
0499 0033C 40              CON(2)  3E0
0500 0033F 140             REL(3)  c7E0L
```

```
Page 011          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0501 00342 04               CON(2)  SREPT
0502 00344 0C0              REL(3)  c7REPT
0503 00347 00               NIBHEX  00
0504 00349 7AA1             GOSUB   addchr
0505 0034D 400              RTNC                if error
0506 00350 676F             GOTO    c71-10
0507 00354
0508 00354 7F91     c7n3    GOSUB   addchr  ajouter_normal (caractère lu)
0509 00358 400              RTNC
0510 0035B 300              LC(1)   0
0511 0035E 109              R1=C                état := 0
0512 00361 665F             GOTO    c71-10
0513 00365
0514 00365 32100    c7ANY   LC(3)   ANYS
0515 0036A B31              B=B-C   X
0516 0036D 4E5              GOC     NoRoom
0517 00370 302              LC(1)   ANY
0518 00373 1540             DAT0=C  P
0519 00377 160              D0=D0+  1
0520 0037A 6D3F             GOTO    c71-10
0521 0037E 110      c7B0L   A=R0
0522 00381 BF4              ASR     W
0523 00384 BF4              ASR     W
0524 00387 BF4              ASR     W           A(A) :=   début buffer
0525 0038A 3481100          LC(5)   STKSIZ
0526 00391 C2               C=C+A   A       C(A) :=   début motif (après pile)
0527 00393 132              ADDEX           A(A) :=   position courante
0528 00396 8A2              ?A=C    A
0529 00399 71               GOYES   c7BL10  en début de buffer
0530 0039B 132              ADDEX
0531 0039E 31E5             LCASC   '^'
0532 003A2 AEA              A=C     B
0533 003A5 7E41             GOSUB   addchr  '^' normal
0534 003A9 400              RTNC            si erreur
0535 003AC 6B0F             GOTO    c71-10
0536 003B0 132      c7BL10  ADDEX
0537 003B3 32100            LC(3)   BOLS    '^' générique
0538 003B8 B31              B=B-C   X
0539 003BB 401              GOC     NoRoom
0540 003BE 306              LC(1)   BOL
0541 003C1 1540             DAT0=C  P
0542 003C5 160              D0=D0+  1
0543 003C8 6FEE             GOTO    c71-10
0544 003CC 6741     NoRoom  GOTO    noroom
0545 003D0 8E0000   c7E0L   GOSUBL  =lookah
0546 003D6 441              GOC     c7EL10  fin de ligne
0547 003D9 3142             LCASC   '$'
0548 003DD AEA              A=C     B
0549 003E0 7311             GOSUB   addchr  '$' normal
0550 003E4 400              RTNC            si erreur
```

```
Page 012          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0551 003E7 60DE             GOTO    c71-10
0552 003EB 32100    c7EL10  LC(3)   EOLS
0553 003F0 B31              B=B-C   X
0554 003F3 48D              GOC     NoRoom
0555 003F6 301              LC(1)   EOL
0556 003F9 1540             DAT0=C  P
0557 003FD 1E0              D0=D0+  1
0558 00400 67BE             GOTO    c71-10
0559 00404 32200    c7REPT  LC(3)   (CLOSS)+(ANYS)
0560 00409 B31              B=B-C   X
0561 0040C 4FB              GOC     NoRoom
0562 0040F 307              LC(1)   CLOS
0563 00412 1540             DAT0=C  P
0564 00416 160              D0=D0+  1
0565 00419 302              LC(1)   ANY
0566 0041C 1540             DAT0=C  P
0567 00420 160              D0=D0+  1
0568 00423 649E             GOTO    c71-10
0569 00427
0570 00427 330000   system  LC(4)   =eSYSER
0571 0042D 02               RTNSC
0572 0042F
0573 0042F          ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
0574 0042F          • getbuf
0575 0042F          •
0576 0042F          • But: allouer un buffer de taille maximum pour la chaine à
0577 0042F          •   compiler.
0578 0042F          • Entrees:
0579 0042F          •   - AVMEMS et AVMEME positionnés
0580 0042F          • Sortie:
0581 0042F          •   - D1 n'est pas modifié
0582 0042F          •   - D(A) = A(A) en entrée
0583 0042F          •   - D0 =   buffer (après la pile)
0584 0042F          •   - R0(2-0) = buffid
0585 0042F          •     R0(7-3) =   début du buffer (partie donnees)
0586 0042F          •   - B(A) = taille du buffer
0587 0042F          • Abime: A(15-5), B-D, D0, R0
0588 0042F          • Appelle: IOFSCR, I/OALL
0589 0042F          • Niveaux: 3 (I/OALL)
0590 0042F          • Detail: le buffer est alloué au maximum de la place
0591 0042F          •   mémoire disponible. STKSIZ quartets sont réservés pour
0592 0042F          •   stocker la pile. La taille renvoyée et le pointeur
0593 0042F          •   D0 tiennent compte de la pile. En revanche, le pointeur
0594 0042F          •   dans R0 est le vrai début du buffer
0595 0042F          • Historique:
0596 0042F          •   88/06/05: PD/JT conception & codage
0597 0042F          ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
0598 0042F
0599 0042F BF0      getbuf ASL     W
0600 00432 BF0             ASL     W
```

```
Page 013          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0601 00435 BF0              ASL     W
0602 00438 BF0              ASL     W
0603 0043B BF0              ASL     W
0604 0043E 133              AD1EX
0605 00441 100              R0=A            R0 := sauvegarde de A et D1
0606 00444          •
0607 00444          • Trouver un id scratch disponible
0608 00444          •
0609 00444 8F00000          GOSBVL  =IOFSCR A, C(A), D1, 1 niveau
0610 0044B 4BD              GOC     system  No available buffer
0611 0044E          •
0612 0044E          • C(X) = buffer id
0613 0044E          •
0614 0044E D5               B=C     A       B(A) := buffer id
0615 00450 1F00000          D1=(5)  =AVMEME
0616 00457 143              A=DAT1  A
0617 0045A 1C0              D1=D1-  (=AVMEME)-(=AVMEMS)
0618 0045D 147              C=DAT1  A
0619 00460 EA               A=A-C   A       A(A) := MEM en quartets
0620 00462 D2               C=0     A
0621 00464 3100             LC(2)   (=LEEWAY)*7
0622 00468 EA               A=A-C   A       A(A) := place restante après LEEWAY
0623 0046A 497              GOC     errmem
0624 0046D 34FFF00          LC(5)   #FFF    4095 nibs pour le buffer
0625 00474 8BA              ?A<=C   A
0626 00477 40               GOYES   getb10
0627 00479 DA               A=C     A       C(A) := 4095 nibs
0628 0047B          • B(A) = id du buffer
0629 0047B          • A(A) = taille demandée
0630 0047B 3481100  getb10 LC(5)    STKSIZ
0631 00482 8B2              ?A<C    A
0632 00485 F5               GOYES   errmem
0633 00487 D6               C=A     A       C(A) := taille demandée
0634 00489 DD               BCEX    A       B(A) = taille demandée, C(A) = bufid
0635 0048B 8F00000          GOSBVL  =I/OALL A-D, D1, D0, 2 niveaux
0636 00492 515              GONC    errmem
0637 00495          •
0638 00495          • D1 = ^ past buffer header
0639 00495          • B(A) = bufsize
0640 00495          • C(6-0) = header (C(1-3) = buffer id)
0641 00495          •
0642 00495 133              AD1EX
0643 00498 130              D0=A            D0 := ^ buffer
0644 0049B BF0              ASL     W
0645 0049E BF0              ASL     W
0646 004A1 BF0              ASL     W       A(7-3) = ^ buffer, A(2-0) = 000
0647 004A4 F6               CSR     A       C(X) = buffid
0648 004A6 A3A              A=A+C   X       A(7-3) = ^ buffer, A(2-0) = buffid
0649 004A9 120              AR0EX
0650 004AC 131              D1=A            D1 := ancien D1 restauré
```

```
Page 014          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0651 004AF BF4              ASR     W
0652 004B2 BF4              ASR     W
0653 004B5 BF4              ASR     W
0654 004B8 BF4              ASR     W
0655 004BB BF4              ASR     W       A(A) := ancien D1 restauré
0656 004BE D9               C=B     A
0657 004C0 D7               D=C     A       D(A) := taille du buffer
0658 004C2 D2               C=0     A
0659 004C4 307              LC(1)   7       pour le header
0660 004C7 E3               D=D-C   A
0661 004C9          •
0662 004C9          • D1 et A(A) ne sont pas modifiés
0663 004C9          • D0 = ^ buffer
0664 004C9          • R0(2-0) = bufid
0665 004C9          • R0(7-3) = ^ buffer
0666 004C9          • D(A) = taille du buffer
0667 004C9          •
0668 004C9 DB               C=D     A       C(A) := taille du buffer
0669 004CB DE               ACEX    A       A(A) := taille ; C(A) := LEN chaine
0670 004CD D7               D=C     A       D(A) := longueur de la chaine
0671 004CF D8               B=A     A       B(A) := taille du buffer
0672 004D1
0673 004D1 3481100          LC(5)   STKSIZ
0674 004D8 E1               B=B-C   A       B(A) := taille (- la pile)
0675 004DA 132              AD0EX
0676 004DD CA               A=A+C   A
0677 004DF 130              D0=A            D0 := ^ après la pile
0678 004E2          •
0679 004E2          • B(A) = taille restante dans le buffer
0680 004E2          • D(A) = taille de la chaine
0681 004E2          • D1 = ^ chaine
0682 004E2          • D0 = ^ buffer
0683 004E2          • R0(2-0) = bufid
0684 004E2          • R0(7-3) = ^ buffer
0685 004E2          •
0686 004E2 03               RTNCC
0687 004E4
0688 004E4 330000   errmem LC(4)   =eMEM
0689 004EA 02               RTNSC
0690 004EC
0691 004EC          ********************************************************
0692 004EC          • marque
0693 004EC          •
0694 004EC          • But: marque la position courante dans la chaine compilée
0695 004EC          •   comme "last token"
0696 004EC          • Entree:
0697 004EC          •   - D0 = ^ buffer
0698 004EC          • Sortie:
0699 004EC          •   - R1 = ^ last token
0700 004EC          • Abime: R1
```

```
Page 015          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0701 004EC          • Niveaux: 0
0702 004EC          • Historique:
0703 004EC          •   88/06/05: PD/JT conception & codage
0704 004EC          ********************************************************
0705 004EC
0706 004EC 136      marque CD0EX
0707 004EF 109             R1=C
0708 004F2 136             CD0EX
0709 004F5 01              RTN
0710 004F7
0711 004F7          ********************************************************
0712 004F7          • addchr
0713 004F7          •
0714 004F7          • But: ajoute un caractère au buffer, c'est à dire le token
0715 004F7          •   CHAR suivi du caractère.
0716 004F7          • Entree:
0717 004F7          •   - A(B) = caractère à écrire
0718 004F7          •   - B(A) = place restante dans le buffer
0719 004F7          •   - D0 =   buffer
0720 004F7          • Sortie:
0721 004F7          •   - Cy = 1 : erreur
0722 004F7          •       C(3-0) = error number
0723 004F7          •   - Cy = 0 : ça s'est bien passé
0724 004F7          •       D0 réactualisé
0725 004F7          • Abime: C(A)
0726 004F7          • Niveaux: 0
0727 004F7          • Historique:
0728 004F7          •   88/06/05: PD/JT conception & codage
0729 004F7          ********************************************************
0730 004F7
0731 004F7 32300    addchr LC(3)    CHARS   size of pattern
0732 004FC B31             B=B-C   X
0733 004FF 441             GOC     noroom
0734 00502 300             LC(1)   CHAR
0735 00505 1540            DAT0=C  P
0736 00509 160             D0=D0+  1
0737 0050C 148             DAT0=A  B
0738 0050F 161             D0=D0+  2       Cy = 0
0739 00512 01              RTN
0740 00514
0741 00514 7204     noroom GOSUB   =ENDPOS
0742 00518 330000          LC(4)   (=id)"(=tel(CMP) "No Room for Pattern"
0743 0051E 02              RTNSC           Cy = 1 : erreur
0744 00520
0745 00520          ********************************************************
0746 00520          • gettab
0747 00520          •
0748 00520          • But: obtenir l'adresse du caractère A(B) dans le tableau
0749 00520          •   pointé par D0.
0750 00520          • Entree:
```

```
Page 016          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0751 00520          •   - A(B) = numéro du caractère
0752 00520          •   - D0 = ^ premier caractère dans le tableau
0753 00520          • Sortie:
0754 00520          •   - D0 = ^ tab [caractère]
0755 00520          •   - C(A) = ancienne adresse (^ tab [0])
0756 00520          • Abime: C(A), A(4-2)
0757 00520          • Niveaux: 0
0758 00520          • Historique:
0759 00520          •   88/06/05: PD/JT conception & codage
0760 00520          ********************************************************
0761 00520
0762 00520 D2       gettab C=0      A
0763 00522 AE6             C=A     B
0764 00525 DA              A=C     A
0765 00527 136             CD0EX
0766 0052A 134             D0=C
0767 0052D C2              C=C+A   A
0768 0052F 136             CD0EX
0769 00532 01              RTN
0770 00534
0771 00534          ********************************************************
0772 00534          • POSADR
0773 00534          •
0774 00534          • But: chercher une chaine générique dans une chaine objet.
0775 00534          • Entree:
0776 00534          •   - D0 = ^ chaine à chercher
0777 00534          •   - A(A) = longueur de la chaine à chercher
0778 00534          •   - R0(7-0) = caractéristiques du buffer (cf. COMPUX/71)
0779 00534          •   - R1(A) = ^ vrai début de la chaine à chercher
0780 00534          • Sortie:
0781 00534          •   Cy = 1 : match found
0782 00534          •       - D0 = ^ occurrence dans la chaine objet
0783 00534          •       - C(A) = longueur de l'occurrence trouvée (en octets)
0784 00534          •   Cy = 0 : match not found ou erreur
0785 00534          • Abime: A-D, R0(12-8), R1, R2, D0, D1
0786 00534          • Appelle: amatch
0787 00534          • Niveaux: 3
0788 00534          • Notes:
0789 00534          •   - La chaine objet doit être dans le "bon" sens, c'est
0790 00534          •     dire celui des fichiers, et pas celui de la M.S. Cela
0791 00534          •     oblige à inverser la chaine à chercher si elle est sur
0792 00534          •     la M.S.
0793 00534          •   - R1(A) contient l'adresse du "vrai" début de la chaine
0794 00534          •     D0 contient l'adresse à partir de laquelle POSADR
0795 00534          •     cherche. Ceci est utile en cas de recherche ne
0796 00534          •     commençant pas au début (ex: substitution globale,
0797 00534          •     paramètre de GENPOS, etc.)
0798 00534          • Remarque: R0(7-0) n'est pas abimé
0799 00534          • Historique:
0800 00534          •   88/06/05: PD/JT conception & codage (de quatre lignes)
```

```
Page 017          Editeur, chaines génériques (xgen.as)
AREUH ASS. V2.4

0001 00534          •   88/06/26: PD/JT conception & codage (du début du reste)
0002 00534          •   88/07/02: PD/JT débogage et ajout de recherche multiple
0003 00534          •   88/10/07: PD/JT débogage
0004 00534          •   88/10/07: PD/JT retrait du compte dans R1(A) pour amatch
0005 00534          •   88/10/08: PD/JT débogage du cas "^"
0006 00534          •   88/10/09: PD/JT renvoie maintenant la longueur en octets
0007 00534          •   88/10/16: PD/JT séparation de UPDTR0
0008 00534          •   88/10/16: PD/JT suppression du point d'entrée POSID
0009 00534          ********************************************************
0010 00534
0011 00534 118      *POSADR C=R0
0012 00537 8F00000          GOSBVL  =CSRC3
0013 0053E 135              D1=f            D1 := ^ buffer
0014 00541          •
0015 00541          • D1 = ^ buffer (début des données)
0016 00541          • D0 =   chaine
0017 00541          • A(A) = longueur de la chaine en octets
0018 00541          •
0019 00541 133              AD1Ex           sauvegarde de A dans D1
0020 00544 DA               A=C     A       A(A) := longueur de la chaine
0021 00546 119              C=R1            C(A) := ^ vrai début
0022 00549 8F00000          GOSBVL  =CSRC5  C(15-11) := ^ vrai début
0023 00550 3481100          LC(5)   STKSIZ
0024 00557 C2               C=A+C   A       C(A) :=   début du pattern
0025 00559 133              AD1Ex           restauration de A(A)
0026 0055C 8F00000          GOSBVL  =CSLC5
0027 00563 109              R1=C            R1(9-5):=pattern; R1(A):=^vrai début
0028 00566
0029 00566 136              CD0Ex           C(A) := ^ chaine à examiner
0030 00569 135              D1=C            D1 := ^ chaine
0031 0056C D6               C=A     A
0032 0056E D7               D=C     A       D(A) := nb de caractères
0033 00570
0034 00570          •
0035 00570          • Si le premier caractère est normal, on rentre dans une
0036 00570          • boucle spéciale pour éviter la grosse machinerie de
0037 00570          • "amatch", et en particulier sa récursivité.
0038 00570          •
0039 00570 119              C=R1
0040 00573 8F00000          GOSBVL  =CSRC5  C(A) := pattern
0041 00578 134              D0=C
0042 0057D 1520             A=DAT0  P       A(0) := type du premier pattern
0043 00581 905              ?A#0    P
0044 00584 E0               GOYES   Poschr  boucle spéciale
0045 00586          •
0046 00586          • Si le premier caractère est "^", on ne teste qu'une
0047 00586          • fois sur le début de ligne.
0048 00586          •
0049 00586
0050 00586 306              LC(1)   BOL
```

```
Page 018          Editeur, chaines génériques (xgen.as)
AREUH ASS. V2.4

0051 00589 906              ?A#C    P
0052 0058C A0               GOYES   pos10
0053 0058E 68F0             GOTO    posbol  boucle spéciale pour "^"
0054 00592
0055 00592 5670     Poschr GOTO     poschr
0056 00596          •
0057 00596          • Tant pis, il faut y aller. On a signé, c'est...
0058 00596          •
0059 00596          •
0060 00596          • Invariant de boucle :
0061 00596          •   R1(9-5) = pattern; R1(A) =   vrai début
0062 00596          •   D(A) = nb de caractères dans la chaine à chercher
0063 00596          •   D1 = ^ premier caractère à chercher
0064 00596          •
0065 00596
0066 00596 119      pos10  C=R1
0067 00599 816              CSRC            C(A) := ^ début pattern
0068 0059C 816              CSRC
0069 0059F 816              CSRC
0070 005A2 816              CSRC
0071 005A5 816              CSRC
0072 005A8 134              D0=C            D0 := ^ début pattern
0073 005AB
0074 005AB 137              CD1EX
0075 005AE 135              D1=C
0076 005B1 812              CSLC            CSLC5
0077 005B4 812              CSLC
0078 005B7 812              CSLC
0079 005BA 812              CSLC
0080 005BD 812              CSLC
0081 005C0 D8               C=D     A
0082 005C2 10A              R2=C            sauvegarde de D et D1
0083 005C5 76F0             GOSUB   amatch
0084 005C9 430              GOC     pos20   match found
0085 005CC
0086 005CC 11A              C=R2            restauration de D et D1
0087 005CF D7               D=C     A
0088 005D1 BF6              CSR     W       CSR 5
0089 005D4 BF6              CSR     W
0090 005D7 BF6              CSR     W
0091 005DA BF6              CSR     W
0092 005DD BF6              CSR     W
0093 005E0 135              D1=C
0094 005E3
0095 005E3 171              D1=D1+  2
0096 005E6 CF               D=D-1   A
0097 005E8 5DA              GONC    pos10
0098 005EB 03       RtnCC  RTNCC            no match
0099 005ED
0100 005ED          pos20
```

```
Page 019          Editeur, chaines génériques (xgen.as)
AREUH ASS. V2.4

0101 005ED          •
0102 005ED          • D1 = ^ fin de l'occurrence dans la chaine objet
0103 005ED          • R2(9-5) = ^ début de l'occurence
0104 005ED          •
0105 005ED 11A              C=R2
0106 005F0 8F00000          GOSBVL  =CSRC5
0107 005F7          •
0108 005F7          • D1 = ^ fin de l'occurrence dans la chaine objet
0109 005F7          • C(A) = ^ début de l'occurrence
0110 005F7          •
0111 005F7 134      pos30  D0=C            D0 := ^ début de l'occurrence
0112 005FA AD0              A=0     M       pour la division par 2
0113 005FD 133              AD1EX           A(A) := ^ fin occurrence
0114 00600 EA               A=A-C   A       C(A) := 2•LEN(occurrence)
0115 00602 81C              ASRB
0116 00605 D6               C=A     A       C(A) := LEN(occurrence)
0117 00607          •
0118 00607          • D0 = ^ occurrence dans la chaine objet
0119 00607          • C(A) = longueur de l'occurrence trouvée (en octets)
0120 00607          •
0121 00607 02               RTNSC           match found
0122 00609
0123 00609          •
0124 00609          • Boucle spéciale sur le caractère
0125 00609          •
0126 00609 119      poschr C=R1            C(9-5) := ^ début pattern
0127 0060C 816              CSRC            CSRC 5
0128 0060F 816              CSRC
0129 00612 816              CSRC
0130 00615 816              CSRC
0131 00618 816              CSRC
0132 0061B 134              D0=C            D0 := ^ début pattern
0133 0061E
0134 0061E 160              D0=D0+  1
0135 00621 14A              A=DAT0  B       A(B) := premier caractère du motif
0136 00624 161              D0=D0+  2
0137 00627          •
0138 00627          • Boucle très rapide
0139 00627          •   D1 = ^ premier caractère à chercher
0140 00627          •   D(A) = nb de caractères dans la chaine à chercher
0141 00627          •   D0 = ^ patern "après" le premier caractère
0142 00627          •
0143 00627 8AB      posc10 ?D=0    A
0144 0062A 1C               GOYES   RtnCC   no match
0145 0062C 14F              C=DAT1  B
0146 0062F CF               D=D-1   A
0147 00631 171              D1=D1+  2
0148 00634 966              ?A#C    B
0149 00637 0F               GOYES   posc10
0150 00639
```

```
Page 020          Editeur, chaines génériques (xgen.as)
AREUH ASS. V2.4

0151 00639 137              CD1EX
0152 0063C 135              D1=C
0153 0063F BF2              CSL     W       CSL 5
0154 00642 BF2              CSL     W
0155 00645 BF2              CSL     W
0156 00648 BF2              CSL     W
0157 0064B BF2              CSL     W
0158 0064E D8               C=D     A
0159 00650 10A              R2=C            R2 := D1 et D(A)
0160 00653 7850             GOSUB   amatch
0161 00657 4D1              GOC     posc20  match found
0162 0065A
0163 0065A 11A              C=R2            restauration de D1 et D(A)
0164 0065D D7               D=C     A
0165 0065F BF6              CSR     W       CSR 5
0166 00662 BF6              CSR     W
0167 00665 BF6              CSR     W
0168 00668 BF6              CSR     W
0169 0066B BF6              CSR     W
0170 0066E 135              D1=C
0171 00671 679F             GOTO    poschr
0172 00675
0173 00675 11A      posc20 C=R2            C(9-5) := ^ deuxième caractère
0174 00678 8F00000          GOSBVL  =CSRC5
0175 0067F CE               C=C-1   A
0176 00681 CE               C=C-1   A       C(A) := ^ début de l'occurrence
0177 00683 637F             GOTO    pos30
0178 00687
0179 00687          •
0180 00687          • Portion de code spéciale pour ne tester qu'une seule
0181 00687          • fois le "^" de début de ligne
0182 00687          •
0183 00687 119      posbol C=R1            C(A) :=   vrai début
0184 0068A 133              AD1EX
0185 0068D 131              D1=A
0186 00690 8A6              ?A#C    A
0187 00693 62               GOYES   Rtncc
0188 00695
0189 00695 160              D0=D0+  1       passer le BOL
0190 00698
0191 00698 137              CD1EX
0192 0069B 135              D1=C
0193 0069E BF2              CSL     W       CSL 5
0194 006A1 BF2              CSL     W
0195 006A4 BF2              CSL     W
0196 006A7 BF2              CSL     W
0197 006AA BF2              CSL     W
0198 006AD D8               C=D     A
0199 006AF 10A              R2=C
1000 006B2 7900             GOSUB   amatch
```

```
Page 021          Editeur, chaines génériques <xgen.as
AREUH ASS. V2.4

1001 006B6 440            GOC     Pos20
1002 006B9 03       Rtncc RTNCC             no match
1003 006BB
1004 006BB 613F     Pos20 GOTO    pos20
1005 006BF
1006 006BF          ********************************************************
1007 006BF          * amatch
1008 006BF          *
1009 006BF          * But: reconnaitre une chaine de caractères
1010 006BF          * Entree:
1011 006BF          *   - D0 = ^ motif courant
1012 006BF          *   - D1 = ^ caractère courant, D(A) = nb de car. restant
1013 006BF          * Sortie:
1014 006BF          *   - Cy = 0 : match not found
1015 006BF          *   - Cy = 1 : match found
1016 006BF          *   - D0 inchangé
1017 006BF          *   - D1 et D(A) actualisés si besoin
1018 006BF          * Abime: A-D, R0(12-8), D0, D1
1019 006BF          * Appelle: omatch, nxtpat
1020 006BF          * Niveaux: 2
1021 006BF          * Algorithme:
1022 006BF          *   tant que pas arrivé à la fin du pattern
1023 006BF          *     si closure
1024 006BF          *       alors
1025 006BF          *         chercher la dernière possibilité (avec omatch)
1026 006BF          *         revenir en arriere jusqu'à ce que
1027 006BF          *               "amatch (reste) = TRUE"
1028 006BF          *         si trouvé alors
1029 006BF          *           alors return TRUE
1030 006BF          *           sinon return FALSE
1031 006BF          *         fin si
1032 006BF          *       sinon si omatch = FALSE alors return FALSE
1033 006BF          *     fin si
1034 006BF          *   fin tant que
1035 006BF          * Historique:
1036 006BF          *   88/06/26: PD/JT conception & codage
1037 006BF          *   88/07/02: PD/JT ajout des closures
1038 006BF          *   88/10/07: PD/JT retrait du compte dans R1(A) pour '''
1039 006BF          *   88/10/14: PD/JT bug GENLEN("AB","A.*B") corrigée
1040 006BF          ********************************************************
1041 006BF
1042 006BF 118      amatch C=R0             Initialisation de "' courant"
1043 006C2 816             CSRC             CSRC3
1044 006C5 816             CSRC
1045 006C8 816             CSRC
1046 006CB DA              A=C     A        A(A) := ^ début de la pile
1047 006CD 816             CSRC             CSRC5
1048 006D0 816             CSRC
1049 006D3 816             CSRC
1050 006D6 816             CSRC
```

```
Page 022          Editeur, chaines génériques <xgen.as
AREUH ASS. V2.4

1051 006D9 816             CSRC
1052 006DC D6              C=A     A
1053 006DE 812             CSLC             CSRC8
1054 006E1 812             CSLC
1055 006E4 812             CSLC
1056 006E7 812             CSLC
1057 006EA 812             CSLC
1058 006ED 812             CSLC
1059 006F0 812             CSLC
1060 006F3 812             CSLC
1061 006F6 108             R0=C             R0(12-8) :=  courant
1062 006F9
1063 006F9 1520     amarec A=DAT0  P
1064 006FD 305             LC(1)  END
1065 00700 902             ?A#C   P
1066 00703 B4              GOYES  return  match found
1067 00705 307             LC(1)  CLOS
1068 00708 902             ?A=C   P
1069 0070B 11              GOYES  amaclo  "closure"
1070 0070D 79F0            GOSUB  omatch
1071 00711 5C3             GONC   return  no match possible
1072 00714 7E71            GOSUB  nxtpat  D0 :=  motif suivant
1073 00718 60EF            GOTO   amarec
1074 0071C
1075 0071C          *
1076 0071C          * Closure trouvée. Les ennuis commencent.
1077 0071C          *
1078 0071C 160      amaclo D0=D0+  CLOSS   passer la closure
1079 0071F DB              C=D     A
1080 00721 D5              B=C     A        B(A) := pos. initiale pour closure
1081 00723 8E0000   acl010 GOSUBL  =lookah
1082 00729 490             GOC     acl100   on a fini la closure
1083 0072C 7AD0            GOSUB   omatch   tant qu'on trouve
1084 00730 42F             GOC     acl010
1085 00733          *
1086 00733          * On est arrivé au maximum. Faut revenir en arrière sur D1
1087 00733          * while D<>B
1088 00733          *   do
1089 00733          *     if amatch (reste) return TRUE
1090 00733          *
1091 00733 7F51     acl100 GOSUB   nxtpat  D0 est bien positionné
1092 00737
1093 00737 6D80     acl150 GOTO    call    appel recursif...
1094 0073B 421      aclret GOC     return  retour si amatch = TRUE
1095 0073E 1C1             D1=D1- 2
1096 00741 E7              D=D+1  A
1097 00743 DB              C=D    A
1098 00745 8B9             ?C<=B  A
1099 00748 FE              GOYES  acl150
1100 0074A          *
```

```
Page 023          Editeur, chaines génériques <xgen.as
AREUH ASS. V2.4

1101 0074A          * on est arrivé au bout sans trouver, alors :
1102 0074A          *
1103 0074A 6300            GOTO    return  et on a Cy = 0
1104 0074E
1105 0074E          *
1106 0074E          * Appels et retours récursifs...
1107 0074E          *
1108 0074E AC0      return A=0     S        sauvegarde de la carry
1109 00751 550             GONC    ret010
1110 00754 B44             A=A+1   S
1111 00757
1112 00757 118      ret010 C=R0
1113 0075A 816             CSRC             CSRC3
1114 0075D 816             CSRC
1115 00760 816             CSRC
1116 00763 DA              A=C     A        A(A) := ^ début
1117 00765 816             CSRC             CSRC5
1118 00768 816             CSRC
1119 0076B 816             CSRC
1120 0076E 816             CSRC
1121 00771 816             CSRC             C(A) := ^ courant
1122 00774 8A2             ?A=C    A
1123 00777 74              GOYES   ret090
1124 00779 137             CD1EX
1125 0077C          * D1 :=   courant : C(A) := ancienne valeur de D1
1126 0077C
1127 0077C          *
1128 0077C          * Correction du 88/10/14 : lorsqu'on a trouvé un
1129 0077C          * match, il ne faut surtout pas restaurer D1, D etc.
1130 0077C          * mais au contraire les préserver car ce sont eux qui
1131 0077C          * donneront la longueur du match.
1132 0077C          *
1133 0077C 948             ?A=0    S        retour sans avoir trouvé
1134 0077F B0              GOYES   ret020
1135 00781 1C9             D1=D1-  10       dépilage sans restauration
1136 00784 1C9             D1=D1-  10
1137 00787
1138 00787 512      * C(A) = ancienne valeur de D1 (celle qu'il faut renvoyer)
                           GONC    ret050  B.E.T.
1139 0078A          *
1140 0078A          * Fin de la correction du 88/10/14
1141 0078A          *
1142 0078A 1C4      ret020 D1=D1-  5        D0
1143 0078D 147             C=DAT1  A
1144 00790 134             D0=C
1145 00793
1146 00793 1C4             D1=D1-  5        B
1147 00796 147             C=DAT1  A
1148 00799 D5              B=C     A
1149 0079B
1150 0079B 1C4             D1=D1-  5        D
```

```
Page 024          Editeur, chaines génériques <xgen.as
AREUH ASS. V2.4

1151 0079E 147             C=DAT1  A
1152 007A1 D7              D=C     A
1153 007A3
1154 007A3 1C4             D1=D1-  5        D1
1155 007A6 147             C=DAT1  A
1156 007A9
1157 007A9          * Ce label est un ajout du 88/10/14, faisant pendant
1158 007A9          * à ce qui se trouve au dessus de ret020.
1159 007A9          *
1160 007A9          ret050
1161 007A9          *
1162 007A9          * C(A) = valeur de D1 à renvoyer
1163 007A9          * D1 = nouveau pointeur de pile réajusté
1164 007A9          *
1165 007A9 137             CD1EX            C(A) := ^ courant réactualisé
1166 007AC
1167 007AC 8F00000         GOSBVL  =CSLC8   C(12-8) := pointeur de pile
1168 007B3 108             R0=C             restauration du pointeur de pile
1169 007B6 7400            GOSUB   ret090   positionne la Cy
1170 007BA 608F            GOTO    aclret   retour de l'appel "récursif"
1171 007BE
1172 007BE 94C      ret090 ?A#0    S
1173 007C1 00              RTNYES           Cy = 1
1174 007C3 01              RTN              Cy = 0
1175 007C5
1176 007C5 118      call   C=R0             12-8 = ^ courant ; 7-2 =  début
1177 007C8 8F00000         GOSBVL  =CSRC8   C(A) := ^ courant
1178 007CF 137             CD1EX            D1 := ^ pile : C(A) := ancien D1
1179 007D2 15D4            DAT1=C  5
1180 007D6 174             D1=D1+  5
1181 007D9
1182 007D9 DF              CDEX    A        D
1183 007DB 145             DAT1=C  A
1184 007DE 174             D1=D1+  5
1185 007E1 DF              CDEX    A
1186 007E3
1187 007E3 DD              CBEX    A        B
1188 007E5 145             DAT1=C  A
1189 007E8 174             D1=D1+  5
1190 007EB DD              CBEX    A
1191 007ED
1192 007ED 136             CD0EX   A        D0
1193 007F0 145             DAT1=C  A
1194 007F3 174             D1=D1+  5
1195 007F6 136             CD0EX   A
1196 007F9
1197 007F9 137             CD1EX            C(A) := ^ courant
1198 007FC 8F00000         GOSBVL  =CSLC8
1199 00803 108             R0=C
1200 00806 62FE            GOTO    amarec
```

```
Page 025          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

1201 0080A
1202 0080A          ********************************************************
1203 0080A          • omatch
1204 0080A          •
1205 0080A          • But: chercher le motif courant dans la chaine
1206 0080A          • Entree:
1207 0080A          •   - D0 = ^ motif courant
1208 0080A          •   - D1 = ^ caractère courant, D(A) = nb de car. restant
1209 0080A          • Sortie:
1210 0080A          •   - Cy = 0 : match not found
1211 0080A          •   - Cy = 1 : match found
1212 0080A          •   - D0 inchangé
1213 0080A          •   - D1 et D(A) actualisés si besoin
1214 0080A          • Abime: A(A), C(W)
1215 0080A          • Appelle: lookah, rhcteg, TBLJMC, gettab
1216 0080A          • Niveaux: 1
1217 0080A          • Detail: cf. "Software tools in Pascal"
1218 0080A          • Historique:
1219 0080A          •   88/06/26: PD/JT conception & codage
1220 0080A          •   88/10/07: PD/JT retrait du compte dans R1(A) pour amatch
1221 0080A          ********************************************************
1222 0080A
1223 0080A 1564     omatch C=DAT0  S       C(S) := motif
1224 0080E A4E             C=C-1   S
1225 00811 5F1             GONC    oma10
1226 00814          •
1227 00814          • Déstructuration complète du code pour gagner le plus de
1228 00814          • cycles possibles
1229 00814          •
1230 00814           omCHAR
1231 00814          • GOSUBL =lookah
1232 00814 8AB             ?D=0    A
1233 00817 83              GOYES   rtncc
1234 00819 14B             A=DAT1  B
1235 0081C          • fin de GOSUBL =lookah
1236 0081C 160             D0=D0+  1
1237 0081F 14E             C=DAT0  B       C(B) := caractère trouvé
1238 00822 180             D0=D0-  1
1239 00825 966             ?A#C    B       A(B) = caractère lu par lookah
1240 00828 72              GOYES   rtncc   match not found
1241 0082A          • GOSUB =rhcteg on avance
1242 0082A 171             D1=D1+  2
1243 0082D CF              D=D-1   A
1244 0082F          • fin de GOSUB =rhcteg
1245 0082F 02              RTNSC           match found
1246 00831
1247 00831 A4E      oma10  C=C-1   S
1248 00834 4E1             GOC     omEOL
1249 00837 A4E             C=C-1   S
1250 0083A 4E1             GOC     omANY
```

```
Page 026          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

1251 0083D A4E             C=C-1   S
1252 00840 412             GOC     omCCL
1253 00843 A4E             C=C-1   S
1254 00846 4B1             GOC     omNCCL
1255 00849 A4E             C=C-1   S
1256 0084C 440             GOC     omEND
1257 0084F 03       rtncc  RTNCC           CLOS ou BOL : match not found
1258 00851
1259 00851 02       omEND  RTNSC           tout ma[rt]ch !
1260 00853
1261 00853          •
1262 00853          • omBOL : enlève le 88/10/07 car le cas '^' est traité à
1263 00853          • part dans POSADR.
1264 00853          •
1265 00853
1266 00853 8C0000   omEOL  GOLONG  =lookah Cy = match
1267 00859
1268 00859 7180     omANY  GOSUB   =rhcteg
1269 0085D 41F             GOC     rtncc
1270 00860 02              RTNSC           match found
1271 00862          omCCL
1272 00862 8E0000   omNCCL GOSUBL  =lookah
1273 00868 46E             GOC     rtncc
1274 0086B 160             D0=D0+  1
1275 0086E 7EAC            GOSUB   gettab  D0 := ^ élément
1276 00872 1564            C=DAT0  S       C(S) := tab [caractère]
1277 00876 134             D0=C
1278 00879 180             D0=D0-  1       D0 := ^ CCL ou NCCL
1279 0087C 1520            A=DAT0  P       A(0) := CCL ou NCCL
1280 00880 303             LC(1)   CCL
1281 00883 902             ?A=C    P
1282 00886 50              GOYES   omCC10
1283 00888 A4E             C=C-1   S       si NCCL : C(S) := not tab [carlu]
1284 0088B 94A      omCC10 ?C=0    S
1285 0088E 1C              GOYES   rtncc   no match
1286 00890 7A40            GOSUB   =rhcteg
1287 00894 02              RTNSC           match found
1288 00896
1289 00896          ********************************************************
1290 00896          • nxtpat
1291 00896          •
1292 00896          • But: positionner D0 sur le motif suivant
1293 00896          • Entree:
1294 00896          •   - D0 :=   motif courant
1295 00896          • Sortie:
1296 00896          •   - D0 :=   motif suivant
1297 00896          • Abime: A(A), C(A), D0
1298 00896          • Appelle: -
1299 00896          • Niveaux: 1
1300 00896          • Detail: cf. "Software tools in Pascal", c'est la nouvelle
```

```
Page 027          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

1301 00896          •   fonction "patsize"
1302 00896          • Historique:
1303 00896          •   88/06/26: PD/JT conception & codage
1304 00896          ********************************************************
1305 00896
1306 00896 1524     nxtpat A=DAT0  S
1307 00898 94C             ?A#0    S
1308 0089D 70              GOYES   ps05
1309 0089F 162             D0=D0+  CHARS
1310 008A2 01              RTN
1311 008A4
1312 008A4 7810     ps05   GOSUB   ps10
1313 008A8 300             CON(3)  CHARS   inutile
1314 008AB 100             CON(3)  EOLS
1315 008AE 100             CON(3)  ANYS
1316 008B1 101             CON(3)  CCLS
1317 008B4 101             CON(3)  NCCLS
1318 008B7 000             CON(3)  0       END
1319 008BA 000             CON(3)  0       BOL
1320 008BD 100             CON(3)  CLOSS
1321 008C0 D0       ps10   A=0     A
1322 008C2 1520            A=DAT0  P       A(A) := motif courant
1323 008C6 D6              C=A     A
1324 008C8 CA              A=A+C   A
1325 008CA CA              A=A+C   A       A(A) := 3*motif
1326 008CC 07              C=RSTK          C(A) := ^ début de la table
1327 008CE C2              C=C+A   A
1328 008D0 136             CD0EX
1329 008D3 15A2            A=DAT0  3
1330 008D7 C2              C=A+C   A
1331 008D9 134             D0=C
1332 008DC 01              RTN
1333 008DE
1334 008DE          ********************************************************
1335 008DE          • rhcteg
1336 008DE          •
1337 008DE          • But: lire un caractère, et passer au suivant (getchr)
1338 008DE          • Entree:
1339 008DE          •   - D1 = ^ chaine
1340 008DE          •   - D(A) = nb de caractères
1341 008DE          • Sortie:
1342 008DE          •   - Cy = 1 : fin de la chaine
1343 008DE          •   - Cy = 0, A(B) = caractère lu
1344 008DE          • Abime: A(B)
1345 008DE          • Appelle: -
1346 008DE          • Niveaux: 0
1347 008DE          • Historique:
1348 008DE          •   88/06/26: PD/JT conception & pompage (getchr et lookah)
1349 008DE          ********************************************************
1350 008DE
```

```
Page 028          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

1351 008DE 8AB      =rhcteg ?D=0    A
1352 008E1 00               RTNYES
1353 008E3 CF               D=D-1   A
1354 008E5 14B              A=DAT1  B
1355 008E8 171              D1=D1+  2
1356 008EB 01               RTN             Cy = 0
1357 008ED
1358 008ED          ********************************************************
1359 008ED          • UPDTR0
1360 008ED          •
1361 008ED          • But: mettre à jour l'adresse du buffer contenue dans
1362 008ED          •   R0(7-3) à partir de son ID contenu dans R0(2-0).
1363 008ED          • Entree:
1364 008ED          •   - R0(2-0) = ID du buffer
1365 008ED          • Sortie:
1366 008ED          •   - Cy = 1 : erreur
1367 008ED          •        - C(3-0) = numéro d'erreur (System Error)
1368 008ED          •   - Cy = 0 : pas d'erreur
1369 008ED          •        - R0(7-0) = caractéristique complète du buffer
1370 008ED          •        - D1 = ^ past buffer header
1371 008ED          • Abime: A, C, D1
1372 008ED          • Appelle: I/OFND0
1373 008ED          • Niveaux: 1
1374 008ED          • Historique:
1375 008ED          •   88/10/16: PD/JT séparation de POSADR
1376 008ED          ********************************************************
1377 008ED
1378 008ED 693B      system GOTO    system
1379 008F1
1380 008F1 118      =UPDTR0 C=R0            C(X) := buffer id
1381 008F4 8F00000          GOSBVL  =IOFND0
1382 008FB 51F              GONC    system  "System Error" + Cy = 1
1383 008FE          •
1384 008FE          • C(X) = ID
1385 008FE          • D1 = ^ past buffer header
1386 008FE          •
1387 008FE 118              C=R0            actualisation de R0(7-2)
1388 00901 8F00000          GOSBVL  =CSRC3
1389 00908 137              CD1EX
1390 0090B 135              D1=C
1391 0090E 8F00000          GOSBVL  =CSLC3
1392 00915 108              R0=C
1393 00918 03               RTNCC           pas d'erreur
1394 0091A
1395 0091A          ********************************************************
1396 0091A          • ENDPOS
1397 0091A          •
1398 0091A          • But: terminer une recherche générique, c'est à dire
1399 0091A          •   effacer le buffer.
1400 0091A          • Entree:
```

```
Page 029          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

1401 0091A          *   - R0(7-2) = caractéristique du buffer
1402 0091A          * Sortie: -
1403 0091A          * Abime: A-C, D1, D0
1404 0091A          * Appelle: I/ODAL
1405 0091A          * Niveaux: 2
1406 0091A          * Historique:
1407 0091A          *   88/07/02: PD/JT conception & codage
1408 0091A          ***************************************************
1409 0091A
1410 0091A 118      =ENDPOS C=R0
1411 0091D 8D00000          GOVLNG  =I/ODAL
1412 00924
```

```
Page 030          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1414 00924
1415 00924          ***************************************************
1416 00924          * CALCab
1417 00924          *
1418 00924          * But: calculer les coefficients a et b permettant de
1419 00924          *   connaitre la longueur de la chaine remplacée. La routine
1420 00924          *   REPLIN les utilise.
1421 00924          * Entree:
1422 00924          *   - R3 = motif de remplacement (dans le sens Math Stack)
1423 00924          *   - MODE71 = 1 si mode HP-71, 0 si mode HP-UX
1424 00924          * Sortie:
1425 00924          *   - les zones "coeffa" et "coeffb" sont remplies.
1426 00924          * Abime: A(S), A(A), B(A), C(S), C(A), D(A), D1
1427 00924          * Niveaux: 1 (utilisation de RSTK)
1428 00924          * Note : Le registre R3 se décompose en
1429 00924          *   - R3(9-5) = adresse du prochain caractère à lire
1430 00924          *   - R3(4-0) = nombre de caractères dans la chaine
1431 00924          * Detail : les coefficients a et b sont respectivement le
1432 00924          *   nombre de caractères constants (normaux) et le nombre de
1433 00924          *   caractères '&' dans la chaine.
1434 00924          *   Dans ces conditions, si "n" est la longueur de
1435 00924          *   l'occurrence trouvée, f(n) = a + bn est la longueur de
1436 00924          *   la chaine après remplacement.
1437 00924          * Historique:
1438 00924          *   88/10/08: PD/JT conception
1439 00924          ***************************************************
1440 00924
1441 00924 118      =CALCab C=R3
1442 00927 D7               D=C     A       D(A) := longueur du motif
1443 00929 8F00000          GOSBVL  =CSRC5  C(A) := ' premier caractère du motif
1444 0092D 1F00000          D1=(5)  =MODE71
1445 00931 1574             C=DAT1  S       C(S) := 0 si mode HP-UX
1446 0093B 135              D1=C            D1 := ' premier caractère du motif
1447 0093E
1448 0093E AC0              A=0     S       A(S) := pas de "\"
1449 00941 D1               B=0     A       a := 0 (caractère normal)
1450 00943 D2               C=0     A       b := 0 (caractère '&')
1451 00945 06               RSTK=C
1452 00947
1453 00947 94E              ?C#0    S       mode HP-71 ?
1454 0094A E3               GOYES   CALC50  oui
1455 0094C          *
1456 0094C          * Mode HP-UX
1457 0094C          *
1458 0094C 8E0000   CALC10 GOSUBL   =getchr
1459 00952 413              GOC     calc99  on est arrivé au bout
1460 00955
1461 00955 94C              ?A#0    S       selon état
1462 00958 32               GOYES   CALC25  "\" avant ce caractère
1463 0095A          * pas de "\"
```

```
Page 031          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1464 0095A 31C5             LC(2)   XESC    "\"
1465 0095E 962              ?A=C    B
1466 00961 41               GOYES   CALC20  état := 1 ; next one
1467 00963 3162             LC(2)   XREP    "&"
1468 00967 966              ?A#C    B
1469 0096A 41               GOYES   CALC30  caractère normal
1470 0096C          * caractère '&' générique
1471 0096C 07               C=RSTK
1472 0096E E6               C=C+1   A       b++
1473 00970 06               RSTK=C
1474 00972 59D              GONC    CALC10  next one (B.E.T.)
1475 00975 B44      CALC20  A=A+1   S       état := 1
1476 00978 53D              GONC    CALC10  B.E.T.
1477 0097B          * Caractère normal
1478 0097B AC0      CALC25  A=0     S       état := 0
1479 0097E E5       CALC30  B=B+1   A       a++
1480 00980 6BCF             GOTO    CALC10
1481 00984
1482 00984 6180     calc99  GOTO    CALC99
1483 00988
1484 00988          *
1485 00988          * Mode HP-71
1486 00988          *
1487 00988 8E0000   CALC50  GOSUBL  =getchr
1488 0098E 477              GOC     CALC99  on a fini
1489 00991          *
1490 00991          * Automate d'états fini :
1491 00991          * - état = 0 : normal
1492 00991          * - état = 1 : précédent caractère etait un "\"
1493 00991          * - état = 2 : caractères génériques OFF
1494 00991          * - état = 3 : état 2 + précédent caractère était un "\"
1495 00991          *
1496 00991 31C5             LC(2)   STOGL   pour économiser des quartets
1497 00995
1498 00995 AC6              C=A     S       C(S) := état de l'automate
1499 00998 A4E              C=C-1   S
1500 0099B 4F1              GOC     CALC60
1501 0099E A4E              C=C-1   S
1502 009A1 442              GOC     CALC61
1503 009A4 A4E              C=C-1   S
1504 009A7 414              GOC     CALC62
1505 009AA 962      CALC63  ?A=C    B
1506 009AD 80               GOYES   CAL630
1507 009AF AC0              A=0     S       état := 0
1508 009B2 4D4              GOC     CALC70  B.E.T.
1509 009B5 A4C      CAL630  A=A-1   S       état := 2
1510 009B8 574              GONC    CALC70  B.E.T.
1511 009BB
1512 009BB 966      CALC60  ?A#C    B
1513 009BE 24               GOYES   CALC70  a++ (B.E.T.)
```

```
Page 032          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1514 009C0 B44              A=A+1   S       état := 1
1515 009C3 54C              GONC    CALC50  B.E.T.
1516 009C6
1517 009C6 962      CALC61  ?A=C    B
1518 009C9 11               GOYES   CAL610  "\"
1519 009CB 3162             LC(2)   SREP    "&"
1520 009CF 962              ?A=C    B
1521 009D2 E0               GOYES   CAL615  "&"
1522 009D4 B44              A=A+1   S       état := 2
1523 009D7 582              GONC    CALC70  a++ (B.E.T.)
1524 009DA AC0      CAL610  A=0     S
1525 009DD 422              GOC     CALC70  a++ (B.E.T.)
1526 009E0 07       CAL615  C=RSTK
1527 009E2 E6               C=C+1   A
1528 009E4 06               RSTK=C
1529 009E6 51A              GONC    CALC50  B.E.T.
1530 009E9
1531 009E9 962      CALC62  ?A=C    B
1532 009EC E0               GOYES   CAL620  "\"
1533 009EE 3162             LC(2)   SREP    "&"
1534 009F2 966              ?A#C    B
1535 009F5 B0               GOYES   CALC70  a++
1536 009F7 58E              GONC    CAL615  spaghetti... mais c'est un B.E.T.
1537 009FA B44      CAL620  A=A+1   S       état := 3
1538 009FD 5A8              GONC    CALC50  B.E.T.
1539 00A00
1540 00A00 E5       CALC70  B=B+1   A       a++
1541 00A02 658F             GOTO    CALC50
1542 00A06
1543 00A06          *
1544 00A06          * Etat final
1545 00A06          * Stockage des coefficients
1546 00A06          *
1547 00A06 1F00000  CALC99  D1=(5)  =coeffa
1548 00A0D D3               C=B     A
1549 00A0F 145              DAT1=C  A       coeffa := B(A)
1550 00A12 174              D1=D1+  5       coeffb
1551 00A15 07               C=RSTK
1552 00A17 145              DAT1=C  A       coeffb := RSTK
1553 00A1A 01               RTN
1554 00A1C
1555 00A1C          ***************************************************
1556 00A1C          * REPLIN
1557 00A1C          *
1558 00A1C          * But: remplacer un motif par un autre dans une chaine ou
1559 00A1C          *   il y a déjà une occurrence du motif.
1560 00A1C          * Entree:
1561 00A1C          *   - R0(7-0) = caractéristique du buffer (chaine compilee)
1562 00A1C          *   - (OUTBS..AVMEMS) = chaine à traiter
1563 00A1C          *   - AVMEME indique la fin de la mémoire disponible
```

```
Page 033          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1564 00A1C          •   - R3 = motif de remplacement (dans le sens Math Stack)
1565 00A1C          •   - D0 = ' occurrence trouvée
1566 00A1C          •   - C(A) = longueur de l'occurrence
1567 00A1C          •   - coeffa et coeffb calculés par CALCab
1568 00A1C          •   - QUERY = 1 si recherche interactive ("?"), 0 sinon
1569 00A1C          •   - A(A) = numéro de la ligne en cours (si QUERY = 1)
1570 00A1C          •   - MODE71 = 1 si chaines génériques HP-71, 0 si HP-UX
1571 00A1C          • Sortie:
1572 00A1C          •   - Cy = 0 : ok
1573 00A1C          •     suivant C(0) :
1574 00A1C          •       0 : quit sans modification
1575 00A1C          •       1 : quit avec quelque chose déjà modifié
1576 00A1C          •       2 : ok sans modification
1577 00A1C          •       3 : ok avec quelque chose de modifié
1578 00A1C          •     (OUTBS..AVMEMS) = zone où est stockée la ligne finale
1579 00A1C          •   - Cy = 1 : erreur
1580 00A1C          •     C(3-0) = numéro d'erreur
1581 00A1C          • Abime: A-D, R0(12-8), R1-R2, R3(10), R4(15-11), R4(9-5),
1582 00A1C          •   D0, D1
1583 00A1C          • Appelle: query, POSADR
1584 00A1C          • Niveaux: 4 (POSADR) ou 5 (query)
1585 00A1C          • Note : Le registre R3 se décompose en
1586 00A1C          •   - R3(9-5) = adresse du prochain caractère à lire
1587 00A1C          •   - R3(4-0) = nombre de caractères dans la chaîne
1588 00A1C          • Algorithme:
1589 00A1C          •   répéter
1590 00A1C          •     procéder au remplacement
1591 00A1C          •     si mode = "?"
1592 00A1C          •       alors
1593 00A1C          •         demander confirmation
1594 00A1C          •         selon réponse
1595 00A1C          •           Q : sortir avec le code "QUIT"
1596 00A1C          •           N : annuler le remplacement
1597 00A1C          •           Y :
1598 00A1C          •         fin selon
1599 00A1C          •     fin si
1600 00A1C          •     rechercher l'occurrence suivante
1601 00A1C          •   jusqu'à ce qu'il n'y ait plus d'occurrence
1602 00A1C          • Historique:
1603 00A1C          •   88/10/08: PD/JT conception
1604 00A1C          •   88/10/14: PD/JT correction du cas "match nul"
1605 00A1C          •   88/10/15: PD/JT correction du cas "R/'.//"
1606 00A1C          ********************************************************
1607 00A1C
1608 00A1C 330000     Errmem LC(4)   =eMEM   Insufficient Memory
1609 00A22 02         errrtn RTNSC           Cy = 1 => erreur
1610 00A24
1611 00A24          =REPLIN
1612 00A24          •
1613 00A24          • Sauvegarder la valeur de A(A) dans R4(9-5). On utilise R4,
```

```
Page 034          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1614 00A24          • les temps sont vraiment durs...
1615 00A24          •
1616 00A24 8F00000          GOSBVL  =ASLW5
1617 00A28 104              R4=A            R4(9-5) = no de la ligne en cours
1618 00A2E          •
1619 00A2E          • Mettre à 0 l'indicateur de modifications
1620 00A2E          •
1621 00A2E 113              A=R3
1622 00A31 2A               P=      10
1623 00A33 A80              A=0     P
1624 00A36 103              R3=A
1625 00A39 20               P=      0
1626 00A3B          •
1627 00A3B          • Boucle tant qu'il y a des occurrences. En fait, c'est
1628 00A3B          • une boucle "repeat..until". Voir à ce propos (différences
1629 00A3B          • entre les boucles WHILE et REPEAT) l'excellent article de
1630 00A3B          • notre excellent camarade Janick Taillandier dans
1631 00A3B          • l'excellent JPC 31.
1632 00A3B          •
1633 00A3B           RPL100
1634 00A3B          •
1635 00A3B          • Assertions :
1636 00A3B          • - D0 = ' occurrence trouvée dans la chaîne à traiter
1637 00A3B          • - C(A) = longueur de l'occurrence (en caractères)
1638 00A3B          • - R0(7-0) = caractéristique du buffer
1639 00A3B          • - R3 = motif de remplacement (dans le sens Math Stack)
1640 00A3B          • - R4(9-5) = numéro de la ligne courante (si mode = "?")
1641 00A3B          •
1642 00A3B          • Procéder au remplacement :
1643 00A3B          •
1644 00A3B          • n = longueur de l'occurrence trouvée (dans RSTK)
1645 00A3B          • Calcul de f(n) = a + b*n
1646 00A3B          •
1647 00A3B 06               RSTK=C          RSTK := n (longueur de l'occurrence)
1648 00A3D AF0              A=0     W
1649 00A40 DA               A=C     A       A(W) := n
1650 00A42 1F00000          D1=(5)  =coeffb
1651 00A49 AF2              C=0     W
1652 00A4C 147              C=DAT1  A       C(W) := b
1653 00A4F 8F00000          GOSBVL  =MPY    résultat dans A, B et C
1654 00A56 AF0              A=0     W       vérification du résultat
1655 00A59 DA               A=C     A
1656 00A5B 8A6              ?A#C    A       <> 1048575 ?
1657 00A5E EB               GOYES   Errmem  non : Not Enough Memory
1658 00A60          • A(A) = B(A) = C(A) := b*n
1659 00A60 1C4              D1=D1-  5       D1=(5) coeffa
1660 00A63 143              A=DAT1  A
1661 00A66 CA               A=A+C   A       A(A) := f(n)    (a + b*n)
1662 00A68 43B              GOC     Errmem  Overflow
1663 00A6B          •
```

```
Page 035          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1664 00A6B          • Assertions :
1665 00A6B          • - RSTK = n
1666 00A6B          • - A(A) = f(n)
1667 00A6B          •
1668 00A6B 07               C=RSTK          C(A) := n
1669 00A6D D8               B=A     A
1670 00A6F E1               B=B-C   A       B(A) := f(n) - n
1671 00A71 8F00000          GOSBVL  =ASLW5  A(9-5) := f(n)
1672 00A78 DA               A=C     A       A(4-0) := n
1673 00A7A 101              R1=A            R1(9-5) := f(n); R1(A) := n
1674 00A7D
1675 00A7D 1F00000          D1=(5)  =AVMEMS
1676 00A84 143              A=DAT1  A       A(A) := AVMEMS
1677 00A87 D6               C=A     A
1678 00A89 C9               C=C+B   A       C(A) := AVMEMS + f(n) - n
1679 00A8B C9               C=C+B   A       C(A) := AVMEMS + 2*(f(n) - n)
1680 00A8D 8B2              ?C>A    A
1681 00A90 40               GOYES   RPL120
1682 00A92 D6               C=A     A       C(A) := MAX (...)
1683 00A94 8F00000   RPL120 GOSBVL  =CSLC5  C(9-5) := MAX (...)
1684 00A9B 136              CD0EX           C(A) := D0
1685 00A9E 10A              R2=C
1686 00AA1
1687 00AA1          •
1688 00AA1          • Assertions :
1689 00AA1          •   R1(9-5) = f(n)      (longueur du remplacement)
1690 00AA1          •   R1(4-0) = n         (longueur de l'occurrence)
1691 00AA1          •   R2(9-5) = MAX (AVMEMS, AVMEMS + 2*(f(n) - n))
1692 00AA1          •   R2(4-0) = D0        (début de l'occurrence)
1693 00AA1          •
1694 00AA1
1695 00AA1          •
1696 00AA1          • Le décalage se produit en plusieurs étapes :
1697 00AA1          • - copier l'occurrence à l'adresse
1698 00AA1          •   MAX (AVMEMS, AVMEMS + f(n) - n)
1699 00AA1          • - copier ce qu'il y a entre la fin de l'occurrence et
1700 00AA1          •   AVMEMS à l'adresse (début de l'occurrence + f(n))
1701 00AA1          • - remplacer l'occurrence par le motif
1702 00AA1          • - mettre à jour AVMEMS
1703 00AA1          •
1704 00AA1
1705 00AA1          •
1706 00AA1          • Il y a suffisamment de mémoire : on demande 2*f(n) ?
1707 00AA1          •
1708 00AA1 119              C=R1
1709 00AA4 8F00000          GOSBVL  =CSRC5  C(A) := f(n)
1710 00AAB C6               C=C+C   A       C(A) := 2*f(n)
1711 00AAD 21               P=      1       P#0 pour ne pas avoir de LEEWAY
1712 00AAF 8F00000          GOSBVL  =MEMCKL Memory Check without LEEWAY
1713 00AB6          • P=0 en sortie de MEMCKL
```

```
Page 036          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1714 00AB6 400              RINC            Bip ! "Insufficient Memory"
1715 00AB9
1716 00AB9          •
1717 00AB9          • Préparation des paramètres pour MOVE•M :
1718 00AB9          • - A(A) = source address = D0
1719 00AB9          • - C(A) = destination address = MAX (AVMEMS, AVMEMS+f(n)-n)
1720 00AB9          • - B(A) = longueur en nibs = 2*n
1721 00AB9          •
1722 00AB9 119              C=R1
1723 00ABC D5               B=C     A
1724 00ABE C5               B=B+B   A       B(A) := 2*n
1725 00AC0 11A              C=R2
1726 00AC3 DA               A=C     A       A(A) := D0
1727 00AC5 8F00000          GOSBVL  =CSRC5  C(A) := MAX (...)
1728 00ACC 8F00000          GOSBVL  =MOVE•M
1729 00AD3
1730 00AD3          •
1731 00AD3          • Préparation des paramètres pour MOVE•M :
1732 00AD3          • - A(A) = source address = D0 + 2*n
1733 00AD3          • - C(A) = destination address = D0 + 2*f(n)
1734 00AD3          • - B(A) = longueur en nibs = AVMEMS - (D0 + 2*n)
1735 00AD3          •
1736 00AD3 119              C=R1            C(A) := n
1737 00AD6 112              A=R2            A(A) := D0
1738 00AD9 C6               C=C+C   A
1739 00ADB CA               A=C+A   A       A(A) := D0 + 2*n
1740 00ADD
1741 00ADD 1B00000          D0=(5)  =AVMEMS
1742 00AE4 146              C=DAT0  A
1743 00AE7 E2               C=C-A   A
1744 00AE9 D5               B=C     A       B(A) := AVMEMS - (D0 + 2*n)
1745 00AEB
1746 00AEB 8F00000          GOSBVL  =CSRC5  C(A) := f(n)
1747 00AF2 D7               D=C     A
1748 00AF4 11A              C=R2
1749 00AF7 CB               C=C+D   A
1750 00AF9 CB               C=C+D   A       C(A) := D0 + 2*f(n)
1751 00AFB          •
1752 00AFB          • On pourrait peut-être gagner quelques cycles si le
1753 00AFB          • déplacement avait lieu à la même adresse, mais les
1754 00AFB          • routines appelées par MOVE•M détectent presque
1755 00AFB          • immédiatement ce cas. Il n'y a donc pas lieu de
1756 00AFB          • s'en inquiéter. Nous n'évitons donc pas ce MOVE•M,
1757 00AFB          • tant pis pour jt (qui se serait bien passé d'un MOVE).
1758 00AFB          •
1759 00AFB 8F00000          GOSBVL  =MOVE•M
1760 00B02          •
1761 00B02          • remplacement de la chaîne à l'adresse D0 par la
1762 00B02          • chaîne à l'adresse R3, le motif original étant à
1763 00B02          • l'adresse MAX (AVMEMS, AVMEMS + f(n) - n)
```

```
Page 037          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1764 00B02            •
1765 00B02 11A              C=R2
1766 00B05 134              D0=C            D0 := ' espace à remplir
1767 00B08
1768 00B08 1F00000          D1=(5)  =MODE71 1 si mode HP-71 ou 0 si mode HP-UX
1769 00B0F 1534             A=DAT1  S       A(S) := 1 si mode HP-71
1770 00B13
1771 00B13 11B              C=R3
1772 00B16 D7               D=C     A       D(A) := nombre de caractères
1773 00B18 8F00000          GOSBVL  =CSRC5
1774 00B1F 135              D1=C            D1 := ' premier caractère dans M.S.
1775 00B22
1776 00B22 94C              ?A#0    S       mode HP-71 ?
1777 00B25 64               GOYES   RPL300
1778 00B27            •
1779 00B27            • Mode HP-UX
1780 00B27            •
1781 00B27            RPL200
1782 00B27            •       A=0     S       A était déjà égal à 0
1783 00B27
1784 00B27            RPL290
1785 00B27            •
1786 00B27            • Le label RPL290 est là pour éviter des GOTO qui branchent
1787 00B27            • à des GOTO.
1788 00B27            •
1789 00B27 8E0000     RPL210 GOSUBL  =getchr
1790 00B2D 493              GOC     rpl500
1791 00B30
1792 00B30 94C              ?A#0    S       précédent caractère = "\"
1793 00B33 D1               GOYES   RPL240
1794 00B35 31C5             LC(2)   xESC
1795 00B39 962              ?A=C    B
1796 00B3C E0               GOYES   RPL220
1797 00B3E 3162             LC(2)   xREP    "&"
1798 00B42 962              ?A=C    B
1799 00B45 11               GOYES   RPL270  rplc
1800 00B47 561              GONC    RPL280  caractère normal (B.E.T.)
1801 00B4A B44        RPL220 A=A+1  S       état := 1
1802 00B4D 59D              GONC    RPL290  next one
1803 00B50
1804 00B50 A4C        RPL240 A=A-1  S       état := 0
1805 00B53 5A0              GONC    RPL280  B.E.T.
1806 00B56
1807 00B56 7062       RPL270 GOSUB  rplc
1808 00B5A 6CCF             GOTO    RPL290
1809 00B5E 148        RPL280 DAT0=A B       ajoute un caractère
1810 00B61 161              D0=D0+  2
1811 00B64 52C              GONC    RPL210  B.E.T.
1812 00B67
1813 00B67 6C80       rpl500 GOTO   RPL500
```

```
Page 038          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1814 00B6B
1815 00B6B            •
1816 00B6B            • Mode HP-71
1817 00B6B            •
1818 00B6B AC0        RPL300 A=0     S
1819 00B6E
1820 00B6E            RPL390
1821 00B6E            •
1822 00B6E            • L'étiquette RPL390 est là pour éviter les GOTO qui
1823 00B6E            • branchent à des GOTO.
1824 00B6E            •
1825 00B6E 8E0000     RPL310 GOSUBL  =getchr
1826 00B74 4F7              GOC     RPL500
1827 00B77
1828 00B77 31C5             LC(2)   STOGL
1829 00B7B AC6              C=A     S       (S) := état
1830 00B7E A4E              C=C-1   S
1831 00B81 4F1              GOC     RPL320
1832 00B84 A4E              C=C-1   S
1833 00B87 442              GOC     RPL321
1834 00B8A A4E              C=C-1   S
1835 00B8D 4E3              GOC     RPL322
1836 00B90
1837 00B90 962        RPL323 ?A=C   B
1838 00B93 80               GOYES   RP3230
1839 00B95 AC0              A=0     S       état := 0
1840 00B98 525              GONC    RPL380  ajouter normal (B.E.T.)
1841 00B9B A4C        RP3230 A=A-1  S       état := 2
1842 00B9E 5C4              GONC    RPL380  ajouter normal (B.E.T.)
1843 00BA1
1844 00BA1 966        RPL320 ?A#C   B
1845 00BA4 74               GOYES   RPL380  ajouter normal
1846 00BA6 B44              A=A+1   S
1847 00BA9 54C              GONC    RPL390  B.E.T.
1848 00BAC
1849 00BAC 962        RPL321 ?A=C   B
1850 00BAF 11               GOYES   RP3210  "\"
1851 00BB1 3162             LC(2)   SREP    "&"
1852 00BB5 962              ?A=C    B
1853 00BB8 E0               GOYES   RP3215  "&"
1854 00BBA B44              A=A+1   S       état := 2
1855 00BBD 5D2              GONC    RPL380  ajoute normal
1856 00BC0 A4C        RP3210 A=A-1  S       état := 0
1857 00BC3 572              GONC    RPL380  ajoute normal (B.E.T.)
1858 00BC6 B44        RP3215 A=A+1  S       état := 2
1859 00BC9 591              GONC    RPL370  rplc (B.E.T.)
1860 00BCC
1861 00BCC 962        RPL322 ?A=C   B
1862 00BCF E0               GOYES   RP3220  "\"
1863 00BD1 3162             LC(2)   SREP    "&"
```

```
Page 039          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1864 00BD5 962              ?A=C    B
1865 00BD8 80               GOYES   RPL370  rplc
1866 00BDA 501              GONC    RPL380  ajoute normal (B.E.T.)
1867 00BDD B44        RP3220 A=A+1  S       état := 3
1868 00BE0 5D8              GONC    RPL390  B.E.T.
1869 00BE3
1870 00BE3 73D1       RPL370 GOSUB  rplc
1871 00BE7 668F       rpl390 GOTO   RPL390
1872 00BEB 148        RPL380 DAT0=A B       ajoute un caractère
1873 00BEE 161               D0=D0+  2
1874 00BF1 55F               GONC    rpl390  B.E.T.
1875 00BF4
1876 00BF4            RPL500
1877 00BF4            •
1878 00BF4            • Après tout ça, il faut actualiser AVMEMS
1879 00BF4            •
1880 00BF4 119               C=R1
1881 00BF7 DA                A=C     A       A(A) := n
1882 00BF9 8F00000           GOSBVL  =CSRC5  C(A) := F(n)
1883 00C00 E2                C=C-A   A
1884 00C02 C6                C=C+C   A       C(A) := 2*(F(n) - n)
1885 00C04 1F00000           D1=(5)  =AVMEMS
1886 00C0B 143               A=DAT1  A
1887 00C0E CA                A=A+C   A
1888 00C10 141               DAT1=A  A       AVMEMS += 2*(F(n) - n)
1889 00C13            •
1890 00C13            • Faut-il poser une question ?
1891 00C13            •
1892 00C13 1F00000           D1=(5)  =QUERY  mode interactif ("?")
1893 00C1A 1574              C=DAT1  S
1894 00C1E 94E               ?C#0    S       Mode interactif ?
1895 00C21 60                GOYES   RPL600  oui : on pose la question
1896 00C23 6ED0              GOTO    RPL800  non : reponse [Y] par défaut
1897 00C27            •
1898 00C27            • Mode interactif :
1899 00C27            •
1900 00C27            • Pour poser la question, il faut qu'il y ait une longueur
1901 00C27            • LIF en début de la chaine. Il faut donc mettre
1902 00C27            • (AVMEMS-OUTBS) en longueur LIF. AVMEMS a été actualisé,
1903 00C27            • entre RPL120 et RPL200.
1904 00C27            •
1905 00C27 AD0        RPL600 A=0     M
1906 00C2A 1B00000           D0=(5)  =AVMEMS
1907 00C31 142               A=DAT0  A
1908 00C34 180               D0=D0-  (=AVMEMS)-(=OUTBS)
1909 00C37 146               C=DAT0  A
1910 00C3A 135               D1=C            D1 := ' début ligne LIF (longueur)
1911 00C3D EA                A=A-C   A       A(A) := longueur en quartets * 4
1912 00C3F 81C               ASRB            A(A) := longueur en octets * 2
1913 00C42 CC                A=A-1   A
```

```
Page 040          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1914 00C44 CC                A=A-1   A
1915 00C46 8F0000            GOSBVL  =SWPBYT
1916 00C4D 1593              DAT1=A  4       longueur LIF
1917 00C51
1918 00C51            •
1919 00C51            • Poser la question, mais sauver d'abord R1 car "query"
1920 00C51            • l'abime.
1921 00C51            • La sauvegarde de R1 est faite en poussant R2 et R3 vers la
1922 00C51            • gauche
1923 00C51            •
1924 00C51 111               A=R1
1925 00C54 11A               C=R2
1926 00C57 D7                D=C     A       D(A) := D0 (cf plus bas)
1927 00C59 8F00000           GOSBVL  =CSLC5
1928 00C60 D6                C=A     A
1929 00C62 10A               R2=C
1930 00C65
1931 00C65 8F00000           GOSBVL  =ASRW5
1932 00C6C 11B               C=R3
1933 00C6F 8F00000           GOSBVL  =CSLC5
1934 00C76 D6                C=A     A
1935 00C78 10B               R3=C
1936 00C7B            •
1937 00C7B            • Mise en place des paramètres de query
1938 00C7B            •
1939 00C7B AF2               C=0     W
1940 00C7E DB                C=D     A       C(W) := ' début occurrence
1941 00C80 142               A=DAT0  A       A(A) := ' OUTBS
1942 00C83 130               D0=A            D0 := ' début ligne
1943 00C86 E2                C=C-A   A       C(A) := delta en nibs = long LIF
1944 00C88 81E               CSRB
1945 00C8B CE                C=C-1   A       C(A) := no colonne (1..max)
1946 00C8D 114               A=R4
1947 00C90 8F0000            GOSBVL  =ASRW5  A(A) := no ligne
1948 00C97            •
1949 00C97            • A(A) := no ligne
1950 00C97            • C(A) := no colonne
1951 00C97            • D0 := ' début ligne (longueur LIF)
1952 00C97            •
1953 00C97 8E0000            GOSUBL  =query
1954 00C9D A87               D=C     P       D(0) := code de retour de query
1955 00CA0            •
1956 00CA0            • Remettre les registres dans l'ordre (restaurer R1)
1957 00CA0            •
1958 00CA0 11B               C=R3
1959 00CA3 DA                A=C     A       A(A) := MAX...
1960 00CA5 8F0000            GOSBVL  =CSRC5
1961 00CAC 10B               R3=C
1962 00CAF 8F0000            GOSBVL  =ASLW5
1963 00CB6
```

```
Page 041          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1964 00CB6 11A              C=R2
1965 00CB9 DA               A=C     A       A(A) := D0
1966 00CBB 8F00000          GOSBVL  =CSRC5
1967 00CC2 10A              R2=C
1968 00CC5 101              R1=A
1969 00CC8            *
1970 00CC8            * Voyons maintenant ce que l'utilisateur a demandé.
1971 00CC8            *
1972 00CC8 A8B              C=D     P       C(0) := code de retour de query
1973 00CCB 8F00000          GOSBVL  =TBLJMC
1974 00CD2 900              REL(3)  RPL700  exit
1975 00CD5 D20              REL(3)  RPL800  yes
1976 00CD8 F10              REL(3)  RPL710  no
1977 00CDB
1978 00CDB            RPL700
1979 00CDB            *
1980 00CDB            * Code de retour = 0 ou 1. C'est exactement la valeur
1981 00CDB            * contenue dans R3(10)
1982 00CDB            *
1983 00CDB 11B              C=R3
1984 00CDE 80FA             CPEX    10      P := code de retour, C(10) := 0
1985 00CE2 80F0             CPEX    0       P := ???, C(0) := code de retour
1986 00CE6 20               P=      0       P := 0
1987 00CE8 90A              ?C=0    P
1988 00CEB A0               GOYES   RPL705  on peut sortir tout de suite
1989 00CED 06               RSTK=C          RSTK := code de retour
1990 00CEF 7DF0             GOSUB   undo    on remet les choses en place
1991 00CF3 07               C=RSTK          RSTK := code de retour
1992 00CF5 03         RPL705 RTNCC
1993 00CF7
1994 00CF7            *
1995 00CF7            * mode interactif et réponse = [N]
1996 00CF7            * Il faut donc remettre les choses en place.
1997 00CF7            *
1998 00CF7 75F0       RPL710 GOSUB   undo    voilà. Les choses sont remises...
1999 00CFB            *
2000 00CFB            * Valeur à ajouter à D0 : +1
2001 00CFB            *
2002 00CFB D2               C=0     A
2003 00CFD E6               C=C+1   A
2004 00CFF 591              GONC    RPL900  B.E.T.
2005 00D02
2006 00D02            *
2007 00D02            * Deux cas :
2008 00D02            *   - mode interactif et réponse = [Y]
2009 00D02            *   - mode non interactif (réponse implicite = [Y])
2010 00D02            * Il y a donc eu modification. R3(10) := 1
2011 00D02            *
2012 00D02 11B        RPL800 C=R3
2013 00D05 2A               P=      10
```

```
Page 042          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

2014 00D07 301              LC(1)   1
2015 00D0A 20               P=      0
2016 00D0C 10B              R3=C            R3(10) := 1
2017 00D0F            *
2018 00D0F            * Mettre dans C(A) le nombre de caractères à ajouter
2019 00D0F            * à D0 pour continuer d'explorer la ligne.
2020 00D0F            *
2021 00D0F 119              C=R1            nb de caractères à ajouter à D0
2022 00D12 8F00000          GOSBVL  =CSRC5  C(A) := f(n) (en caractères)
2023 00D19            *
2024 00D19            * C(A) := taille du bloc que l'on vient de passer c'est à
2025 00D19            * dire la valeur de l'incrément de D0 pour la recherche
2026 00D19            * suivante.
2027 00D19            *
2028 00D19
2029 00D19            RPL900
2030 00D19            *
2031 00D19            * C(A) = taille du bloc que l'on vient de passer c'est à
2032 00D19            * dire la valeur de l'incrément de D0 pour la recherche
2033 00D19            * suivante. Attention : la taille du bloc est en
2034 00D19            * caractères !
2035 00D19            *
2036 00D19            * Rechercher maintenant une nouvelle occurrence
2037 00D19            *
2038 00D19 112              A=R2            A(A) := D0
2039 00D1C CA               A=A+C   A
2040 00D1E CA               A=A+C   A       A(A) := nouveau D0
2041 00D20 130              D0=A            D0 := ^ chaine à chercher
2042 00D23
2043 00D23            *
2044 00D23            * Vérifier si la nouvelle position de D0 est plausible
2045 00D23            * c'est à dire pas derrière AVMEMS)
2046 00D23            * Assertions :
2047 00D23            * A(A) = D0 = nouvelle position de début de recherche
2048 00D23            * Stocker dans R4(15-11) la valeur actuelle de D0 pour
2049 00D23            * comparaison après POSADR, et ceci afin de détecter
2050 00D23            * deux "null match" de suite.
2051 00D23            *
2052 00D23 11C              C=R4
2053 00D26 8F00000          GOSBVL  =CSLC5
2054 00D2A D6               C=A     A
2055 00D2C 8F00000          GOSBVL  =CSRC5
2056 00D30 10C              R4=C
2057 00D33            RPL920
2058 00D33            *
2059 00D33            * Est-on arrivé au bout de la chaine ?
2060 00D33            *
2061 00D33 1F00000          D1=(5)  =AVMEMS
2062 00D40 147              C=DAT1  A       C(A) := AVMEMS
2063 00D43 8B6              ?A>C    A
```

```
Page 043          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

2064 00D46 F5               GOYES   RPL999  oui : Fini !
2065 00D48            *
2066 00D48            * Cas standard : il reste assez de place pour une
2067 00D48            * nouvelle recherche.
2068 00D48            *
2069 00D48 1C4              D1=D1-  5       D1=(5)  =OUTBS
2070 00D4B 147              C=DAT1  A
2071 00D4E 137              CD1EX           C(A) := ^ OUTBS, D1 := OUTBS
2072 00D51 173              D1=D1+  4
2073 00D54 137              CD1EX           A(A) := OUTBS + 4 ; D1=(5) OUTBS
2074 00D57            *
2075 00D57            * 88/10/15 : correction du bug :
2076 00D57            *   GENRPLC$(A$,"","","")
2077 00D57            *   Le vrai début doit être trafiqué, car on ne doit
2078 00D57            *   plus trouver de match en début de chaine. Avant,
2079 00D57            *   on trouvait un match à chaque tour sur l'exemple
2080 00D57            *   ci-dessus.
2081 00D57            *
2082 00D57 CE               C=C-1   A       C'est de l'escroquerie !
2083 00D59 109              R1=C            R1 := ^ vrai début de la chaine
2084 00D5C
2085 00D5C AF2              C=0     W
2086 00D5F 174              D1=D1+  5       D1=(5) =AVMEMS
2087 00D62 147              C=DAT1  A
2088 00D65 E2               C=C-A   A
2089 00D67 81E              CSRB            C(A) := longueur de ce qu'il reste
2090 00D6A DA               A=C     A       A(A) := longueur en caractères
2091 00D6C
2092 00D6C 8E2C7F           GOSUBL  =POSADR
2093 00D72            *
2094 00D72            * Si pas d'occurrence, sortir. La Cy n'a pas été modifiée.
2095 00D72            *
2096 00D72 523              GONC    RPL999  Pas d'occurrence : fini
2097 00D75            *
2098 00D75            * C(A) = LEN (occurrence trouvée)
2099 00D75            * D0 = position de l'occurrence trouvée
2100 00D75            *
2101 00D75 8AE              ?C#0    A
2102 00D78 F1               GOYES   rpl100  ok, on peut remplacer
2103 00D7A 136              CD0EX           D0 := LEN(occurrence), C(A) := D0
2104 00D7D            *
2105 00D7D            * Restauration de l'ancien D0 qui était dans R4(15-11).
2106 00D7D            *
2107 00D7D 114              A=R4
2108 00D80 810              ASLC
2109 00D83 810              ASLC
2110 00D86 810              ASLC
2111 00D89 810              ASLC
2112 00D8C 810              ASLC            A(A) := ancien D0
2113 00D8F 8A2              ?A=C    A
```

```
Page 044          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

2114 00D92 90               GOYES   RPL950  stop ! match nul. 1 partout...
2115 00D94 136              CD0EX           on remet les choses en place
2116 00D97 63AC       rpl100 GOTO   RPL100  et on y va pour le remplacement
2117 00D9B
2118 00D9B            *
2119 00D9B            * Cas spécial : on tombe sur une occurrence de longueur
2120 00D9B            * nulle derrière une occurrence déjà trouvée. On
2121 00D9B            * l'ignore donc.
2122 00D9B            *
2123 00D9B E4         RPL950 A=A+1   A
2124 00D9D E4               A=A+1   A
2125 00D9F 130              D0=A            D0 := A(A) := nouveau D0
2126 00DA2 569              GONC    RPL920  B.E.T.
2127 00DA5
2128 00DA5            RPL999
2129 00DA5            *
2130 00DA5            * La valeur de retour est 2 ou 3. C'est R3(10) + 2
2131 00DA5            *
2132 00DA5 11B              C=R3
2133 00DA8 80FA             CPEX    10      P := code de retour, C(10) := 0
2134 00DAC 80F0             CPEX    0       P := ???, C(0) := code de retour
2135 00DB0 20               P=      0       P := 0
2136 00DB2 B06              C=C+1   P
2137 00DB5 B06              C=C+1   P
2138 00DB8 03               RTNCC           Cy := 0 ; C(0) := 2 ou 3
2139 00DBA
2140 00DBA            ****************************************************
2141 00DBA            * rplc
2142 00DBA            *
2143 00DBA            * But: copie le texte qui a été sauvegardé derrière la
2144 00DBA            *   chaine normale à l'endroit pointé par D0.
2145 00DBA            * Entree:
2146 00DBA            *   - D0 = destination address
2147 00DBA            *   - R1(4-0) = longueur du motif original (en octets)
2148 00DBA            *   - R2(9-5) = source address
2149 00DBA            * Sortie:
2150 00DBA            *   - D0 est réactualisé
2151 00DBA            * Abime: A, B(A), C(A), D(A), D0
2152 00DBA            * Appelle: MOVE*M, ASRW5, CSRC5, CSLC5
2153 00DBA            * Niveaux: 2 (MOVE*M)
2154 00DBA            * Note: D1 est sauvegarde lors de l'appel
2155 00DBA            * Historique:
2156 00DBA            *   88/10/08: PD/JT conception & codage
2157 00DBA            *   88/10/09: PD/JT débogage
2158 00DBA            ****************************************************
2159 00DBA
2160 00DBA 137        rplc  CD1EX
2161 00DBD 8F00000          GOSBVL  =CSLC5  C(9-5) := D1
2162 00DC4            *
2163 00DC4            * Pour MOVE*M
```

```
Page 045          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

2164 00DC4          * A(A) = source address
2165 00DC4          * B(A) = length in nibs
2166 00DC4          * C(A) = destination address
2167 00DC4          *
2168 00DC4 111              A=R1
2169 00DC7 D8               B=A     A
2170 00DC9 C5               B=B+B   A       B(A) := length in nibs
2171 00DCB 112              A=R2
2172 00DCE 8F00000          GOSBVL  =ASRW5  A(A) := source address
2173 00DD5 136              CD0EX           C(A) := destination address
2174 00DD8 8F00000          GOSBVL  =MOVE+M
2175 00DDF C9               C=C+B   A       C(A) := destination + length
2176 00DE1 134              D0=C            D0 est réactualisé
2177 00DE4 8F00000          GOSBVL  =CSRC5  restauration de D1
2178 00DEB 135              D1=C
2179 00DEE 01               RTN
2180 00DF0
2181 00DF0          ****************************************************************
2182 00DF0          * undo
2183 00DF0          *
2184 00DF0          * But: remet le texte comme il était avant la dernière
2185 00DF0          *   modification.
2186 00DF0          * Entree:
2187 00DF0          *   - R1(4-0) = longueur du motif original (en octets)
2188 00DF0          *   - R1(9-5) = longueur du motif remplacé (en octets)
2189 00DF0          *   - R2(4-0) = destination
2190 00DF0          *   - R2(9-5) = source
2191 00DF0          * Sortie:
2192 00DF0          *   - AVMEMS est réactualisé
2193 00DF0          * Abime: A, B(A), C, D(A), D0, D1
2194 00DF0          * Appelle: MOVE+M, CSRC5, ASRW5
2195 00DF0          * Niveaux: 2 (MOVE+M)
2196 00DF0          * Historique:
2197 00DF0          *   88/10/16: PD/JT conception & codage
2198 00DF0          ****************************************************************
2199 00DF0
2200 00DF0 119      undo    C=R1
2201 00DF3 D7               D=C     A       D(A) := n
2202 00DF5 C7               D=D+D   A       D(A) := 2*n
2203 00DF7 8F00000          GOSBVL  =CSRC5  C(A) := f(n)
2204 00DFE C6               C=C+C   A       C(A) := 2*f(n)
2205 00E00 112              A=R2            A(A) := D0
2206 00E03 CA               A=A+C   A       A(A) := D0 + 2*f(n) (source addr)
2207 00E05
2208 00E05 1B00000          D0=(5)  =AVMEMS
2209 00E0C 146              C=DAT0  A       C(A) := AVMEMS
2210 00E0F E2               C=C-A   A       C(A) := AVMEMS - (D0 + 2*f(n))
2211 00E11 D5               B=C     A       B(A) := length of bloc
2212 00E13
2213 00E13 11A              C=R2            C(A) := D0
```

```
Page 046          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

2214 00E16 CB               C=C+D   A       C(A) := D0 + 2*n
2215 00E18          *
2216 00E18          * A(A) = source address = D0 + 2*f(n)
2217 00E18          * B(A) = length of bloc in nibs = AVMEMS - (D0 + 2*f(n))
2218 00E18          * C(A) = dest address = D0 + 2*n
2219 00E18          *
2220 00E18 8F00000          GOSBVL  =MOVE+M
2221 00E1F
2222 00E1F 119              C=R1
2223 00E22 D5               B=C     A       B(A) := n
2224 00E24 C5               B=B+B   A       B(A) := 2*n
2225 00E26
2226 00E26 112              A=R2
2227 00E29 D6               C=A     A       C(A) := D0
2228 00E2B
2229 00E2B 8F00000          GOSBVL  =ASRW5  A(A) := MAX...
2230 00E32          *
2231 00E32          * A(A) = source address = MAX..
2232 00E32          * B(A) = length of bloc in nibs = 2*n
2233 00E32          * C(A) = dest address = D0
2234 00E32          *
2235 00E32 8F00000          GOSBVL  =MOVE+M
2236 00E39
2237 00E39          *
2238 00E39          * AVMEMS += 2*(n - f(n))
2239 00E39          *
2240 00E39 1B00000          D0=(5)  =AVMEMS
2241 00E40 119              C=R1
2242 00E43 DA               A=C     A       A(A) := n
2243 00E45 8F00000          GOSBVL  =CSRC5  C(A) := f(n)
2244 00E4C EA               A=A-C   A       A(A) := n - f(n)
2245 00E4E 146              C=DAT0  A       C(A) := AVMEMS
2246 00E51 C2               C=C+A   A
2247 00E53 C2               C=C+A   A
2248 00E55 144              DAT0=C  A       AVMEMS += 2*(n - f(n))
2249 00E58 01               RTN
2250 00E5A
2251 00E5A                  END
```

```
Page 047          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   **** SYMBOL TABLE ****

=ASLW5      Extrn Ukn       -  1616 1671 1962
=ASRW5      Extrn Ukn       -  1931 1947 2172 2229
=AVMEME     Extrn Ukn       -  0615 0617
=AVMEMS     Extrn Ukn       -  0617 1675 1741 1885 1906 1908 2061 2208 2240
 ANY        00002 Abs  0015 -  0196 0517 0565
 ANYS       00001 Abs  0026 -  0193 0514 0559 1315
 BOL        00006 Abs  0019 -  0213 0337 0540 0850
 BOLS       00001 Abs  0024 -  0210 0537
=CALCab     00924 Rel  1441 -
=COMP71     002AB Rel  0438 -
=COMPUX     00000 Rel  0112 -
=CSLC3      Extrn Ukn       -  1391
=CSLC5      Extrn Ukn       -  0826 1683 1927 1933 2053 2161
=CSLC8      Extrn Ukn       -  1167 1198
=CSRC3      Extrn Ukn       -  0812 1388
=CSRC5      Extrn Ukn       -  0822 0840 0906 0974 1443 1709 1727 1746 1773
                            +  1882 1960 1966 2022 2055 2177 2203 2243
=CSRC8      Extrn Ukn       -  1177
 CAL610     009DA Rel  1524 -  1518
 CAL615     009E0 Rel  1526 -  1521 1536
 CAL620     009FA Rel  1537 -  1532
 CAL630     009B5 Rel  1509 -  1506
 CALC10     0094C Rel  1458 -  1474 1476 1480
 CALC20     00975 Rel  1475 -  1466
 CALC25     0097B Rel  1478 -  1462
 CALC30     0097E Rel  1479 -  1469
 CALC50     00988 Rel  1487 -  1454 1515 1529 1538 1541
 CALC60     009BB Rel  1512 -  1500
 CALC61     009C6 Rel  1517 -  1502
 CALC62     009E9 Rel  1531 -  1504
 CALC63     009AA Rel  1505 -
 CALC70     00A00 Rel  1540 -  1508 1510 1513 1523 1525 1535
 CALC99     00A06 Rel  1547 -  1482 1488
 CCL        00003 Abs  0016 -  0244 1280
 CCLS       00101 Abs  0027 -  0239 0261 0294 1316
 CHAR       00000 Abs  0013 -  0734
 CHARS      00003 Abs  0029 -  0731 1309 1313
 CLOS       00007 Abs  0020 -  0340 0361 0562 1067
 CLOSS      00001 Abs  0030 -  0559 1078 1320
 Cmpend     002E0 Rel  0468 -  0458
=ENDPOS     0091A Rel  1410 -  0372 0741
 END        00005 Abs  0018 -  0165 1064
 EOL        00001 Abs  0014 -  0231 0555
 EOLS       00001 Abs  0025 -  0228 0552 1314
 Errmem     00A1C Rel  1608 -  1657 1662
=FINDA      Extrn Ukn       -  0133 0494
 FiLeNd     00E5A Rel  2251 -
=I/OALL     Extrn Ukn       -  0635
=I/OCON     Extrn Ukn       -  0171
=I/ODAL     Extrn Ukn       -  1411
```

```
Page 048          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   **** SYMBOL TABLE ****

=IOFND0     Extrn Ukn       -  1381
=IOFSCR     Extrn Ukn       -  0603
 InvPat     0008D Rel  0190 -  0184
 Invpat     00121 Rel  0236 -  0243 0267 0275
=LEEWAY     Extrn Ukn       -  0621
=MEMCKL     Extrn Ukn       -  1712
=MODE71     Extrn Ukn       -  1444 1768
=MOVE+M     Extrn Ukn       -  1728 1759 2174 2220 2235
=MOVED2     Extrn Ukn       -  0357
=MPY        Extrn Ukn       -  1653
 NCCL       00004 Abs  0017 -  0249
 NCCLS      00101 Abs  0028 -  1317
 NoRoom     003CC Rel  0544 -  0516 0539 0554 0561
 Nroom      00EE5 Rel  0218 -  0195 0212 0230 0241
=OUTBS      Extrn Ukn       -  1908
=POSADR     00534 Rel  0811 -  2092
 Pos20      006BB Rel  1004 -  1001
 Poschr     00592 Rel  0855 -  0844
=QUERY      Extrn Ukn       -  1892
=REPLIN     00A24 Rel  1611 -
 RP3210     00BC0 Rel  1856 -  1850
 RP3215     00BC6 Rel  1858 -  1853
 RP3220     00BDD Rel  1867 -  1862
 RP3230     00B9B Rel  1841 -  1838
 RPL100     00A3B Rel  1633 -  2116
 RPL120     00A94 Rel  1683 -  1681
 RPL200     00B27 Rel  1781 -
 RPL210     00B27 Rel  1789 -  1811
 RPL220     00B4A Rel  1801 -  1796
 RPL240     00B50 Rel  1804 -  1793
 RPL270     00B56 Rel  1807 -  1799
 RPL280     00B5E Rel  1809 -  1800 1805
 RPL290     00B27 Rel  1784 -  1802 1808
 RPL300     00B6B Rel  1818 -  1777
 RPL310     00B6E Rel  1825 -
 RPL320     00BA1 Rel  1844 -  1831
 RPL321     00BAC Rel  1849 -  1833
 RPL322     00BCC Rel  1861 -  1835
 RPL323     00B90 Rel  1837 -
 RPL370     00BE3 Rel  1870 -  1859 1865
 RPL380     00BEB Rel  1872 -  1840 1842 1845 1855 1857 1866
 RPL390     00B6E Rel  1820 -  1847 1863 1871
 RPL500     00BF4 Rel  1876 -  1813 1826
 RPL600     00C27 Rel  1905 -  1895
 RPL700     00CDB Rel  1978 -  1974
 RPL705     00CF5 Rel  1992 -  1988
 RPL710     00CF7 Rel  1998 -  1976
 RPL800     00D02 Rel  2012 -  1896 1975
 RPL900     00D19 Rel  2029 -  2004
 RPL920     00D39 Rel  2057 -  2126
```