JPC

MAI 1987

NUMERO 44

Le numéro 40 F

A PROPOS DU CLUB

P. David — Editorial — *1*

PPC Paris se réunit — *2*
Ah ! Vous écrivez ! — *2*
Courrier du coeur — *3*
Courrier des lecteurs — *4*

HP28

P. David — Des courbes et des paramètres — *6*

HP41

E. Gengoux — Tarification SNCF (Acte II) — *8*

HP75

P. Picheret — Graphiques sous VisiCalc — *12*

HP71

J. Baudier — Etats d'âme — *18*

F. Stengel — Les objets en Forth — *18*

J.M. Kefaloucos — Horloge Astronomique — *24*
P. Ehrenberg & V. Klann — Dérivées symboliques — *25*
S. Tendon — Graphes de Connett — *28*

Le coin des Lhex — *44*

EDITORIAL

Chers amis,

J'ai été content de vos réactions à propos de FINPUT, le Lex pour HP-71 paru dans notre dernier JPC. Vous avez été nombreux à manifester votre intérêt pour notre travail. Je vous en remercie.

Nous sommes heureux de vous annoncer le changement d'adresse du Club. Désormais, la seule et unique adresse pour tout votre courrier est :

PPC Paris Chapter
B.P. 604
75028 Paris Cedex 01

L'opération Eprom est un relatif succès. Douze ont été commandées. Quand vous lirez ces lignes, douze d'entre vous seront en possession de leur nouvel module. Et notre travail sera enfin terminé ! Cette opération nous aura conduit à remanier un bon nombre de Lex. Les mises à jour paraîtront dans les JPC à venir, ainsi que les nouvelles fonctions non encore publiées.

Enfin, et ce n'est pas le moindre, nous nous voyons obligés d'augmenter le prix au numéro ainsi que la cotisation annuelle. Celle-ci passe à 350 Francs, 300 pour les étudiants. Nous n'avons pas pris cette décision de gaieté de coeur, mais je suis sûr que vous la comprendrez. La dernière augmentation date de février 1985, c'est à dire il y a plus de 2 ans. Nous n'avons pas pu reculer davantage cette décision.

Ce mois-ci, vous remarquerez la signature de Stefano Tendon, d'Italie, et de Peter Ehrenberg et Volker Klann, d'Allemagne. Qu'ils soient remerciés pour leurs contributions à ce Journal.

Bonne lecture, et à bientôt

Pierre David (37)

PPC PARIS SE REUNIT UNE FOIS PAR MOIS

Comme vous le savez peut être déjà, PPC Paris se réunit une fois par mois, en plein coeur de Paris. Amenez votre matériel, votre bonne volonté et vos idées ! Plus vous en apporterez, et plus vous en trouverez chez vos collègues de PPC.

Ces réunions se déroulent de manière très libre, aucun ordre du jour, discussion ou autre n'étant imposé. Un membre du bureau est toujours présent. Ainsi, si vous désirez remettre votre article tout frais au Journal, si vous avez des suggestions à faire, si vous voulez vous procurer des anciens numéros de JPC, ce sera en principe toujours possible.

Si donc cela vous intéresse, n'hésitez plus un seul instant, venez nous rejoindre tous les premiers samedis de chaque mois (sauf en période de vacances scolaires) au :
Centre de Jeunesse et de Loisirs Jean Verdier
11 rue de Lancry
75010 Paris

et en montant au deuxième étage, vous entendrez des éclats de rire et des discussions passionnées vers la salle 215. Attention, toutefois, de venir entre 16 et 19h.

Pour l'accès en métro, trois possibilités s'offrent à vous :
- Métro Strasbourg Saint Denis :
Sortie porte St Martin / Bd St Denis, coté pairs
- Métro République :
Sortie Bd St Martin, coté pairs
- Métro Jacques Bonsergent :
Sortie Bd Magenta, coté impairs.

Ah, j'oubliais ! JPC est (souvent) distribué en avant première lors de ces réunions... A bon entendeur, salut !

Les dates des prochaines réunions sont :
Samedi 16 mai 1987
Samedi 6 juin 1987

Pierre David (37)

AH ! VOUS ECRIVEZ

Vous vous sentez en verve, mais vous ne savez pas sous quelle forme "l'équipe de rédaction" souhaite recevoir votre prose. C'est ici que se trouvent les réponses à vos questions.

Dans la mesure du possible, vous devez nous envoyer vos écrits sur support magnétique (carte, cassette ou disquette). Soyez sans crainte, nous vous retournerons vos biens après copie.

Si vous ne pouvez pas utiliser de support magnétique, ou ne pouvez vous rendre aux réunions, alors et alors seulement faites le sur papier.

Que ce soit sur une feuille de papier, ou sur support magnétique, ne dépassez pas 50 caractères par ligne.

Pour nous épargner du travail, insérez dans votre texte les commandes de formattage suivantes (et non les commandes du formatteur HP) :

"^" centre un titre, par exemple :
^TITRE

"\" (CHR$(92)) marque le début et la fin d'un paragraphe. Par exemple :

\Début de paragraphe exprimant le contenu de vos idées qui, même si vous en doutez, intéressera certains des membres du Club. Surtout si vous vous sentez débutant. Les articles pour débutants écrits par des débutants sont ceux qui manquent le plus. Fin de paragraphe.\

N'oubliez pas de mettre les accents. Utilisez le jeu de caractères Roman8. Les possesseurs de HP71 utiliseront les redéfinitions de touches ci-dessous, ainsi que le fichier CHARLEX listé dans le coin des Lhex en fin de journal.

Jean-Jacques Dhénin (177)

```
DEF KEY 'fW', CHR$(197);      (é)
DEF KEY 'fE', CHR$(193);      (ê)
DEF KEY 'fR', CHR$(201);      (è)
DEF KEY 'fY', CHR$(203);      (ù)
```

```
DEF KEY 'fU', CHR$(195);      (û)
DEF KEY 'fI', CHR$(209);      (î)
DEF KEY 'fO', CHR$(194);      (ô)
DEF KEY 'f/', CHR$(92);       (\)
DEF KEY 'fA', CHR$(192);      (â)
DEF KEY 'fS', CHR$(200);      (à)
DEF KEY 'fD', CHR$(205);      (ë)
DEF KEY 'fJ', CHR$(207);      (ü)
DEF KEY 'fK', CHR$(221);      (ï)
DEF KEY 'f*', CHR$(124);      (|)
DEF KEY 'fC', CHR$(181);      (ç)
```

PPC Paris
BP 604
75028 Paris Cedex 01

Vend :
Interfaces vidéo HP82163B neuves (dans leur boîte, avec documentation) : 600 F seulement.

Des lots de 100 cartes magnétiques pour HP-41, HP-67, HP-97 et même HP-65 : 100 F seulement.

Des lecteurs de cartes magnétiques HP82400A pour HP-71 : 500 F pièce.

Pierre David
33 Bd St Martin
75003 Paris
Tél : (1) 48 87 68 93

Vend :
Imprimante 80 colonnes HP82905B + papier : 2500 F.
Convertisseur HP82166A : 1000 F.

Pierre Picheret
Calmon Aiguefonde
81200 Mazamet

Vend :
un HP150B (256Ko) avec HP-Link, Pascal 2.0, Basic, Forth 83 et Programmer's Toolkit : le tout 9000 F à débattre.

Georges Julem
118 rue d'Issy
92100 Boulogne

Recherche tous programmes d'électronique, en français de préférence, pour HP-67 / 97 ou HP-15.

Jean-Pierre Perez
46 60 91 89

Vend 2 modules 4 K octets pour HP-71.

Alain Gillet
Tél : 43-67-54-44 (après 18h)

Vend :
Imprimante 82161A, plusieurs rouleaux de papiers : 2000 F, Lecteur de code-barre: 500 F. Livres : Programmation synthétique (Dr. Wickes) : 60 F, La programmation synthétique c'est facile (K. Jarret) : 60F, Time module solution 1 (HP) : 60 F, Au fond de la 41 (Dodin) : 60F.

Eric Gengoux
8 rue de Furstenberg
75006 Paris

Vend :
Lecteur de cassettes HP82162 : 2000 F.

Jean-Louis Paré
(1) 47 83 56 08

Recherche manuel d'emploi pour HP-80.

Laurent Eymard
22 rue Lechantre
02100 Saint Quentin

Recherche module Maths pour HP-41 et des contacts dans la région de Rouen.

COURRIER DES LECTEURS

Rappelons la nouvelle adresse de PPC Paris :

PPC Paris Chapter
B.P. 604
75028 Paris Cedex 01

Guy Toublanc
21 rue Henri Becquerel
59120 Loos

APPEL A LA RESISTANCE

Non, ce n'est pas une incitation guerrière, mais un cri de ralliement que je lance ! En effet, la résistance des matériaux, le calcul des structures et du béton armé sont pratiquement ignorés dans JPC (un seul article pour HP-41 de Philippe Davase dans JPC 19).

Si la RPN a été favorisée avec les programmes HP et de la SOCOTEC qui sont une référence, il reste encore bien des chapitres inexplorés. Pour le BASIC, il existe des ouvrages dont les programmes sont adaptables aux HP-75 et HP-71, mais ils sont :

- rédigés dans un basic très rudimentaire,

- limités par leurs possibilités, abusant de menus conçus plutôt pour une secrétaire ignorant tout de la résistance des matériaux, d'où un énorme gaspillage de mémoire,

- peu fiables, certains donnent des résultats FAUX. L'absence de référence à des formules ou à des règlements ne permet ni de les contrôler ni de les actualiser.

Donc il y a beaucoup à faire, d'autant que l'usage des abaques et tables multiples est d'un âge révolu : l'informatique est une nécessité première en résistance des matériaux, calcul des stuctures et de béton armé.

Il n'est pas question de rivaliser avec les logiciels professionnels que nous ne pouvons pas utiliser. Je vous propose donc de faire le bilan de :

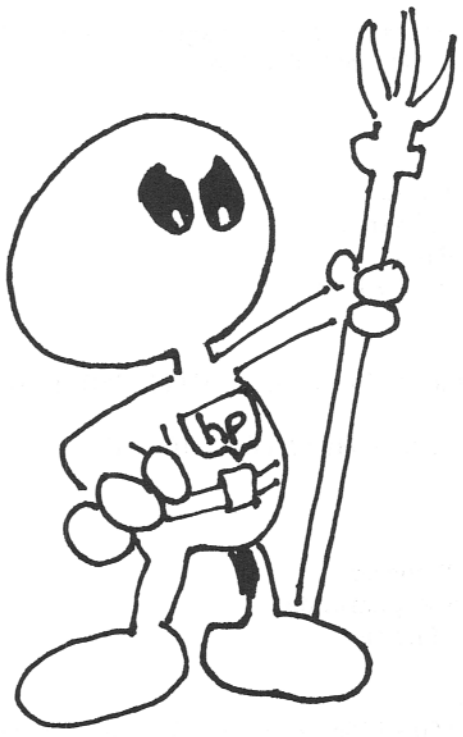
- ce que vous possédez

- ce que vous voudriez

- ce que vous pouvez ou voudriez publier et pour quelle machine.

Un regroupement serait peut-être souhaitable lorsqu'un travail est trop important pour un seul. Si vous avez des propositions à faire, une méthode à suggérer, alors contactez moi.

Guy Toublanc (276)



HP-28C

P. David

Des courbes et des paramètres 6

DES COURBES ET DES PARAMETRES

Si d'aventure vous êtes venus trainer vos guêtres sur les moquettes du dernier Sicob, vous avez peut-être visité le stand Hewlett-Packard.

Attiré par l'attroupement, vous vous êtes certainement approché. Et devant vos yeux ébahis, un démonstrateur expliquait le fonctionnement du HP-28C.

Et en l'écoutant en détail, vous l'avez entendu dire : «tracer des courbes paramètrées, ce n'est pas bien dur. Je suis certain que le Club PPC va bientôt publier un petit programme pour y arriver...».

Ce brave démonstrateur ne croyait pas si bien dire.

Le petit programme que je vous présente aujourd'hui peut rendre quelques services. Aussi me suis-je décidé à le publier. Le but est de tracer des courbes paramètrées, c'est à dire de la forme :
y = f (t)
x = f (t)

Pour y arriver, il faut d'abord rentrer ceci :

```
« -> y          l'expression y = f(t)
     x          l'expression x = f(t)
     m1         t minimum
     m2         t maximum
   « CLLCD      efface l'écran
     DRAX       trace les axes si besoin
     m2 m1 -    calcul de l'incrément
     100 /
     m1         t minimum
     m2         t maximum
     FOR t
       t        t courant
       'T' STO  stocké dans la variable globale T
       x EVAL   évaluation de x = f (t)
       y EVAL   évaluation de y = f (t)
       R->C     transformé en complexe pour PIXEL
       PIXEL    allumage du point
       DUP      duplication de l'incrément
     STEP       absorbption de l'incrément
     DROP       destruction du dernier incrément
   »
»
```

Puis stockez-le dans une variable :

```
'PARAM [STO]
```

Pour l'utiliser, vous devrez au préalable remplir la pile avec, au :

niveau 4 : l'expression y = f(t) ('symbole')
niveau 3 : l'expression x = f(t) ('symbole')
niveau 2 : t initial (nombre réel)
niveau 1 : t final (nombre réel)

Note : les expressions doivent être des fonctions de la variable globale T. Je sais, ça n'est pas aussi propre que la fonction intégrée DRAW. Mais essayez donc de tracer des courbes paramètrées avec DRAW !

Essayons avec un petit exemple : une ellipse.
x = 6cos(t)
y = sin(t)

Effacez la variable T, passez en mode radians et annulez les paramètres de tracé :

```
'T [PURGE]
[MODE] [RAD]
[PLOT] [NEXT] 'PPAR [PURGE]
```

Et allons-y :

```
'SIN(T     [ENTER]
'6*COS(T   [ENTER]
0          [ENTER]
6.28       [ENTER]
[USER] [PARAM]
```

Et votre ellipse se trace petit à petit... N'est-ce pas beau ?

Après l'exécution de ce programme, nous pouvons voir que la variable PPAR a été créée. Ceci est le résultat de l'exécution des fonctions graphiques. PPAR est nécessaire à leur exécution, donc elles le créent. Quoi de plus simple ?

Notons, pour finir, un petit «truc» bien utile si vos fonctions à tracer ont la mauvaise habitude de diviser par 0, ou de prendre des tg(90). Armez l'indicateur binaire 59, cela masquera les erreurs Infinite Result.

Pierre David (37)

HP-41

E. Gengoux — Tarification SNCF (Acte II) — *8*

ENCORE UN BILLET DOUX

L'été dernier paraissaient dans votre Journal préféré quatre programmes pour HP-41 (et un sous programme équivalent pour le 71, afin de ne pas exciter de jalousies...) qui vous disaient tout ou presque sur le calcul du prix des billets de chemin de fer, et montraient - enfin, je l'espère - qu'il n'est point besoin de manger des nouilles tous les jours pour pouvoir partir (j'entends le Rédac'Chef qui rigole...).

La hausse des tarifs du 30 avril m'amène à publier la mise à jour que voici...

Qu'est-ce qui a changé dans les «règles du jeu» ? Les prix de base, bien sûr, mais aussi un petit détail, qui nous a obligé à reprendre le programme "T2" : le «minimum de perception» en 1ère classe n'est plus exactement égal à 1,5 fois celui de la 2ème. D'où apparition de deux nouveaux paramètres en R11 et R12, disparition de R10 et adaptation des lignes 65 et suivantes.

Comme les principes exposés dans le précédent article n'ont pas changé, non plus que le mode de calcul de la «distance palière» ou l'existence de trois tranches selon la distance (à ce propos, la comparaison des ancienne et nouvelle formules montre qu'au delà de 595 km, ça coûte même moins cher. Si, si, c'est vrai !), il n'est pas utile de vous infliger de nouvel exposé. Reportez-vous à JPC 36 (juillet août 86) pour les détails et bon(s) voyage(s) quand même.

Eric Gengoux (108)

```
01*LBL "SNCF" ! Programme principal (menus)
02 "PRIX BILLET"
03 AVIEW
04 STOP      ! Faire [R/S]
05 "DIST="
06 PROMPT    ! distance en km
07 STO 15
08 XEQ "D2" ! calcul dist. "palière"
09 "CLASSE="
10 PROMPT    ! 1 ou 2
11 STO 14
12 "REDU %"
13 PROMPT    ! 0 si plein tarif
14 STO 00
15 "AD=1/EN=2" ! ADulte (>= 12 ans)
16 PROMPT    ! ENfant si <12 ans
17 STO 12
18 "NBRE="
19 PROMPT
20 STO 13
21 "SP=1/AR=2" ! SP: aller simple...
22 PROMPT
23 STO 11
24 XEQ "T2" ! calcul prix total
25 END
```

```
01*LBL "PBASE" ! Charge R01 à R12 avec les bons prix...
02 "PBASE 5/87"
03 AVIEW
04 0.02001
05 CLRGX
06 10.00     ! T.C. 2ème cl. "1ère tranche" (<299)
07 STO 01
08 .445      ! Taux 2ème cl. "1ère tranche"
09 STO 02
10 299       ! limite sup. "1ère tranche"
11 STO 03
12 20.50     ! T.C. 2ème cl. "2ème tranche"
13 STO 04
14 .41       ! Taux 2ème cl. "2ème tranche"
15 STO 05
16 349       ! limite sup. "2ème tranche"
17 STO 06
18 38.00     ! T.C. 2ème cl. "dernière tranche"
19 STO 07
20 .36       ! Taux 2ème cl. "dernière tranche"
21 STO 08
22 .60       ! Plafonnement 60% T.C. courtes dist.
23 STO 09
24 48        ! Mini de perception 2ème (4.80 F)
25 STO 11
26 70        ! Mini de perception 1ère (7.00 F)
27 STO 12
28 "PBASE OK" ! Maintenant, [XEQ] [SNCF]...
29 AVIEW
30 BEEP
31 END
```

```
01*LBL "D2" ! Calcul distance tarifaire
02 RCL 15    !(i.e. "médiane" du "palier")
03 ENTER^
04 6
05 X>Y?
06 X<> 15
07 RCL 15
08 ENTER^
09 50
10 X<=Y?
11 GTO 04
```

```
12 RCL 15
13 ENTER^
14 2
15 /
16 FRC
17 X=0?
18 GTO 05
19 .5
20 ST- 15
21 GTO 03
22*LBL 05
23 .5
24 ST+ 15
25 GTO 03
26*LBL 04
27 RCL 15
28 ENTER^
29 200
30 X<=Y?
31 GTO 06
32 RCL 15
33 ENTER^
34 10
35 /
36 STO 08
37 INT
38 STO 09
39 ST- 08
40 10
41 ST* 09
42 RCL 08
43 ENTER^
44 .5
45 X<=Y?
46 GTO 07
47 RCL 09
48 ENTER^
49 2
50 +
51 STO 15
52 GTO 03
53*LBL 07
54 RCL 09
55 ENTER^
56 7
57 +
58 STO 15
59 GTO 03
60*LBL 06
61 RCL 15
62 ENTER^
63 10
64 /
65 INT
66 10
67 *
68 4
69 +
70 STO 15
71*LBL 03
72 RTN
73 END
```

```
01*LBL "T2" ! Calcul du prix
02 "MAI87"  ! dernière mise à jour
03 RCL 06   ! Choix formule selon distance
04 RCL 15
05 X>Y?
06 GTO 03
07 RCL 03
08 RCL 15
09 X>Y?
10 GTO 02
11 GTO 01
12*LBL 03   ! "Dernière tranche", R15>R06
13 RCL 07
14 RCL 08
15 RCL 15
16 *
17 +
18 STO 16
19 GTO 05
20*LBL 02   ! "2ème tranche", R03<R15<=R06
21 RCL 04
22 RCL 05
23 RCL 15
24 *
25 +
26 STO 16
27 GTO 05
28*LBL 01   ! "1ère tranche", R15<=R03
29 RCL 02   ! Formule avec "plafonnement T.C."
30 RCL 15   ! si TC>=R09*(TK*d) )
31 *
32 STO 16
33 RCL 09
34 *
35 RCL 01
36 X>Y?
37 GTO 04   ! Plafonnement...
38 RCL 01
39 ST+ 16
40 GTO 05
41*LBL 04   ! Le voilà...
42 RCL 16
43 RCL 09
44 *
45 ST+ 16
46*LBL 05   ! Application réduction
47 RCL 16
48 ENTER^
49 100
50 ENTER^
51 RCL 00
52 -
```

```
 53 %
 54 RCL 12
 55 /
 56 STO 16
 57*LBL c    ! Prix*1.5 si 1ère classe
 58 XEQ a    ! arrondi tarifaire
 59 RCL 14
 60 2
 61 X=Y?
 62 GTO b
 63 1.5
 64 ST* 16
 65*LBL b    ! SP/AR et minimum de perception
 66 RCL 11
 67 ST* 16
 68 1
 69 RCL 14
 70 X=Y?
 71 GTO 11
 72 GTO 12
 73*LBL 11   ! Minimum de perception 1ère cl.
 74 RCL 16
 75 RCL 10
 76 X>Y?
 77 STO 16
 78 GTO 13
 79*LBL 12   ! Minimum de perception 2ème cl.
 80 RCL 16
 81 RCL 20
 82 X>Y?
 83 STO 16
 84*LBL 13   ! Fin et affichage
 85 RCL 13
 86 ST* 16
 87 RCL 16
 88 RTN
 89*LBL a    ! Arrondi tarifaire
 90 .00001   ! seuil de déclenchement
 91 ST+ 16   ! Prix à arr. pris -> R16
 92 RCL 16   ! tranche aux 20 cmes. supérieurs
 93 20
 94 X>Y?
 95 GTO 07
 96 RCL 16   ! tranche aux 50 cmes. supérieurs
 97 30
 98 X>Y?
 99 GTO 08
100 RCL 16   ! tranche au Franc supérieur
101 1
102 +
103 INT
104 STO 16
105 RTN
```

HP-75

P. Picheret	Graphiques sous VisiCalc	*12*
P. Picheret	Exemples de graphiques	*15*
P. Picheret	Programme "VISICM2"	*30*

GRAPHIQUES SOUS VISICALC

Cette discussion a pour but un des éléments essentiels du HP75, outil dit vénérable par certains, mais qui mérite vraiment une étude particulière. Il s'agit de VisiCalc et spécialement des "extensions de commande" de ce logiciel.

Nous examinerons ensemble une de ces extensions possibles : il s'agit d'un module graphique destinée à la table traçante HP7470. Cependant, il me semble aisé de paramètrer ce programme de manière à le rendre portable sur d'autres périphériques graphiques. Nous ne nous servirons que des commandes de base du langage HPGL. Le résultat se trouve dans le programme VISICM2.

Nous nous servirons de la feuille BUDGET tout au long de cette discussion. C'est la feuille de démonstration livrée avec le logiciel.

BUDGET	A	B	C	D	E	F
	Jan	Feb	Mar	Apr	May	Jun
1 Income	2000	2000	2000	2000	2000	2000
2						
3 House	400	400	400	400	400	400
4 Insur	100	100	100	100	100	100
5 Taxes	600	600	600	600	600	600
6 Food	200	175	180	155	195	140
7 Util	110	95	87	81	68	55
8						
9 TotExp	1410	1370	1367	1336	1363	1295
10 Remain	590	630	633	664	637	705

Pour fonctionner, VISICM2 doit trouver en mémoire vive le fichier TEXT VISIMSGS. Celui-ci contient les libellés des messages utilisés pour dialoguer avec l'utilisateur. Ce fichier devra contenir les lignes suivantes :

```
 22 DFGHIMPRVW-ET
 80 T.graph
 81 Empil Jumel Rond Cumul
100 Choix
101 1 Vert Horz 4 Suite
```

LES EXTENSIONS DE COMMANDE

Lorsque VisiCalc fonctionne, on accède à une série de commandes en appuyant sur la touche [/].

La ligne de commande de VisiCalc correspond à la ligne 22 du fichier TEXT VISIMSGS qui est listée plus haut. Par défaut, elle vaut :

```
DFGHIMPRVW-
```

Ce sont les commandes dont on se sert depuis le niveau supérieur de VisiCalc. Les caractères apparaissant après le "-" correspondent aux mnémoniques des extensions de commandes disponibles. La première lettre correspond à la commande contenue dans le fichier VISICM1, la seconde à la commande contenue dans VISICM2, et ainsi de suite. L'extension de commande est donc activée en appuyant sur la touche correspondant au mnémonique choisi, pourvu que le programme Basic correspondant soit en mémoire. Le *E* figurant dans la ligne 22 du fichier VISIMSGS décrit plus haut correspond au programme VISICM1 livré avec le produit. Le *T* comme Tracé correspond au programme VISICM2 qui nous occupe.

L'écriture d'un programme Basic correspondant à une extension de commandes est un peu plus délicate que celle d'un programme ordinaire. Il est impératif de se référer au *Manuel de Référence du Programmeur VisiCalc* (ce manuel, livré en anglais avec le module existe en traduction française à demander à HP). Il décrit succintement la marche à suivre pour écrire votre Basic, ainsi que les quelques 80 mots clefs auxquels il est fait largement appel puisqu'il sont faits pour interagir avec la feuille de travail.

DESCRIPTION DES GRAPHIQUES

Les graphiques peuvent être présentés de quatre manières différentes sur la feuille : un graphique horizontal, deux graphiques superposés, deux juxtaposés et enfin quatre simultanément.

Quels types de graphiques peuvent être réalisés à partir de VISICM2 ?

Le type circulaire que les anglais appellent *pie*. Une restriction cependant : pour que le graphique reste bien circulaire, on ne peut utiliser que la première ou la quatrième des représentations décrites plus haut.

Les histogrammes dit *graphiques en barres* ou *en cheminées*. Les histogrammes peuvent être simples, juxtaposés ou cumulés. Ils sont utilisables dans n'importe quel type de présentation. Toutefois, suivant le nombre de valeurs de la variable, il faudra choisir judicieusement le type de présentation.

Dans les exemples, nous trouvons un graphique circulaire où nous voyons la représentation du mois de Janvier (colonne A) de la feuille BUDGET pour 5 valeurs [House], [Insur], [Taxes], [Food] et [Util] (lignes 3, 4, 5, 6 et 7). Les valeurs numériques prises en considération sont situées à l'intersection de la colonne et des lignes. La somme des valeurs représente 100 %, soit le cercle entier, et chacune d'elles est représentée proportionellement à sa contribution. La variable représentée, c'est à dire une colonne, et les valeurs correspondantes (les lignes) peuvent être choisies nominativement. Notons qu'il ne peut y avoir plus de 6 valeurs représentées pour une variable.

Dans un histogramme simple, nous pouvons tracer le graphique sur une étendue quelconque de colonnes (ici de [Jan] à [Jun]) en commencant, si nous le désirons, à la n^{ième} de ces colonnes. Nous pouvons aussi désigner la ligne sur laquelle portera l'étude (ici [Remain]). L'échelle sera calculée de manière complètement automatique.

Notons que la variable étudiée est une ligne de la feuille de travail et les valeurs de la variable correspondent aux intersections avec les colonnes dont l'étendue a été définie globalement. Dans les histogrammes jumelés ou cumulés il suffira de désigner les différentes lignes dans l'ordre souhaité.

Enfin, pour un camembert (Humm, un graphique circulaire !), il n'est pas possible de considérer une ligne comme étant une variable dans cette extension de commande. De même, dans un histogramme, nous ne pouvons pas considérer une colonne comme variable. S'il vous est utile de tracer ces sortes de graphiques, au demeurant aussi valables que ceux fournis par ce programme, il faudrait écrire une autre extension de commande qui permuterait lignes et colonnes de votre tableau ou une partie de celui-ci. Vous obtiendrez ainsi un graphique circulaire représentant, par exemple, les différentes valeurs de [Income].

MODE D'EMPLOI

Rappelons qu'il est obligatoire d'avoir en mémoire le programme VISICM2 et le fichier texte VISIMSGS dont la ligne 22 a été modifiée et où on a ajouté les lignes 80, 81 et 100, 101.

Vous décidez de tracer le graphique circulaire donné en exemple. L'extension de commande va être employée. Vous êtes donc dans la feuille VisiCalc BUDGET, votre curseur est positionné en A1 vous appuyez sur `[/]` ce qui vous donne accès a la ligne de commande :

```
DFGHIMPRVW-ET
```

vous pressez alors `[T]` et vous verrez apparaître :

```
Choix : 1 Vert Horz 4 Suite
```

vous choisissez alors votre présentation. Tapez `1` pour avoir le diagramme sur toute la page. Les autres possibilités sont *V* pour une présentation verticale (superposée), *H* pour une présentation horizontale (juxtaposée), *4* pour quatre diagrammes simultanés. *S* est utilisé pour enchaîner le tracé des diagrammes successifs. Le programme vous demande ensuite quel type de graphique vous désirez tracer en affichant au LCD :

```
Empil Jumel Rond Cumul
```

Vous pressez `[R]`. Puis le programme vous demandera quelle est votre variable en affichant : `First col [Jan]`.

Par la suite, vous entrerez les coordonnées qui vous seront demandées en format système (A1) ou utilisateur ([Jan^Income]) en fonction de celui que vous avez spécifié.

Par défaut, le programme vous propose la colonne courante. Vous pouvez accepter en pressant `[RTN]`. Mais pouvez demander une autre colonne en tapant `[B]` ou [Fev] suivi de `[RTN]`, par exemple. Puis, pour la 1ère valeur la ligne courante : `Select row [Income]`. On peut accepter par `[RTN]` ou modifier. Ensuite, les valeurs suivantes vous sont réclamées par le programme qui affiche `Select row B`. Ce *B* n'est pas une coordonnée valable pour les lignes. Il faut répondre par une coordonnée de ligne. Par exemple 3 ou [House]. Tant que nous répondrons à cette demande en fournissant

une coordonnée de ligne, notre réponse sera acceptée. Pour arrêter la saisie, répondez simplement par [RTN] (vous ne pouvez spécifier plus de 6 lignes). Le tracé commence immédiatement. Vous n'aurez de nouveau la main qu'à la fin du graphe.

Les autres types de graphiques répondent au même schéma. Par exemple, si vous souhaitez tracer un histogramme simple dans la partie supérieure de la feuille et un histogramme jumelé dans la partie inférieure vous taperez :

```
[/]      accès aux commandes
[T]      pour l'extension tracé
[H]      choix de présentation horizontale
[J]      pour jumellé
```

Le programme affiche :

```
First col: [Jan]
```

Validez ou modifiez. Puis :

```
Last col: [Jun]
```

Validez ou modifiez. La 1ère colonne proposée est la colonne courante. La seconde est la dernière colonne du tableau (correspondant à MAXCOL). Il n'y a plus qu'à définir les données observées, en répondant par :

```
[RTN]  Pour [Income]
3 ou [House] et [RTN]
4 ou [Insur] et [RTN]
     ......
7 ou [Util]  et [RTN]
```

Quand le programme vous rend la main, vous pouvez taper [/][T][S][J] le *S* correspond à Suite, puis :

```
[RTN] pour [Jan]
[RTN] pour [Jun]
10 pour [Remain]
[RTN]
```

ce qui termine l'interrogation puisqu'on nous proposait B et que nous n'avons pas répondu par une coordonnée valable.

Remarquez qu'un histogramme simple est un histogramme jumelé ou cumulé portant sur une seule ligne. Il suffira donc de presser [RTN] dès la seconde demande de ligne.

Quelque soit la présentation choisie, le premier graphique tracé est celui qui se trouve le plus bas et le plus à gauche sur la feuille. Dans le cas de quatre schémas, les autres seront tracés dans le sens des aiguilles d'une montre.

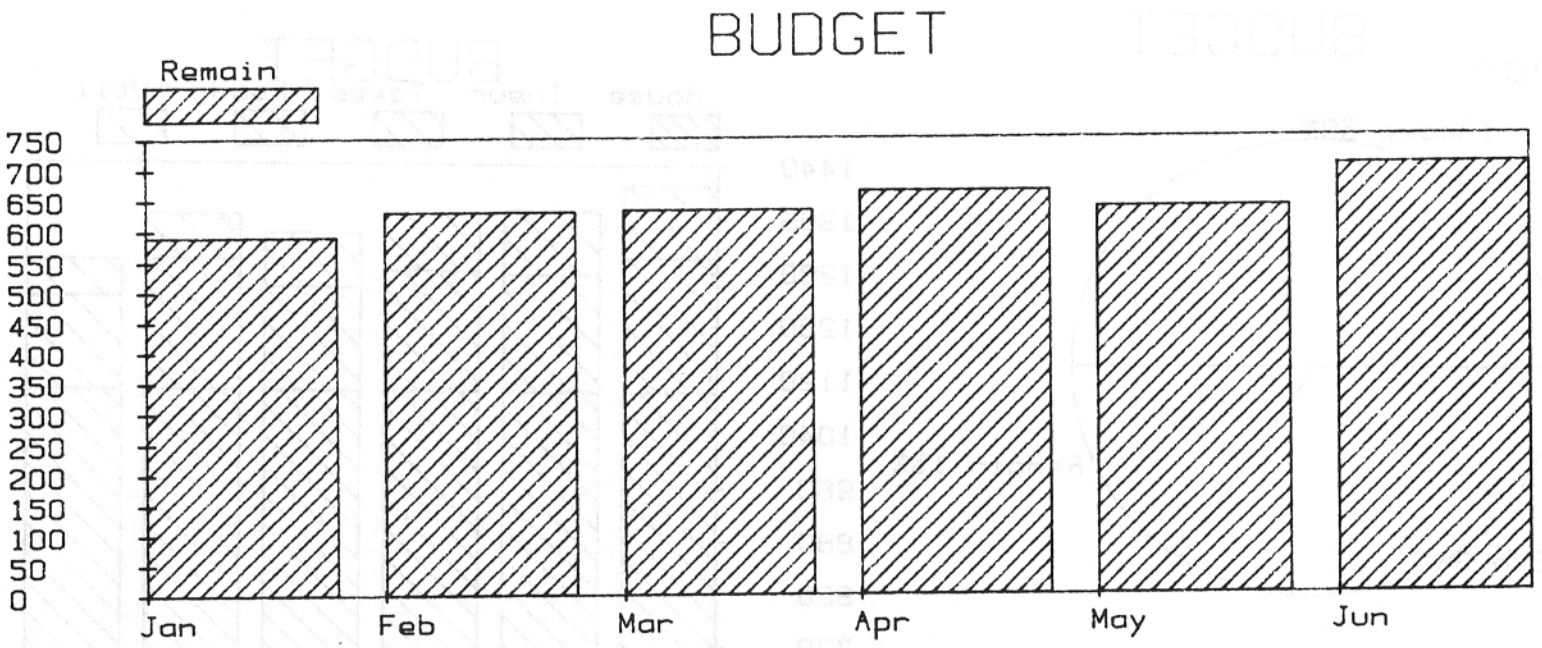
Vous pouvez mélanger ces présentations. Prenons un exemple : vous voulez tracer deux graphiques superposés dans la moitié gauche de la feuille et un seul graphique dans la moitié droite. En lançant l'extension de commandes, il faut demander la présentation 4. Après le premier tracé, en bas à gauche de la feuille nous choisirons *Suite* pour effectuer le second suivant les mêmes critères. Pour le troisième, il suffira de choisir *Vertical*, puis de ne pas effectuer le premier tracé vertical qui viendrait se superposer aux deux graphiques existants (ceci se fait en répondant [RTN] lors du choix du type de graphique). Il n'y a plus qu'à demander *Suite* et le graphique correspondant occupera la partie droite de la feuille.

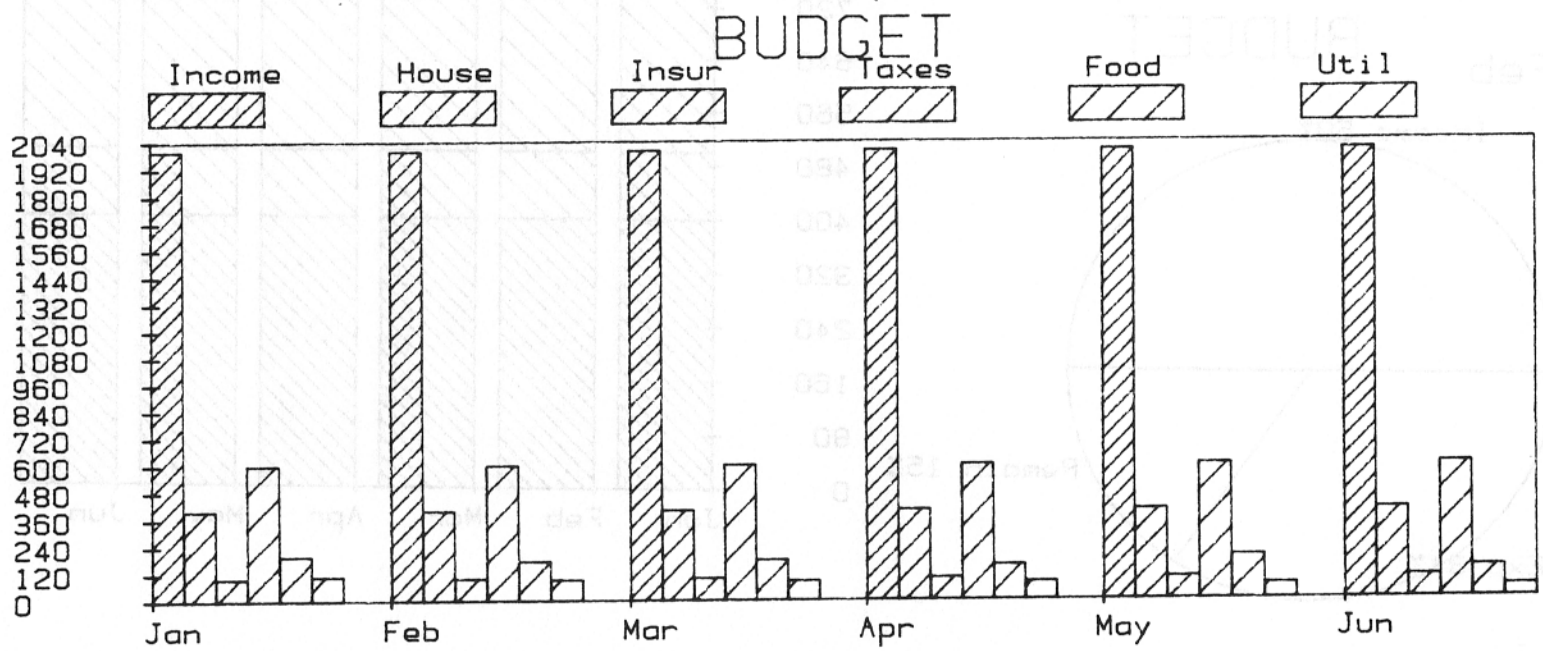
CONCLUSION

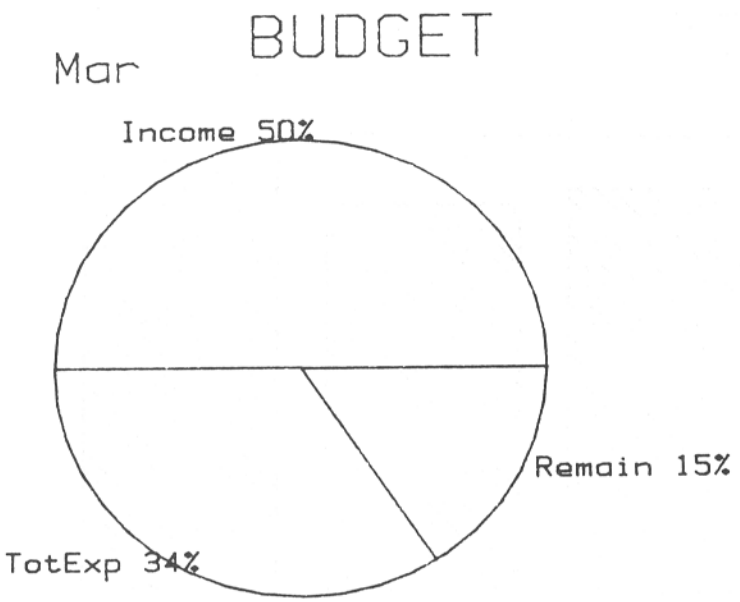
Les exemples proposés dans cet article ont été réalisés sur table traçante 7470. Le programme fonctionne également très bien, sans aucune modification, avec une interface video Pac Screen. Seule la relative lenteur du tracé est plus perceptible.

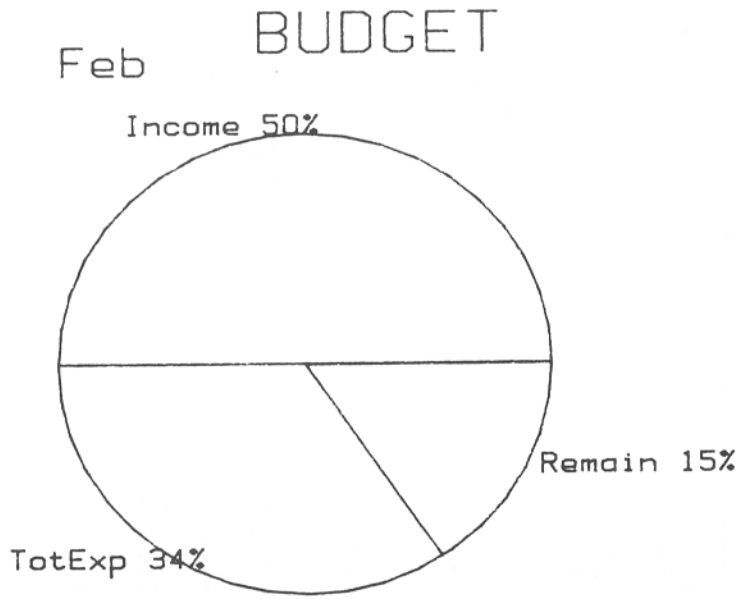
Ce programme illustre les possibilités énormes du VisiCalc du HP-75 qu'on ne trouve dans aucun des tableurs et autres logiciels intégrés du marché. Il est dommage qu'il n'existe pas d'équivalent sur HP-71.

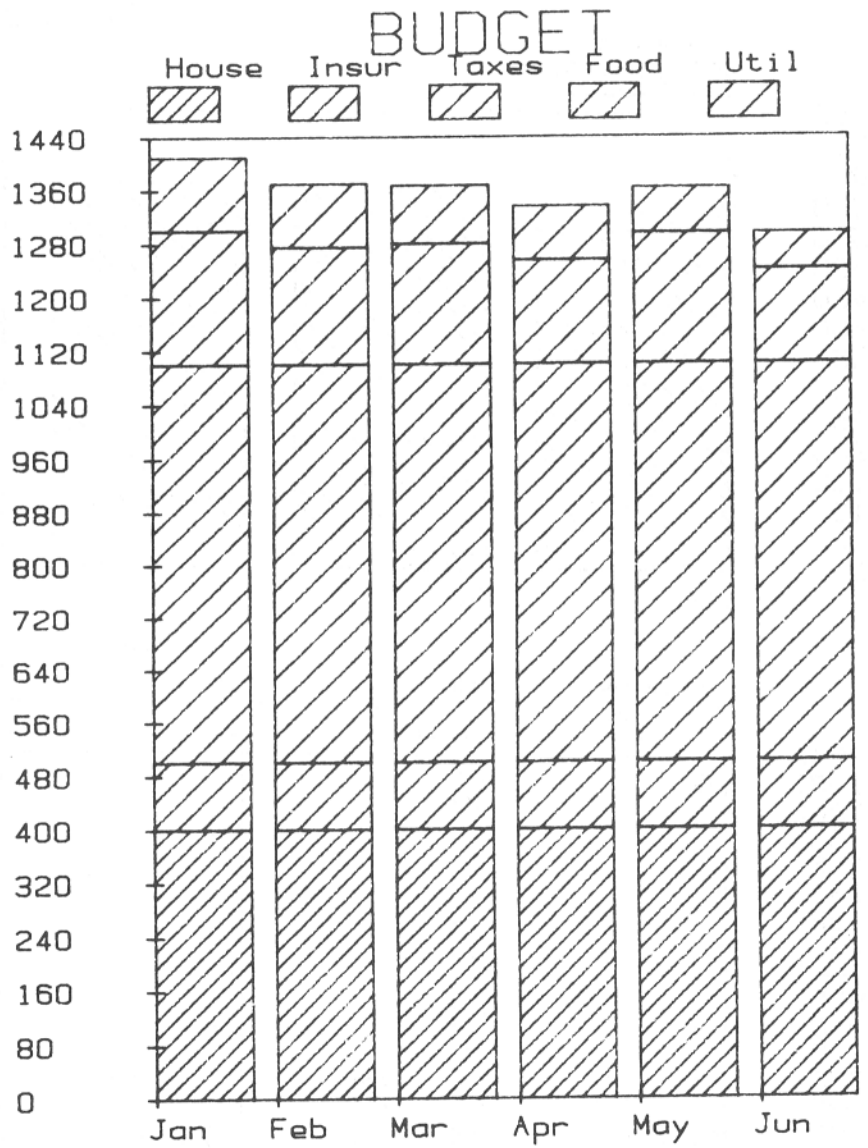
Pierre Picheret (309)


BUDGET
Remain
750
700
650
600
550
500
450
400
350
300
250
200
150
100
50
0
Jan
Feb
Mar
Apr
May
Jun


BUDGET
Income
House
Insur
Taxes
Food
Util
2040
1920
1800
1680
1560
1440
1320
1200
1080
960
840
720
600
480
360
240
120
0
Jan
Feb
Mar
Apr
May
Jun


BUDGET
Mar
Income 50%
Remain 15%
TotExp 34%


BUDGET
Feb
Income 50%
Remain 15%
TotExp 34%


BUDGET
House
Insur
Taxes
Food
Util
1440
1360
1280
1200
1120
1040
960
880
800
720
640
560
480
400
320
240
160
80
0
Jan
Feb
Mar
Apr
May
Jun

ASSEMBLEUR

J. Baudier	Etats d'âme	*18*

FORTH

F. Stengel	Les objets en Forth	*18*

BASIC

J.M. Kéfaloucos	Horloge Astronomique	*24*
P. Ehrenberg & V. Klann	Dérivées symboliques	*25*
S. Tendon	Graphes de Connett	*28*
E. Gengoux	Programme "SNCF" (voir rubrique HP-41)	*30*
P. Picheret	Programme "VISICM2" pour HP-75	*30*
J.M. Kéfaloucos	Programme "HORA"	*35*
P. Ehrenberg	Programme "DIFF"	*39*
S. Tendon	Programme "CONNETT"	*42*

LE COIN DES LHEX

44

ETATS D'AME

Chers amis de PPC, bonjour ! Je viens juste de recevoir le numéro de décembre / janvier de JPC et ceci m'amène à me poser une question.

Cette question est la suivante : où en est l'assembleur pour HP-71 ? Je veux bien croire que PPC Paris est fort dans le domaine du langage assembleur, mais ceci est uniquement dû au fait qu'un petit groupe seulement pratique, au sein du Club, l'assembleur à un haut niveau. Les autres membres sont perdus devant la complexité de l'assembleur, complexité qu'il est quasiment impossible de surmonter tout seul dans son coin.

C'est pourquoi je crie au secours et propose de créer une rubrique «initiation à l'assembleur» ou, tout au moins, d'essayer d'avoir dans chaque numéro de JPC un article d'initiation à l'assembleur. Il ne s'agit en aucun cas de niveller par le bas l'assembleur publié dans notre Journal.

Je sais que ceci n'est pas simple, car il faut pouvoir intéresser un maximun de lecteurs sans pour autant laisser tomber les autres. Mais je sais que beaucoup d'entre nous sont intéressés par l'assembleur mais ne savent pas par quel bout attaquer le problème.

Je m'engage donc à écrire dans un prochain numéro de JPC un article d'initiation dans lequel j'essaierai de faire passer les quelques notions d'assembleur que je possède.

J'espère que cette tentative sera concluante, que vous répondrez présent à l'appel et que vous me donnerez votre avis.

Voilà, c'est tout pour aujourd'hui. Je vous dis donc à bientôt...

Jacques Baudier (192)

LES OBJETS EN FORTH

Tout d'abord, je tiens à préciser que l'idée de cet article n'est pas de moi. En effet, le nom des mots accessibles à l'utilisateur est tiré de l'article "Object oriented FORTH" de Dick Pountain paru dans la revue *BYTE* d'août 1986. Néanmoins, tout le reste du code est de moi (notamment le vocabulaire vt).

Définissons ce qu'est un objet : c'est la réunion, d'une part d'une structure de données, et d'autre part de méthodes permettant d'y accéder.

Appliquons cette définition à un cas simple : définissons un type COMPLEX formé de la réunion de deux champs REAL et IMAG et des méthodes :

- COM@ permettant de déposer un complexe sur la pile.
- COM! permettant de prendre un complexe sur la pile.

```
TYPE> COMPLEX
  5 VAR REAL ( Les deux champs )
  5 VAR IMAG
OPS>
  { COM@ REAL @ IMAG @ } ( Les méthodes )
  { COM! IMAG ! REAL ! }
ENDTYPE> COMPLEX
```

Déclarons maintenant un objet de type COMPLEX : `COMPLEX Ac`, et utilisons le :

```
0 1 Ac COM! SP0 Ac COM@
```

Examinons les mots utilisés :

TYPE> ... OPS> ... ENDTYPE>
(déclarations de type) (---)

Utilisé sous la forme suivante :

```
TYPE> Nom
  déclarations1
OPS>
  déclarations2
ENDTYPE> Nom
```

Où les *déclarations1* sont les mots qui, soit serviront à définir les champs de la structure de données, soit seront des définitions internes au type, et donc inaccessibles de l'extérieur.

Pour créer un objet du type *Nom* il suffit de procéder comme suit : `Nom Obj`.

Pour utiliser cet objet, il faudra procéder comme suit : `Obj Méthode`.

(*Méthode* est un mot de *déclarations2* déclaré par `(`).

Cette syntaxe est impérative. En effet, essayez avec l'exemple précédent : `Ac +`

Votre HP-71 vous répondra par l'injure :

`«Undefined operation» !`

VAR
(champ variable) (n ---)

Utilisé sous la forme suivante :

```
n VAR Nom
```

Sert a déclarer un champ numérique de *n* quartets de long.

`{ ... }`
(---)

Utilisé sous la forme suivante :

```
{ Nom ... }
```

Crée une méthode de nom *Nom*. C'est le seul moyen d'employer les champs declarés par `VAR`.

Les présentations faites, passons aux choses sérieuses. Ce vocabulaire permet :

- de composer des types,
- de créer des tableaux de types.

Pour la composition, il est préférable de ne pas imbriquer plus de 9 types (cf. OSTACK). Pour créer des tableaux de types il suffit d'utiliser la déclaration suivante :

```
n ARRAY-OF Nom Nomtableau
```

Pour ensuite accéder au i^{ème} objet de `Nomtableau` il suffit de faire :

```
i NomTableau Méthode
```

On pourra alors définir le type suivant :

```
TYPE> COMSTACK ( Pile complexe )
  5 VAR STKPTR
  100 CONSTANT STKLEN
  STKLEN ARRAY-OF COMPLEX STACK
OPS>
  { INIT 0 STKPTR ! }
  { PUSH 1 STACKPTR + STACK COM !
        1 STACKPTR +! }
  { POP STACKPTR @ STACK COM@
        -1 STACKPTR +! }
  { EMPTY? STACKPTR @ 0= }
  { FULL? STACKPTR @ 1+ STKLEN = }
ENDTYPE> COMSTACK
```

Et l'utiliser par :

```
COMSTACK MACHIN
0 1 MACHIN PUSH
2 3 MACHIN PUSH
MACHIN POP MACHIN POP
```

Examinons les mots restant :

ARRAY-OF
(array of) (n ---)

Utilisé sous la forme suivante :

```
n ARRAY-OF NomType NomTableau
```

Pour accéder au i^{ème} élément de *NomTableau* avec la méthode *Méthode*, il faut faire :

```
i NomTableau Méthode
```

STR
(str) (n ---)

Utilisé sous la forme suivante :

```
n STR NomChaîne
```

Permet de déclarer un champ chaîne de n caractères de long. *Attention* : `STR` n'est pas au point. En effet, `STR` ne marche que dans le cas où il n'est pas utilisé à l'intérieur d'un tableau.

VAR[
(var-bracket) (---)

Utilisé sous la forme suivante :

```
VAR[ n ] NomTableau Méthode
```

Permet de compiler directement l'adresse du n^{ème} objet de *NomTableau*.

SELF
(self) (---)

Utilisé sous la forme suivante :

```
SELF Méthode
```

Permet d'utiliser une méthode d'un type à l'intérieur d'une autre méthode du même type. On a, par exemple :

```
TYPE> DEBILE
  5 VAR TOTO
OPS>
  { METH1 ." METHODE 1" }
  { METH2 SELF METH1 ."  ET METHODE 2" }
ENDTYPE> DEBILE
```

Si on effectue alors DEBILE DEB DEB METH2, on obtiendra : «METHODE 1 ET METHODE 2».

Passons maintenant à la rubrique : comment ça marche ?

La première chose à dire est que les mots de définition TYPE> ENDTYPE> OPS> et INCLUDE> modifient le vocabulaire. Vérifiez-le en entrant le type COMPLEX donné en exemple ; listez votre vocabulaire. Seul sera visible le mot COMPLEX ; tout le reste étant caché. Ceci pose un problème pour l'exécution de FORGET. Je n'ai malheureusement pas réussi à écrire un mot comme TYPEFORGET>.

Examinons le mécanisme de cette manipulation. Pour cela regardons le vocabulaire après modification :

```
Reste du dictionnaire
COMPLEX  ( celui défini par TYPE> )
REAL
IMAG
COM@     ( fin de chaîne : (lfa)=0 )
COM!     ( (lfa) pointe vers COM! )
COMPLEX  ( défini par ENDTYPE>
         ( (lfa) pointe vers reste du
           dictionnaire )
```

ENDTYPE> va court-circuiter le vocabulaire courant et créer une chaine secondaire avec les méthodes. INCLUDE>, quant à lui, va étendre la chaîne secondaire par celle de l'objet suivant c'est ce qui fait que seul le dernier INCLUDE> est pris en compte. Pour garder en mémoire tous ces détails, ces mots utilisent OSTACK comme suit :

Etat de la pile OSTACK après l'exécution des mots suivants :

```
TYPE> Nom:

Pointeur : [ 20 (décimal)                    ]
           [ NFA du mot précédent Nom        ]
           [ LFA de Nom                      ]

OPS>     :

Pointeur : [ 21 (décimal)                        ]
           [ 0                                   ]
           [ LFA du premier mot méthode de Nom   ]
           [ NFA de mot précédent Nom            ]
           [ LFA de Nom                          ]

INCLUDE> Nomi :

Pointeur : [ 21 (décimal                         ]
           [ NFA du dernier mot méthode de Nomi  ]
           [ LFA du dernier mot méthode de Nom   ]
           [ NFA du mot précédent Nom            ]
           [ LFA de Nom                          ]
```

Pour pouvoir accéder aux méthodes de COMPLEX, une clé est nécessaire : le début de la chaîne des méthodes. Cette clé est stockée dans le champ données de COMPLEX.

Examinons la structure des définitions :

Définition d'un type :

```
lfa : ?????
nfa : 'Nom du type';
cfa : pointe vers le DOES> de ENDTYPE> (CR-OBJ)
pfa : [ clef vers les méthodes ]
      [ taille du champ données ]
      [ premier octet de la déclaration de type ]
```

Objet dans le cas normal :

```
lfa : ?????
nfa : 'Nom de l'objet'; IMMEDIAT
cfa : pointe vers EX-OBJ-NOR (à travers CR-OBJ)
pfa : [ clef d'accès aux méthodes ]
      [ données ]
```

Objet déclaré dans un type :

```
lfa : ?????
nfa : 'Nom de l'objet'; IMMEDIAT
cfa : pointe vers EX-OBJ-TY (à travers CR-OBJ)
pfa : [ position du début du champ données dans
        dans celui du type ]
      [ clef d'accès aux méthodes ]
```

Pour les tableaux, la structure est similaire aux objets, il faut simplement remplacer EX-OBJ-NOR par EX-AR-NOR et EX-OBJ-TY par EX-AR-NOR.

Les objets, ainsi que les tableaux, sont des mots immédiats parce qu'ils ont une action qui varie selon les drapeaux STATE et VAR-EX. En effet, lors de la compilation d'un objet, il faut compiler l'adresse des données ainsi que la méthode utilisée. Observons la structure :

Dans le cas d'un objet normal :
[LITERAL] [adr. des données] [méthode]

Dans le cas d'un objet dans un type :
[LIT-TY] [pos. des données] [méthode]
Où n LIT-TY remplace n LITERAL O@ +

Dans le cas d'un tableau normal :

Si VAR-EX=0 (VAR[ne vient pas d'être exécuté)

```
[ LITERAL ] [ adresse des données ]

Si VAR-EX#0

[ LITERAL ] [ adr. de l'objet dans le tableau ]
[ méthode ]
```

Dans le cas d'un tableau défini dans un type, il suffit de remplacer CALC-ADR-NOR par CALC-ADR-TY.

Lors de l'exécution, le coeur du vocabulaire se trouve dans la pile OSTACK. C'est sur son sommet que l'on trouve, à tout moment, l'objet en cours de traitement. C'est elle donc qui va limiter les possibilités de ce vocabulaire. En effet, on ne peut imbriquer qu'au plus 9 (la taille de OSTACK) objets. Les mots de manipulation de cette pile sont OO >O O> O@. A part OO ils sont en tout point similaires à >R R> R@. OO remet à zéro la pile.

Encore quelques explications sur certains mots du vocabulaire vt.

SEARCHW (nfa --- cfa)
S'utilise sous la forme :

```
nfa SEARCHW Nom
```

Ce mot va chercher dans la chaîne commençant en nfa le mot Nom. Renvoie le cfa de Nom s'il a été trouvé ; provoque un ABORT" Undefined operation" s'il ne l'a pas trouvé (la fin de chaîne est indiquée par (lfa)=0).

DOES>

Une vieille connaissance. Ce mot remplace l'ancien DOES> pour pouvoir l'utiliser dans des structures IF ... ELSE ... THEN. Essayez avec l'ancien : ça coince !

STR

Ce mot est un mot infâme parce qu'il bricole avec les cfa et DOES> de manière peu orthodoxe. Si vous réussissez à comprendre son fonctionnement tout seul, alors vous serez certainement capable de le réécrire mieux (c'est à dire de telle sorte qu'il fonctionne aussi dans un tableau !).

Voilà, j'espère ne pas avoir été trop long et que vous n'aurez aucun problème à entrer ce vocabulaire.

Frank STENGEL (297)

Les primitives Forth :

```
      FORTH

* Les mots en langage machine ne sont pas
* commentés car ceux-ci (à part LIT-TY) sont
* très simples dans leur fonctionnement. Quand
* à LIT-TY son action est assez particulière,
* en effet la séquence COMPILE LIT-TY , équivaut
* à :  [COMPILE] LITERAL , O@ +



DEPTH  EQU    10     * A modifier pour imbriquer +

       WORD   'OSTACK'  * ( --- )
       RTNCC
PT     CON(5) STK
STK    BSS    5*DEPTH

       WORD   '>O'      * ( n --- )
       A=DAT1 A
       D1=D1+ 5
       CD1EX
       D=C    A
       D1=(5) PT
       C=DAT1 A
       CD1EX
       D1=D1+ 5
       DAT1=A A
```

```
CD1EX
DAT1=C A
C=D    A
D1=C
RTNCC

WORD   'O@'     * ( --- n)
CD1EX
D=C    A
D1=(5) PT
C=DAT1 A
CD1EX
A=DAT1 A
C=D    A
D1=C
D1=D1- 5
DAT1=A A
RTNCC

WORD   'O>'     * ( --- n)
CD1EX
D=C    A
D1=(5) PT
C=DAT1 A
CD1EX
A=DAT1 A
D1=D1- 5
CD1EX
DAT1=C A
C=D    A
D1=C
D1=D1- 5
DAT1=A A
RTNCC

WORD   'OO'     * ( --- )
AD1EX
LC(5) STK
D1=(5) PT
DAT1=C A
AD1EX
RTNCC

WORD   'LIT-TY'   * ( --- n)
CD1EX
D=C    A
D1=(5) PT
C=DAT1 A
CD1EX
C=DAT1 A
A=DAT0 A
A=A+C A
D0=D0+ 5
C=D    A
D1=C
D1=D1- 5
DAT1=A A
RTNCC
```

Et maintenant la partie Forth :

```
: TYPES ;

VOCABULARY vt IMMEDIATE
vt DEFINITIONS

" OWRDS" ASSEMBLE

VARIABLE OFSET
VARIABLE VAR-EX
VARIABLE CR-TYPE?

HEX

: SEARCHW
  >R BL WORD COUNT 2DUP 1- 2* + 80 TOGGLE
  BEGIN
   2DUP R@ COUNT 1F AND S= NOT
   WHILE
   R> 5- @ ?DUP 0= ABORT" Undefined operation"
   >R
  REPEAT
  2DROP R> 1 TRAVERSE 2+ ;

: EX-OBJ-NOR
  DUP @ SEARCHW
  STATE @ 0=
  IF
   SWAP 5+ SWAP EXECUTE
  ELSE
   SWAP 5+ [COMPILE] LITERAL ,
  THEN ;

: EX-OBJ-TY
  DUP 5+ @ SEARCHW
  STATE @ 0=
  IF
   SWAP @ O@ + SWAP EXECUTE
  ELSE
   SWAP @ COMPILE LIT-TY , ,
  THEN ;

: DOES>
  E601C , F8 C, E713B , ; IMMEDIATE

: CR-OBJ
  CR-TYPE? @
  IF
   CREATE
   OFSET @ , DUP @ , 5+ @ OFSET +! IMMEDIATE
   DOES> EX-OBJ-TY
  ELSE
   CREATE
   DUP @ , 5+ @ NALLOT IMMEDIATE
   DOES> EX-OBJ-NOR
  THEN ;
```

```
: STr
  DOES> @ O@ + COUNT ;

: O>D O> DROP ;

: CALC-ADR-NOR
  DUP >R 5+ @ * R> 5+ 5+ + ;

: CALC-ADR-TY
  DUP >R 5+ 5+ @ * R> @ + O@ + ;

: EX-AR-TY
  DUP 5+ @ SEARCHW >R
  STATE @ 0=
  IF
   CALC-ADR-TY R> EXECUTE
  ELSE
   VAR-EX @
   IF
    0 VAR-EX ! CALC-ADR-TY [COMPILE] LITERAL R> ,
   ELSE
    [COMPILE] LITERAL COMPILE CALC-ADR-TY R> ,
   THEN
  THEN ;

: EX-AR-NOR
  DUP @ SEARCHW >R
  STATE @ 0=
  IF
   CALC-ADR-NOR R> EXECUTE
  ELSE
   VAR-EX @
   IF
    0 VAR-EX !
    CALC-ADR-NOR [COMPILE] LITERAL R> ,
   ELSE
    [COMPILE] LITERAL COMPILE CALC-ADR-NOR R> ,
   THEN
  THEN ;

FORTH DEFINITIONS

DECIMAL

: VAR vt
  CREATE
  OFSET @ DUP , + OFSET !
  DOES> @ O@ + ;

: STR vt
  CREATE
  DUP 2* OFSET @ 2+ DUP , + OFSET ! ,
  DOES> [']  STr 5+ 5+ OVER 5- !
  DUP @ O@ + SWAP 5+ @ OVER 2- ! 0 ;

: TYPE> vt
  CREATE
  1 CR-TYPE? ! 0 OFSET ! O0 0 VAR-EX ! 1 OKFLG !
  LATEST 5- DUP >O @ >O
  20 >O ;

: OPS> vt
  O> 20 <> ABORT" ERR:Illegal TYPE> structure"
  HERE >O 0 >O 21 >O ;

: INCLUDE> vt
  O> 21 <> ABORT" ERR:Illegal TYPE> structure"
  O> DROP ' 5+ @ >O 21 >O ;

: ENDTYPE> vt
  0 OKFLG ! 0 CR-TYPE? !
  O> 21 <> ABORT" ERR:Illegal TYPE> structure"
  CREATE
  LATEST 5- @ ,
  OFSET @ ,
  O> O> !
  O> LATEST 5- !
  O> ,
  DOES> CR-OBJ ;

: { vt [COMPILE] : COMPILE >O ;

: } vt COMPILE O>D [COMPILE] ; ; IMMEDIATE

: ARRAY-OF vt
  ' CREATE
  CR-TYPE? @
  IF
   OFSET @ ,
   5+ DUP @ ,
   5+ @ DUP ,
   * OFSET +! IMMEDIATE
   DOES> EX-AR-TY
  ELSE
   5+ DUP @ ,
   5+ @ DUP ,
   * NALLOT IMMEDIATE
   DOES> EX-AR-NOR
  THEN ;

: VAR[ vt
  1 VAR-EX ! 0 STATE ! ; IMMEDIATE

: SELF vt O@ ;
```

HORLOGE ASTRONOMIQUE

HORA est l'abréviation de HORloge Astronomique, mais signifie aussi "heure" en latin. Ce programme, conçu à l'origine pour la mesure du temps en astronomie (ne pas confondre avec l'astrologie !) peut rendre service à tout utilisateur de HP-71, en transformant celui-ci en horloge / calendrier / réveil facile à régler et à utiliser.

Commencez par allumer le flag 1 en période d'heure d'hiver, ou le flag 2 en période d'heure d'été. Le numéro du flag indique la différence entre l'heure légale en France et l'heure T.U. (Temps Universel) égale à l'heure du méridien de Greenwich, donc un seul des flags 0, 1 et 2 doit être allumé en même temps.

A la mise en route, le HP-71 affiche la date et l'heure de façon continue, sous la forme : 06/12/86 13:02:00 en appuyant sur [R], vous passez en mode "réglage" : vous ajoutez ou retranchez des heures, minutes ou secondes en appuyant sur [H], [M] ou [S], suivi du nombre voulu de [+] ou de [-]. Exemple : pour avancer d'une heure (changement de fuseau horaire), appuyez sur [H], puis sur [+]. Pour retarder de 2 minutes, appuyez sur [M], puis deux fois sur [-]. Tant que le HP-71 est en mode réglage, les touches [D] et [A] permettent respectivement de modifier la date et les paramètres d'une alarme (fréquence et durée du beep, beep faible ou fort : répondre par [g][F] ou [F]. Pour terminer le réglage, appuyez sur [ENDLINE] : le HP-71 repasse en mode horloge calendrier.

Vous pouvez régler le HP-71 sur l'heure d'été, l'heure d'hiver ou l'heure T.U., en appuyant respectivement sur [E], [H] ou [U] : l'horloge est décalée du nombre d'heures voulu, le flag correspondant allumé, les autres éteints.

[A] permet de programmer une alarme : saisir la date et l'heure, éventuellement un message à afficher ; le signal sera un beep répété jusqu'à un appui sur n'importe quelle touche. Il ne sera déclenché que si le programme n'est pas arrêté entretemps. Eteindre éventuellement le HP-71 en appuyant sur [O] : le programme continuera à l'allumage.

La touche [f][D] provoque l'affichage de la date en continu sous le format : Vendredi 5 décembre 1986. [f][H] donne l'heure au format : 12h 13mn 40s. Dans ces deux cas, terminer par [ENDLINE].

[J] donne le jour julien (système de datation utilisé en astronomie, défini comme le numéro du jour depuis le 1er janvier 4713 avant J.C. : ainsi le 1er janvier 1982 est le jour julien 24.444.971, le 2 janvier est le 24.444.972, etc.). [S] donne le temps sidéral, utilisé en astronomie et en navigation, basé sur un jour de 23h 56mn : intervalle de temps séparant deux passages consécutifs d'une étoile au même méridien. Pour ces deux sous programmes, il suffit de répondre aux questions (année, mois etc.) en appuyant sur [ENDLINE] pour renvoyer la date et l'heure actuelles, ou en indiquant une date et une heure différentes.

[W] affiche le jour de la semaine après saisie de la date (équivalent de la fonction DOW du module Time HP-41).

[f][S] affiche en continu l'heure sidérale. Terminer par [ENDLINE]. En mode horloge calendrier, [ENDLINE] permet de sortir proprement du programme. Dans les autres modes, il faut deux appuis successifs sur cette touche.

Jean-Michel Kéfaloucos (140)

NDLR : ce programme utilise les fonctions de trois Lex : KEYWAIT, STRUC1 et INDATLEX.

STRUC1 sera bientôt remplacé par STRUC2, offrant plus de possibilités de programmation structurée.

INDATLEX est supprimé, comme indiqué dans JPC 39 (Novembre 1986), page 16. En conséquence, vous pouvez soit ignorer la ligne 430, soit la transformer pour faire appel au sous programme INDATE.

CALCUL SYMBOLIQUE (ACTE I)

Le calcul symbolique est un domaine peu exploré sur le HP-71B. Beaucoup d'entre vous ont fait connaissance avec ce sujet grâce au HP-28C présenté dans le JPC 41.

Pourtant, nos collègues allemands ont fait paraître dès mai 1985 dans *Prisma*, le Journal de leur Club (CCD), un programme de dérivation d'expressions.

Le but est de calculer formellement la dérivée d'une expression. Ce qui nous intéresse, c'est la fonction dérivée, et non sa valeur en quelque point. Autrement dit, nous voulons savoir que la dérivée de $sin(x)/x$ est $(xcos(x)-sin(x))/x^2$, et non calculer des valeurs numériques.

HISTORIQUE

La première version du programme que nous présentons ici est dûe à Peter Ehrenberg, et est parue dans *Prisma* de mai 1985. Tout était écrit en Basic. Et Peter lançait un appel aux amateurs d'assembleur.

Peu de temps après, Volker Klann répondit à cet appel. Il a transcrit en assembleur, sous forme de fichier Bin, le sous programme le plus utilisé.

Peter et Volker nous ont très gentiment accordé l'autorisation de publier leurs travaux dans JPC. Remercions-les vivement.

UTILISATION

Si vous avez le module Forth / Assembleur, nous verrons plus loin le sous programme en assembleur qui vous permettra d'aller plus vite. Si vous ne possèdez pas ce module, ne lisez pas le chapitre suivant.

Dans tous les cas, rentrez le programme Basic "DIFF" listé plus loin.

Toutes les expressions doivent être des fonctions de la variable x. Elles peuvent comporter des constantes (de A à Z, ainsi que E et PI). Les minuscules et majuscules sont indifférenciées.

Les expressions peuvent contenir des opérateurs standard (*, /, +, -, ^) ainsi que les fonctions *sin*, *cos*, *tan*, *ln*, *exp*, *sqr*, *asin*, *acos*, *atan*, *cot*, *cosh*, *sinh*, *tanh*, *coth*, *asinh*, *acosh*, *atanh* et *acoth*.

Essayons de dériver $sin(x)/x$. Faites `RUN DIFF`. A la question «`f(x)=`», rentrez `SIN(X)/X`, puis appuyez sur la touche `[ENDLINE]`.

Environ 10 secondes après, votre HP-71 vous affiche :

```
f'=(COS(X)*X-SIN(X))/X^2
```

Impressionant, non ?

Faites maintenant `[f][CONT]` pour dériver une autre expression.

SI VOUS VOULEZ ALLER PLUS VITE

... et si vous avez le module Forth / Assembleur ou si vous pouvez venir aux réunions !

Effacez le sous programme "MINPRIO" dans le Basic, c'est à dire les lignes 160 à 360.

Rentrez ensuite le programme source ci-dessous, assemblez-le. Ca y est.

Voilà, le premier acte du calcul symbolique est de nos amis allemands. A quand la suite ?

```
        BIN    'DIFBIN'

=ADRS40 EQU    #0F52B
=AVE=D1 EQU    #188B8
=D1=AVE EQU    #18651
=DEST   EQU    #0F7B0
=ENDBIN EQU    #0764B
=EXPEXC EQU    #0F186
=HDFLT  EQU    #1B31B
=STORE  EQU    #0F5F8

=S-R1-2 EQU    #2F88B

=tCOM   EQU    #F1
=tEOL   EQU    #F0
=tLITRL EQU    #C4
=tPRMEN EQU    #F8
=tPRMST EQU    #F4
=tSUB   EQU    #C1
=tSVAR  EQU    #2D
```

```
        CHAIN  MINPR    premier sous-programme
        GOVLNG =ENDBIN
        REL(2) MINPRC
        CON(2) =tSUB
 MINPR  CON(5) -1       pas d'autres sous-prog.

        CON(2) =tLITRL
        NIBASC 'MINPRIO' nom du sous-programme
        CON(2) =tPRMST
        CON(2) =tSVAR
        NIBASC 'T'      premier parametre
        CON(2) =tCOM
        NIBASC 'P'      deuxieme parametre
        CON(2) =tCOM
        NIBASC 'O'      troisieme parametre
        CON(2) =tCOM
        NIBASC 'L'      quatrieme parametre
        CON(2) =tPRMEN
MINPRC  CON(2) =tEOL

* L=LEN(T$)

        SETHEX
        P=     0
        LC(3)  #074     code de T$
        B=C    X
        GOSBVL =ADRS40
        D0=D0+ 2
        C=DAT0 A
        D0=C            D0 := ^ T$
        A=0    A
        A=DAT0 4        A[3:0] := longeur T$
        GOSUB  CONPUS   M.S. := LEN(T$)

* P=0 @ K=0 @ O=6

        C=0    A
        R0=C            R0[A] recoit P
        D1=C            D1 recoit K (var. locale)
        C=C+1  B        C[B] := 1
        R2=C            R2[A] := I
        P=     2
        C=P    15       C[S] := 15
        R3=C            R3 = 2???????????00001
        C=C+1  S        C[S] := 3
        C=C+C  S        C[S] := 6
        A=C    S        parametre O dans A[S]

* premiere partie de "FOR I=1 TO L"
* teste si on doit quitter la boucle

        C=DAT0 4        C[A] := LEN(T$)
        ?C#0   A        LEN(T$) = 0 ?
        GOYES  L240     oui, executer la boucle
        GOTO   L350     non, passer la boucle

* deuxieme partie de "FOR I=1 TO L"
* calcule l'adresse du premier element de T$

 L240   C=C-1  A
        D=C    A        compteur dans D
        C=C+C  A
        D0=D0+ 4
        ADOEX
        A=A+C  A
        D0=A            D0 := ^ T$[1,1]

* troisieme partie de "FOR I=1 TO L"
* debut d'execution de la boucle

 L250   A=DAT0 B        K$ est dans A[B]
        P=     0

* IF K$='(' THEN K=K+1 @ GOTO 340

        LCASC  '('
        ?A#C   B        IF K$#'(' THEN 270
        GOYES  L270
        D1=D1+ 1        K=K+1
        GOTO   L340

* IF K$=')' THEN K=K+1 @ GOTO 340

 L270   LCASC  ')'
        ?A#C   B        IF K$#')' THEN 280
        GOYES  L280
        D1=D1- 1        K=K-1
        GOTO   L340

* IF K#0 THEN 340

 L280   CD1EX           C[A] := K
        D1=C
        ?C#0   A        IF K#0 THEN 340
        GOYES  L340

* IF K$='+' THEN ...

        LCASC  '+'
        ?C#A   B        IF K$#'+' THEN 300
        GOYES  L300
        P=     1
        GOTO   L339     fin de boucle

* IF K$='-' THEN ...

 L300   LCASC  '-'
        ?C#A   B        IF K$#'-' THEN 310
        GOYES  L310

        P=     2
        GOTO   L339     fin de boucle

 L310   C=R3
        B=C    W        B[W] = 2???????????00001

* IF K$='*' THEN ...
```

```
        LCASC  '*'
        P=     3        valeur pour 0
        GOSUB  L337
        P=     0

* IF K$='/' THEN ...

        LCASC  '/'
        P=     4        valeur pour 0
        GOSUB  L337
        P=     0

* IF K$='/' THEN ...

        B=B+B  S        B[S] := 4
        B=B+1  S        B[S] := 5

        LCASC  '^'
        P=     5        valeur pour 0
        GOSUB  L337

        GOTO   L340     fin de boucle

* ... AND (O>N OR P=1) THEN ... (n=2 ou n=5)

 L337   ?C#A   B        C vaut '*' ou '/' ou '^'
        RTNYES
        ?A>B   S        O>2 ?
        GOYES  L338
        C=R0            C recoit P
        ?C#B   A        B[A] := 1
        RTNYES
 L338   C=RSTK

* ... O=N @ P=I

 L339   C=P    15
        A=C    S        nouvelle valeur pour O
        C=R2
        R0=C

* NEXT I

 L340   D0=D0- 2        adresse du prochain
         *              element de T$
        C=R2
        C=C+1  A
        R2=C            I=I+1
        D=D-1  A        decremente le compteur
        GOC    L350     sortie de boucle

        GOTO   L250

 L350   P=     0
        LC(5)  #2F6E9   adresse des flags 0 a 3
        CD1EX
        B=C    A
        A=DAT1 1        lit les flags

* CFLAG 1

        LC(1)  #D       masque
        A=A&C  B

* IF K#0 THEN CFLAG 0 @ SFLAG 1

        ?B=0   A
        GOYES  L360     K=0
        LC(1)  2        masque
        A=A!C  B        SFLAG 1
        LC(1)  #E
        A=A&C  B        CFLAG 0
 L360   DAT1=A 1        reecrit les flags 0 a 3

* met a jour les valeurs de P, O et L

        A=0    A
        ASLC            A[S] = 0
        GOSUB  CONPUS
        A=R0
        GOSUB  CONPUS
        LCASC  'P'
        GOSUB  POPSTO
        LCASC  'O'
        GOSUB  POPSTO
        LCASC  'L'
        GOSUB  POPSTO

        GOVLNG =ENDBIN  fin du sous-programme

* sous-programme CONPUS

 CONPUS GOSBVL =HDFLT
        SETHEX
        GOSBVL =D1=AVE
        D1=D1- 16       decremente le pointeur
        DAT1=A W        copie A vers la pile
 CONPU1 GOVLNG =AVE=D1  met a jour le pointeur
         *              et retour

* sous-programme POPSTO

 POPSTO A=C    B        copie le nom dans A[B]
        GOSUB  POPST1
        CON(2) 0        reserve un octet
        CON(2) =tEOL
 POPST1 C=RSTK
        D0=C
        DAT0=A B        copie le nom dans l'octet
         *              reserve
        GOSBVL =D1=AVE
        RSTK=C          sauvegarde de D1
        GOSBVL =EXPEXC
        GOSBVL =DEST
        A=0    A
        D1=(5) =S-R1-2
        DAT1=A A
```

```
C=RSTK          relit D1
RSTK=C
D1=C
A=DAT1 W
GOSBVL =STORE
C=RSTK          relit D1 encore une fois
D1=C
D1=D1+ 16       incremente le pointeur
GOTO   CONPU1

END
```

GRAPHES DE CONNETT

Tous ceux qui ont un HP-71B et une imprimante ThinkJet peuvent désormais dessiner des graphes de Connett.

L'idée est très simple, et est basée sur la formule x^2+y^2.

Le programme examine tous les points de la région que vous désirez explorer, évalue la formule pour chacun d'entre eux, et prend la partie entière du résultat. Si cet entier est pair, le programme noircira le point correspondant, sinon il le laissera blanc.

Si vous désirez, vous pouvez inverser le dernier test, et noircir le point si le résultat est impair. Ce n'est qu'une question de goût personnel.

L'algorithme original, tel que publié dans la traduction italienne de la revue *Scientific American*, était :

entrer x **et** y
entrer l **et** r
pour i **de** l **à** r
  pour j **de** l **à** r
    x_1 := x + (l*i/r)
    y_1 := y + (l*j/r)
    z := x_1^2 + y_1^2
    c := INT(z)
    si c est pair
      alors noircir le point (i, j)
    fin si
  fin pour
fin pour

La première version du programme Basic sur le HP-71 créait un graphe de 640 points par 640 points en à peu près 15 heures. J'ai donc appliqué les optimisations suivantes :

x et y sont les coordonnées du coin inférieur gauche d'un carré de l unités de coté, avec une résolution de r points.

x, y, l, r, i et j sont toutes supposées être des valeurs entières.

Nous voulons évaluer la parité de l'expression :

$$\mathrm{INT}((x + li/r)^2 + (y + lj/r)^2),$$

soit :

$$\mathrm{INT}(x^2 + y^2 + (2rlxi + l^2i^2 + 2rlyj + l^2j^2) / r^2)$$

Maintenant, soit a et b. L'égalité INT(a+b)=a+INT(b) est vraie si a est entier. Or, x^2+y^2 est certainement un entier. Donc, nous pouvons réduire le problème à la parité de l'expression :

$$x^2 + y^2 + \mathrm{INT}((2rlxi + l^2i^2 + 2rlyj + l^2j^2) / r^2)$$

Le numérateur du second terme peut être considéré comme la somme de deux polynômes en i et en j. Ils peuvent donc être évalués par l'algorithme de Hörner :

$$x^2 + y^2 + \mathrm{INT}(((2rli + l^2i)i + (2rly + l^2j)j) / r^2)$$

maintenant, posons :

$a = x^2 + y^2$
$b = 2rlx$
$c = 2rly$
$d = l^2$
$e = r^2$
$f = (2rlx + l^2i)i = (b + di)i$

a, b, c, d et e sont des invariants pour les deux boucles, c'est à dire que i et j n'apparaissent pas dans ces expressions. f est invariant dans la boucle interne (j n'apparaît pas dans l'expression). Ceci nous amène à :

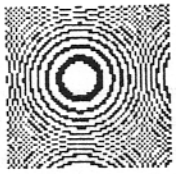
$$a + \mathrm{INT}((f + (c + dj)j) / e)$$

Nous pouvons évaluer la partie entière plus vite en utilisant la division entière :
a + (f + (c + dj)j) DIV e

Et nous pouvons connaître la parité en cherchant le reste de la division par 2 :

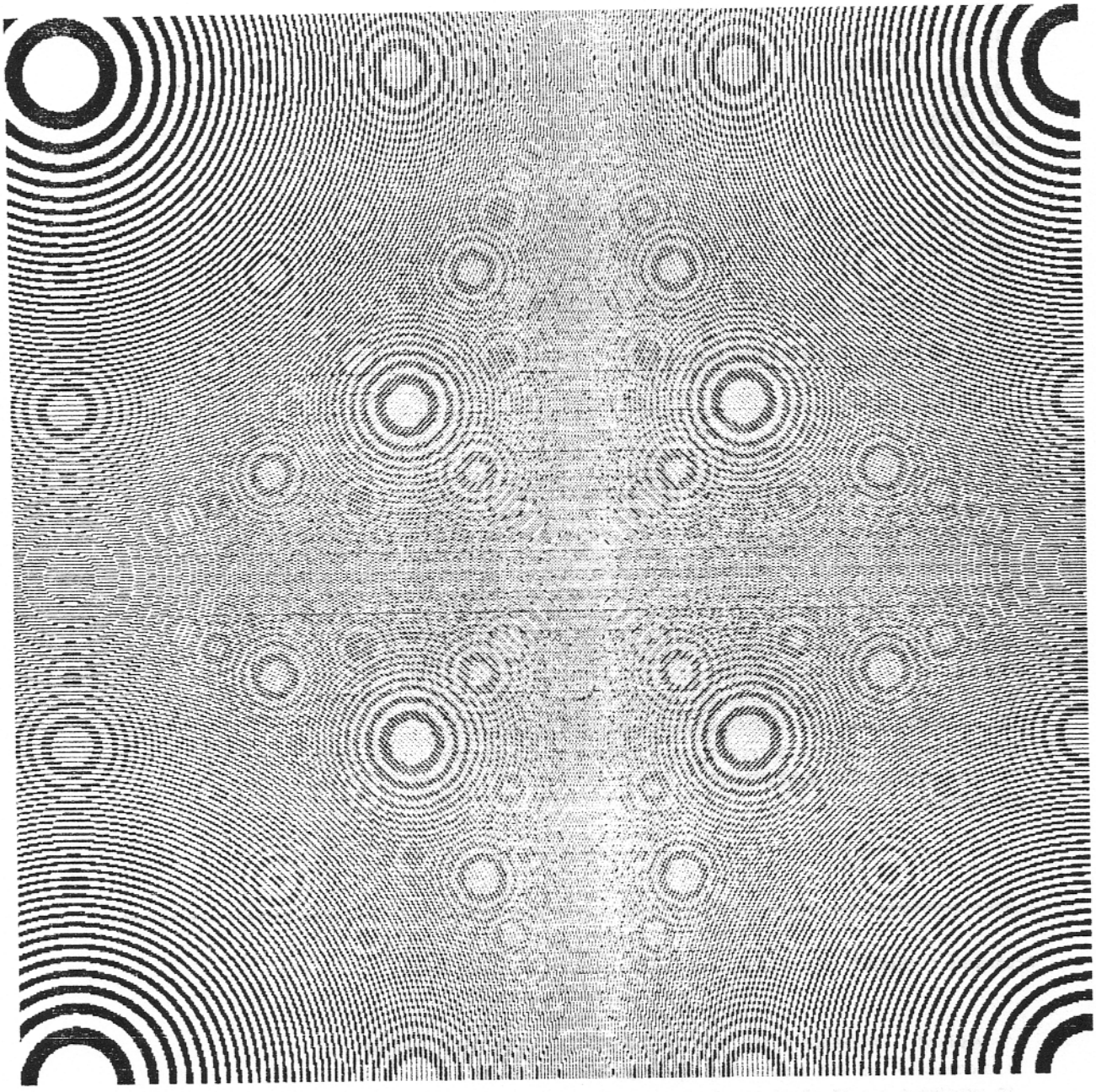
MOD (a + (f + (c + dj)j) DIV e, 2)

Rassemblez toutes ces remarques ensemble, et vous obtenez le programme Basic. Cette version optimisée crée un graphe de 640 points par 640 en environ 6 heures et 40 minutes, soit 225 % plus vite.



64 3 3 9

Stefano Tendon (241)
Italie



640 -26 -26 26

Programme "SNCF" (Tarification SNCF mise a jour, voir rubrique HP-41)

```
================================================================
  10 SUB TAX587(C,R,B,D,P)
   - SUB TAX587(Classe, Réduction%, Code AD/ENF/CH., Dist.km, [prix])
     Niveau tarifaire 30 avril 1987
     (pour prix AR, mult. par 2 le résultat)
  50 OPTION BASE 1
  60 DIM D0(3),P0(3),P1(3),P8(3),P9(3),M9(2)
   - Prix et seuils de base utilisés
  80 D0(1)=299 @ D0(2)=349 @ D0(3)=INF
  90 P0(1)=10 @ P0(2)=20.50 @ P0(3)=38.00
 100 P1(1)=.42 @ P1(2)=.4 @ P1(3)=.36
 110 P8(1)=.2 @ P8(2)=.5 @ P8(3)=1
 120 P9(1)=20 @ P9(2)=30 @ P9(3)=INF
 130 M9(1)=4.8 @ M9(2)=7.0 @ T=.6
 140 GOSUB 'DISTP'
 150 GOSUB 'PRIARR'
 160 END

================================================================
 170 'PRIARR': IF P0(1)>T*P1(1)*D2 THEN P=(1+T)*P1(1)*D2 @ GOTO 220
 180 FOR I=1 TO 3
 190 IF D2<D0(I)+1 THEN P=P0(I)+P1(I)*D2 @ GOTO 220
 200 NEXT I
 210 IF B=3 OR B=6 THEN P=.5*P @ GOTO 250
 220 P=P*(1-R/100)
 230 IF C=1 THEN P=1.5*P
 240 IF B=2 THEN P=.5*P
 250 FOR I=1 TO 3
 260 IF P<P9(I) THEN P=CEIL(P/P8(I))*P8(I) @ GOTO 280
 270 NEXT I
 280 M=M9(2) @ IF C=1 THEN M=M9(1)
 290 IF P<M THEN P=M
 300 RETURN

================================================================
 310 'DISTP': IF D<=7 THEN D9=7 @ D2=6.5 @ GOTO 360
 320 IF D<=49 THEN D9=2*IP(D/2)+1 @ D2=D9-.5 @ GOTO 360
 330 IF D<=199 THEN D9=5*IP(D/5)+4 @ D2=D9-2 @ GOTO 360
 340 IF D<=1949 THEN D9=10*IP(D/10)+9 @ D2=D9-4.5 @ GOTO 360
 350 D9=10*IP(D/10)+9 @ D2=D9-4.5
 360 RETURN
```

****

Programme "VISICM2" (Extensions graphiques de VISICALC pour HP-75)

```
   - visicm2    PICHERET PEIRE  PPCT211 PPCP309 10/02/85 VERs 21/1/87
     necessite les ROM VISICALC et hp75 I/O
  10 INTEGER P,C,R,F5,V5,V0,F1,X0,Y0,X1,Y1,Z0,Z1,Z2,S1,S4,S5,S6,S7
  20 RCLVAR
  30 T2$=CHR$(13)&" ÊËÀÂ`Ùè" @ M2=0
```

```
 40 DIM B0$[96],M$[96],T$[96]
 60 OPTION BASE 1
 61 DIM P1(4,4),P2(2),Q(4)
 62 INTEGER T0,T8,T,T2,T3(6)
 63 M1=100 @ GOSUB 8795
 64 T=GETSTATUS(0) @ ON K GOSUB 115,116,117,118,119
 65 PUTSTATUS 0,T
 66 FOR M=1 TO 2
 67 FOR N=1 TO 4
 70 READ P1(M,N)
 71 NEXT N
 72 NEXT M
 75 GOTO 151

=============================================================
 80 DATA 250,279,5050,4200,850,479,4850,3140
 82 DATA 250,4000,5050,7479,850,4220,4850,6879
 84 DATA 5250,279,10250,4200,5850,479,9850,3140
 86 DATA 5250,4000,10250,7479,5850,4220,9850,6879
101 DATA 250,279,5050,7479,850,1079,4850,6679
104 DATA 5250,279,10250,7479,5850,1079,9850,6679
107 DATA 250,279,10250,4200,1850,479,9850,3140
110 DATA 250,4000,10250,7479,1850,4220,9850,6879

=============================================================


=============================================================
113 DATA 250,279,10250,7479,1850,1079,9850,6679

=============================================================
115 RESTORE 113 @ T=0 @ RETURN
116 RESTORE 101 @ T=120 @ RETURN
117 RESTORE 107 @ T=121 @ RETURN
118 RESTORE 80 @ T=122 @ RETURN
119 IF T=0 THEN 63 ELSE GOSUBX T @ GOTO 65
120 RESTORE 104 @ T=0 @ RETURN
121 RESTORE 110 @ T=0 @ RETURN
122 RESTORE 82 @ T=123 @ RETURN
123 RESTORE 84 @ T=124 @ RETURN
124 RESTORE 86 @ T=0 @ RETURN

155 IF PRT$#":G1" THEN P1$=PRT$ @ PRINTER IS ":G1"

=============================================================
400 DEF FNP7(X) = 2*PI*X/U1

=============================================================
401 DEF FNP8(X,Y)
421 IF 0#POS("NAERROR",GETVALUE$(X,Y,1)) THEN FNP8=0 @ GOTO 451
426 FNP8=VAL(GETVALUE$(X,Y,1))
451 END DEF

=============================================================


=============================================================
491 DEF FNP9(X,Y) ! CALCUL du PAS GRADUTIONS
```

```
 492 D=-1*(MAX((X-Y)/X,Y/X)/2-1)
 493 D=D*(X/10^INT(LOG10(X)))

=============================================================
 494 DATA 5,6,8,10,12,16,20,25,50,60,80,100

=============================================================
 495 RESTORE 494 @ FOR Y=1 TO 12 @ READ L @ L=L/10
 496 IF D<L THEN 498
 497 NEXT Y
 498 FNP9=L*10^(INT(LOG10(X))-1)
 499 END DEF

=============================================================
 500 M1=80 @ GOSUB 8795 @ T=K
1010 M$=MESSAGE$(95)
1020 B0$=COORD$(C,0,S5)
1030 GOSUB 9115 @ T0=0
1040 T0=COL(B0$)
1060 IF NOT T0 THEN BEEP @ GOTO 1010
1061 IF T=3 THEN T8=T0 @ GOTO 1115
1070 M$=MESSAGE$(97)
1080 B0$=COORD$(MAXCOL,0,S5)
1090 GOSUB 9115 @ T8=0
1100 T8=COL(B0$)
1110 IF NOT T8 THEN BEEP @ GOTO 1070
1115 M$=MESSAGE$(94)
1120 FOR T2=1 TO 6
1130 IF T2=1 THEN B0$=COORD$(0,R,S5) ELSE B0$=CHR$(T2+64)
1140 GOSUB 9115 @ IF B0$=CHR$(T2+64) THEN T2=T2-1 @ GOTO 1500
1150 IF ROW(B0$) THEN T3(T2)=ROW(B0$) ELSE 1130
1160 NEXT T2 @ T2=T2-1

1510 GOSUB 5000 @ IF T=3 THEN 3500
1670 T4=FNP9(U1,U)
1680 T9=T4*CEIL(U1/T4)
1690 T1=0


2102 GOSUB 9983
2104 P1(3,1)=T0
2105 P1(3,2)=T1
2106 P1(3,3)=T8+.8
2107 P1(3,4)=T9
2108 GOSUB 6038
2110 FOR Q1=0 TO 2 STEP 2

2113 P2(1)=P1(3,Q1+1)
2114 P2(2)=P1(3,Q1+2)
2115 GOSUB 6700 @ NEXT Q1
2121 GOSUB 6600

2130 FOR M=T0 TO T8 @ U=P1(3,2)
2131 FOR N1=1 TO T2
2132 IF T=2 THEN 2140
2133 P2(1)=M @ P2(2)=U @ Q1=0 @ GOSUB 6700 @ GOSUB 4170
2134 P2(1)=M+.8 @ A=T3(N1) @ P2(2)=FNP8(M,A)+U @ U=P2(2) @ Q1=2 @ GOSUB 6700 @ GOTO 2144
2140 P2(1)=M+(N1-1)*(.8/T2) @ P2(2)=U @ Q1=0 @ GOSUB 6700 @ GOSUB 4170
```

```
2141 P2(1)=M+N1*(.8/T2) @ A=T3(N1) @ P2(2)=FNP8(M,A) @ Q1=2 @ GOSUB 6700
2144 GOSUB 6500
2145 NEXT N1
2146 NEXT M
2210 FOR M=T1 TO T9 STEP T4
2220 P2(1)=P1(3,1) @ P2(2)=M @ Q1=0 @ GOSUB 6700
2230 PRINT "PA";Q(1);Q(2);"YT;"
2240 PRINT "CP-8,-.25;LB";M;CHR$(3)
2250 NEXT M
2251 U=INT((P1(2,3)-P1(2,1))/T2) @ FOR N1=1 TO T2
2252 Q(1)=P1(2,1)+U*(N1-1) @ Q(2)=P1(2,4)+100
2253 Q(3)=Q(1)+MIN(1000,U/2) @ Q(4)=Q(2)+200 @ GOSUB 6500
2254 PRINT "pa";Q(1);Q(2)
2260 PRINT "CP1,1.2;LB";GETLABEL$(0,T3(N1));CHR$(3) @ NEXT N1
2270 PRINT "pa";(P1(2,1)+P1(2,3))/2;P1(2,4);"SI";L9/5;H9/5
2280 PRINT "CP";-LEN(ACTIVE$)/2;1;"LB";ACTIVE$;CHR$(3)
2499 GOTO 9990

     GRAPH X,Y

3489 GOTO 9990

3500 P1(3,1)=-1.1 ! CAMEMBERT
3530 P1(3,2)=-1
3540 P1(3,3)=1.756
3550 P1(3,4)=1
3560 GOSUB 6038
3565 P2(1)=0 @ P2(2)=0 @ Q1=0 @ GOSUB 6700 @ PRINT "PA";Q(1);Q(2);"PD"
3570 FOR O1=0 TO 2*PI+PI/20 STEP PI/20
3580 P2(1)=COS(O1) @ P2(2)=SIN(O1) @ Q1=0 @ GOSUB 6700
3590 PRINT "PA";Q(1);Q(2);"PD"
3600 NEXT O1 @ PRINT "PU";
3605 N=T0 @ P2(1)=0 @ P2(2)=0 @ Q1=0 @ GOSUB 6700 @ A=0
3607 FOR N1=1 TO T2
3610 GOSUB 7000 @ U=A+T1/2
3615 P2(1)=COS(FNP7(U))
3616 P2(2)=SIN(FNP7(U)) @ Q1=2 @ GOSUB 6700
3618 IF FNP7(U)<PI/2 OR FNP7(U)>PI*1.5 THEN K=1 ELSE K=-9
3619 PRINT "PA";Q(3);Q(4);
3620 PRINT "CP";K;0;"LB";GETLABEL$(0,T3(N1));" ";STR$(IP(T1/U1*100));"%";CHR$(3)
3625 A=A+T1
3635 P2(1)=COS(FNP7(A))
3640 P2(2)=SIN(FNP7(A)) @ Q1=2 @ GOSUB 6700
3650 PRINT "PA";Q(1);Q(2);"PD";Q(3);Q(4);"PU"
3670 NEXT N1 @ PRINT "SI";L9/7;H9/7;"PA";P1(2,1);P1(2,4);"CP1,1";
3675 PRINT "LB";GETLABEL$(T0,0);CHR$(3)
3999 GOTO 2270
4170 IF N1#1 THEN RETURN
4171 PRINT "PA";Q(1);Q(2);"XT;"
4190 PRINT "CP-.33,-1;LB";GETLABEL$(M,0);CHR$(3) @ RETURN

5010 T1=0
5020 U=INF @ U1=EPS
5030 FOR N=T0 TO T8
5040 T4=0
5050 FOR N1=1 TO T2
5060 GOSUB 7000
5070 IF U>T1 THEN U=T1
```

```
5080 T4=T4+T1
5090 IF T#2 THEN 5110
5100 IF U1<T1 THEN U1=T1
5110 NEXT N1
5120 IF T=2 THEN 5140
5130 IF U1<T4 THEN U1=T4
5140 NEXT N @ RETURN
   -   INITIALISATION
6038 X2=(P1(3,3)-P1(3,1))/(P1(2,3)-P1(2,1)) @ P1(4,1)=P1(3,1)-X2*(P1(2,1)-P1(1,1))
6040 X3=(P1(3,4)-P1(3,2))/(P1(2,4)-P1(2,2)) @ P1(4,2)=P1(3,2)-X3*(P1(2,2)-P1(1,2))
6041 P1(4,3)=P1(3,3)+X2*(P1(1,3)-P1(2,3))
6042 P1(4,4)=P1(3,4)+X3*(P1(1,4)-P1(2,4))
6060 PRINT "IN;VS10;IP p1(1,1),p1(1,2),p1(1,3),p1(1,4)"
6070 L9=2 @ H9=3 ! @ PRINT "SI",L9/10,H9/10,";"
6080 PRINT "SP1;"
6090 RETURN

     Hachures

6511 PRINT "IW";Q(1);Q(2);Q(3);Q(4)
6520 FOR Y=Q(2)-(Q(3)-Q(1)) TO Q(4) STEP 100*SQR(N1)
6530 PRINT "PU";Q(1);Y;"PD";Q(3);Y+(Q(3)-Q(1))
6540 NEXT Y
6550 PRINT "PU;IW;"
   -  trace la boite x1,y1 ; x2,y2

6620 PRINT "PA;PU";Q(1);Q(2)
6622 PRINT "PA;PD";Q(3);Q(2);Q(3);Q(4);Q(1);Q(4);Q(1);Q(2)
6623 PRINT "PU";
6640 RETURN
   - P2(x.y)en unit utilisateur sont transforme en unit plotter , charger dans q(1+q1,2+q1)
6701 FOR O=1 TO 2
6711 P2(O)=(P1(1,2+O)-P1(1,O))/(P1(4,2+O)-P1(4,O))*(P2(O)-P1(4,O))+P1(1,O)
6721 NEXT O
6722 Q(1+Q1)=INT(P2(1)) @ Q(2+Q1)=INT(P2(2))
6723 RETURN

7000 A$=GETVALUE$(N,T3(N1),1) @ T1=POS("NAERROR",A$) @ IF T1#0 THEN T1=0 ELSE T1=VAL(A$)
7010 RETURN

8795 IF NOT M2 THEN M2=M1+1
8796 SETWIN @ DISP MESSAGE$(M1);": ";MESSAGE$(M2) @ T$=UPRC$(CHOICE$(M2)) @ M2=0
8810 K$=WAITKEY$(-1) @ K=POS(T$,UPRC$(K$)) @ IF K THEN RETURN
8830 IF POS(" ",K$) THEN POP ELSE 8810
8835 IF K$=" " THEN P=0
8852 GOTO 9990

9115 SETWIN @ DISP M$;": "; @ CLRLCD LEN(M$)+3
9121 W0,P0=1 @ F0=0
9123 SETWIN LEN(M$)+3
9125 K$=INLINE$(B0$,W0,P0,T2$,F0)
9130 B0$=INPBUF$ @ W0=INPWIN @ P0=INPPTR @ F0=IRFLAG @ L=LEN(B0$)+1
9180 K=POS(T2$,K$) @ IF K>3 THEN 9123
9210 IF K=1 THEN RETURN
9220 IF K=2 THEN POP @ GOTO 9990
9230 IF K=3 THEN POP @ P=0 @ GOTO 9990
9399 RETURN
9982 IF V0 THEN CURSOFF
```

```
9983 SETWIN @ DISP MESSAGE$(2) @ RETURN
9990 PRINT "sp;" @ PRINTER IS DSP$ @ V5=3 @ STOVAR @ END
```

****

Programme "HORA" (Horlogue astronomique. Necessite KEYWAIT et STRUC1)

==========

```
  10 'HORA':
   - Horloge astronomique.
  20 DELAY 0,0 @ STD
  30 DESTROY ALL
  40 USER OFF @ LC OFF
  50 F=FLAG(1)+2*FLAG(2)
  60 IF NOT F AND NOT FLAG(0) THEN DISP '[E]té ou [H]iver? ' @ K$=KEYWAIT$
  70 IF K$='H' THEN SFLAG 1
  80 IF K$='E' THEN SFLAG 2
  90 REPEAT
 100 REPEAT
 110 D$=DATE$
 120 DISP D$[7,8]&D$[3,6]&D$[1,2]&" "&TIME$
 130 K$=KEY$
 140 UNTIL K$#''
 150 IF K$="E" THEN CALL DECAL(2,F)
 160 IF K$="H" THEN CALL DECAL(1,F)
 170 IF K$="U" THEN CALL DECAL(0,F)
 180 IF K$="R" THEN GOSUB 'REGLAGE'
 190 IF K$="A" THEN GOSUB 'ALARME'
 200 IF K$="J" THEN GOSUB 'JOURJUL'
 205 IF K$="O" THEN OFF
 210 IF K$="S" THEN GOSUB 'TEMPSID'
 220 IF K$="W" THEN GOSUB 'JOURSEM'
 230 IF K$='fD' THEN GOSUB 'DATE'
 240 IF K$='fH' THEN GOSUB 'HEURE'
 250 IF K$='fS' THEN GOSUB 'HORSID'
 260 UNTIL K$='#38'
 270 PUT '#43'
 280 END
```

==========

```
 300 'REGLAGE':
   - Heure, date et paramètres d'alarme.
 310 DESTROY T
 320 DESTROY K
 330 REPEAT
 340 REPEAT
 350 DISP TIME$
 360 T$=KEY$
 370 UNTIL T$#''
 380 IF T$="H" THEN T=3600
 390 IF T$="M" THEN T=60
 400 IF T$="S" THEN T=1
 410 IF T$="-" THEN K=-1
 420 IF T$="+" THEN K=1
```

```
 430 IF T$="D" THEN SETDATE INDATE$
 440 IF T$="A" THEN GOSUB 480
 450 ADJABS T*K
 460 UNTIL T$='#38'
 470 RETURN
 480 INPUT 'fréquence beep? ',STR$(F1);F1
 490 INPUT 'durée beep? ',STR$(D);D
 495 IF FLAG(-25) THEN B$='F' ELSE B$='f'
 500 INPUT 'beep Fort/faible? ',B$;B$
 510 IF B$='F' THEN SFLAG -25 ELSE CFLAG -25
 520 RETURN

 ================================================================
 600 'DATE':
 610 REPEAT
 620 CALL DATE(D$)
 630 DISP D$
 640 T$=KEY$
 650 UNTIL T$#''
 660 RETURN

 ================================================================
 700 'HEURE':
 710 REPEAT
 720 CALL HEURE(H$)
 730 DISP H$
 740 T$=KEY$
 750 UNTIL T$#''
 760 RETURN

 ================================================================
1000 'JOURJUL':
   - Jour julien.
1010 CALL INDATE(A,M,J,H)
1020 CALL JJ(A,M,J,H,U)
1030 FIX 1 @ DISP "Jour julien: "&STR$(J)
1040 PAUSE
1050 RETURN

 ================================================================
1100 'TEMPSID':
   - Temps sidéral.
1110 GOSUB 'JOURJUL'
1120 IF K$='J' THEN RETURN
1130 FIX 4
1140 INPUT "Longitude? ",STR$(-2.2014);L
   - Longitude de Paris: -2.2014
1150 CALL TS(H,U,L,O)
1160 DISP "Temps sidéral: ";STR$(O)
1170 PAUSE
1180 RETURN

 ================================================================
1500 'JOURSEM':
   - Jour de la semaine.
1510 CALL INDATE(A,M,J,H)
1520 CALL JJ(A,M,J,H,U)
1530 CALL JS(U,J$)
```

```
1540 DISP J$
1550 PAUSE
1560 RETURN

============================================================
2000 'HORSID':
   - Horloge sidérale.
2010 FIX 4
2015 REPEAT @ T$=KEY$
2020 CALL DATEL(A,M,J,H,L)
2050 CALL JJ(A,M,J,H,U)
2060 CALL TS(H,U,L,O)
2070 DISP "Temps sidéral: ";STR$(O)
2080 UNTIL T$#''
2090 RETURN

============================================================
2500 'ALARME':
2510 IF NOT F1 THEN F1=1000
2520 IF NOT D THEN D=1
2530 CALL INDATE(A,M,J,H)
2540 CALL JJ(A,M,J2,H,U)
2550 LINPUT 'Message? ';M$
2560 POKE '2F6E7','8'
   - allume l'indicateur d'alarme
2570 CALL DATEL(A,M,J,H,L)
2580 CALL JJ(A,M,J1,H,U)
2590 IF J2-J1<0 THEN RETURN
2600 S=(J2-J1)*24*3600-1
2610 ON TIMER #1,S GOSUB 'SONNERIE'
2620 RETURN

============================================================
2800 'SONNERIE':
2810 POKE '2F6E7','0'
   - éteint (en principe) l'indicateur d'alarme
2820 DISP M$
2830 REPEAT
2840 BEEP F1,D @ T$=KEY$
2850 UNTIL T$#''
2860 OFF TIMER #1
2870 RETURN

============================================================
3000 SUB DECAL(D,F)
   - Décalage horaire.
3010 ADJABS (D-F)*3600
3020 RESET @ SFLAG D @ F=D
3030 END SUB

============================================================
3500 SUB INDATE(A,M,J,H)
   - Saisie de la date.
3510 D$=DATE$ @ T$=TIME$
3520 INPUT "Année? ","19"&D$[1,2];A
3530 INPUT "Mois? ",D$[4,5];M
3540 INPUT "Jour? ",D$[7,8];J
3550 INPUT "Heure? ",T$[1,2]&"."&T$[4,5]&T$[7,8];H
```

```
3560 END SUB

============================================================
4000 SUB DATEL(A,M,J,H,L)
   - Date et longitude.
4010 D$=DATE$ @ T$=TIME$
4020 A=VAL("19"&D$[1,2])
4030 M=VAL(D$[4,5])
4040 J=VAL(D$[7,8])
4050 H=VAL(T$[1,2]&"."&T$[4,5]&T$[7,8])
4060 L=-2.2014
4070 END SUB

============================================================
4500 SUB JJ(A,M,J,H,U)
   - Calcul du jour julien pour une date et une heure données.
4510 A=A-(M<=2) @ M=M+12*(M<=2)
4520 IF A>=1582 THEN V=INT(1%A) @ W=2-V+INT(V/4)
4530 U=INT(365.25*A)+INT(30.6001*(M+1))+J+1720994.5+W
4540 J=U+HR(H)/24
4550 END SUB

============================================================
5000 SUB TS(H,U,L,O)
   - Calcul du temps sidéral.
5010 Q=(U-2415020)/36525
5020 S=.276919398+100.0021359*Q+.000001075*Q^2
5030 S=FP(S)*24
5040 I=S+H*1.002737908
5050 P=I*60-HR(L)*4 @ O=HMS(P/60)
5060 O=O-24*(O>=24)
5070 END SUB

============================================================
5500 SUB DATE(D$)
   - Renvoie la date sous forme de chaîne.
5510 CALL DATEL(A,M,J,H,L)

============================================================
5520 DATA Janvier,Février,Mars,Avril,Mai,Juin,Juillet,Août,Septembre,Octobre,Novembre,Décembre

============================================================
5530 RESTORE 5520
5540 FOR I=1 TO M
5550 READ M$
5560 NEXT I
5570 D$=STR$(J)&" "&M$&" "&STR$(A)[3,4]
5580 CALL JJ(A,M,J,H,U)
5590 CALL JS(U,J$)
5600 D$=J$&" "&D$
5610 END SUB

============================================================
6000 SUB HEURE(H$)
   - Renvoie l' heure sous forme de chaîne.
6010 H$=TIME$
6020 H$=H$[1,2]&"h "&H$[4,5]&"mn "&H$[7,8]&"s"
6030 END SUB
```

```
============================================================
6500 SUB JS(U,J$)
   - Détermine le jour de la semaine.
6510 J=MOD(U+2,7)

============================================================
6520 DATA Dimanche,Lundi,Mardi,Mercredi,Jeudi,Vendredi,Samedi

============================================================
6530 FOR I=0 TO J
6540 READ J$
6550 NEXT I
6560 END SUB

============================================================
7000 END ALL
```

```
*****************************************************************************************************

Programme "DIFF" (Recherche de derivees symboliques)


  10 DESTROY ALL
  30 DIM K$[91]
  40 INPUT 'f(x)=',F1$;K$
  50 K$=UPRC$(K$) @ L=LEN(K$)
  60 DIM F$[L],F1$[MIN(91,10*L)]
  70 F$=K$ @ CALL DIFF(F$,F1$)
  80 IF FLAG(0) THEN BEEP 4000,.01 @ DISP "f'=";F1$; @ PAUSE @ GOTO 40
  90 BEEP 1400,.05
 100 IF FLAG(1) THEN DISP 'Parenthesis Omitted'
 110 IF FLAG(2) THEN DISP 'Syntax in Formula'
 120 IF FLAG(3) THEN DISP 'Formula too Complex'
 130 F1$=K$ @ WAIT 3 @ GOTO 40

============================================================
 160 SUB MINPRIO(T$,P,O,L)
 210 INTEGER I,K
 220 DIM K$[1]
 230 CFLAG 1 @ L=LEN(T$) @ P=0 @ K=0 @ O=6
 240 FOR I=1 TO L
 250 K$=T$[I,I]
 260 IF K$='(' THEN K=K+1 @ GOTO 340
 270 IF K$=')' THEN K=K-1 @ GOTO 340
 280 IF K#0 THEN 340
 290 IF K$='+' THEN O=1 @ P=I @ GOTO 340
 300 IF K$='-' THEN O=2 @ P=I @ GOTO 340
 310 IF K$='*' AND (O>2 OR P=1) THEN O=3 @ P=I @ GOTO 340
 320 IF K$='/' AND (O>2 OR P=1) THEN O=4 @ P=I @ GOTO 340
 330 IF K$='^' AND (O=6 OR P=1) THEN O=5 @ P=I
 340 NEXT I
 350 IF K#0 THEN CFLAG 0 @ SFLAG 1
 360 END SUB

============================================================
```

```
380 SUB DIFF(F$,F1$)
410 ON ERROR GOTO 1670
420 INTEGER P,D,L,O1,O2

============================================================
430 DEF FNK$[91](T$,O)
440 CALL MINPRIO(T$,P,O2,L)
450 IF O2<O OR T$[1,1]='-' THEN FNK$='('&T$&')' ELSE FNK$=T$
460 END DEF

============================================================
470 DEF FNN(T$)

490 INTEGER I,L @ DIM K$[1]
500 CFLAG 4 @ I=1 @ L=LEN(T$) @ IF L>12 THEN FNN=0 @ END
510 IF T$[1,1]='-' OR T$[1,1]='+' THEN I=2 ELSE I=1
520 K$=T$[I,I]
530 IF K$#'.' THEN 550
540 IF FLAG(4) THEN FNN=0 @ END ELSE SFLAG 4 @ GOTO 560
550 IF '0'>K$ OR K$>'9' THEN FNN=0 @ END
560 IF I<L THEN I=I+1 @ GOTO 520
570 FNN=1
580 END DEF

============================================================
590 SFLAG 0 @ CFLAG 2,3

============================================================
600 'START':
610 CALL MINPRIO(F$,P,O,L)
620 IF NOT FLAG(0) THEN END
630 IF P=L OR P=1 AND O>2 THEN CFLAG 0 @ SFLAG 2 @ END
640 IF P>1 THEN 'ABLUV'

660 IF O#6 THEN F$=F$[2] @ L=L-1
670 IF F$#'X' THEN 700
680 IF O=2 THEN F$='-X' @ F1$='-1' ELSE F1$='1'
690 END

============================================================
700 IF ('A'>F$ OR F$>'Z' OR L>1 AND F$#'PI') AND NOT FNN(F$) THEN 740
710 F1$='0'
720 IF O=2 THEN F$='-'&F$
730 END

============================================================
740 P=POS(F$,'(')
750 IF P#1 THEN 'SUCHEFKT'
760 F$=F$[2,L-1] @ L=L-2
770 IF O#2 THEN 'START'
780 CALL DIFF(F$,F1$)
790 IF F1$#'0' THEN F$='-'&FNK$(F$,3) @ F1$='-'&FNK$(F1$,3)
800 END

============================================================
810 'SUCHEFKT':
820 IF P=0 OR P=L THEN CFLAG 0 @ SFLAG 2 @ END
830 DIM U$[L-P],U1$[(L-P)*10],V$[P-1]
```

```
840 U$=F$[P+1,L-1] @ CALL DIFF(U$,U1$)
850 V$=F$[1,P-1]
860 IF 'SIN'=V$ THEN O1=3 @ F1$='COS('&U$&')' @ GOTO 'A'
870 IF 'COS'=V$ THEN O1=4 @ F1$='SIN('&U$&')' @ GOTO 'A'
880 IF 'TAN'=V$ THEN O1=1 @ F1$='COS('&U$&')^2' @ GOTO 'A'
890 IF 'LN'=V$ THEN O1=1 @ F1$=FNK$(U$,5) @ GOTO 'A'
900 IF 'EXP'=V$ THEN O1=3 @ F1$='EXP('&U$&')' @ GOTO 'A'
910 IF 'SQR'=V$ THEN O1=1 @ F1$='.5*SQR('&U$&')' @ GOTO 'A'
920 IF 'ASIN'=V$ THEN O1=1 @ F1$='SQR(1-'&U$&')' @ GOTO 'A'
930 IF 'ACOS'=V$ THEN O1=2 @ F1$='SQR(1-'&U$&')' @ GOTO 'A'
940 IF 'ATAN'=V$ THEN O1=1 @ F1$='1+'&FNK$(U$,5)&'^2)' @ GOTO 'A'
950 IF 'COT'=V$ THEN O1=2 @ F1$='SIN('&U$&')^2' @ GOTO 'A'
960 IF 'SINH'=V$ THEN O1=3 @ F1$='COSH('&U$&')' @ GOTO 'A'
970 IF 'COSH'=V$ THEN O1=3 @ F1$='SINH('&U$&')' @ GOTO 'A'
980 IF 'TANH'=V$ THEN O1=1 @ F1$='COSH('&U$&')^2' @ GOTO 'A'
990 IF 'COTH'=V$ THEN O1=2 @ F1$='(SINH('&U$&')^2' @ GOTO 'A'
1000 IF 'ASINH'=V$ THEN O1=1 @ F1$='SQR(1+'&FNK$(U$,5)&'^2)' @ GOTO 'A'
1010 IF 'ACOSH'=V$ THEN O1=1 @ F1$='SQR(1-'&FNK$(U$,5)&'^2)' @ GOTO 'A'
1020 IF 'ATANH'=V$ THEN O1=1 @ F1$='1-'&FNK$(U$,5)&'^2' @ GOTO 'A'
1030 IF 'ACOT'=V$ THEN O1=2 @ F1$='1+'&FNK$(U$,5)&'^2' @ GOTO 'A'
1040 IF 'ACOTH'=V$ THEN O1=1 @ F1$='1+'&FNK$(U$,5)&'^2' @ GOTO 'A'

================================================================
1050 'A':
1060 IF U1$='0' THEN F1$='0' @ GOTO 'EF'
1070 IF O1<3 THEN 1110
1080 IF U1$#'1' THEN F1$=FNK$(U1$,3)&'*'&F1$
1090 IF O1=4 THEN F1$='-'&FNK$(F1$,3)
1100 GOTO 'EF'
1110 IF O1=1 THEN F1$=FNK$(U1$,3)&'/'&FNK$(F1$,5) ELSE F1$='-'&FNK$(U1$,3)&'/'&FNK$(F1$,5)

================================================================
1120 'EF': IF O=2 THEN F$='-'&F$ @ F1$='-'&FNK$(F1$,3)
1130 END

================================================================
1140 'ABLUV':
1150 DIM U$[P-1],V$[L-P],U1$[(P-1)*10],V1$[(L-P)*10]
1160 U$=F$[1,P-1] @ CALL DIFF(U$,U1$)
1170 IF NOT FLAG(0) THEN END
1180 V$=F$[P+1,L] @ CALL DIFF(V$,V1$)
1190 ON O GOTO 'ADD','SUB','MUL','DIV','POT'

================================================================
1200 'ADD': 'SUB':
1210 IF V1$='0' THEN 1280
1220 IF U1$='0' THEN 1260
1230 IF O=2 THEN F1$=U1$&'-'&FNK$(V1$,3) @ END
1240 F1$=U1$&'+' @ IF V1$[1,1]='-' THEN F1$=F1$&'('&V1$&')' ELSE F1$=F1$&V1$
1250 END

================================================================
1260 IF O=2 THEN F1$='-'&FNK$(V1$,3) ELSE F1$=V1$
1270 END

================================================================
1280 IF U1$='0' THEN F1$='0' ELSE F1$=U1$
1290 END
```

```
============================================================
1300 'MUL': 'DIV':
1310 IF V1$='0' THEN 1420
1320 IF U1$='0' THEN 1390
1330 IF U1$='1' THEN F1$=V$ ELSE F1$=FNK$(U1$,3)&'*'&FNK$(V$,3)
1340 IF V1$='1' THEN 1370
1350 F1$=F1$&CHR$(43+(O-3)*2)&FNK$(U$,3)&'*'&FNK$(V1$,3)
1360 GOTO 1430
1370 IF O=4 THEN F1$=F1$&'-'&FNK$(U$,3) ELSE F1$=F1$&'+'&U$
1380 GOTO 1430
1390 IF O=4 THEN F1$='-' @ U$=FNK$(U$,3) ELSE F1$=''
1400 IF V1$='1' THEN F1$=F1$&U$ ELSE F1$=F1$&U$&'*'&FNK$(V1$,3)
1410 GOTO 1430
1420 IF U1$='0' THEN F1$='0' ELSE IF U1$='1' THEN F1$=V$ ELSE F1$=FNK$(U1$,3)&'*'&FNK$(V$,3)
1430 IF O=4 AND F1$#'0' THEN F1$=FNK$(F1$,3)&'/'&FNK$(V$,5)&'^2'
1440 END

============================================================
1450 'POT':
1460 IF V1$='0' THEN 1580
1470 IF U1$#'0' THEN 1520
1480 IF U$='E' THEN F1$='E^'&FNK$(V$,6) @ GOTO 1500
1490 F1$=FNK$(U$,5)&'^'&FNK$(V$,6)&'*LN('&U$&')'
1500 IF V1$#'1' THEN F1$=F1$&'*'&FNK$(V1$,3)
1510 END

============================================================
1520 F1$=FNK$(U$,5)&'^'&FNK$(V$,6)&'*('
1530 IF V1$#'1' THEN F1$=F1$&FNK$(V1$,3)&'*'
1540 F1$=F1$&'LN('&U$&')+'
1550 IF U$=V$ THEN F1$=F1$&'1' ELSE F1$=F1$&FNK$(V$,5) @ IF U$#'1' THEN F1$=F1$&'/'&FNK$(U$,5)
1560 F1$=F1$&')'
1570 END

============================================================
1580 IF U1$='0' THEN F1$='0' @ END
1590 F1$=FNK$(V$,3)&'*'&FNK$(U$,5)&'^'
1600 IF FNN(V$) THEN 1620
1610 F1$=F1$&'('&V$&'-1)' @ END
1620 IF V$='1' THEN F1$='1' @ END
1630 IF V$='0' THEN F1$='0' @ END
1640 V1$=STR$(VAL(V$)-1)
1650 IF V1$='1' THEN F1$=V$&'*'&FNK$(U$,3) ELSE F1$=F1$&FNK$(V1$,3)
1660 END

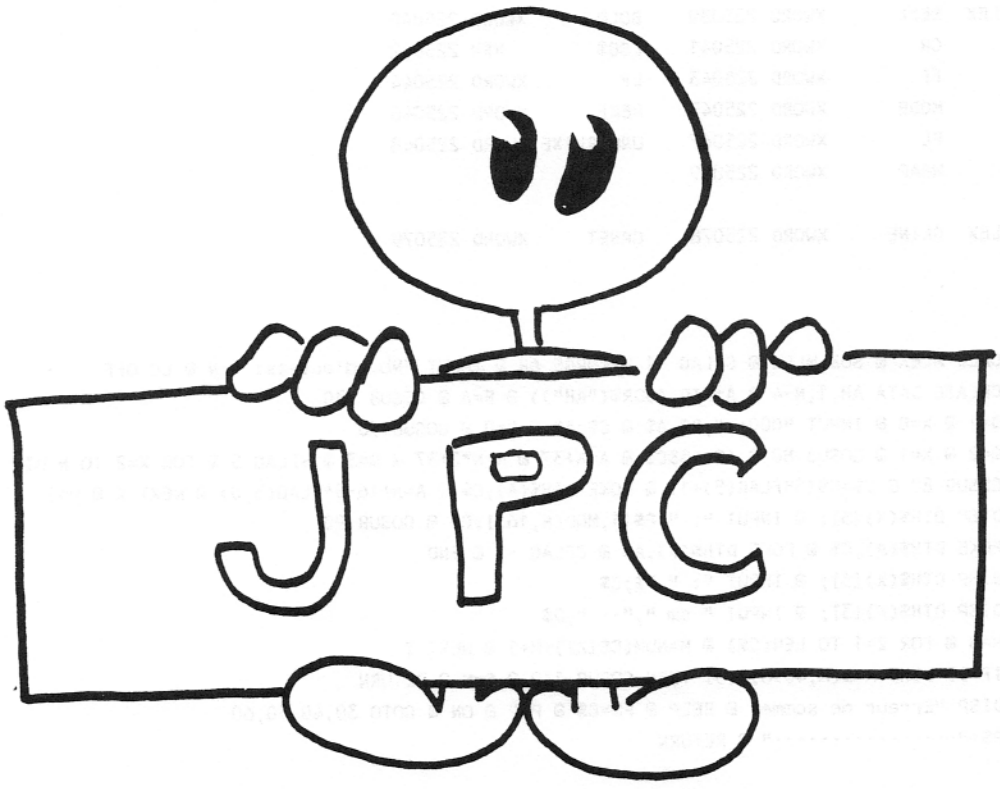
============================================================
1670 IF ERRN=24 OR ERRN=37 THEN SFLAG 3 ELSE BEEP 1200,.1 @ DISP 'Err L';ERRL;ERRM$ @ WAIT 3
1680 CFLAG 0
1690 END SUB
```

****

Programme "CONNETT" (Trace des graphes de Connett. Necessite PRINTLEX et GRAPHLEX)

- Ce programme imprime des graphes de Connettt sur une ThinkJet
  Les graphes de Connett sont expliqués dans un article de A. K. Dewdney
  Référence : Scientific American, traduction italienne, Novembre 1986, Numéro 219
  Par Stefano Tendon (PC241,SIG9)
  Cantone Delle Asse 5
  29100 Piacenza - Italie

```
100 PRINTER IS :PRINTER @ PWIDTH INF
110 REAL X,Y,L,A,B,C,D,E,F @ INTEGER I,J,K,R
120 INPUT "R, X, Y, L ? ";R,X,Y,L
130 DIM G$[80],B$[80] @ FOR K=1 TO 80 @ B$[K,K]=CHR$(0) @ NEXT K
140 PRINT ESC$("*r640S");ESC$("*rA");
150 A=X*X+Y*Y
160 B=2*R*L*X
170 C=2*R*L*Y
180 D=L*L
190 E=R*R
200 FOR I=1 TO R
210 G$=B$
220 F=(B+D*I)*I
230 FOR J=1 TO R
240 IF MOD(A+(F+(C+D*J)*J) DIV E,2) THEN GPSET J
250 NEXT J
260 PRINT ESC$("*b80W");G$;
270 NEXT I
280 PRINT ESC$("*rB")
290 LF 2 @ PRINT R;X;Y;L @ LF 4
300 END
```



LE COIN DES LHEX

Comme de coutume, cette rubrique contient la liste des codes hexadécimaux des fichiers Lex parus ce mois-ci.

Rappelons ce qu'est un fichier Lex : c'est un programme pour le HP-71, en assembleur, qui apporte de nouvelles fonctions. Celles-ci sont utilisables directement, ou dans des programmes Basic.

Pour bénéficier de ces nouvelles fonctions, vous n'avez pas besoin de programmer vous-même en assembleur, ni de posséder un module Forth/Assembleur.

Il suffit de recopier le petit programme basic "MAKELEX" ci-dessous, de le lancer et de recopier les codes du fichier Lex désiré. Quand vous avez fini, les nouvelles fonctions sont accessibles, après avoir éteint et rallumé votre HP-71.

Si l'erreur "Erreur de somme" apparaît, vérifiez la ligne que vous avez introduite.

Vous trouverz donc, outre le Lex CHARLEX nécessaire à la rédaction de votre article (voir "Ah ! Vous écrivez !"), les Lex utilisés par les programmes Basic.

```
CHARLEX

KEYWAIT   KEYWAIT$    XFN  82001

STRUC1    END WHILE XWORD 225066    WHILE     XWORD 225067
          REPEAT    XWORD 225068    UNTIL     XWORD 225069
          LEAVE     XWORD 225070

PRINTLEX  BELL      XWORD 225039    BOLD      XWORD 225040
          CR        XWORD 225041    ESC$        XFN 225042
          FF        XWORD 225043    LF        XWORD 225044
          MODE      XWORD 225045    PERF      XWORD 225046
          PL        XWORD 225047    UNDERLINE XWORD 225048
          WRAP      XWORD 225049

GRAPHLEX  GLINE     XWORD 225078    GPSET     XWORD 225079
```

```
 10 CALL MLEX @ SUB MLEX @ SFLAG -1 @ PURGE AH @ INPUT "Nb. d'octets: ";N @ LC OFF
 20 CREATE DATA AH,1,N-4 @ A=HTD(ADDR$("AH")) @ B=A @ GOSUB 130
 30 Q=1 @ X=0 @ INPUT "000: ",P$;A$ @ C$=A$ @ S=0 @ GOSUB 90
 40 Q=2 @ X=1 @ GOSUB 80 @ A$=A$&C$ @ A=A+37 @ N=N*2+37 @ Q=3 @ SFLAG 5 @ FOR X=2 TO N DIV 16-1
 50 GOSUB 80 @ C$=C$[5*FLAG(5)+1] @ POKE DTH$(A),C$ @ A=A+16-5*FLAG(5,0) @ NEXT X @ Q=4
 60 DISP DTH$(X)[3]; @ INPUT ": ",P$[1,MOD(N,16)];C$ @ GOSUB 90
 70 POKE DTH$(A),C$ @ POKE DTH$(B),A$ @ CFLAG -1 @ END
 80 DISP DTH$(X)[3]; @ INPUT ": ",P$;C$
 90 DISP DTH$(X)[3]; @ INPUT " sm ","---";D$
100 M=S @ FOR Z=1 TO LEN(C$) @ M=NUM(C$[Z])+M+1 @ NEXT Z
110 IF D$=DTH$(MOD(M,4096))[3] THEN GOSUB 130 @ S=M @ RETURN
120 DISP "Erreur de somme" @ BEEP @ P$=C$ @ POP @ ON Q GOTO 30,40,50,60
130 P$="----------------" @ RETURN
```

```
CHARLEX  ID#E1  624 octets

     0123456789ABCDEF sm

000: 34841425C4548502 35E
001: 802E008051624078 6B6
002: 5E4001E000000000 9FA
003: FE0000000800001F D54
004: F31BF961400032BF 0E7
005: 38F14A11DB10AD23 481
006: 07D532BFB8FD7911 834
007: 11AD754D7A101743 BB7
008: 11014D1CB15D0000 F22
009: 71450375FF864834 29F
00A: 5655581008355654 5F6
00B: 5810002455565870 943
00C: 0026555658700836 C95
00D: 5556581008364545 FEB
00E: 4A30000A49724000 33E
00F: 0808094A2C180814 6A7
010: A464242008355455 A01
011: 581000054C714000 D47
012: 0C3142404C700832 0A3
013: 41414A70002078A0 3FB
014: 2F30000000000000 726
015: 0000000000000000 A36
016: 0000000000000000 D46
017: 0000000000000000 056
018: 0000000000000000 366
019: 0000000000000000 676
01A: 0000000000000000 986
01B: 0000000000000000 C96
01C: 0000000000000000 FA6
01D: 0000000000000000 2B6
01E: 0000000000000000 5C6
01F: 0000000000000000 8D6
020: 0000000000000000 BE6
021: 000000000000080C F11
022: 1A28080008080A2C 27B
023: 180008040E340800 5C4
024: 08001E3018000000 8FE
025: 0000000000000000 C0E
026: 0000000000000000 F1E
027: 0000000000000000 22E
028: 0201000000010200 544
029: 0000000201020000 859
02A: 0001000100000002 B6D
02B: 0102010000000000 E81
02C: 0000000000000000 191
02D: 045E755142400101 4DD
02E: 0101010000000000 7F0
02F: 0000000000000000 B00
030: 0000070507000000 E23
031: 000000000083444C4 161
032: 44400D7901112D70 4C1
033: 050D750509700000 80B
034: 0D70000000384540 B4E
035: 4020014E322E3140 EA2
036: 084E794142400000 1F2
037: 000000000002E4559 530
038: 3200000000000000 845
039: 0000000000000026 B5D
03A: 5556587008365556 EBC
03B: 5810083645464830 20D
03C: 0832414248700024 54E
03D: 5655587008345655 8AB
03E: 5810083446454830 BFA
03F: 0C3042414C700024 F4F
040: 5556587008355654 2AC
041: 5810083546444830 5FB
042: 0C3142404C700025 951
043: 5455587008355455 CAB
044: 5810083544454830 FF9
045: 0C3140414C700875 35B
046: 14141870000A4972 6AC
047: 40000E3159454E30 A0C
048: 0C7A0F7949400024 D84
049: 5554587000084A71 0E0
04A: 40000C523A262D10 441
04B: 0424587458400875 798
04C: 1415187000094A70 AE8
04D: 4000083544454830 E2C
04E: 0C3140414C300C74 194
04F: 5655545000054C71 4EB
050: 40000            5E4

KEYWAIT  ID#52  55 octets

     0123456789ABCDEF sm

000: B454957514944502 366
001: 802E009051624078 6BF
002: 3700025101000000 9E2
003: F710000000000000 D10
004: 0E1000FFB4549575 09B
005: 14944542101FF001 3FB
006: 361081371098F2C6 76B
007: 0045111913511813 AA7
008: 48D8ACA18F127006 E43
009: BDF              F12

STRUC1   ID#E1  656 octets

     0123456789ABCDEF sm

000: 3545255534130202 341
001: 802E000151624078 692
002: 425001E246400000 9D3
003: FB30087000000000 D1D
004: 06D000D2309F100D 094
005: 6101A300D520C030 3F1
006: 0D900FA300D554E4 783
007: 44249C4541465546 AE9
008: 4B25540554144544 E43
009: 955E44594C454975 1CA
00A: 8494C454341FF306 550
00B: 001304758494C454 8AD
00C: C81408E6F6022554 C2E
00D: 05541445C3150ECE FB3
00E: 4C454146554CF160 330
00F: BBE6F60254E44402 6C9
010: 758494C454CFFD04 A75
011: 008E3001361B698F DEF
012: 21441367D124E02F 163
013: 3092094371203130 4A2
014: 22311E208DA93908 81C
015: 7D1220321088FAB8 BA3
016: 11D81378B70DC98B F47
017: B9CDB1B198F21441 2F2
018: 3671814311B698F2 666
019: 1461366E4013610A 9C6
01A: 1B198F21422F3097 D45
01B: E8111A13686F927A 0D7
01C: 2042213610A8F95E 44E
01D: F011A1348FB7EF01 7F2
01E: 121308D7E47084A8 B71
01F: 6F0086D0087A001F EF6
020: 0B7F22030214B0E0 26A
021: 6A0C1FD55F214713 601
022: 517E143F434412E0 978
023: 8A2400203403008F CD1
024: 200780113679B058 02C
025: 213610A1B198F214 39C
026: 22F30977D011A136 711
027: 8D84A80D2E610A86 AB3
028: D011F765F2147661 E32
029: 020321088FAB8111 19E
02A: 37C2D72031FE8F99 549
02B: A804A03160658E16 8C0
02C: 1331E34231521916 C12
02D: E011AE610A6D1022 F96
02E: 312423916F011ACE 316
02F: 8AAD010A713069AF 6B7
030: 16365BE8F681F08F A64
031: 19DB0042E5A091CC E05
032: 011091C400302183 149
033: D014AD8132C01300 4BC
034: 18D31F808DE3F801 85E
035: 361B198F214401FD BE8
036: 1003D1001361B698 F45
037: F21447FCF4E02F30 2F0
038: A20943A0314063BD 66B
039: 87D1220321088FAB 9F2
03A: 811D81378B7DDC98 DA0
03B: BB6DDB1B198F2144 15A
03C: 1641461367A3F560 4C6
03D: 641E1B198F214210 83B
03E: 22F30A68CD701009 BBF
03F: 01001322F30A714F F24
040: 816136696E7E0009 29C
041: E000773F7C2F4412 625
042: F30A94321309943A 99F
043: 031506A0DD2E610A D26
044: 86D011F765F21476 0AC
```

```
045: 61020321088FAB81 41D
046: 1137C2D71B198F21 7A7
047: 461362031FE8F99A B3D
048: 804A0315066BC161 EAC
049: 331E24231521912D 20C
04A: 3223154239120322 548
04B: 3144239127122313 888
04C: 423912A0716E68AF C16
04D: 11AE610A6FEF11AC FD3
04E: E10A8AE1E704E6BC 399
04F: C8D30350038FA2C2 72C
050: 0FE1E34E0000208D AB7
051: 5303018503397584 E07
052: 94C454298F324508 183
053: FCE2508DE6A208D3 541
054: 9450F            65E

PRINTLEX ID#E1  671 octets

     0123456789ABCDEF sm

000: 052594E445C45485 378
001: 802E001151624078 6CA
002: 245001E721300000 A08
003: F170000000000000 D36
004: 0F0200DB0071200D 0A2
005: 610D2200DD105420 405
006: 0F820BB200DF20BC 7AC
007: 200D63061300D140 AFE
008: 16300DC4050400D3 E63
009: 5008400D6602A400 1BB
00A: D72454C4C472724F 54F
00B: 4C44482334259275 8B4
00C: 4353442A236464B2 C1B
00D: 3C464C27D4F44454 FAE
00E: D2705542564E2305 319
00F: C4F2F55E4445425C 6BC
010: 494E403775251405 A1D
011: 131FF038F8BF3031 DB6
012: 54966811718FA2C2 13C
013: 00E2EF1EDDF008DB 510
014: 2E208D271308FE7A 8AD
015: 2041F7BEF68108FE C60
016: 7A204FD79DF8D6CA 038
017: 208DE6A208D30350 3BB
018: 3354021818FDFC20 743
019: 14B8DCF2508D0745 AE2
01A: 0844660085420048 E2D
01B: F73321F0831D0310 19D
01C: 48D393901311C414 507
01D: 7D5137C910807135 877
01E: E6D7700007D534B1 BFE
01F: 000C906D606D015B F7C
020: 001700007DA34C10 2DB
021: 00C2061180613710 622
022: 8012086490118060 968
023: 18D84A8014A310E9 CF2
024: 62000133FFFACEFF 0B8
025: 7D4F170C3FFFECEF 4B0
026: F73DF490753F1F07 85A
027: E2F1E030FFFA9EFF C4D
028: 74006CAF1F659F2A FFC
029: E214D7F0F1D00140 389
02A: 194E821371091351 6E5
02B: C12031B114D8F064 A63
02C: A18408D7B1811360 DDA
02D: 68F83DB013713413 156
02E: 51C110AC2D81098F 4EB
02F: 261B111A1351C111 84A
030: 913418131B114C07 BAB
031: 13457A36EFFAFDFF F91
032: 746F797E1C06EE0B 34A
033: 6EFF71EFF14A8F50 723
034: 450D0570E45318F8 AA4
035: 71F08FBC6315FC10 E4E
036: 1732F6D00101783E 1C4
037: 1A0111CC50F68816 547
038: A9071EFFAEDFF776 934
039: 0AF2A969765E0431 CC5
03A: 03A6A8FE1DE08F12 07E
03B: DE03A5B162B60035 40B
03C: A728FC2B9015CA13 7B1
03D: 6066DCDCBDFFE4DF BB6
03E: F735ED031C4540E4 F51
03F: 7D206C3E048F871F 2F7
040: 08F2EA21044F0AF2 697
041: 329999F2008D91FB A36
042: 0AE7D81B178F2367 DDA
043: B162C615C6166815 158
044: 815310322D4D1A6A 4D6
045: 1481618110D5DE20 849
046: AEB14C161310114C BCE
047: 1361340618E13606 F24
048: 672D61DFF4DCFF14 2FA
049: A8F504505D1720DB 690
04A: B162C67323076363 9FC
04B: 6613C47E3F10117F D88
04C: 1331BE95F2146859 10E
04D: 8BE618491318FDF8 4BE
04E: E07B1F1331333105 82A
04F: 7D2F879D01113164 BAA
050: 7D1F6C2D87CFFA0C F8D
051: FF722D4F0748C4B1 33F
052: 624604777C4B1624 6B0
053: 64406CFF2FBFF77F A8D
054: C411795C5B162371 E09
055: 3347A4C5B1623703 17F
056: 34F              22F

GRAPHLEX ID#E1  302 octets

     0123456789ABCDEF sm

000: 7425140584C45485 365
001: 802E002151624078 6B8
002: 062001EE4F400000 A19
003: F020000000000000 D41
004: 095100DD00EC000D 0C4
005: 974C494E454E4974 45B
006: 05355445F41FF8FB 801
007: C63157017F018D91 B7F
008: FB08D074508DE6A2 F2D
009: 0305A0E10A8FD963 2C1
00A: 0781011A90A218FD 643
00B: C6305F011A49D8D2 9DC
00C: 71308DC5E208F871 D70
00D: F032760AB58F155F 10F
00E: 041216A142136E21 467
00F: 34D015A3C4C4C416 7F8
010: 30131358D3939057 B58
011: FFF77FFF7FAF1007 F46
012: A4F1188AC70E4590 2E5
013: 8BA40DA74006D31C 685
014: CD8AF0D481C81C81 A44
015: CD23070EF1ED31F4 DFA
016: E2C6132C2134AE2B 197
017: 666600A66CD5AFAE 555
018: 514E0E6D14C13001 8D0
019: 3FEFFCFEFF7D2F10 CD8
01A: 21361083051B178F 03E
01B: 2631010971BE1401 399
01C: 64119A0E5CE8AC40 741
01D: E4118134101706F1 AA2
01E: F088F21111478AE7 E34
01F: 0E6145CACC11A8BA 1F1
020: F0111E2E61451121 55B
021: 00792F1C4147111C 8CD
022: A1011188BEF41C41 C5C
023: 4710A1C4D2E610B1 FE6
024: 1179FE111E410111 353
025: 88BE52113E410311 6C5
026: A8B2BD111147CA10 A5C
027: 11188B21C8D84A80 DE8
028: F                E2F
```

PPC PARIS CHAPTER

Association régie par la loi de 1901, enregistrée
à Paris le 2 décembre 1982 sous le numéro 82/3240

BULLETIN D'ADHESION

Nom |_|_|_|

Le Journal JPC est le bulletin de liaison entre les membres de l'Association "PPC-PC", régie par la loi de 1901. Le Club est éditeur du JPC, et son siège social est au 56, rue Jean-Jacques Rouseau, 75001 Paris.

La maquette de ce numéro a été préparée et réalisée par Pierre David, Jean-Jacques Dhénin et Janick Taillandier, grâce à un système comprenant un HP71B, deux lecteurs de disquettes HP9114A, une imprimante HP2225B et une imprimante LaserJet.

Numéro ISSN : 0762 - 381X

ENGLISH SUMMARY

JPC 44 - MAY 1987

Let us begin with some news about our Club.

Those who have participated in JPCLEX Eprom will receive their module and the documentation for JPCLEX with this issue of JPC. The User's Manual has more than 200 pages. An english version will be available as soon as possible. It is worth noting that all this operation was concluded without any delay for the Journal.

We are sorry to be obliged to slightly increase th subscription price to 350 FRF (300 FRF for students). As in the past, the price is the same for "local" and foreign members.

Also, please note our new address :

PPC Paris Chapter
B.P. 604
75028 Paris Cedex 01
France

In this issue of JPC you will find :

For the HP-28C, a program to draw function defined by :
x = f(t)
y = f(t)
This very nice program is a good illustration of the graphic capabilities of the HP-28C.

The HP-41 program is a new version of a program first published in JPC 36. It computes the price of french railways ticket. The HP-71 program is given with other Basic programs.

The HP-75 program is a major achievement : it allows you to plot, from VisiCalc, various graphs of the data in the worksheet. You have pie, line or bar graphs and may mix 1, 2 or 4 graphs on the same page. It works with HP-GL devices such as HP-7470 plotter or Pac Screen video interface.

The first article on the HP-71 is an "object extension" for HP-71 Forth. This work is based on an article by Dick Pountain published in *Byte* (August 1987).

The program HORA provides various time and date functions.

The programs described after this were originally published in *Prisma*, the journal of the German Club (CCD). They allow you to compute symbolic differentiation "a la" HP-28C. Thanks to the authors Peter Ehrenberg and Volker Klann for allowing us to publish their work.

Finally, using the program by Stefano Tendon, you will be able to plot Connet's graphs. The algorithm was described in the italian translation of *Scientific American*. Stefano has implemented and optimized it. He explains us what he has done.

Happy JPC reading.