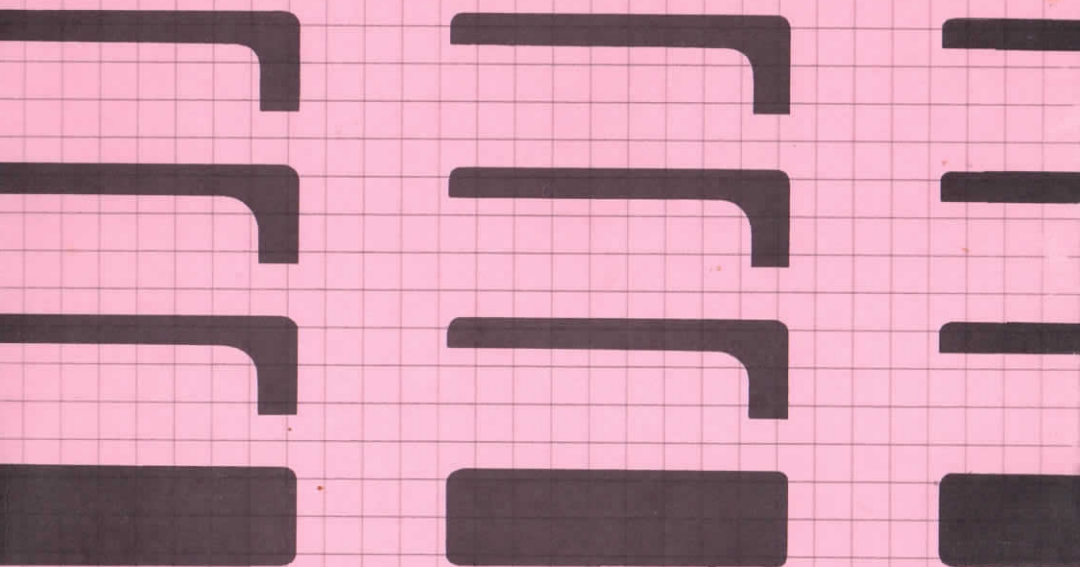


MICRO REVUE

LA REVUE DE L'INFORMATIQUE PORTABLE



1986 Bimestriel

Edité par PPC-T

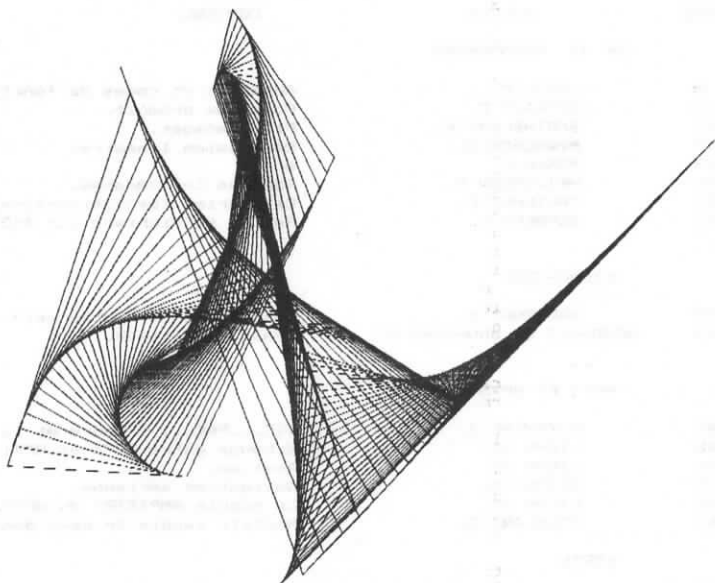
°11 MARS-AVRIL 1986

50 F.

TABLE DES MATIERES

PAGE	AUTEUR	CONTENU
HP 41 PROGRAMMES		
8	LAVILLE L.	Adresses et codes de fonctions.
10	DEVEAUX F.	Polynôme produit.
12	BARTHELEMY P.	File manager.
17	MONACHON D.	Régression linéaire.
18	MOSER F.	Arab.
19	HEILBRONN P.	Romains <--->Arabes.
22	CAHIERRE P.	Calendrier revolutionnaire.
23	SERGEANT E.	Resize de fichiers sur 41C-CV.
MICRO-CODE		
37	VAUDENAY S.	Transfert ROM-RAM au sein d'un MLDL.
41	GOUBAULT DE BRUGIERE A.	Desas + Toulmath.
HP71 ET HP75		
46	DUTERTRE J.	TIC...PAC... l'heure et la date.
48	LIEHN JC.	Echange entre HP75 et HP71.
51	LIEHN JC.	Analyse.
55	SIMON JM.	Navigation aérienne.
59	LIEHN JC.	Le module AMPISTAT du HP71.
61	TOUBLANC G.	Rectili : angle de deux demi plans.
FORTH		
64	PETREMANN M.	Forth et le langage machine.
HP 1X		
76	SERGEANT E.	Calcul de racines par la methode de BAIRSTOW.
LE MAGAZINE DU CLUB		
82	LES JOURNAUX-SUR LE MARCHE	
84	NOUVELLES BREVES-PETITES ANNONCES	
86	REACTIONS	
89	S.O.S	
90	ARGUS DE L'OCCASION HP	
92	ORGANISATION PPC-T	
93	BULLETIN D'ADHESION	

TABIE DES MATIERES



MICRO-REVUE

(T84)

EDITORIAL

Chers amis,

Les clubs sont-ils en crise ? Certains indices peuvent le laisser penser. L'arrivée des journaux se fait de plus en plus tardive, et MICRO-REVUE se conduit plutôt bien sur ce sujet. Nous n'avons reçu de PPC-T que le numéro de Décembre 85 (nous sommes en Avril), CHHU ne donne pas de nouvelles, le club anglais n'a rien fait depuis l'abandon de son éditeur, sauf publier une liste des adhérents, JPC n'a plus reparu depuis le numéro de Décembre-Janvier.

Nous même avons été obligés d'augmenter la cotisation, et malgré cela nous avons du mal à joindre les deux bouts, d'où une réduction que nous espérons provisoire de notre nombre de pages.

Un examen, sans réelle prétention statistique, de notre taux de renouvellement de cotisation a fait apparaître un taux de renouvellement en baisse au cours du dernier semestre de 1985, pour se stabiliser au niveau de 50%. Comme il y a peu de nouveaux adhérents, cela se traduit par une baisse rapide du nombre de nos adhérents (environ 310 à jour à cette date).

Sur quelques bulletins de non réabonnement reçus, je relève 1 mécontent, 5 qui ne s'intéressent plus au matériel HP, 4 sans motif indiqué, 5 utilisateurs de compatibles PC. Très peu de critiques du contenu de la revue, quelques-unes viennent d'utilisateurs de HP-41 qui se trouvent délaissés (un peu à tort, il me semble).

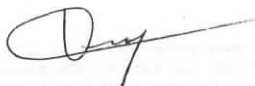
Cette situation n'est pas sans m'inquiéter. Si nous faisons du mauvais travail, nous pourrions nous améliorer. Si c'est la clientèle qui disparaît... que faire ? Il nous semble que plusieurs phénomènes se télescopent :

- * Le prix des calculatrices, en particulier HP, n'a pas diminué à proportion du reste de matériel informatique. Du coup un compatible PC revient au même prix qu'un HP-71. Bien sûr, ce n'est pas le même usage, mais tous ceux qui se tournaient vers les calculatrices uniquement pour leur prix n'y trouvent plus leur compte.

- * L'aspect "découverte" de la programmation synthétique de la HP-41 a motivé des générations de programmeurs, mais il n'y a plus grand chose à découvrir de ce côté. Le HP-71 est une machine ouverte et son examen n'est qu'une question d'argent. Le rôle des clubs est moins évident.

Si cette situation continue, elle va devenir dramatique pour nous tous. Il est temps que tout le monde se mobilise pour trouver de nouveaux adhérents. Dans ce but, nous avons décidé de donner une prime aussi bien aux nouveaux adhérents qu'à ceux qui renouvellent leur cotisation, sous la forme d'un cadeau.

A bientôt, et bon courage !
Jean-Daniel Dodin



ADRESSE :
MICRO-REVUE, PPC-T, 77 rue du Cagire 31100 Toulouse France.

EDITION :
Edité par le club de Programmation Personnelle sur Calculatrice de Toulouse (PPC-T).

REDACTION-ADMINISTRATION :
Rédacteur en chef, directeur de la publication : Jean-Daniel Dodin ; secrétaires de rédaction : Gilles Barret et Jean-Yves Pasquier ; administration : Jean-François Sibille.

VENTE ET DISTRIBUTION :
Vente en librairie et par correspondance. Diffusion par l'éditeur. De plus, MICRO-REVUE est disponible à un tarif très avantageux pour les autres clubs et les associations qui le désirent, ainsi que pour les professionnels et les libraires qui souhaitent le revendre.

PARUTION :
Bimestriel, paraît en début de bimestre civil (Janvier, Mars, Mai, Juillet, Septembre, Novembre), 6 numéros par an.

TARIFS (franco de port) :
Prix de vente au numéro : 50F (60F par avion).
Abonnement individuel : 250F (300F par avion). Les envois à destination de l'Afrique n'arrivent pas toujours à destination, nous ne pouvons en aucun cas en assumer la responsabilité, dans la mesure du possible nous donner une adresse en Europe.

SOUSSION D'ARTICLES :
Tous les membres du club sont invités à fournir des articles pour MICRO-REVUE. L'original à nous faire parvenir doit être tapé à la machine avec un ruban bien noir, recto seul, simple interligne, avec une largeur maximale de 15 cm. Un envoi sur disquette IBM simple face, sur cassette HP82161A est recommandé. Les articles manuscrits sont acceptés s'ils sont calligraphiés sur papier blanc quadrillé 5x5, sans sauter de ligne, recto seul, au stylo noir. Nous faire parvenir cartes ou cassettes (qui vous seront remplacés) pour les programmes, ainsi qu'un listing. Cet original sera utilisé tel que pour la publication, soignez-le ! Ceux qui prévoient à l'avance d'écrire des articles sont invités à nous demander un "Kit Auteur" qui leur donnera les modèles et toutes les instructions nécessaires.

Les articles qui nous sont fournis sont certifiés originaux par leurs auteurs. Ni la revue, ni le club ni les auteurs ne sont responsables de l'usage qui sera fait des programmes qui sont uniquement fournis à titre documentaire, les utilisateurs sont seuls responsables de l'adaptation des programmes à leur propre usage. Ceci est spécialement vrai des programmes professionnels. Le fait, pour les auteurs, de nous envoyer un article ou un programme vaut autorisation de publication dans MICRO-REVUE ou dans toute autre publication du club, sans qu'aucune rémunération soit due à l'auteur de ce fait.

(c) 1985 REPRODUCTION INTERDITE

Les droits de reproduction des programmes et articles de MICROREVUE restent la propriété des auteurs et de la revue. En dérogation aux mentions ci-dessus, la reproduction des programmes est autorisée pour l'usage personnel des lecteurs, à l'exclusion de toute reproduction commerciale.

Les auteurs délèguent à la revue le droit d'autoriser la reproduction des articles et programmes dans d'autres revues à caractère non commercial.

PROGRAMME

The programme is designed to provide a comprehensive overview of the current state of the art in the field of artificial intelligence. It covers the following topics:

- 1. Introduction to Artificial Intelligence
- 2. The History of Artificial Intelligence
- 3. The Current State of Artificial Intelligence
- 4. The Future of Artificial Intelligence
- 5. The Role of Artificial Intelligence in Society
- 6. The Ethical Implications of Artificial Intelligence
- 7. The Legal Implications of Artificial Intelligence
- 8. The Economic Implications of Artificial Intelligence
- 9. The Social Implications of Artificial Intelligence
- 10. The Environmental Implications of Artificial Intelligence

HP-41

The HP-41 programme is designed to provide a comprehensive overview of the current state of the art in the field of artificial intelligence. It covers the following topics:

- 1. Introduction to Artificial Intelligence
- 2. The History of Artificial Intelligence
- 3. The Current State of Artificial Intelligence
- 4. The Future of Artificial Intelligence
- 5. The Role of Artificial Intelligence in Society
- 6. The Ethical Implications of Artificial Intelligence
- 7. The Legal Implications of Artificial Intelligence
- 8. The Economic Implications of Artificial Intelligence
- 9. The Social Implications of Artificial Intelligence
- 10. The Environmental Implications of Artificial Intelligence

PROGRAMMES

The programme is designed to provide a comprehensive overview of the current state of the art in the field of artificial intelligence. It covers the following topics:

- 1. Introduction to Artificial Intelligence
- 2. The History of Artificial Intelligence
- 3. The Current State of Artificial Intelligence
- 4. The Future of Artificial Intelligence
- 5. The Role of Artificial Intelligence in Society
- 6. The Ethical Implications of Artificial Intelligence
- 7. The Legal Implications of Artificial Intelligence
- 8. The Economic Implications of Artificial Intelligence
- 9. The Social Implications of Artificial Intelligence
- 10. The Environmental Implications of Artificial Intelligence

ADRESSES ET CODES DE FONCTIONS

Vous devez déjà connaître ce programme si vous possédez le n°8 de Micro-Revue. Dans le cas contraire je vais faire un bref rappel. Ce programme qui permet de déterminer :

- soit le code hexadécimal d'une fonction standard ou d'un périphérique connecté
- soit l'adresse hexadécimale de sa position dans la ROM (interne ou externe)
- soit les deux en même temps,

est basé sur la particularité du registre Q. En effet celui-ci laisse dans les octets trois, deux et un, zéro respectivement le code hexa et l'adresse, consécutif à l'appel de la fonction PASN du module X-FONCTIONS. Mais là où intervient la différence avec la précédente version, c'est que celle-ci contrairement à l'autre n'oublie pas d'éliminer les codes de la fonction assignée temporairement dans les registres d'assignations situés à partir de R0C0.

Mode D'emploi : - XEQ ALPHA AFCN ALPHA

- Entrez le nom de la fonction comme demandé puis R/S.
- En cas d'impossibilité la machine affiche INEXISTANTE, puis redemande le nom de la fonction.
- Sinon elle affiche A:DR H:EX E:H&A et attend votre réponse. Tapez (A) pour l'adresse, tapez (H) pour le code ou alors tapez (E) pour les deux valeurs. La machine affiche ensuite le résultat.
- Pour une nouvelle fonction tapez R/S.

Analyse Ligne à Ligne :

Lignes 01 à 05 : initialisation du registre d (registre des drapeaux), par une ligne de texte synthétique.

Lignes 06 à 09 : demande le nom de la fonction puis l'affiche.

Lignes 10 et 11 : sauve le registre t (registre contenant les bits codant les positions primaires des touches non shiftées portant un assignement) en X.

Lignes 12 à 14 : assigne la fonction choisie à la touche RDN et récupère le contenu de Q.

Lignes 15 à 17 : sauve le nom de la fonction en L et Z.

Lignes 18 à 22 : élimine l'assignation temporaire et restore le nom de la fonction.

Lignes 23 et 24 : rétablit le contenu du registre t.

Lignes 25 à 30 : en cas d'impossibilité.

Lignes 31 à 35 : sauve le nom de la fonction en X et affiche le choix des options.

Lignes 38 à 45 : établit le choix de l'utilisateur.

Lignes 46 et 47 : stocke le contenu de Q dans M.

Ligne 48 : efface les octets inintéressants de Q.

Lignes 49 à 55 : place les valeurs voulues en N.

Ligne 56 : constante pour la boucle 05.

Lignes 57 à 59 : stocke le nom de la fonction en M et prépare l'affichage final.

Lignes 60 et 61 : exécute une fois la boucle 05.

Lignes 62 à 64 : exécute deux fois la boucle 05.

Ligne 65 : sépare les deux valeurs demandées à l'affichage final.

Lignes 66 à 95 : transforme les valeurs en hexadécimal et les rajoutent à l'affichage.

Flags : 00 - option 0 : adresse
01 - option 1 : code
02 - option 2 : code et adresse
05 - fin de la boucle 05 après le deuxième passage
06 - désactive l'affichage dans la boucle 05 pour l'option 2

Entrées pour LB (PFC ROM) ou tout autre équivalent :

Ligne 03 : 244, 101, 6, 192, 128 (lève les flags : 25 26 29 31 37 38 40 41 48)
 Ligne 04 : 144, 117 Ligne 05 : 145, 126 Ligne 11 : 206, 122
 Ligne 14 : 206, 121 Ligne 24 : 206, 122 Ligne 29 : 159, 18
 Ligne 32 : 206, 117
 Ligne 33 : 255, 65, 58, 68, 82, 32, 72, 58, 69, 88, 32, 69, 58, 72, 38, 65
 Ligne 35 : 159, 71 Ligne 36 : 243, 11, 23, 15 Ligne 41 : 159, 47
 Ligne 47 : 145, 117 Ligne 52 : 145, 118 Ligne 56 : 27, 17
 Ligne 58 : 145, 117

01 LBL "ACFCN"	32 X<> M	63 SF 06	94 CF 01
02 LBL 01	33 "A:DR H:EX E:H&A"	64 XEQ 05	95 END
03 "eJ" " "	34 AVIEW	65 "t-"	
04 RCL M	35 TONE I	66 LBL 04	
05 STO d	36 " " "	67 SF 05	
06 "NOM FONCTION"	37 EMER	68 LBL 05	
07 STOP	38 LBL 03	69 RDN	
08 AOFF	39 RDN	70 ATOX	
09 AVIEW	40 GETKEY	71 I6	
10 CLX	41 TONE 7	72 /	
11 X<> t	42 POSA	73 X+Y?	
12 22	43 X<0?	74 X<Y?	
13 PASN	44 GTO 03	75 55	
14 X<> Q	45 SF IND X	76 X<Y?	
15 ASTO L	46 X<> Z	77 48	
16 ASHF	47 STO M	78 +	
17 ASTO Z	48 "t-xx"	79 XTOA	
18 CLA	49 FS?C 00	80 FRC	
19 22	50 "t-xx"	81 I6	
20 PASN	51 "	82 *	
21 ARCL L	52 STO N	83 X+Y?	
22 ARCL T	53 "t-xx"	84 X<Y?	
23 X<> Z	54 FS?C 02	85 55	
24 X<> t	55 "t-xx"	86 X<Y?	
25 FS?C 25	56 EI	87 48	
26 GTO 02	57 R	88 +	
27 "INEXISTANTE"	58 STO M	89 XTOA	
28 AVIEW	59 "t: "	90 FS?C 05	
29 TONE 8	60 FC?C 02	91 GTO 05	
30 GTO 01	61 GTO 04	92 FC?C 06	
31 LBL 02	62 SF 05	93 AVIEW	

Bonne programmation à tous

PRODUIT

Franck DEVEAUX
Faux-Fresnay
51 230 FERÉ CHAMPENOISE
T 678

Faux-Fresnay, le 30/7/85

Monsieur Jean Daniel Dodin
MICRO REVUE
77, rue du Cagire
31100 Toulouse

Cher Monsieur,

dans le numéro de Juillet-août, j'ai été très intéressé par le programme de monsieur Alain Herremann, concernant le produit d'éléments de $\mathbb{R}[X]$.

Néanmoins, je pense que dans ce genre de calculs, la rapidité est essentielle. J'ai donc tenté de rédiger un programme répondant à cet objectif.

Celui-ci se compose de 3 parties.

La première concerne le stockage des données — jusqu'au pas 19 —

La seconde le calcul des coefficients du polynome produit — à noter qu'ils sont tous calculés simultanément —

et la troisième, l'affichage du résultat.

Les détails de l'utilisation du programme sont rassemblés dans le tableau qui suit.

$$\begin{aligned} \cdot P(x) &= P_m x^m + P_{m-1} x^{m-1} + \dots + P_0 \\ \cdot Q(x) &= q_n x^n + q_{n-1} x^{n-1} + \dots + q_0 \\ \cdot P.Q(x) &= r_{m+n} x^{m+n} + \dots + r_0 \end{aligned}$$

Presser:	affichage	commentaire
XEQ α PQ α	1+DEG P	introduire le degré de P, +1.
(m+1) R/S	2+DEG P.Q	introduire le degré de P, plus celui de Q, +2.
(2+m+n) R/S	0,0000	introduire les coefficients de Q(x) dans l'ordre des puissances décroissantes.
q _n R/S	q _n	
q _{n-1} R/S	q _{n-1}	
q ₀ R/S	q ₀	il faut maintenant introduire les coefficients de P(x)
P _m R/S	P _m	
P _{m-1} R/S	P _{m-1}	
2nd α	P _{m-1}	} en cas d'erreur, en mode USER
P _{m-1} R/S	P _{m-1}	
P ₀ R/S	R _{m+n} X _{m+n}	monôme de plus haut degré de P(x).Q(x).
R/S	R _{m+n-1} X _{m+n-1}	
R/S	R ₀	

Pour obtenir un e nouvelle fois l'affichage du résultat faire A - en mode USER - Puis (R/S).

Exemple numérique : $(x+1)(x^3+3x^2+3x+1)$

1+deg P	→ 2	R/S	
2+deg P.Q	→ 6	R/S	
coef de x^3	→ 1	R/S	
" x^2	→ 3	R/S	
" x	→ 3	R/S	
" 1	→ 1	R/S	
" x	→ 1	R/S	
" 1	→ 1	R/S	
		R/S	1x4
		R/S	4x3
		R/S	6x2
		R/S	4x1
		R/S	1,0000

en 4"75.

01 * LBL a	24 +	47 RCL d
02 ISG 0	25 STO 0	48 3
03 T (FQ)	26 SIGN	49 LAST X
04 GTO 10	27 * LBL 09	50 STO a
05 * LBL T PQ	28 RCL IND N	51 -
06 T+DEG P	29 RCL IND M	52 CHS
07 PROMPT	30 ST* Y	53 FIX Q
08 CLRG	31 X<>Y	54 CF 29
09 STO 00	32 ST+IND 0	55 * LBL 08
10 T+DEG P.Q	33 ISG 0	56 CLA
11 PROMPT	34 T (FQ)	57 ARCL IND a
12 STO 0	35 DSE M	58 T- X
13 STO N	36 GTO 09	59 ARCL X
14 CLX	37 E	60 ISG a
15 * LBL 10	38 ST+ 0	61 T
16 STOP	39 ST- N	62 PROMPT
17 STO IND 0	40 RCL 00	63 DSE X
18 DSE 0	41 ST- 0	64 GTO 08
19 GTO 10	42 STO M	65 X<>Y
20 RCL 00	43 RCL N	66 STO d
21 STO M	44 X≠Y?	67 RCL IND a
22 RCL N	45 GTO 09	68 END
23 E	46 * LBL A	

- 130 octets
- 1 carte magnétique
- SIZE 2(deg P.Q + 2)

HAPPY PROGRAMMING

- Franck DEVEAUX -

P.S. Je serais intéressé par un programme plus rapide.

FILE MANAGER

I - PRESENTATION -

FILE MANAGER est un macro-langage de gestion de fichiers à accès direct. Les instructions de FILE MANAGER sont d'un niveau supérieur à celles du module XF, ce qui permet à l'utilisateur de se consacrer au problème à résoudre sans perdre de temps sur des questions propres au XF.

En effet, le module XF apporte un ensemble puissant d'instructions de gestion de fichiers, mais oblige à des manipulations d'assez bas niveau (en particulier la gestion des pointeurs).

II - JEU d'INSTRUCTIONS -

- CHR X : calcule le nombre de registres nécessaires pour un fichier de X caractères. Doit précéder immédiatement CREATE.
- CREATE <file> : crée le fichier <file>, avec comme taille le contenu du registre X.
- OPEN <file> : <file> devient le fichier travail : ses pointeurs sont remis à 0.
- STRUCT X : annonce la définition de X champs portant sur le fichier courant
- FIELD <field> : définit un champ par son nom (6 caract. max). toutes les instructions FIELD doivent se suivre et suivre immédiatement CREATE.

ex : STRUCT 3
FIELD NOM
FIELD PRENOM
FIELD ADR

partition mémoire		structure du fichier	
RO0=3		{	0 nom
RO1="NOM"	1		1 prénom
RO2="PRENOM"			2 adr
RO3="ADR"		{	3 nom
	2		4 prénom
			5 adr

WITH <value> : positionne sur la fiche de numéro <value>
<value> peut être :
- un registre mémoire noté Rxx
- un registre de la pile, noté X,Y,Z,T
- une constante numérique

GET <field> : positionne sur un champ de la fiche définie par WITH

NEXT : passe à la fiche suivante, quelle que soit la position actuelle des pointeurs

FORWARD : la recherche se fera à partir de la position actuelle des pointeurs (option par défaut)

WHOLE : La recherche se fera à partir du début du fichier

LIST : liste le fichier source

DUMP : imprime le contenu du fichier travail

FDUMP <field> : imprime le contenu du champ <field> de chaque fiche

TYPE <field> : définit le type de la chaîne que l'on veut chercher

ex : TYPE PRENOM

si TYPE est employé avec l'option FORWARD il faut s'assurer que le pointeur est déjà sur un champ du type <field>

SEARCH <string> : cherche la chaîne string

III - SYNTAXE -

Les macro-instructions doivent être immédiatement précédées du signe \$. Sans symbole spécial, elles seraient traitées comme des instructions de base, c'est-à-dire simplement copiées.

Les blancs en début de ligne ne sont pas autorisés, car ils seraient significatifs. Le paramètre est séparé de l'instruction par un espace.

IV - MODE D'EMPLOI -

FILE MANAGER est un pré-processeur, c'est-à-dire un logiciel recevant en entrée du texte source qui combine instructions de base (celles du langage HP 41) et macro-

instructions (celles de FILE MANAGER) et produisant en sortie des instructions de base.

Le code produit est ensuite utilisable seul.

Le fichier source est du type ASCII en X-Memory

La sortie se fait sur l'imprimante HP 82143 A

Le logiciel est facilement adaptable pour que l'entrée se fasse au clavier et la sortie sur l'affichage de la HP 41.

Il est évidemment regrettable que l'on ne puisse pas créer directement en sortie un programme en mémoire.

V - PASSAGE AU MODE INTERPRETEUR -

Pour passer à un interpréteur, il faudrait :

- ne pas altérer le source
- exécuter les séquences correspondantes aux macro-instructions au lieu de les imprimer

En fait, ces contraintes aboutiraient à un temps d'exécution inacceptable.

De plus, il vaut peut être mieux laisser l'utilisateur optimiser le code selon ses considérations personnelles.

Pierre BARTHELEMY
12, square Hopkinson
13004 MARSEILLE

T-659

01*LBL "FLMAN"	65*LBL 89	121*LBL 07	178*LBL 53
02 CF 05	66 FC? 08	122 CF 22	179 CF 05
03 CF 06	67 5	123 ANUM	180 RTN
04 FIX 0	68 FS? 08	124 FS? 22	
05 ADV	69 7	125 XEQ 18	181*LBL 26
06 "SOURCE FILE IS "	70 XEQ 12	126 FS? 22	182 SF 05
07 ACA	71 RCL 00	127 GTO 06	183 RTN
08 "SRC FILE ?"	72 INT	128 ASTO X	
09 ANUM	73 1 E3	129 "X"	184*LBL 32
10 STOP	74 /	130 ASTO Y	185 5
11 AOFF	75 1	131 X=Y?	186 XEQ 12
12 ACA	76 +	132 GTO 06	187 PRA
13 PRBUF		133 "RCL "	188 "0"
14 .		134 ARCL X	189 PRA
15 SEEKPTA		135 PRA	190 "SEEKPTA"
16 ADV			191 PRA
			192 RTN
17*LBL 04	77*LBL 01	136*LBL 06	193*LBL 49
18 SF 25	78 STO L	137 "I"	194 7
19 GETREC	79 RCL IND X	138 PRA	195 XEQ 12
20 FC? 25	80 ASTO Y	139 --	196 PRA
21 RTN	81 X=Y?	140 PRA	197 "CRFLAS"
22 CF 25	82 GTO 02	141 RCL 00	198 PRA
23 36	83 LASTX	142 INT	199 RTN
24 ATOX	84 ISG X	143 CLA	
25 X=Y?	85 GTO 01	144 ARCL X	200*LBL 82
26 GTO 05		145 XEQ 09	201 ANUM
27 X=Y?	86*LBL 02	146 --	202 20
28 RCL Z	87 LASTX	147 PRA	203 2
29 XEQ 10	88 1	148 "INT"	204 +
30 SF 25	89 -	149 PRA	205 7
31 CF 07	90 INT	150 "SEEKPT"	206 /
32 XEQ IND X	91 FS? 08	151 PRA	207 INT
33 FS? 07	92 RTN	152 RTN	208 CLA
34 SF 25	93 X=0?	153*LBL 38	209 ARCL X
35 FC? 25	94 GTO 03	154 "RCLPT"	210 XEQ 09
36 XEQ 17	95 "RCLPT"	155 PRA	211 RTN
37 GTO 04	96 PRA	156 RCL 00	
	97 CLA	157 INT	212*LBL 54
38*LBL 05	98 FIX 0	158 CLA	213 SF 07
39 RCLPT	99 ARCL X	159 ARCL X	214 .
40 INT	100 XEQ 09	160 XEQ 09	215 SEEKPT
41 SEEKPT	101 --	161 --	216 SIGN
42 GETREC	102 PRA	162 PRA	
43 PRA	103 "SEEKPT"	163 "INT"	
44 GTO 04	104 PRA	164 PRA	
		165 "I"	
45*LBL 97	105*LBL 03	166 PRA	
46 ANUM	106 "GETREC"	167 --	
47 STO 00	107 PRA	168 PRA	
48 1	108 RTN	169 RCL 00	
49 +		170 CLA	
50 SIZE?	109*LBL 35	171 ARCL X	
51 X<>Y	110 5	172 XEQ 09	
52 X>Y?	111 XEQ 12	173 --	
53 PSIZE	112 82	174 PRA	
54 SIGN	113 POSA	175 "SEEKPT"	
55 RTN	114 X<0?	176 PRA	
	115 GTO 07	177 RTN	
56*LBL 11	116 ANUM		
57 RCL Y	117 "RCL "		
58 7	118 ARCL X		
59 XEQ 12	119 XEQ 09		
60 RCL Z	120 GTO 06		
61 ASTO IND X			
62 1			
63 +			
64 RTN			

217*LBL 08	276*LBL 14		
218 SF 25	277 LASTX		
219 GETREC	278 INT		
220 FC? 25	279 1		
221 SEEKPT	280 -		
222 FC? 25	281 10		
223 ADV	282 /		
224 FC? 25	283 ST+ 00	340*LBL 16	381*LBL 76
225 RTN	284 SF 06	341 FS? 30	382 SF 08
226 PRA	285 RTN	342 *NO TYPE*	383 XEQ 09
227 GTO 00		343 FC? 05	384 CLA
	286*LBL 72	344 GTO 00	385 ARCL X
	287 8	345 *0*	386 XEQ 09
228*LBL 09	288 XEQ 12	346 PRA	387 *SEEKPT*
229 ALENG	289 PRA	347 *SEEKPT*	388 PRA
230 1	290 FC? 06	348 PRA	389 *LBL 00*
231 -	291 XEQ 16		390 PRA
232 AROT	292 FC? 06	349*LBL 00	391 *SF 25*
233 ATOX	293 RTN	350 *POSFL*	392 PRA
234 PRA	294 RCL 00	351 PRA	393 *GETREC*
235 RTN	295 FRC	352 CF 05	394 PRA
	296 10	353 CF 06	395 *FC?25*
236*LBL 10	297 *	354 RTN	396 PRA
237 ATOX	298 FC? 05		397 *RTN*
238 ATOX	299 GTO 15	355*LBL 78	398 PRA
239 *	300 CLA	356 *0*	399 *RCLPT*
240 1 E3	301 ARCL X	357 PRA	400 PRA
241 /	302 XEQ 09	358 *SEEKPT*	401 RCL 00
242 FRC	303 RCL Z	359 PRA	402 CLA
243 100	304 *SEEKPT*	360 *SF 25*	403 ARCL X
244 *	305 PRA	361 PRA	404 XEQ 09
245 INT		362 *LBL 00*	405 +-*
246 RTN	306*LBL 15	363 PRA	406 PRA
	307 *LBL 00	364 *GETREC*	407 *SEEKPT*
247*LBL 12	308 PRA	365 PRA	408 PRA
248 FS? 30	309 *ASTO X	366 *FC?25*	409 *GTO 00*
249 *TO PARAM*	310 PRA	367 PRA	410 PRA
250 RCLPT	311 *SF 25*	368 *RTN*	411 CF 00
251 INT	312 PRA	369 PRA	412 RTN
252 SEEKPT	313 *GETREC	370 *ACA*	
253 X<>Y	314 PRA	371 PRA	413*LBL 17
254 DELCHR	315 *FC?C 25	372 *FS?17*	414 FS? 30
255 GETREC	316 PRA	373 PRA	415 *ERREUR*
256 RTN	317 *RTN*	374 *GTO 00*	416 TONE 0
	318 PRA	375 PRA	417 RCLPT
257*LBL 47	319 *ASTO Y	376 *PRBUF*	418 INT
258 6	320 PRA	377 PRA	419 SEEKPT
259 XEQ 12	321 *X=Y?	378 *GTO 00*	420 GETREC
260 RCL 00	322 PRA	379 PRA	421 PRA
261 INT	323 *RTN*	380 RTN	422 *INCONNU*
262 STO 00	324 PRA		423 PRA
263 1 E3	325 *RCLPT*		424 STOP
264 /	326 PRA		
265 1	327 RCL 00		425*LBL 18
266 +	328 INT		426 CLA
	329 CLA		427 ARCL X
267*LBL 13	330 ARCL X		428 XEQ 09
268 STO L	331 XEQ 09		429 RTN
269 RCL IND X	332 +-*		430 .END.
270 ASTO Y	333 PRA		
271 X=Y?	334 *SEEKPT*		
272 GTO 14	335 PRA		
273 LASTX	336 *GTO 00*		
274 ISC X	337 PRA		
275 GTO 13	338 CF 06		
	339 RTN		

REGRESSION

OLIVIER MONACHON
Secrétaire T-13

TOULMATH
REGRESSION LINEAIRE

Voici ma contribution à TOULMATH :

Dans le numéro 8 de Micro-Revue, le programme de régression linéaire n'était pas tout à fait à mon goût, en effet celui-ci utilisait le registre 0 (du registre ALPHA) ce qui n'était pas très utile. Voici donc une nouvelle version du programme qui n'utilise plus le registre ALPHA, mais la pile et les registres statistiques (oeuf corse !!!).

J'ai utilisé des labels locaux pour les différentes fonctions, mais la version en ROM pourra avantageusement les remplacer par des labels globaux, plus utiles en cas d'appel en tant que routines d'un autre programme.

Pour les valeurs contenues dans les registres 00 à 05, on calcule la pente -a- de la droite de régression, et l'ordonnée à l'origine -b- ainsi que le coefficient de corrélation -c- pour l'équation $y=ax+b$.

Voici les fonctions associées à chaque label:

RLIN: place le premier registre statistique en R00 et efface R00 à R05

- A: calcule a, la pente de la droite (1)
- B: calcule b, l'ordonnée à l'origine (1)
- C: calcule c, le coefficient de corrélation (1)
- D: calcule x, abscisse estimée, y étant en X (2)
- E: calcule y, ordonnée estimée, X étant en Y (2)

(1) résultat en X, le contenu précédent de X est copié en Y, Z et T.

(2) résultat en X, rien de sauvé, hélas...

```
01*LBL "RLIN"  
IREG 00 CLE RTN
```

```
05*LBL B  
XEQ A MEAN LASTX * - RTN
```

```
12*LBL C  
XEQ A SDEV LASTX * X<>Y / RTN
```

```
20*LBL D  
XEQ A MEAN LASTX RT RT - X<>Y / + RTN
```

```
31*LBL E  
XEQ A MEAN LASTX X<>Y RT - CHS * + RTN
```

```
42*LBL A  
RCL 04 RCL 05 * RCL 00 RCL 02 * -  
RCL 01 RCL 05 * RCL 00 XT2 - / END
```

BIS...

CHER MONSIEUR

C EST AVEC PLAISIR QUE
J AI DECOUVERT PAR
HAZARD VOTRE BOUQUIN, ET
LA REVUE PPC-T.
ON Y TROUVE DES IDEES
FANTASTIQUES ET UTILES.
A L ATTENTION DE
P. HEILBRONN, JE JOINS
UN PRGM DE CONVERSION
CHIFFRES ROMAINS EN
CHIFFRES ARABES.
ON PEUT TRADUIRE PLUS DE
12 CARACTERES EN AJOU-
TANT DES REGS A 01 ET 02

SI VOUS POUVIEZ ME FAIRE
PARVENIR UN KIT AUTEUR,
J AI QUELQUES PRGM PRO,
ET DES JEUX AMUSANTS.

AMICALEMENT,

FabriceM... (41CX)

77-66

KUWAIT-CITY, LE 24/11/85

FREDERIC MOSER C/O C.B.
P.O. BOX 1364
HAMALLI - KUWAIT

CE PRGM CONVERTIT LES
CHIFFRES ROMAINS EN
CHIFFRES ARABES.

FabriceM... (41CX)

PRP "ARAB"

20:53 11/24
01*LBL "ARAB"
02 CLRG
03 ,
04 X<F
05 CF 29
06 FIX 0
07 "CHIFFRE ?"
08 TONE 9
09 AOH
10 PROMPT
11 ASTO 01
12 ASHF
13 ASTO 02
14 AOFF
15 CLA

16*LBL 01
17 "IM"
18 SF 01
19 XEQ 17

20*LBL 02
21 "XM"
22 SF 01
23 XEQ 17

24*LBL 03
25 "CM"
26 SF 01
27 XEQ 17

28*LBL 04
29 "M"
30 CF 01
31 XEQ 17

32*LBL 05
33 "CD"
34 SF 01
35 XEQ 17

36*LBL 06
37 "D"
38 CF 01
39 XEQ 17

40*LBL 07
41 "IC"
42 SF 01
43 XEQ 17

44*LBL 08
45 "XC"
46 SF 01
47 XEQ 17

48*LBL 09
49 "C"
50 CF 01
51 XEQ 17

52*LBL 10
53 "XL"
54 SF 01
55 XEQ 17

56*LBL 11
57 "L"
58 CF 01
59 XEQ 17

60*LBL 12
61 "IX"
62 SF 01
63 XEQ 17

64*LBL 13
65 "X"
66 CF 01
67 XEQ 17

68*LBL 14
69 "IV"
70 SF 01
71 XEQ 17

72*LBL 15
73 "V"
74 CF 01
75 XEQ 17

76*LBL 16
77 "I"
78 CF 01
79 XEQ 17

80 ARCL 01
81 ALENG
82 X=0?
83 GTO 18
84 GTO 01

85*LBL 17
86 ASTO X
87 ASTO L
88 CLA

89 ARCL 01
90 ARCL 02
91 POSA
92 X=0?

93 GTO a
94 SF 02
95 ATOX
96 RDN

97 FS?C 01
98 ATOX
99 ASTO 01
100 ASHF

101 ASTO 02
102 SF 00

103*LBL a
104 FS?C 02
105 XEQ IND L
106 FS?C 00

107 GTO 01
108 RTN

109*LBL "M"
110 E3
111 ST+ 00
112 RTN

113*LBL "IM"
114 999
115 ST+ 00
116 RTN

117*LBL "XM"
118 990
119 ST+ 00
120 RTN

121*LBL "CM"
122 9 E2
123 ST+ 00
124 RTN

125*LBL "D"
126 5 E2
127 ST+ 00
128 RTN

129*LBL "CD"
130 4 E2
131 ST+ 00
132 RTN

133*LBL "C"
134 E2
135 ST+ 00
136 RTN

137*LBL "IC"
138 99
139 ST+ 00
140 RTN

141*LBL "XC"
142 90
143 ST+ 00
144 RTN

145*LBL "L"
146 50
147 ST+ 00
148 RTN

149*LBL "XL"
150 40
151 ST+ 00
152 RTN

153*LBL "X"
154 10
155 ST+ 00
156 RTN

157*LBL "IX"
158 9
159 ST+ 00
160 RTN

161*LBL "V"
162 5
163 ST+ 00
164 RTN

165*LBL "IV"
166 4
167 ST+ 00
168 RTN

169*LBL "I"
170 E
171 ST+ 00
172 RTN

173*LBL 18
174 CLA
175 ARCL 00
176 TONE 9 (83)

177 AVIEW
178 FIX 4
179 SF 29
180 STOP

181 GTO "ARAB"
182 END
422 BYTES

END
FabriceM... (41CX)

ROMAINS<->ARABES

PHILIPPE HEILBRONN

**6, RUE FURSTENBERG
75006 PARIS**

PARIS, 1e 24 Octobre 1985

Monsieur Jean-Daniel DODIN,

MICRO-REVUE
77, rue du Cagire,
31100 Toulouse

Cher Monsieur,

Dans la rubrique 'PROGRAMMES' de MICRO-REVUE N° 7 P8 de Juillet-Août 1985, vous avez publié une lettre et un programme intitulé "ROMain" que je vous avais adressés concernant la conversion d'une date écrite en "ARABES" en une date écrite en "ROMAINS". En P.S., j'avais indiqué que je serais heureux de connaître un programme qui fasse l'opération inverse. Or, j'ai récemment reçu un programme écrit il y a 'quelques années' par un membre de PPC-T:

Monsieur François LESIEUR (T 541)
10 Rond-Point du VERMANDOIS

77176 SAVIGNY le TEMPLE

Le programme qui utilise de la la programmation synthétique permet de faire la conversion en un temps toujours inférieur à 9 secondes (sans imprimante attachée). La période couverte par le programme va de l'an 1 après J.C. à l'an 3999 après J.C. Le programme détecte les entrées de fautes d'écriture (comme IIII=4? ou bien comme IIX=0?).

Les instructions synthétiques permettent la lecture octet par octet du registre ALPHA. Les lignes 29 et 175 n'ont d'autre but que de le décaler d'un octet en ajoutant une octet vide (à 0). La ligne 39 sert à éviter des résultats fantaisistes en introduisant comme référence un caractère que l'on ne peut pas entrer directement au clavier.

Pour lancer le programme, il suffit de faire [XEQ] R-A; la machine se met en mode ALPHA, en émettant un TONE synthétique 89. Introduire alors la date avec les conventions M=1000, D=500, C=100, L=50, X=10, V=5, I=1, puis [R/S]. Le résultat obtenu, il suffit, pour une autre date, de l'entrer directement au clavier en mode ALPHA et de presser [R/S], en écrasant le résultat précédent. Le test des lignes 9 et 10 permet d'imprimer les résultats soit en largeur normale soit en caractères double WIDE (SF 12).

Le programme (en-tête et END inclus) comporta 287 octets et utilise les quatre registres R0, R1, R2 et R3.

Je me permets de vous adresser ci-inclus deux présentations du listing, l'une en 6/11, l'autre en 8/11, ainsi que DEUX cartes magnétiques.

J'ajoute que le programme tourne sur une HP-41 sans la présence d'aucun module.

Vous en souhaitant bonne réception, je vous prie de croire, cher Monsieur, à l'expression de mes sentiments les meilleurs.

HAPPY PROGRAMMING.

Philippe Heilbronn

Philippe Heilbronn (PPC-T 551)

Conversion d'une date écrite en "ROMAINS" en une date écrite en "ARABES".

```

001BL "R-A"
002 FIX 0
003 CF 29
004 CLA
005 TONE 9 (S89)
006LBL 08
007 AON
008 STOP
009 FS7 55
010 PRA
011 AOFF
012 RCL M
013 RCL N
014 RCL O
015
016 STO 00
017 CLA
018LBL 05
019 X#Y?
020 GTO 10 a
021 RDN
022 STO 05
023LBL 10 a
024 RDN
025 STO M
026 CLX
027 ENTER
028LBL 04
029
030 X<> N
031 X=Y?
032 GTO 04
033 X<> M
034 STO L
035 ASTO Y
036 RDN
037 "M"
038 ASTO 01
039 "0"
040 ASTO 02
041 ASTO 03
042 XEQ 05
043 "C"
044 ASTO 01
045 "D"
046 ASTO 02
047 "M"
048 ASTO 03
049 XEQ 06
050 "X"
051 ASTO 01
052 "I"
053 ASTO 02
054 "C"
055 ASTO 03
056 XEQ 06
057 "I"
058 ASTO 01
059 "U"
060 ASTO 02
061 "X"
062 ASTO 03
063 XEQ 06
064
065 "ERREUR"
066 ARCL 00
067 X=Y?
068 ASHF
069 ASTO X
070 TONE 9 (S89)
071 VIEW X
072 GTO 08
073 LBL 06
074 E1
075 ST+ 00
076 CLX
077 RCL 01
078 X=Y?
079 GTO 10 b
080 CLX
081 RCL 02
082 X=Y?
083 GTO 01
084 RDN
085 RTN
086LBL 10 b
087 XEQ 07
088 RCL 01
089 X=Y?
090 GTO 10 c
091 CLX
092 RCL 02
093 X=Y?
094 GTO 03
095 CLX
096 RCL 03
097 X=Y?
098 GTO 02
099 CLX
100 E
101 ST+ 00
102 RDN
103 RTN
104LBL 10 c
105 XEQ 07
106 RCL 01
107 X=Y?
108 GTO 10 d
109 CLX
110 2
111 ST+ 00
112 RDN
113 RTN
114LBL 10 d
115 CLX
116 3
117 GTO 09
118LBL 03
119 CLX
120 4
121 GTO 09
122LBL 02
123 CLX
124 9
125 GTO 09
126LBL 01
127 XEQ 07
128 RCL 01
129 X=Y?
130 GTO 10 e
131 CLX
132 5
133 ST+ 00
134 RDN
135 RTN
136LBL 10 e
137 XEQ 07
138 RCL 01
139 X=Y?
140 GTO 10 f
141 CLX
142 6
143 ST+ 00
144 RDN
145 RTN
146LBL 10 f
147 XEQ 07
148 RCL 01
149 X=Y?
150 GTO 10 g
151 CLX
152 7
153 ST+ 00
154 RDN
155 RTN
156LBL 10 g
157 CLX
158 8
159LBL 09
160 ST+ 00
161LBL 07
162 R"
163 R"
164
165 LAST X
166 X#Y?
167 GTO 10 h
168 RDN
169 X=Y?
170 RTN
171 X<>Y
172LBL 10 h
173 STO M
174 CLX
175
176 X<> N
177 X<> M
178 STO L
179 ASTO Y
180 RDN
181 END

```

PERIODE COUVERTE PAR
LE PROGRAMME :

de l'an 1 après J.C.:

"1"

à l'an 3999 après J.C.:

MMCMXCIX

"3999"

EXEMPLES :

=====

MDXV

"1515"

MDCCLXXXIX

"1789"

MDCCLXXXVIII

"1888"

MMMDCCCLXXXVIII

"3888"

Philippe Heilbronn (PPCT-551)

Mr CAHIERRE Patrick (T340)
Petit Kerbiguet
56380 GUER

Le 18.11.1985

Si un jour vous avez l'occasion de lire des documents de l'époque révolutionnaire, vous serez peut-être embarrassés par des dates du genre "3 Prairial An IV".....
Voici un court PRGM qui vous aidera à surmonter cet embarras.

Heureusement, le calendrier révolutionnaire était assez simple, il comportait 12 mois de 30 jours. En fin d'année, 5 jours (6 pour les années bissextiles) étaient ajoutés pour obtenir 365,25 jours. Le début de l'ère républicaine était fixé le 22 Sept. 1792 (1er Vendémiaire An I). Dès lors tout devient simple avec le TIME MODULE, il suffit de rentrer le PRGM suivant. (=112 octets)
Il rentre juste sur une piste de carte magnétique. Attention !!! Le prgm ne vérifie pas les entrées, vous pouvez mettre n'importe quoi !!
TABLE DE CONVERSION DES MOIS :

Vendémiaire	1
Brumaire	2
Frimaire	3
Nivôse	4
Pluviôse	5
Ventôse	6
Germinal	7
Floréal	8
Prairial	9
Messidor	10
Thermidor	11
Fructidor	12

Jours sans-culottides: Mois = 13

Jour = 1
2
3
4
5

Jour de la révolution = 6

(Tous les 4 ans)

REVGRE

LISTING DU PROGRAMME : REV-GRE

```
SF 16
PRP "REV-GRE"
10:32AM 12.11
01LBL "REV-GRE"
"JOUR ?" PROMPT
"MOIS ?" PROMPT E -
30 * + "AN ?" PROMPT
4 / INT LASTX 4 *
E - 365 * + +
21,091792 X<>Y DATE+
STO Y "SAMVENVJEMERMAR"
"BLUNDIM" -3 X<>Y DOW
* AROT RDN ASHF ASHF
ASHF "b" ADATE AVIEW
END
```

Exemple d'utilisation:

19 Brumaire An VIII
(Coup d'état de BONAPARTE)

41C: JOUR ? réponse 19 R/S
41C: MOIS ? réponse 2 R/S
41C: AN ? réponse 8 R/S

41C : DIM 10.11.1799

Voilà c'est tout simple....En espérant
rendre service à quelqu'un....

RESIZE

T334.

Ce programme permet de remplacer la fonction de la 41CX "RESZFL" qui modifie la taille d'un fichier ASCII ou DATA, pour les possesseurs de 41CV ou 41C, avec cependant quelques limitations :

- Il doit y avoir 1 seul fichier DATA ou ASCII dans la Mem. étendue.
- On ne peut dimensionner le fichier que de 255 registres maxi si vous avez des X-M, et 124 si vous avez un X-F.

Avant de passer au programme, voyons, ou pour certains revoyons, comment est organisée la structure d'un fichier en mémoire étendue. Chaque fichier est composé de 2 registres d'en-tête.

- le premier contient le nom du fichier sur 6 caractères
- le second est codé de la manière suivante selon les fichiers.

a) fichiers programmes : 10, 00, 00, 00, BB, BS, SS.

BBB est le nombre d'octets du programme sauvegardé.
SSS est la taille du fichier

b) fichiers données : 2A, AA, 00, 00, RR, RS, SS

- le "2" signale le fichier données
- AAA est l'adresse absolue du second registre d'en-tête.
- RRR est l'adresse absolue du pointeur d'enregistrement
- SSS est le nombre de registres du fichier.

c) fichiers ASCII : 3A, AA, 00, CC, RR, RS, SS

- le "3" signale un fichier ASCII
- CC est le pointeur de caractères.
- RRR est le pointeur d'enregistrement.
- SSS et AAA ont la même fonction que dans le fichier données.

A la fin du dernier fichier, un registre contient un code HEXA "FF, FF, FF, FF, FF, FF" qui signale la fin des fichiers.

Il s'agit donc pour le programme, de modifier le second registre d'en-tête du fichier, en y modifiant les deux premiers digits qui correspondent au SIZE. Vous comprendrez qu'il est quasiment impossible avec la programmation synthétique de travailler sur les 3 digits de leur agencement.

Il faut ensuite placer le code de fin de partition après le dernier registre du fichier.

Pour faire cela, le programme seul ne suffit pas, il faut aussi vous munir d'un programme de Conversion
Hexa → DECIMAL DECIMAL → Hexa.

Exemple: introduisez le programme "RSZF", puis effacez tous les fichiers de la mémoire étendue. Créez un fichier DATA de taille 8 registres. Maintenant nous voulons agrandir ce fichier et le faire passer de 8 à 15 registres par exemple, ou dans le cas général à X registres.

- Calculez $189 - X$ qui donnera l'adresse absolue du dernier registre du fichier. Ici: $189 - 15 = 174$
- Transformez ce nombre en Hexa. Ici AE
- Gardez le digit de gauche, en décimal: $A \rightarrow 10$
- Rajoutez au digit de droite un "2": E2, puis transformez ces deux digits en décimal: 226.
- Mettre le nombre X à l'affichage puis R/S, ou xEQ "RSZF".
au premier arrêt introduire 10 puis R/S
au second arrêt introduire 226 puis R/S
- A la fin vérifiez la Validité de l'opération: xEQ "EMNR".

Cependant une bonne connaissance de votre X-F et la "Quick reference Card" peuvent vous éviter l'usage d'un programme de conversion. En effet, reprenons notre exemple:

$$189 - 15 = 174$$

Cherchez sur la carte des codes la case

GTO	IND
247	IND46
174	

. En lisant ligne-Colonne vous obtenez AE.

Gardez le "A", puis cherchez la case ligne E colonne 2: (décimal 226). Voilà! Procédez ensuite normalement.

Si vous possédez uniquement un X-F, pour des fichiers de taille 124 registres, il ne sera pas utile de placer le code de fin de partition, et de toute façon il n'y aura pas la place. Donc dès que "-3" apparaîtra à l'affichage, interrompez l'exécution.

1	LbL "RSZF"	25	XTOA
	LbL D		STOP
	CF 00		XTOA
	LbL 00		0
5	RCL c		XTOA
	CLA	30	RCL c
7	"245,1,105,11,226,0"		ASTO c
	ASTO c		"247,255,255,255,255,255,255,255"
	CLA		RCL M
10	"247,32,190,0,0,11,208,0"		STO 00
	FS ? 00	35	X < Y
	XEQ 01		STO c
	-1		CLST
	AROT		RTN
15	X < Z		LbL A
	XTOA	40	SF 00
	RCL M		GTO 00
	X < 00		LbL 01
	X < Z		CLA
20	STO c		"247,48,190,0,0,11,208,0"
	"245,1,105,0,0,0"	45	RTN
	-3		END.
	AROT		
	STOP		

Mode d'emploi

- mettre X à l'affichage. En mode USER appuyer sur A pour un fichier ASCII et sur D pour un fichier Données. A l'arrêt introduire les digits en décimal. (voir exemple)

ACFCN

ROW1: LINES1-1



ROW2: LINES2-5



ROW3: LINES5-6



ROW4: LINES6-10



ROW5: LINES11-15



ROW6: LINES15-20



ROW7: LINES21-25



ROW8: LINES25-27



ROW9: LINES27-29



ROW10: LINES30-33



ROW11: LINES33-33



ROW12: LINES33-36



ACFCN

ROW13: LINES36-42



ROW14: LINES42-47



ROW15: LINES47-50



ROW16: LINES50-53



ROW17: LINES53-56



ROW18: LINES57-60



ROW19: LINES61-64



ROW20: LINES65-70



ROW21: LINES70-76



ROW22: LINES77-82



ROW23: LINES83-89



ROW24: LINES89-94



ROW25: LINES94-95



ARAB

ROW1: LINES1-2



ROW2: LINES3-7



ROW3: LINES7-8



ROW4: LINES8-14



ROW5: LINES15-19



ROW6: LINES19-23



ROW7: LINES23-27



ROW8: LINES27-31



ROW9: LINES32-35



ROW10: LINES36-40



ROW11: LINES41-44



ROW12: LINES45-48



ARAB

ROW13: LINES49-53



ROW14: LINES53-57



ROW15: LINES57-61



ROW16: LINES61-65



ROW17: LINES66-69



ROW18: LINES70-73



ROW19: LINES74-77



ROW20: LINES78-81



ROW21: LINES82-86



ROW22: LINES86-91



ROW23: LINES91-95



ROW24: LINES96-101



ARAB

ROW25: LINES101-105



ROW26: LINES106-109



ROW27: LINES109-113



ROW28: LINES113-116



ROW29: LINES117-118



ROW30: LINES119-121



ROW31: LINES122-125



ROW32: LINES125-129



ROW33: LINES129-131



ROW34: LINES131-134



ROW35: LINES135-137



ROW36: LINES138-141



ARAB

ROW37: LINES141-145



ROW38: LINES145-149



ROW39: LINES149-151



ROW40: LINES152-155



ROW41: LINES155-158



ROW42: LINES159-162



ROW43: LINES163-165



ROW44: LINES166-169



ROW45: LINES170-175



ROW46: LINES176-181



ROW47: LINES181-182



R-A

ROW1: LINES1-2



ROW2: LINES3-9



ROW3: LINES9-14



ROW4: LINES14-21



ROW5: LINES22-28



ROW6: LINES29-33



ROW7: LINES33-38



ROW8: LINES38-42



ROW9: LINES42-46



ROW10: LINES47-50



ROW11: LINES51-55



ROW12: LINES55-59



R-A

ROW13: LINES59-63



ROW14: LINES63-65



ROW15: LINES66-71



ROW16: LINES71-76



ROW17: LINES77-83



ROW18: LINES84-90



ROW19: LINES90-97



ROW20: LINES98-104



ROW21: LINES105-110



ROW22: LINES111-117



ROW23: LINES118-125



ROW24: LINES125-130



ROW25: LINES131-137



ROW26: LINES137-143



ROW27: LINES144-150



ROW28: LINES150-157



ROW29: LINES158-165



ROW30: LINES166-173



ROW31: LINES173-177



ROW32: LINES178-181



RLIN

ROW1: LINES1-2



ROW2: LINES2-7



ROW3: LINES8-13



ROW4: LINES14-21



ROW5: LINES21-28



ROW6: LINES29-34



ROW7: LINES35-42



ROW8: LINES43-51



ROW9: LINES52-57



REVGRE

ROW1: LINES1-1



ROW2: LINES1-3



ROW3: LINES4-6



ROW4: LINES7-11



ROW5: LINES11-19



ROW6: LINES20-25



ROW7: LINES25-27



ROW8: LINES27-29



ROW9: LINES29-29



ROW10: LINES29-30



ROW11: LINES31-36



ROW12: LINES37-42



ROW13: LINES43-43



RSZF

ROW1: LINES1-2



ROW2: LINES2-7



ROW3: LINES7-10



ROW4: LINES10-12



ROW5: LINES12-16



ROW6: LINES16-20



ROW7: LINES21-23



ROW8: LINES23-29



ROW9: LINES29-32



ROW10: LINES32-36



ROW11: LINES36-41



ROW12: LINES42-44



ROW13: LINES44-46



MICROCODE

TRANSFERT ROM-RAM AU SEIN D'UN MLDL

Imaginez un instant que vous n'avez, en tout et pour, qu'une HP 41, un MLDL de 4K avec le MLE1H et votre esprit tordu, mais malicieux pour résoudre le problème suivant: vous avez achevé un module de 4K que vous avez mis sur EPROM, et bien sur, vous trouvez une BUG. Evidemment, ce module n'est plus dans votre MLDL. Le problème est donc de charger le contenu des EPROM dans la RAM du MLDL, de corriger la bug, puis de remettre le module sur EPROM.

A première vue, ce n'est pas trop difficile: on utilise le MLDL comme lecteur d'EPROM et on fait un COPYROM... Cependant, quand vous utilisez le MLDL comme lecteur d'EPROM, vous devez retirer le MLE1H, et de ce fait, vous n'avez plus droit à la fonction COPYROM (sauf, bien entendu, si elle se trouve dans votre module). Quand on réfléchis bien au problème, c'est un vrai casse-tête...

Si vous ne m'avez pas bien suivi jusqu'ici, ne réfléchissez plus! Voici une solution: un programme de 43 registres qui nécessite un SIZE $\phi 1\phi$ (vous remarquerez un PSIZE à la ligne 3, mais ce n'est pas indispensable pour les malheureux qui n'ont pas le bonheur d'avoir le module X-FUNCTIONS, c'est juste pour faire joli!).

Suivez-moi bien, son principe de fonctionnement n'est pas évident:

1- Le programme réclame le MLE1H, puis demande le port dans lequel se trouve le MLDL, de manière à calculer l'adresse des EPROM. Nous appellerons X la page de cette adresse et Y celle des RAM. Le programme recopie ensuite le MLE1H dans les RAM du MLDL, de manière à

conserver les fonctions ROM-REG et GOLONGA, puis charge un petit programme microcodé maison analogue à COPYROM durant une phase appelée "PREPARATION".

2- Le programme réclame ensuite votre module ("EPROM S.V.P."), et dans la phase "CHARGEMENT" transfère la partie de votre module se trouvant entre XFE7 et XFFF dans les registres avec ROM-REG et lance le programme microcode précité qui se trouve en YFDF avec GOLONGA. Ce programme fait un "X000XFE6Y000 COPYROM".

3- On vous reréclame ensuite le MLE1H, et, dans la phase "TRANSFERT", le programme charge, par un REG-ROM, le petit bout de module resté sagement dans la mémoire de la 41 en YFE7.

Quelques détails, maintenant, sur le petit programme microcode. Il est chargé à partir des NNN du programme:

ligne 32	F710000069FDF015
ligne 35	F7144E273DOE41DO
ligne 38	F7150E270F8AF330
ligne 41	F71D068EABC29840
ligne 44	F7123A11806DABAO
ligne 47	F71BA30000000000

Voici ce qu'il donne, une fois désassemblé:

YFDF 04E C=O ALL	C=O W
YFEO 09C R=5	PT=5
YFE1 3DO LD R F	LC F
YFE2 390 LD R E	LC E
YFE3 1DO LD R 7	LC 7
YFE4 10E A=C ALL	A=C W
YFE5 09C R=5	PT=5
YFE6 0F8 READ 3(X)	C=REGN 3(X)
YFE7 2BC RCR 7	RCR 7
YFE8 330 FETCH S&X	CXISA
YFE9 106 A=C S&X	A=C X
YFEA 23A C=C+1 M	C=C+1 M
YFEB 2BC RCR 7	RCR 7
YFEC 0A6 AC EX S&X	AC EX X
YFED 040 WRIT S&X	WROME
YFEE 23A C=C+1 M	C=C+1 M
YFEF 046 C=O S&X	C=O X
YFF0 006 A=O S&X	A=O X
YFF1 36A ?A≠C R-	?A≠C WPT
YFF2 3AO ?NC RTN	RTN NC
YFF3 3A3 JNC - OC	GONC - OC retour en YFE7

Vous êtes toujours là? Que vous ayez compris ou non, vous pouvez maintenant charger votre module dans vos RAM. Pour la correction de la bug et la mise sur EPROM, je vous laisse vous débrouiller.

Mis à part la correction d'EPROM, ce programme a d'autres utilités. Ce qui m'a poussé à faire ce programme en est une illustration originale.

Il y a un an de cela, j'avais fait un petit module de 4K qui nécessitait de se trouver en page 9. J'avais besoin d'utiliser ce module en même temps que le lecteur de carte, et comme je n'avais pas de lecteur d'EPROM, je devais utiliser mon MLDL. Cependant, pour mettre les EPROM du MLDL en page 9, il fallait mettre celui-ci dans le port 1, ce qui était impossible avec le lecteur de carte. La solution était de mettre mon module dans les RAM du MLDL qui fut mis en page 9 grâce aux switches.

Une autre utilité est de pouvoir lire des modules de 8K avec un MLDL: on charge une page dans les RAM, et on utilise l'autre page sous forme d'EPROM.

La raison qui m'a poussé à ressortir ce programme des fonds de mon tiroir, et à vous en parler aujourd'hui concerne bien entendu le module FORTH 41. Mis à part le fait qu'il fasse 8K, ce qui est déjà une raison d'utiliser ce programme si vous n'avez pas de lecteur d'EPROM, vous n'êtes pas sans savoir qu'il nécessite 4K en page 4, les 4 autres K pouvant se trouver n'importe où. Or, un lecteur d'EPROM classique placera ces 4K en page 5, ce qui est ennuyeux si vous avez un module TIME, ou une HP-41CX. Une solution consiste à charger les 4K en page 4 dans les RAM du MLDL, et les 4 autres K en EPROM. Ce n'est pas la seule solution, et elle a ses contraintes:

on ne peut pas utiliser le MLDL en même temps que le module FORTH, et on doit avoir ce module sous forme de 2X4K, ce qui fait tout de même 4 EPROM...

Je signale à ceux que cela interesse les autres solutions:

- se passer de module TIME (ce qui est impossible dans le cas d'une 41CX) et utiliser un lecteur d'EPROM "classique" de 8K ou plus;
- utiliser d'autres appareils (un MLDL hollandais, par exemple, ou un lecteur d'EPROM spécial, mais Jean-Daniel Dodin aura l'occasion de vous en reparler). Je n'est pas personnellement essayé, mais cela doit marcher.

Pour en revenir à notre programme, vous trouverez sans doute d'autres applications.

A bientôt.

Serge Vaudenay

```

01*LBL "EPR-RAM"
02 10
03 PSIZE
04 "MLE 1H S.V.P."
05 BEEP
06 PROMPT
07 "0"
08 XROM 04,47
09 ASTO X
10 "PORT"
11 AVIEW
12 1
13 TONE 0
14 XROM 04,29
15 "PREPARATION"
16 AVIEW
17 TONE 9
18 XROM 04,45
19 2
20 *
21 7
22 +
23 STO 03
24 XROM 04,30
25 ASTO L
26 "F000"
27 ARCL L
28 "FFF"
29 ARCL Y
30 XROM 04,26
31 XROM 04,38
32 "0+ip"
33 STO 04
34 XROM 04,38
35 "N=AP"
36 STO 05
37 XROM 04,38
38 "s0"
39 STO 06
40 XROM 04,38
41 "+00"
42 STO 07
43 XROM 04,38
44 "A"
45 STO 08
46 XROM 04,38
47 "
48 STO 09
49 "FDF"
50 XROM 04,47
51 4
52 XROM 04,28
53 "EPROM S.V.P."
54 BEEP$
55 PROMPT
56 "CHARGEMENT"
57 AVIEW$
58 TONE 9
59 RCL 03
60 XROM 04,30
61 ASTO X
62 "FE7"
63 ARCL X
64 "FFF"
65 4
66 XROM 04,19
67 "0%"
68 XROM 04,47
69 ASTO X
70 RCL 03
71 XROM 04,38
72 "000000"
73 ARCL Y
74 "000"
75 XROM 04,09
76 "FDF"
77 XROM 04,47
78 XROM 04,02
79 "MLE 1H S.V.P."
80 BEEP$
81 PROMPT
82 "TRANSFERT"
83 AVIEW$
84 TONE 9
85 "FE7"
86 XROM 04,47
87 4
88 XROM 04,28
89 "0.K."
90 CLST$
91 +
92 AVIEW$
93 CLAR
94 BEEP$
95 BEEP$
96 .END.$

```

DESAS+TOULMATH

GOUBAULT de BRUGIERE Alain

27 avenue de GRINONT

78400 CHATOU

T441

le 4 avril 1985

Gérard JDD,

Voilà plus de trois ans que je m'adonne à la programmation en langage machine sur HP41 et c'est réellement formidable. Je salue mes compliments aux concepteurs du module ZENROM. Je m'en suis devenu indispensable tout d'abord à l'usage du MC. Seul reproche, la fonction de désassemblage nécessite une imprimante et je ne peux donc pas l'utiliser. C'est pourquoi j'ai écrit un court programme de désassemblage que je joins à ma lettre. Il est basé sur les 2 fonctions DECROT et MINEMO de l'EPRM de Stéphane Baugier (Nervi Stéphane!). Je n'interviendrai sûrement d'autres lectures. Mes réalisations en matière de MC se sont surtout portées sur des fonctions mathématiques qui je l'espère, seront incluses dans le module TOULMATHS. Je t'envoie la liste de ces fonctions et une brève description pour publication.

Nom du programme DESAS

Auteur

GOUBAULT T441

Date de révision

11/04/85

Description :

Programme de désassemblage du microcode en ROM.

Configuration : HP41 + EPRM box

Langage LU

Taille 94 octets

Size 0 registers

Périphériques

MLEH de

S. Baugier

Module XF

Points d'entrée :

DESAS demande l'introduction de l'adresse de début du désassemblage.

DES Cette adresse se trouve déjà dans le registre ALPHA.

Mode d'emploi :

Mettre une adresse en HEXA (4 digits) dans le registre ALPHA et XEQ DES ou suivre la procédure suivante :

XEQ DESAS → affichage ADDR? ---

→ introduction l'adresse en HEXA (seules les touches 0 à 9, A à F et la touche de correction sont valides)

Le programme affiche READS. Le désassemblage peut commencer. Tout ce qui sur RIS affiche la mémoire de l'instruction à l'adresse courante (cette adresse est automatiquement incrémentée). En mode USER [A] affiche l'adresse, le code 244 et le caractère associé (l'adresse n'est pas incrémentée).

Listing du programme : DESAS

```

01 LBL DESAS 23 LBL 00
CF 21 X# 0?
4 AVEW
ADDR? GTO 01
AVEW 27 LBL A
CLA SF 10
HEXPR LBL 01
DECODE FS? C 09
T--- RDN
RCL M JCD ROT
CLA FS? C 10
STO M AVEW
3 CHS MNENO
AROT CLA
ARCL Z
ARCL T
ALENG
SF 09
SF 21 GTO 00
42 END
CODE
READY

```

Commentaires :

Le clavier 09 est armé à chaque fois qu'une mnémotechnique est affichée.
Le clavier 10 est armé à chaque appui de la touche [A] en mode USER.
Une mnémotechnique sur 2 mots machine est affichée en entier.

Voici quelques fonctions écrites en NC pour le projet : TOULMATHS

Fonctions hyperboliques

Sinus (SH), Cosinus (CH), Secante (SEH), Cosecante (CEH), Amplitude (GD), tangente (TH) et cotangente (COTH) et leurs inverses, respectivement : ASH, ACH, ASEH, ACEH, AGD, ATH et ACOTH.
Ces fonctions s'utilisent de la même façon que SIN ou ATAN, etc...

Fonctions vectorielles

Produit scalaire PSCAL

$||X|| = m \cdot ddd \cdot m \cdot m$
avec Rm est la première registre où commence le vecteur u.
Rdd est la première registre où commence le vecteur v.
mm est la dimension des deux vecteurs.

Le résultat du produit scalaire de u par v est dans X, l'ancien X est sauvegardé dans lastX

Norme NORM

$||X|| = m \cdot ddd$
avec Rm est la première registre où commence le vecteur.
Rdd est la dernière registre où finit le vecteur.
Le résultat est dans X, l'ancien X est sauvegardé dans lastX.

Multiplication d'un vecteur par une matrice MB3U

$$\|X\| = m \cdot ddd \cdot mnn \quad \|Y\| = \alpha$$

Le mot de contrôle est identique à celui pour PSCAL.

Le résultat est $r = \alpha \cdot u$.

La pile n'est pas modifiée.

Combinaison linéaire de deux vecteurs COLNR

$$\|X\| = m \cdot ddd \cdot mnn \quad \|Y\| = bbb \cdot fff$$

$$\|Z\| = \alpha \quad \|T\| = \beta$$

Le mot de contrôle dans X est identique à celui pour PSCAL. Celui dans Y est identique à celui pour NORM (vecteur w).

Le résultat est $w = \alpha u + \beta v$.

La pile n'est pas modifiée.

Somme ou différence de deux matrices M+M, M-M

Mode opératoire identique à COLNR sauf que l'on ne donne pas les coefficients α et β en Z et T.

Transposition d'une matrice TRASM

$$\|X\| = m \cdot ddd \cdot mnn \quad \|Y\| = bbb \cdot fff \cdot mnn$$

La matrice M est stockée du registre ms au registre ddd (mnn colonnes, mnn lignes).

La matrice N est stockée du registre bbb au registre fff (mnn colonnes, mnn lignes).

Le résultat est $N = {}^t M$.

La pile n'est pas modifiée.

Multiplication d'une matrice par un vecteur MB3U

$$\|X\| = m \cdot ddd \cdot mnn \quad \|Y\| = bbb \cdot fff$$

La matrice M est stockée à partir du registre ms (mnn colonnes, (fff-bbb+1) lignes).

Le vecteur u est stocké à partir du registre ddd (mnn composants).

Le vecteur v est stocké du registre bbb au registre fff.

Le résultat est $r = M \cdot u$.

La pile n'est pas modifiée.

Produit des éléments diagonaux d'une matrice

DETER

$$\|X\| = m \cdot ddd \cdot mnn$$

La matrice est stockée du registre ms au registre ddd (mnn colonnes).

Le résultat est dans X, l'ancien X est sauvegardé dans TstX.

Effacement d'une zone mémoire MCLR

$$\|X\| = m \cdot ddd$$

Les registres ms à ddd sont effacés (fonctionne comme CLRGX du H4410 mais les registres sont contigus).

Multiplication de deux matrices

M*N

$$\|Z\| = m \times d \quad d \times m$$

$$\|Y\| = b \times b \quad b \times m$$

$$\|X\| = x \times x \quad x \times x$$

La matrice M est stockée du registre mms au registre ddd (mmm colonnes)
La matrice N est stockée du registre bbb au registre fff (mmm colonnes, mmm lignes)
La matrice L est stockée du registre xxx au registre yyy (mmm colonnes)
Le résultat est L = M.N
X est dans laot X, la file est descendue

Notes

Les erreurs sont détectées et l'exécution des fonctions est suspendue (ALPHA DATA, NONEXISTENT, OUT OF RANGE, DATA ERROR)
Certains résultats partiels peuvent toutefois être enregistrés.

Les matrices sont stockées ligne par ligne dans des registres contigus.

Cela permettra aux lecteurs de NR d'être au courant de ce qui est déjà réalisé, mais aussi ce que pour inclure les fonctions qui les intéressent dans leurs propres programmes qu'ils voudraient soumettre à ce projet.

Dans le dernier numéro de NR, tu parles d'une diffusion sous forme d'Epson. J'espère bien qu'il y aura assez de demandes pour en faire un module car un lecteur d'Epson (qui plus est de 12 ko) gêne terriblement la portabilité du HP 41 (A moins qu'il n'existe des lecteurs très compact, mais pour 12 ko je demande à voir).

Enfin qu'il en soit, je souhaite la bonne réussite de ce projet. Elle dépend de la participation de tous ; aussi, que les lecteurs de NR envoient à Eric leurs observations, il y aura toujours quelque chose d'intéressant.

H. P.

H. P. Guibaud

HP-71

HP-75

TIC...PAC... TIC...PAC...

J.Dutertre T416

Le programme TICPAC transforme une HP75, avec un Pacscreen et un moniteur, en une pendule digitale calendrier fort acceptable. Je vois d'ici les esprits chagrins remarquer qu'il suffit d'appuyer sur "TIME" pour en faire autant. Bien, c'est ça, vous n'êtes ni myope, ni presbyope, ni astigmaté ou que sais-je: tournez la page. S'il n'existait jamais un moyen compliqué de faire les choses simples, ou serait le plaisir. D'ailleurs cette notice est sans intérêt: le programme fonctionne tout seul, faites "RUN" et vous verrez bien.

Maintenant, si vous aimez aussi les choses compliquées, ajoutez un carillon "Westminster", ou faites lui chanter Matines et Vêpres, à vous le soin, avec mes amitiés.

```

10 ! TICPAC
20 ! J.Dutertre
30 !
40 D0$.D2$.J0$.T4$.T5$.T6$=""
50 J$="DIMLUNMARMERJEUENSAM"
60 PRINTER IS ":g1" E PRINT "in.sc 0.10.0.10.si 8.8"
70 D$=DATE$ E IF D2$=D$ THEN 130
80 D0=INT((DATE/1000)) E D1=DATE-1000*D0 E D0=D0-1 E IF D0<84
   THEN D1=D1-1
90 D=MOD((INT(D0/4)+365*D0+D1+1.7)*3+1) E J1$=J$[D.D+2]
100 D1$=D$[7.8]&" "&D$[4.5]&" "&D$[1.2]
110 X=1 E Y=7 E K0$=D0$ E K1$=D1$ E GOSUB 190 E D0$=D1$ E
   D2$=DATE$ E PRINT "si 10.8"
120 X=3 E Y=4 E K0$=J0$ E K1$=J1$ E GOSUB 190 E J0$=J1$ E
   D2$=DATE$ E PRINT "si 8.8"
130 T$=TIME$ E T0$=T$[8.8] E IF T0$="0" AND T0$="5" THEN 130
140 T1$=T$[1.2] E T2$=T$[4.5] E T3$=T$[7.8] E Y=1
150 IF T3$#T6$ THEN X=7 E K0$=T6$ E K1$=T3$ E GOSUB 190 E
   T6$=T3$
160 IF T2$#T5$ THEN X=4 E K0$=T5$ E K1$=T2$ E GOSUB 190 E
   T5$=T2$
170 IF T1$#T4$ THEN X=1 E K0$=T4$ E K1$=T1$ E GOSUB 190 E
   T4$=T1$
180 GOTO 70
190 PRINT "pu.pa";X;".":Y E PRINT "sp 0.pd.lb"&K0$&CHR$(3)
200 PRINT "pu.pa";X;".":Y E PRINT "sp 1.pd.lb"&K1$&CHR$(3) E
   RETURN

```

ECHANGE ENTRE
HP75 ET HP71

Jean Claude LIEHN T 488
8 rue de l'écaille
51100 Reims
Tel: 26 85 50 52

Jean Daniel DODIN
77 rue du Cagire
31100 Toulouse

Reims, le 1er mars 1986

Cher Président,

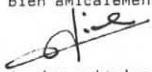
Voici trois propositions d'article pour MR. Il s'agit:

- D'un ensemble de deux programmes permettant la communication de données numériques entre le 75 et le 71.
- D'un programme d'analyse de données pour le 75 facilement transposable sur le 71. (Dans la série: il existe encore des calculteurs qui calculent!)
- D'un article sur le module AMPISTAT du 71. (Voir HP7x)

Je pense que tout cela peut intéresser les lecteurs.

Bravo toujours aux gens de Toulouse pour MR. La couverture du numéro 10 est très réussie. Continuer toujours à soutenir le 75. Le 71 n'est pas si supérieur que cela. Il est nettement plus lent et son clavier est invivable. Avec le recul, vos conclusions du n°11 du journal (en mai 84) me paraissent pertinentes, comparées à l'admiration un peu bête des gens de Paris. Pour ceux qui ont besoin d'un bon BASIC scientifique, un 75 d'occasion me paraît effectivement une bonne affaire. Vous faites bien de le dire.

Bien amicalement.



PS: Si vous avez à traduire de l'anglais des articles touchant mes préoccupations (statistiques et analyse numérique), je peux vous aider.

ECHANGE ENTRE HP75 ET HP71

Jean Claude LIEHN T 488
8 rue de l'écaillie
51000 Reims
Tel: 26 85 50 52

le 26 02 86

ECHANGE DE DONNEES NUMERIQUES ENTRE LE HP75 ET LE HP71

Les deux programmes WSDA7571 (pour le 75) et FR75HPAF (pour le 71) réalisent les tâches suivantes:

- 1) Saisie d'enregistrements numériques sur le 75 et écriture:
 - soit dans une feuille de calcul, en commençant à la cellule voulue.
 - soit dans un fichier de type DATA du 75.
- 2) Transfert de données d'une feuille de calcul vers un fichier DATA du 75 ou réciproquement.
- 3) Transfert de données d'une feuille de calcul ou d'un fichier DATA du 75 vers un fichier de format HPAF du 71.

Le format HPAF est le format officiel pour le 71. Il est utilisé par les modules FINANCE, CURVE FIT et AMPISTAT ainsi que par divers programmes du catalogue HP. Il est décrit dans les documentations de ces programmes ainsi que dans le Software Developers' Handbook. Il n'a jamais existé rien de tel pour le 75.

Pour permettre le transfert entre le 75 et 71, il faut faire CONTROL OFF sur le 71 et l'insérer dans la boucle ayant le 75 comme contrôleur. Le 75 lance lui-même l'exécution du programme FR75HPAF du 71. Il faut les modules VISICALC et I/O sur le 75 et le module HPIL sur le 71.

J'utilise personnellement ces programmes pour traiter avec les programmes statistiques du 71 des données saisies et éditées dans VISICALC. Ces manœuvres un peu lourdes n'auraient pas de raison d'être si le 75 avait été doté des logiciels qu'il méritait ou si le 71 était un peu plus ergonomique.

Je ne donne pas de mode d'emploi, il suffit de suivre le menu. Je signale simplement que la saisie des données est terminée en tapant E.

WSDA7571

WSDA7571 B 3110 14:09 03/01/86

```

10 *****
   * US07571 version du 26/08/85
   *****
20 INTEGER 0,01,02,L,L1,L2,L3,L4,C,C1,C2,C3,C4
30 REAL X,U
40 *****
   * ! Display message and get answer
   *****

50 DEF FNR1(M,C)
60 DISP M
70 0=POS(C#UPR(C#KEY$)) # IF 0=0 THEN 70
80 FNR1=0 # END DEF
90 *****
   * ! Read from file
   *****

100 DEF FNR1(L,C)
110 SEEK # 1,L,C # READ # 1 : U # FNR1=U # END DEF
120 *****
   * ! Read from worksheet
   *****

130 DEF FNR2(L,C) = VAL(GETVALUE$(C,L,1))
140 *****
   * ! Read from keyboard
   *****

150 DEF FNR3(L,C)
160 DISP "CE Record:"L+3-1;" Item "C+C3-1; # INPUT " : "R$
170 ON ERROR GOTO 200
180 IF UPR(C#R$)="C" THEN END
190 FNR3=VAL(R$)
200 OFF ERROR
210 END DEF
220 *****
   * ! Read value according to input option
   *****

230 DEF FNR(L,C,D)
240 IF 0=1 THEN FNR=FNR1(L,C)
250 IF 0=2 THEN FNR=FNR2(L,C)
260 IF 0=3 THEN FNR=FNR3(L,C)
270 END DEF
280 *****
   * ! Write on file
   *****

290 DEF FNR1(L,C,U)
300 IF C=1 THEN PRINT # 1,L : U # GOTO 340
310 SEEK # 1,L,C
320 ADVANCE # 1 : 1,X
330 INSERT # 1 : U
340 FNR1=1
350 END DEF
360 *****
   * ! Write on worksheet
   *****

370 DEF FNR2(L,C,U)
380 PUTVALUE C,L,STR$(U)
390 FNR2=1
400 END DEF
410 *****
   * ! Send to HP71
   *****

420 DEF FNR3(L,C,U)

```

```

430 OUTPUT "c1" ; STR$(U) @ FMS-1 @ END DEF
440 *****
    * I Write on the proper file or device according to the output option
*****

450 DEF FMS(L,C,U,D)
460 IF D=1 THEN FMS=FMS(L,C,U)
470 IF D=2 THEN FMS=FMS(L,C,U)
480 IF D=3 THEN FMS=FMS(L,C,U)
490 END DEF

500 *****
    * I Main program
*****

510 D1,D2=0
515 *****
    * I Enter input and output options
*****

520 D1=FMS("from "BCHB(198)A"file "BCHB(215)A"orksheet "BCHB(203)A"eyboard ",FMS")
530 D2=FMS("to "BCHB(198)A"file "BCHB(215)A"orksheet "BCHB(203)A"PP1 ",FMS")
535 *****
    * I Look for unallowed transfer
*****

540 IF D1=1 AND D2=1 THEN DELAY 1 @ DISP "Not from File to File " @ DELAY 0 @ GOTO 530
550 IF D1=2 AND D2=2 THEN DELAY 1 @ DISP "Not Worksheet to Worksheet" @ DELAY 0 @ GOTO 530
555 IF D1=3 AND D2=3 THEN DELAY 1 @ DISP "Not Keyboard to HP71" @ DELAY 0 @ GOTO 530
556 *****
    * I Get file names and field limits
*****

560 IF D1=1 THEN INPUT "Origin File : ",DIR$(3,"B"); F1@ ASSIGN # 1 TO F1@
570 IF D1=2 THEN INPUT "Origin Worksheet : ",DIR$(3,"U"); U1@ WORKSHEET U1@
580 IF D2=1 THEN INPUT "Destination File : ",DIR$(3,"B"); F2@ ASSIGN # 1 TO F2@
590 IF D2=2 THEN INPUT "Destination Worksheet : ",DIR$(3,"U"); U2@ WORKSHEET U2@
600 IF D2=3 THEN INPUT "HP71 File Name : ",F2@
610 IF D1=1 THEN L1=1 @ L2=LASTLN(F1@) @ C1=1 @ C2=LINELEN(L1)
620 IF D1=2 THEN L1=1 @ L2=MAXROW @ C1=1 @ C2=MAXCOL
630 IF D1=3 THEN L1=1 @ L2=1000 @ C1=1 @ INPUT "Number of items per record : "C2
640 IF D1#3 THEN INPUT "From line ",STR$(L1);L1
650 IF D1#3 THEN INPUT "To line ",STR$(L2);L2
660 IF D1#3 THEN INPUT "From column ",STR$(C1);C1
670 IF D1#3 THEN INPUT "To column ",STR$(C2);C2
680 L3,C3=1
690 IF D2=1 THEN L3=LASTLN(F2@)+1 @ C3=1 @ INPUT "Write from line ",STR$(L3);L3
700 IF D2=2 THEN L3=MAXROW+1 @ C3=1
710 IF D2=2 THEN INPUT "Write from row ",STR$(L3);L3
720 IF D2=2 THEN INPUT "Write from column ",STR$(C3);C3
725 *****
    * I Initialize HP71 and send HPBF file name and size
*****

730 IF D2=3 THEN REMOTE "c1" @ OUTPUT "c1" ; "RUN PR75HPBF" @ LOCAL
740 IF D2=3 THEN OUTPUT "c1" ; F2@
750 IF D2=3 THEN OUTPUT "c1" ; STR$(C2-C1+1)
760 IF D2=3 THEN OUTPUT "c1" ; STR$(L2-L1+1)
770 *****
    * I Perform transfer
*****

780 FOR L=L1 TO L2
790   FOR C=C1 TO C2
800     V=FMS(L,C,D1)
810     X=FMS(L+L3-L1,C+C3-C1,U,D2)
820     NEXT C
830 NEXT L
840 END

```

```

10 ! ***** FR75HPAF *****
11 !
15 OPTION BASE 1
20 DISP '      OK HP75 !'
30 ENTER :LOOP :F$ @ DISP 'FILENAME = ' :F$
40 ENTER :LOOP :N1$ @ DISP 'N ITEMS = ' :VAL(N1$)
50 ENTER :LOOP :N2$ @ DISP 'N RECORDS = ' :VAL(N2$)
60 P=VAL(N1$)
70 N=VAL(N2$)
71 REAL X(P)
75 A$='HPAF' @ FOR I=1 TO P @ A$=A$&'N' @ NEXT I
80 DISP F$:'(' :P :',' :N :')':A$
90 ON ERROR GOTO 95
92 PURGE F$
95 OFF ERROR
100 CREATE DATA F$,N+3,P*8
110 ASSIGN #1 TO F$
115 PRINT #1,0:A$
120 PRINT #1,1:N
130 PRINT #1,2:0
140 FOR I=1 TO N
145 FOR J=1 TO P
150 ENTER :LOOP :A$
160 X(J)=VAL(A$)
170 NEXT J
175 PRINT #1,I+2:X()
177 NEXT I
180 DISP 'TRANFER OK TANK YOU'

```

ANALYSE

ANALYSE EN COMPOSANTES PRINCIPALES SUR LE HP75 (et le 71)

Les programmes statistiques disponibles sur la machine de la série 70 ne dépassent pas le niveau des tests usuels simples. Pourtant la puissance de ces machines, en particulier avec les modules mathématiques permet d'envisager des applications plus avancées.

Voici un programme d'analyse en composantes principales pour le 75 muni des modules MATH et I/O. Une adaptation pour le 71 est facile, elle a déjà été réalisée par Michel Weil (T640) qui est à l'origine de quelques corrections que j'ai apportées à la version initiale de ce programme.

Ce programme travail sur des enregistrements de données strictement numériques écrites ligne par ligne dans un fichier de type DATA du 75 (un enregistrement par ligne). Il utilise le sous-programme SPEIGEN qui calcule les valeurs et vecteurs propres d'une matrice symétrique. Une version modifiée de la méthode de JACOBI est utilisée.

Le programme calcule :

- La moyenne et l'écart-type de chaque variable.
- La matrice de corrélation.
- Les valeurs propres et les pourcentages d'inertie.
- Les vecteurs propres.
- Les projections des variables (avec une représentation graphique simplifiée).

SPEIGEN

SPEIGEN B 1179 14:23 03/01/86

```

10 SUB "SPEIGEN" (U(),D())
15 *****
   # ! DIAGONALISATION D'UNE MATRICE SYMETRIQUE
*****

16 REM EN ENTREE: U CONTIENT LA MATRICE
17 REM EN SORTIE: D CONTIENT LES VALEURS PROPRES, U CONTIENT LES VECTEURS PROPRES.
20 N=UBND(U,1)
150 U2=FNORM(U) # U2=U2*U2
160 E=.000001*U2/N
170 K2=N*(N-1)/2 # (Z=K2 # N1=N-1
180 FOR K=1 TO N1
190     K1=K+1
200     FOR P=K1 TO N
210         U2,U3=0
220         FOR L=1 TO N
230             U2=U2+U(L,K)*U(L,P)
240             U3=U3+(U(L,K)+U(L,P))*(U(L,K)-U(L,P))
250         NEXT L
260         U4=U2*2 # U5=ABS(U4)
280         IF U5<E AND U3=0 THEN 400
290         U6=ABS(U3)
300         IF U5<4*U6 THEN !+U5/U6 # C=1/SQR(1+!+1) # S=1+C ELSE A=4*U5/U6 # S=1/SQR(1+A*A) # C=A*H
310         C=SQR((1+C)/2) # S=S/(2*C)
320         IF U3<0 THEN F=C # C=S # S=F
330         IF U4<0 THEN S=-S
340         FOR L=1 TO N
350             F=U(L,K)
360             U(L,K)=F+C*U(L,P)*S
370             U(L,P)=F*S+U(L,P)*C
380         NEXT L
390         K2=K2 # GOTO 430
400         K2=K2-1
420         IF K2<0 THEN 460
430     NEXT P
440 NEXT K
450 GOTO 180
460 FOR K=1 TO N
470     D(K)=0
480     FOR L=1 TO N
490         D(K)=D(K)+U(L,K)*U(L,K)
500     NEXT L
510     D(K)=SQR(D(K))
520 NEXT K
530 FOR K=1 TO N
540     FOR L=1 TO N
550         U(L,K)=U(L,K)/D(K)
560     NEXT L
570 NEXT K

```


ANACOMPR

ANACOMPR B 2203 15:32 03/01/86

```

10 *****
   * 1 ANACOMPR: ANALYSE EN COMPOSANTES PRINCIPALES 30 08 84 *
*****

20 OPTION BASE 1
30 REAL B(128),M(10),E(10),C(10,10),V(10,10),P(10)
40 *****
   * 1 SIMULATION ACCES DIRECT *
*****

50 DEF FND(L,C)
60 SEEK # 1,L,C # READ # 1 : VO # FND=VO
70 END DEF
80 *****
   * 1 DECLARATION FICHIER DE DONNEES *
*****

90 INPUT "DATAFILE NAME : "; AN # ASSIGN # 1 TO AN
100 N=LASTLN?(FILEN(1)) # N1=LINELEN(1,1)
110 REDIM B(N),M(N),C(N),C(N,1),V(N,1),P(N)
120 *****
   * 1 REDUCTION ET CENTRAGE *
*****

130 PRINT # PRINT "*** ANALYSE EN COMPOSANTES PRINCIPALES ***" # PRINT
140 PRINT " MOYENNES ET ECART TYPE" # PRINT
150 FOR I=1 TO N1
160   FOR J=1 TO N # B(J)=FND(J,I) # NEXT J
170   T1=SUM(B) # T2=DOT(B,B)
180   M(I)=T1/N # E(I)=SQRT((T2-T1*T1/N)/N)
190   PRINT USING " " VAR # ".DO,3X,"M = ",.00000.D00,3X,"E1 = ",3X,.00000.D00" : I,M(I),E(I)
200 NEXT I # PRINT
210 *****
   * 1 CALCUL DE LA MATRICE DE CORRELATION *
*****

220 FOR I=1 TO N1 # FOR J=1 TO I
230   S=0
240   FOR K=I TO N
250     S=S+(FND(K,I)-M(I))*(FND(K,J)-M(J))/E(I)
260   NEXT K
270   C(I,J),C(J,I)=S/N
280 NEXT J # NEXT I
290 ASSIGN # 1 TO *
300 PRINT " MATRICE DE CORRELATION " # PRINT
310 MAT PRINT USING "S(XX,NO.D00)" : C # PRINT
320 *****
   * 1 RECHERCHE DES VALEURS PROPRES *
*****

330 CALL "SPEIGEN" ( C(I),P(I) )
340 PRINT " VALEUR PROPRE ET POURCENTAGE D'INERTIE" # PRINT
350 FOR I=1 TO N1
360   PRINT USING "3X,"UP = ",.00.D00,3X,"% INERTIE = ",.00.D0" : P(I):P(I)*100/SUM(P)
370 NEXT I
380 PRINT # PRINT " VECTEURS PROPRES" # PRINT
390 MAT PRINT USING "S(XX,NO.D00)" : C
400 *****
   * 1 PROJECTION DES VARIABLES *
*****

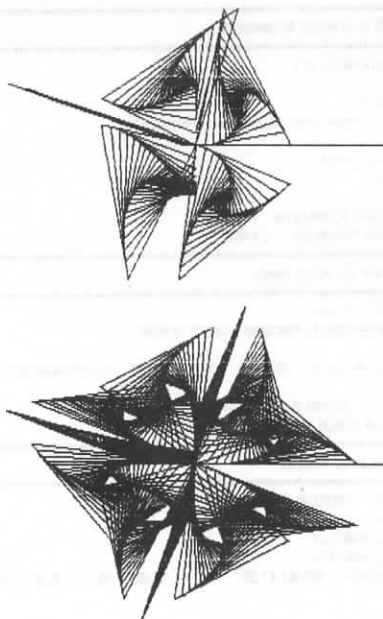
405 PRINT # PRINT " PROJECTION DES VARIABLES" # PRINT
410 FOR I=1 TO N1
425   C1=C(I,1)+SQRT(P(1))
426   C2=C(I,2)+SQRT(P(2))
430   PRINT USING " " VARIABLE # ".DO," AXE 1 : ",M.D0," AXE 2 : ",M.D0" : I,C1,C2
440 NEXT I

```

```

500 *****
    * I GRAPHE DES VARIABLES
    *****
510 DIM AB(51,21),I1,I2
520 DEF FNG(L,C,B)
530 AB(L,C)=B# FNG=0 # END DEF
550 FOR L=1 TO 51 # FOR C=1 TO 51 # AB(L,C)=0 # NEXT C # NEXT L
560 FOR L=1 TO 51 # Y=FNG(L,26,"") # NEXT L
570 FOR C=1 TO 51 # Y=FNG(26,C,"") # NEXT C
580 Y=FNG(26,26,"")
600 FOR I=1 TO M1
610 C1=INT(C(I,1)*SQR(P(1))*25*26)
620 C2=51-INT(C(I,2)*SQR(P(2))*25*26)
630 Y=FNG(C2,C1,STR$(I))
640 NEXT I
1000 *****
    * I IMPRESSION
    *****
1004 PRINT CHR$(12)
1005 PRINT CHR$(27)"0190"
1010 FOR L=1 TO 51
1020 FOR C=1 TO 51
1030 PRINT AB(L,C);
1040 NEXT C
1050 PRINT
1060 NEXT L

```



NAVIGATION

NAVIGATION

Je voudrais vous soumettre ci-après un bien modeste programme qui, toutefois, ne rends de grands services lors de la préparation de mes navigations aériennes.

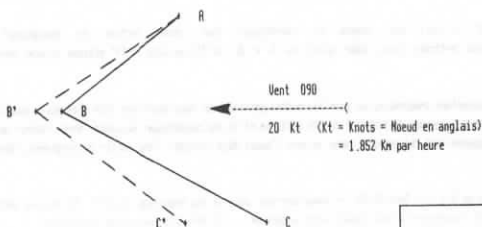
Ce programme ne s'intéresse pas à la préparation 'statique' de la navigation: c'est-à-dire que le tracé de la route a déjà été reporté sur la carte et les caps magnétiques calculés. De plus, étant donné que je pratique la navigation à vue, les distances entre repères ont été déterminées.

Ces éléments constituent l'étape préalable qui précède la préparation 'dynamique' dans laquelle notre 71 excelle tout particulièrement (mais a-t-il vraiment beaucoup de mérites dans ce cas précis...). En effet, avant tout voyage aérien, il est absolument impératif de se renseigner auprès d'un Centre Météo afin de connaître les conditions atmosphériques en cours !

Les données qui sont transmises sont diverses. Parmi celles-ci nous intéressent particulièrement, dans ce programme, la direction du vent et sa force: la direction du vent est toujours exprimée en degrés à partir de son origine: un vent du 180 est un vent qui vient du Sud et qui souffle plein Nord. La force du vent quant à elle est exprimée en noeuds...

Ne soyez pas surpris: en Aviation ne régnent que des unités de barbares... Je ne citerais que les distances en milles, les altitudes en pieds et les vitesses en noeuds !!! (Les pleins d'essence se font quand même en litres et non pas en gallons).

Le meilleur moyen pour illustrer le fonctionnement du programme est de prendre un exemple: imaginons une navigation partant du point A et se rendant au point C en passant par B. Nous aurons donc déterminé sur la carte que la distance entre A et B est de 35 Kts et que pour aller de B à C il faut prendre un cap magnétique théorique au 230, c'est-à-dire en fait s'il n'y a pas de vent. Il faut préciser que ce cap magnétique théorique est le résultat de ce que l'on appelle la Route Vraie, qui est déterminée sur la carte à l'aide d'un rapporteur, à laquelle a été appliquée la déclinaison magnétique du lieu (par ex. - 5° Ouest en Région Parisienne). Puis de B à C, nous aurons calculé une distance de 50 Kts avec une route au 120.



Malheureusement, tout cela n'est que théorie, car au moment du départ, le météorologiste nous a indiqué un vent du 090 soufflant à 20 noeuds. Si nous conservons les caps magnétiques calculés sur la carte, de A nous nous retrouverions en B' et de B' en C': gênant, non ?

Il va donc falloir rectifier ces caps et recalculer les temps de vol en fonction du vent, car ces derniers se trouvent augmentés lorsque l'on vole avec le vent dans le nez et diminués dans le cas contraire.

Le programme nous interrogera donc tout d'abord sur la direction du vent qui devra être entrée en degrés, puis sur sa force en noeuds. Ces 2 données, direction et force, sont supposées définitives et sans changement sur le parcours entre les repères:

elles ne seront donc entrées qu'une seule fois au début de l'exécution. Enfin, nous indiquerons la distance entre A et B (35 Kns) et le cap magnétique (230).

Il faudra alors choisir pour l'édition des résultats entre l'affichage sur l'écran et l'impression sur une imprimante. A ce propos, je dois vous indiquer que je ne sers pour cela de l'imprimante thermique HP 82162 A car celle-ci est portable, peu encombrante et bien suffisante pour cette utilisation.

Les résultats suivants apparaîtront donc, soit sur l'écran, soit sur l'imprimante :
(pour l'écran, appuyez à chaque fois sur une touche)

T.S.U	=	14'
Dérive (X)	=	14.4°
Dérive sur axe	=	- 9.3°
Correction temps	=	- 11"/"
Temps corrigé	=	11' 25"
Nouveau cap magn.	=	221°

(au lieu de 230)

Le T.S.U, ou Temps Sans Vent, est calculé en multipliant la distance par le facteur de base (fb). Ce facteur de base est un rapport établi en divisant 60 par la vitesse de croisière de l'avion (sans vent). Les avions sur lesquels je vole ayant une vitesse moyenne de 150 Kns/h., le facteur de base est donc $60 / 150 = 0.4$. Celui-ci, ayant été placé dans une variable en début de programme, peut-être facilement modifié et adapté à tous types d'avions. Ce T.S.U est donc le temps que mettrait l'avion pour parcourir la distance de 35 Kns séparant A de B s'il n'y avait pas de vent...

Ensuite nous avons besoin de connaître quelle serait la dérive maximale, appelée X, si l'avion se trouvait en plein travers, à 90°, par rapport au vent. Celle-ci serait de 14.4°. Ce résultat nous sert après pour déterminer la correction d'axe que nous devons effectuer pour conserver notre route entre A et B. Cette correction s'appelle la dérive sur l'axe et elle est ici de - 9.3°, c'est-à-dire qu'au lieu de garder un cap au 230, qui nous mènerait en B', nous allons suivre un cap magnétique au 221; ce que nous confirmera l'ordinateur avec son dernier résultat.

Pour les puristes, que vous êtes certainement, je dois préciser que 221 est un résultat arrondi et que dans la pratique le cap à suivre serait même arrondi à 220 car il est très difficile en vol à vue de conserver une route à 5° près.

La correction de temps donnée ensuite par le programme est exprimée en secondes par minutes de vol, bien entendu en plus ou en moins selon que l'on vole contre le vent ou avec celui-ci dans le dos. Ici le vent nous pousse donc nous irons plus vite et mettrons moins de temps pour aller de A à B que s'il n'y avait pas de vent. La correction de temps est alors négative et sera à soustraire du résultat T.S.U.

En fait, jusqu'à maintenant, ce ne sont que des résultats intermédiaires qui ont été indiqués, pour contrôle du bon fonctionnement du programme, et seuls vous intéresseront pour notre navigation les résultats finaux exprimés dans les deux dernières lignes.

Le temps corrigé, quant à lui, est obtenu en soustrayant (sur cette portion de navigation) au T.S.U (14') : $14 \times (-11) = 154$. Nous mettrons donc, pour aller de A à B, 11'25" au lieu de 14' puisque le vent nous est favorable.

Pour l'autre partie de cette navigation imaginaire, ou pour les autres parties, car nous pourrions très bien aller ensuite de C à D puis de D à E..., le programme nous interroge directement sur la distance et le cap magnétique suivants sans revenir sur la force et la direction du vent qu'il suppose sans changement, comme je vous l'avais déjà indiqué. Pour sortir du programme, répondez donc 0 pour la distance !

Ah ! J'oubliais, pour se rendre de B à C il faut 24'09" en temps corrigé avec un cap magnétique au 113. Si tels ne sont pas les résultats que vous délivre votre 7i, revérifiez à tout hasard votre programme... Et attention aux vents trop forts !...

J'espère que cet exposé un peu long vous aura intéressé et que certains d'entre vous trouveront, si ce n'est l'utilisation de ce programme, peut-être l'envie de l'appliquer en situation réelle "sur le terrain" (d'aviation bien sûr...).

A bientôt dans les airs de notre chère vieille France

Jean-Marie SIMON P 160 T 516

PROGRAMME NAU

```

100 DESTROY ALL @ STD @ DELAY .5,125
110 Z=CHR$(27)*"BSG"CHR$(0)CHR$(0)CHR$(23)CHR$(21)CHR$(23)
120 ! Définition du symbole degré pour l'ignorante
130 IMAGE 15a,ddd,d,10a
140 IMAGE 13a,2x,add
150 IMAGE 16a,dd,a,x,zz,a
160 IMAGE 15a,ddd,d,a
170 IMAGE 13a,2x,add,3a
180 IMAGE 15a,dd,"",zz,a
190 A=0 @ B=.4 ! Facteur de base
200 ! -----
210 INPUT "Dir. vent : ";U
220 IF U>360 THEN 210
230 INPUT "Force vent : ";F
240 IF F>50 THEN DISP "Vent trop fort..." @ WAIT 2 @ GOTO 230
250 INPUT "Distance : ";D
260 IF D=0 THEN DISP "Fin..." @ WAIT 2 @ PUT "R43" @ END
270 IF D>100 THEN 250
280 INPUT "Cap magn. : ";R1
290 IF R1>360 THEN 280
300 DISP "Ecran / Ignorante"
310 A$=KEY$ @ IF A$="" OR A$("<") OR A$(">") THEN 310
320 IF A$="I" THEN RESTORE IO @ PRINTER IS=PRINTER
330 ! ----- C A L C U L S -----
340 T=U*B
350 O1=B*(90*(2+T))
360 O2=O1+SIN(U-R1)
370 C1=O1+COS(U-R1)
380 C2=R1+C1/60*1
390 R2=R1+O2 @ R2=IP(R2*.5+SGN(R2)) @ IF R2>360 THEN R2=R2-360
400 R2$=STR$(R2)
410 IF LEN(R2$)<3 THEN LET R2$="0"R2$ @ GOTO 410
420 IF A$="I" THEN PRINTER IS=PRINTER @ GOSUB "IMPR" ELSE GOSUB "ECRAN"
430 GOTO 250
440 ! ----- I M P R E S S I O N -----
450 "IMPR":
460 IF A=1 THEN 520
470 U$=STR$(U)
480 IF LEN(U$)<3 THEN LET U$="0"U$ @ GOTO 480
490 PRINT "Direction vent = ";U$:Z$
500 PRINT "Force = ";F:"kt"
510 FIX 1 @ PRINT "F. de base (Fb) =";B @ PRINT "-----" @ PRINT @ A=1
520 STD @ PRINT "Distance = ";D:"km"
530 R1$=STR$(R1)
540 IF LEN(R1$)<3 THEN LET R1$="0"R1$ @ GOTO 540
550 PRINT "Cap magn. = ";R1$:Z$ @ PRINT
560 STD @ PRINT "T.S.U = ";
570 IF T>60 THEN PRINT IP(T/60):"h";
580 PRINT IP(MOD(T,60)*.5):"" @ PRINT
590 FIX 1 @ PRINT USING 130:"Derive (X) =";.01,Z$
600 IF ABS(O2)<3 THEN PRINT "Der. s/axe négligeable" @ GOTO 620
610 PRINT USING 130:"Der. s/axe =";.02,Z$
620 IF ABS(IP(C1))<5 THEN PRINT "Corr. temps négligeable" @ PRINT @ GOTO 640
630 STD @ PRINT USING 140:"Corr. temps =";IP(C1):"/h""
640 STD @ PRINT USING 150:"T. corrige =";IP(C2):"/h""
650 PRINT "Nouv. Cap magn. = ";R2$:Z$
660 PRINT @ PRINT "-----"
670 RETURN
680 ! ----- E C R A N -----

```

```

690 "ECRAN":
700 DELAY @ B=Z=CHR$(1)
710 DISP "T.S.U = ";
720 IF T>60 THEN DISP IP(T/60):"h";
730 DISP IP(MOD(T,60)*.5):""
740 FIX 1 @ DISP USING 160:"Derive (X) =";.01,Z$
750 IF ABS(O2)<3 THEN DISP "Der. s/axe négligeable" @ GOTO 770
760 DISP USING 160:"Der. s/axe =";.02,Z$
770 IF ABS(IP(C1))<5 THEN DISP "C. Temps négligeable" @ GOTO 790
780 STD @ DISP USING 170:"C. Temps =";IP(C1):"/h""
790 STD @ DISP USING 180:"T. corrige =";IP(C2):"/h""
800 DISP "Nouv. Cap magn. = ";R2$:Z$
810 DELAY .5,125 @ Z=CHR$(1) @ RETURN

```

FORMULES UTILISEES

$$Fb = \frac{60}{Vit. croisière}$$

$$\text{Dérive (X)} = Fb \cdot Vit. Vent$$

(en degrés) (en Kns/H)

$$Vit. Vent = (2 \cdot Vit. Vent) - 10X$$

(en Kns/H) (en noeuds)

$$\text{Dérive sur l'axe} = X \cdot \sin \beta$$

(en degrés)

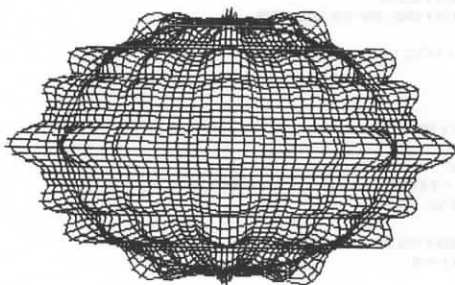
β étant l'angle au vent (angle formé par l'axe de l'avion avec la direction du vent).

$$\text{Correction temps} = X \cdot \cos \beta$$

(en " / de vol)

$$T.S.U = Fb \cdot \text{Distance (en Kns)}$$

$$\text{Temps corrigé} = T.S.U : (\text{Correct. temps} \cdot T.S.U)$$



AMPISTAT

Jean Claude LIEHN T 488
8 rue de l'écaille
51100 Reims
Tel: 26 85 50 52

le 26 02 86

LE MODULE AMPISTAT DU HP 71

AMPI signifie American Micro Products Inc., nom de la firme qui a développé ce module. Cette origine ne se voit pas au niveau du module qui a toutes les apparences d'un module HP, par contre le manuel (217 pages en anglais) n'a pas le format bizarre des manuels du 71. La qualité de ce manuel est cependant au standard HP: bien fait et clair, avec des exemples détaillés pour toutes les fonctions.

Ce module de 32 K permet la saisie des données, la réalisation de calculs statistiques sur ces données et le calcul des distributions usuelles, le tout s'enchaînant sous forme de menu.

LA SAISIE ET L'EDITION DES DONNEES:

Le programme de saisie et d'édition des données est très voisin de celui du module CURVE FIT, c'est à dire qu'il est assez bien conçu. Il présente certaines des fonctions de base d'un tableur. Les colonnes peuvent être nommées, ce qui rend la saisie plus agréable. Le touche END LINE permet de passer automatiquement à une cellule voisine de son choix, le tableau de données pouvant être rempli ligne par ligne ou colonne par colonne. On peut supprimer ou ajouter une ligne ou une colonne. Mieux, on peut remplir une colonne par des valeurs calculées à partir des valeurs contenues dans le tableau. On peut enfin trier les données. Parler de micro-tableur ne me paraît pas exagéré.

Les tableaux statistiques permettent la saisie de données groupées ou non.

Les données peuvent être écrites ou lues dans un fichier HPAF en mémoire centrale ou sur une mémoire de masse.

LES TRAITEMENTS STATISTIQUES

Sur ces données, les calculs suivants peuvent être effectués:

Statistique descriptive:

- Calcul des moyennes arithmétique, géométrique et harmonique.
- Calcul des moments d'ordre 1, 2, 3 et 4.
- Calcul des coefficients d'aplatissement (Kurtosis) et d'asymétrie (Skewness).
- Constitution d'un histogramme.

Comparaison de distributions:

- Comparaison d'un histogramme observé à une des lois de probabilité usuelles.

Comparaison de deux moyennes:

- Test de Student (t-test) apparié ou non.

Comparaison de plusieurs moyennes:

- Analyse de variances à un ou deux facteurs.

Comparaison de fréquences et de répartitions (Analyse d'un tableau de contingence)

- test du KHI-DEUX.

Regression et Corrélation linéaire.

- Regression linéaire multiple avec ou sans coefficient constant.

Statistique non paramétrique:

- Test U de Mann-Withney.
- Test de Kruskal-Wallis.

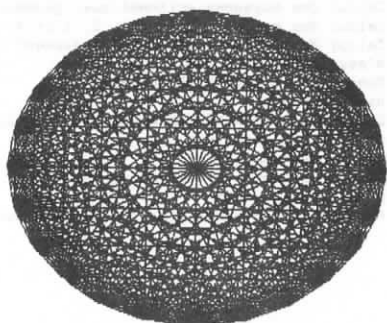
Calculs des distributions usuelles:

Pour les lois suivantes, sont données les probabilités $p(u)$, $p(x < u)$ et $p(x > u)$. Pour certaines lois, les relations inverses (trouver la valeur de u pour $p(x < u)$ donné) sont fournies.

- Distribution de Student.
- Distribution F.
- KHI-DEUX.
- Loi Normale.
- Loi de Weibull.
- Loi exponentielle.
- Loi binomial.
- Loi de Poisson.

Enfin l'ensemble des calculs statistiques et des lois de probabilité sont réalisés par des sous-programmes bien documentés, ce qui permet d'envisager d'utiliser ce logiciel comme une "boîte à outils".

Mon impression est donc très favorable et je pense que ce logiciel peut permettre du travail sérieux pour un coût assez raisonnable. J'ajoute que je n'ai pas testé l'ensemble des programmes et je ne prétends pas que tous les calculs sont correctement faits, mais l'ensemble inspire confiance. Mon seul regret est l'absence d'intégration des modules du 71. A quoi bon développer des modules MATH extrêmement puissants et qui font du 71 et du 75 des machines très originales si, pour une application typique du calcul matricielle, on n'utilise pas cette puissance? En acceptant de rendre ce module statistique dépendant du module mathématique, on aurait pu aller beaucoup plus loin en 32K et, en tout cas, beaucoup plus vite.

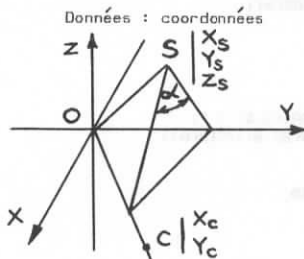


RECTILI

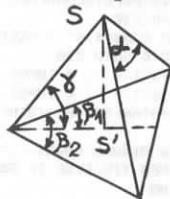
RECTILI

TOUBLANC Guy (T 719)
21 rue H. Becquerel
59120 LOOS

Le programme, de Gilbert Blanchet, "ANGLE" (MR8p25) permet de déterminer l'angle de deux demi-plans (rectiligne du dièdre).
Voici une version pour HP71.



données : angles



EXEMPLES

1) option : coordonnées et
degrés sexagesimaux

```
RUN (S)exa ou (D)eci 5
(C)oord. ou (A)ngles C
Yc= 4 Xc= 12
Xs= 5 Ys= 9 Zs= 18
alpha = 79°10'50"
autre dièdre (O/N) 0
```

2)

```
(C)oord. ou (A)ngles C
Yc= 0
Xs= 5 Ys= 7 Zs= 18
alpha = 95°34'00"
autre dièdre (O/N) N
(FIN)
```

3) option : angles et
degrés décimaux

```
RUN (S)exa ou (D)eci D
(C)oord. ou (A)ngles A
beta1= 25 beta2= 35 gamma = 45

alpha = 47.4051
autre dièdre (O/N) N
(FIN)
```

RECTILIGNE

```

100 I Rectiligne du dièdre / Guy Toubianc (T 719) 13/2/86
105 DISP "RECTILIGNE" @ DESTROY ALL
110 DEGREES @ CFLAG 0,-15
115 INPUT "O(S)exa ou (D)éci ", "S";R$
120 IF R$="D" THEN SFLAG 0 ELSE IF R$="S" THEN 115
125 INPUT "(C)oord. ou (A)ngles ", "C";R$
130 IF R$="A" THEN 210 ELSE IF R$="C" THEN 125
135 INPUT "Yc = ", "0";R$ @ Y=VAL(R$)
140 IF Y=0 THEN B=0 @ GOTO 150 ELSE INPUT "Xc = ";X
145 B=ATAN(Y/X)
150 INPUT "Xs = ";X @ INPUT "Ys = ";Y
155 INPUT "Zs = ";Z @ Z=ATAN(Z/SQR(X^2+Y^2)) @ B1=ATAN(X/Y)
160 A=ABS(ATAN(X/Y/SIN(Z))+ATAN(1/SIN(Z)/TAN(B+B1)))
165 IF FLAG(0) THEN 200
170 B=FP(A)*360 @ B1=IP(B/60) @ B=IP(B-B1*60)
175 A=IP(A) @ IF B>9 THEN B$="" ELSE B$="0"
180 IF B1>9 THEN B1$="" ELSE B1$="0"
185 DISP CHR$(4)&" = "&STR$(A)&CHR$(1)&B1$&STR$(B1);
190 DISP ""&B$&STR$(B)&CHR$(34)
195 WAIT 3 @ GOTO 230
200 FIX 4 @ DISP CHR$(4)&" = "&STR$(A)
205 WAIT 3 @ STD @ GOTO 230
210 DISP CHR$(5)&"I = "; @ INPUT B1
215 DISP CHR$(5)&"Z = "; @ INPUT B2 @ DISP CHR$(6)&" = ";
220 INPUT G @ A=ATAN(TAN(B1)/SIN(G))+ATAN(TAN(B2-B1)/SIN(G))
225 GOTO 165
230 INPUT "autre dièdre (O/N) ", "0";R$
235 IF R$="0" THEN 125 ELSE IF R$="N" THEN 230
240 CFLAG 0 @ END

```

FORTH ET LE LANGAGE MACHINE

PAR J. L. LAFONT

Professeur à l'Université de
Paris VI

et à l'École Supérieure de
Génie des Télécommunications

et à l'École Supérieure de
Génie des Télécommunications

Le langage FORTH est un langage de programmation qui a été développé par J. L. LAFONT. Il est basé sur le langage machine et permet de programmer des machines à microprocesseurs. Il est très simple et efficace, et permet de programmer des machines à microprocesseurs de manière très simple et efficace. Il est très simple et efficace, et permet de programmer des machines à microprocesseurs de manière très simple et efficace.

Le langage FORTH est un langage de programmation qui a été développé par J. L. LAFONT.

Il est basé sur le langage machine et permet de programmer des machines à microprocesseurs. Il est très simple et efficace, et permet de programmer des machines à microprocesseurs de manière très simple et efficace.

FORTH

Le langage FORTH est un langage de programmation qui a été développé par J. L. LAFONT. Il est basé sur le langage machine et permet de programmer des machines à microprocesseurs. Il est très simple et efficace, et permet de programmer des machines à microprocesseurs de manière très simple et efficace.

Il est basé sur le langage machine et permet de programmer des machines à microprocesseurs. Il est très simple et efficace, et permet de programmer des machines à microprocesseurs de manière très simple et efficace.

Le langage FORTH est un langage de programmation qui a été développé par J. L. LAFONT. Il est basé sur le langage machine et permet de programmer des machines à microprocesseurs. Il est très simple et efficace, et permet de programmer des machines à microprocesseurs de manière très simple et efficace.

Il est basé sur le langage machine et permet de programmer des machines à microprocesseurs. Il est très simple et efficace, et permet de programmer des machines à microprocesseurs de manière très simple et efficace.

Le langage FORTH est un langage de programmation qui a été développé par J. L. LAFONT. Il est basé sur le langage machine et permet de programmer des machines à microprocesseurs. Il est très simple et efficace, et permet de programmer des machines à microprocesseurs de manière très simple et efficace.

Il est basé sur le langage machine et permet de programmer des machines à microprocesseurs. Il est très simple et efficace, et permet de programmer des machines à microprocesseurs de manière très simple et efficace.

FORTH ET LE LANGAGE MACHINE

FORTH ET LE LANGAGE MACHINE

par

Marc PETREMANN

Secrétaire de l'ASSOCIATION JEDI

8, rue Poirier de Narçay 75014 PARIS

La réputation de rapidité du langage FORTH n'est plus à faire. Cependant, au cours de vos diverses expériences, vous avez certainement abordé des problèmes dont le temps d'exécution était parfois critique et qui auraient trouvé une solution élégante s'ils avaient été développés en assembleur plutôt qu'en langage FORTH. De tels cas critiques apparaissent fréquemment lors de la création de routines graphiques ou de processus s'exécutant en temps réel, telle la synthèse numérique de sons, saisie de données à partir d'un convertisseur A/D D/A, etc...

IMPLANTATION DU CODE MACHINE

Dans le présent exposé, je ferai fréquemment appel à des exemples tournant sur le μP 6809, mais dans la mesure du possible, je donnerai aussi quelques exemples en 8080 et 6502.

1er cas, vous ne disposez pas d'assembleur sur votre version FORTH. Dans ce cas, vous devez faire appel aux mots `_` et `C_` pour implanter le code binaire de votre sous-programme assembleur, avec un avantage par rapport à BASIC, c'est de ne pas avoir à vous préoccuper de l'adresse d'implantation de ce code.

Le mot `_` plante un nombre huit bits dans le dictionnaire.

Le mot `C_` plante un nombre seize bits dans le dictionnaire.

Le pointeur dictionnaire est automatiquement incrémenté de une ou deux unités en fonction du nombre compilé. Exemple:

```
HERE .      affiche l'adresse du pointeur de dictionnaire => adr
3 C,        compile 3 à l'adresse adr.
HERE .      affiche la nouvelle adresse du pointeur de diction-
              naire, soit adr+1
```

on vérifie que 3 est bien implanté à adr en tapant:

```
HERE 1- C@ .      affiche 3
```

Ainsi peut-on implanter du code machine directement dans le dictionnaire FORTH, et même réaliser l'appel de ce code directement depuis FORTH, à l'aide d'un en-tête qu'il suffit de taper.

Mais voyons d'abord comment est défini une primitive dont la définition est déjà écrite en code machine. Tiens, essayons avec DUP :

```
' DUP        délivre le pfa de DUP => adr
' DUP CFA    délivre le CFA de DUP => soit adr-2
' DUP CFA @  délivre le contenu du CFA de DUP.
```

On constate que le contenu du CFA de DUP pointe sur le PFA de DUP. Ceci n'est pas le cas des mots définis par "deux-points" (:), VARIABLE, CONSTANT, USER ou tout autre mot de définition existant sur votre système FORTH.

Nous allons implanter une première définition, qui lors de son exécution, sera chargée d'exécuter un sous-programme écrit en code machine:

définition de CALL en 6809 (tous systèmes 6809):

```
VARIABLE CALL -2 ALLOT HEX
AD C, 71 C,
AE C, A1 C,
6E C, 94 C,
CALL DUP 2- ! DECIMAL
```

équivalent assembleur 6809:

```
CALL JSR ([U++])
LDX ,Y++
JMP ([X])
```

Explications:

- la séquence VARIABLE CALL crée un en-tête CALL dans le dictionnaire FORTH.
- -2 ALLOT récupère l'espace mémoire occupé par le contenu de la variable CALL.
- puis on passe en hexadécimal et on compile le code objet du programme défini ci-dessus.
- CALL DUP 2- ! met l'adresse du PFA de CALL dans le CFA de CALL ce qui transforme la variable CALL en un mot équivalent à une primitive définie en code machine, tel DUP, LIT ou (LOOP) ...

On utilisera CALL en tapant adr CALL, où adr est l'adresse du code machine à exécuter.

Remarque: l'en-tête CALL a été défini à l'aide de la séquence VARIABLE -2 ALLOT, mais aurait aussi bien pu être défini par CREATE. Si j'ai choisi cette solution, c'est parce qu'elle est applicable à tous les standards. En effet, CREATE, en 79-STANDARD, crée un en-tête de même type qu'une variable classique, mais sans affecter de zone paramétrique:

```
CREATE CALL est équivalent à
VARIABLE CALL -2 ALLOT en 79-STANDARD
```

alors qu'en FIG, CREATE définit un en-tête directement exécutable comme une primitive "code-machine":

```
CREATE CALL est équivalent à
Ø VARIABLE CALL -2 ALLOT CALL DUP 2- !
en FIG-STANDARD.
```

Autre remarque: vous constaterez de visu qu'il n'est pas nécessaire d'héberger en mémoire le programme source réalisant la compilation du code machine contrairement au langage BASIC qui doit faire appel à une zone de DATAS, ceci par l'intermédiaire d'une boucle de type FOR... NEXT ou WHILE ... REPEAT.

PASSAGE DE PARAMETRES

Un programme dont la définition contient un programme écrit en code machine peut traiter des données qui ont été préalablement

déposées sur la pile de données ou stockées dans une constante ou une variable FORTH.

Exemple: cas d'une donnée stockée dans une variable FORTH. Vous voulez créer un jeu vidéo, mais l'incrément du score est une opération dont le temps d'exécution est critique, c'est à dire qu'elle provoque un ralentissement significatif de l'exécution des autres routines. Voici tout d'abord comment une telle routine est définie en assembleur:

```
SCORE EQU    ENDMEM-200
INCSC LDD    >SCORE
        ADDD  #1
        STD   >SCORE
        RTS
```

en assembleur 6809. Après assemblage, on peut reprendre le code objet généré, (ou si vous êtes doués, faire l'assemblage "à la main") et définir le mot FORTH INC-SCORE comme suit:

```
VARIABLE SCORE
VARIABLE INC-SCORE -2 ALLOT HEX
FC C, SCORE, ( LDD >SCORE )
C3 C, 0001, ( ADDD #1 )
FD C, SCORE, ( STD >SCORE )
39 C, ( RTS )
DECIMAL
```

Le mot INC-SCORE sera exécuté en tapant la séquence INC-SCORE CALL. Exemple:

```
Ø SCORE ! SCORE ?      affiche 0
INC-SCORE CALL SCORE ?  affiche 1
```

Voici un autre exemple, réalisant la somme du contenu de deux variables, stockant le résultat dans une troisième variable:

```
VARIABLE X1
VARIABLE X2
VARIABLE X3
VARIABLE SOMMEX1X2 -2 ALLOT HEX
FC C, X1, ( LDD >X1 )
F3 C, X2, ( ADDD >X2 )
FD C, X3, ( STD >X3 )
39 C, DECIMAL
```

et l'exécution de SOMMEX1X2 donne:

```
25 X1 ! affecte 25 à X1
3 X2 ! affecte 3 à X2
SOMMEX1X2 CALL
X3 ? affiche 28 (hourrah)
```

On peut faire une première remarque au vu des différents exemples concernant certains codes revenant fréquemment, tel 39 pour RTS qui est l'équivalent code machine de RETURN. On pourrait déjà à ce stade, par souci de simplification, définir un mot RTS qui sera chargé d'implanter le code hexadécimal 39 dans le dictionnaire:

```
HEX : RTS ( --- ) 39 C, ; DECIMAL
```

On pourrait faire de même avec les autres mnémoniques utilisés ci-dessus, tels LDD, ADDD ou STD:

```

HEX
: >LDD ( adr ---) FC C, , ;
: >ADD ( adr ---) F3 C, , ;
: >STD ( adr ---) FD C, , ;

```

et on réécrit SOMMEX1X2 comme suit:

```

VARIABLE X1
VARIABLE X2
VARIABLE X3

VARIABLE SOMMEX1X2 -2 ALLOT ( ---)
X1 >LDD X2 >ADD X3 >STD
RTS

```

Le signe > qui précède nos "mnémoniques" indique le mode d'adressage étendu. Bien entendu, pour les autres modes d'adressage, le code opérateur change. On peut envisager de sélectionner un mode d'adressage en faisant précéder ledit mnémonique par le symbole de sélection et le mnémonique compile alors le code correspondant:

voici ce que donne un véritable assembleur FORTH défini pour le µP 6809 (version eFORTH de FHL):

```

VARIABLE X1 VARIABLE X2 VARIABLE X3
CODE SOMMEX1X2
X1 ldd X2 addd X3 std NEXT END-CODE

```

Le mot NEXT est une macro-instruction qui peut-être définie comme suit:

```

: NEXT ( ---)
AE C, A1 C, 6E C, 94 C, ;

```

Ainsi, l'exécution de NEXT compile automatiquement le retour au système FORTH. La même macro-instruction peut-être définie en assembleur FORTH:

```

: NEXT ( ---)
,Y++ ldx ,X ) jmp ;

```

En assembleur 8080 sous FORTH, NEXT sera défini par:

```

: NEXT >NEXT JMP ;

```

où >NEXT est l'adresse de la routine redonnant le contrôle au système FORTH. Pour connaître cette adresse sur votre système, il suffit de reprendre les deux derniers octets d'une routine définie en code machine, tel le mot ØBRANCH ou (DO). Pour pointer sur les deux derniers octets de ce mot, faites VLIST et repérez le mot qui le précède dans l'ordre d'apparition. Exemple:

si en faisant VLIST, I précède (DO), on trouve les deux derniers octets de (DO) en tapant:

```

' I NFA 2- @

```

C'est cette valeur qui sera affectée à >NEXT (sur système Z80 ou 8080). Ceci peut aussi être valable pour d'autres µP, mais je n'en ai pas fait l'essai.

Les mots suivants se terminent pratiquement à coup sûr par un branchement à >NEXT:

(DO), CMOVE, U/, AND, D+, MINUS, DMINUS, etc...

TRAITEMENT DES DONNEES DEPOSEES SUR LA PILE DE DONNEES

Mais le moyen le plus simple (si!) pour traiter des données est de les récupérer sur la pile de données. En effet, sur la majorité

des systèmes FORTH, les données transitent d'une routine FORTH à une autre en passant par la pile système ou la pile utilisateur. Sur le 6809, les données transitent par la pile U. Ainsi, la récupération des données peut être réalisée de diverses manières:

- par dépileage: PULU D dépile la valeur située au sommet de la pile de données et l'affecte au registre D.
- par l'adressage indexé: LDD ,U++ affecte la valeur située à U (pointeur de pile) au registre D et effectue une double post-incrémentation du pointeur de pile.

Exemple: définition de +!

en assembleur:

```
PSTORE    LDX    ,U++
          LDD    ,U++
          ADDD   ,X
          STD    ,X
          JMP    >NEXT
```

en FORTH sans assembleur, µP 6809:

```
VARIABLE +! -2 ALLOT HEX
AE C, C1 C,
EC C, C1 C,
E3 C, 84 C,
ED C, 84 C,
7E C, >NEXT , +! DUP 2- !
```

en FORTH avec assembleur et NEXT défini en macro-instruction:

```
CODE +!
,u++ ldx    ,u++ ldd    ,X addd
,X std     NEXT     END-CODE
```

en FORTH, avec assembleur 8080 (compatible Z80):

```
CODE +! ( n adr ---)
H POP    D POP    M A MOV    E ADD    M A MOV
H INX    M A MOV    D ADC    A M MOV    NEXT C;
```

En assembleur 6502:

```
CODE +!
CLC,      BOT X) LDA, SEC ADC,
BOT X)   STA,  BOT INC,
Ø= IF,
        BOT 1+ INC,
THEN,
BOT X)   LDA, SEC 1+ ADC, BOT X) STA,
POPTWO JMP,
```

(et je vous épargne le 8086 et le 68000)

Remarquez dans le cas de +! sur 6502, l'usage de directives d'assemblage IF, et THEN, qui ont pour but de compiler et résoudre un branchement conditionnel exprimé par Ø=, ce qui évite l'usage de labels.

La lecture de cette manière particulière d'assembler ne doit guère poser de problèmes. Pour la majorité des mnémoniques, les opérandes sont placés avant l'opérateur. Ainsi, sur le 6809, on aura:

```
,u++ ldx    qui équivaut à    LDX    ,U++
3 # lda     qui équivaut à    LDA    #3
ØØ99 CONSTANT NEXT
NEXT jmp    qui équivaut à    jmp    NEXT
```


Les pseudo-structures de contrôle utilisées en assembleur FORTH remplacent avantageusement les labels dans la majorité des cas, rendant le code source "structuré". Exemple:

Définition de U/ en assembleur conventionnel:

```

LDD 2,U
LDX 4,U
STX 2,U
STD 4,U
ASL 3,U
ROL 2,U
LDX #$10
USLL1 ROL 5,U
      ROL 4,U
      LDD 4,U
      SUBD ,U
      ANDCC #$FE
      BMI USLL2
      STD 4,U
      ORCC #1
USLL2 ROL 3,U
      ROL 2,U
      LEAX -$1,X
      BNE USLL1
      LEAU 2,U
      JMP NEXT

```

et en assembleur FORTH:

```

CODE U/  HEX
2 ,u ldd  4 ,u ldx  2 ,u stx
4 ,u std  3 ,u asl
2 ,u rol  10 # ldx
begin
  5 ,u rol  4 ,u rol  4 ,u ldd
  0 ,u subd  FE # andcc
  mi if
    4 ,u std  1 # orcc
  then
    3 ,u rol  2 ,u rol
    -1 ,x leax
  ne until
  2 ,u leau  NEXT jmp
END-CODE

```

Si vos primitives d'assemblage FORTH ont été vectorisées, il est possible de visualiser le code objet généré:

- en redéfinissant C_u et _u on peut afficher ou imprimer les octets compilés lors de l'assemblage.
- en redéfinissant HERE, on peut mettre en valeur les branchements en cours de résolution, en adressage absolu, relatif ou simultanément.

Si un sous-programme assembleur se termine par RTS (ou RET en Z80), on peut utiliser ce sous-programme depuis FORTH par l'intermédiaire de CALL ou comme sous-programme depuis un autre sous-programme assembleur. Exemple:

```

CODE PUSHD ( --- n )
d pshu rts  END-CODE

```

```

CODE I ( --- I)
0 ,s ldd ' PUSHD jsr NEXT jmp END-CODE
CODE J ( --- J)
4 ,s ldd ' PUSHD jsr NEXT jmp END-CODE
CODE K ( --- K)
8 ,s ldd ' PUSHD jsr NEXT jmp END-CODE

```

Dans ce cas, l'en-tête de PUSHD devient label dans les définitions de I, J et K (définitions pour 6809 - THOMSON ou VEGAS). Le passage de paramètres d'un programme à un sous-programme peut être réalisé par passage de données via la pile ou les variables, ceci depuis l'interpréteur FORTH vers un sous-programme assembleur, ou inversement, ou entre deux programmes assembleurs.

Cette nouvelle méthode d'assemblage permet un contrôle point par point du code généré, s'interface facilement entre les autres définitions de FORTH et permet d'utiliser toutes les ressources du système FORTH. Il est tout à fait logique de ne point recourir à l'assembleur pour les routines n'exigeant aucune rapidité d'exécution, tels les menus de présentation (là où il y a des fonctions KEY ...) ce qui ne contraint pas le programmeur à écrire en langage machine une application de bout en bout (chose éprouvante et ô combien laborieuse).

EXTENSION ASSEMBLEUR AU FORTH

J'ai eu souvent l'occasion de constater que la majorité des systèmes FORTH implantés sur les micro-ordinateurs ne disposaient que d'un assembleur minimal, voire pas d'assembleur du tout. Ayant déjà soulevé brièvement le problème de la définition d'un assembleur sous FORTH dans une précédente page, je soumet quelques utilitaires permettant l'implantation dudit assembleur. Bien entendu, comme chaque micro a ses propres mnémoniques, je donnerai pour cette partie que des directives générales de création et la bibliographie où les listings source des assembleurs pour la majorité des processeurs existants.

Définitions communes suggérées pour commencer son assembleur:

```

VOCABULARY ASSEMBLER IMMEDIATE
FORTH DEFINITIONS
: CODE
CREATE ( remplacer par <BUILDS en FIG-standard>
HERE DUP 2- !
(COMPILE) ASSEMBLER ; IMMEDIATE
: END-CODE ( ---)
CURRENT @ CONTEXT ! ;
: ;CODE ( ---)
?CSP COMPILE (;CODE)
(COMPILE) (
(COMPILE) ASSEMBLER
(COMPILE) SMUDGE ; IMMEDIATE

```

ASSEMBLER DEFINITIONS

(et on continue avec les mnémoniques d'assemblage).

Le mot CODE permet de créer un en-tête dans le dictionnaire. Ce mot est suivi de mnémoniques d'assemblages. Les opérations sur labels et adresses sont autorisées (+, -, SWAP, DUP etc ...). La séquence d'assemblage se termine par END-CODE. Pour exemples, voir plus

haut.

Les mnémoniques sont ensuite implantés dans le vocabulaire ASSEMBLER. Si votre FORTH fait la différence entre les mots tapés en caractères majuscules de ceux tapés en minuscules, il faut définir de préférence les mnémoniques en minuscules, ceci permettant la discrimination entre nombre hexadécimal et mnémoniques (ADDA et adda sur 6809, DE et de en Z80, etc...).

Liste des ouvrages et documents pour les différents assembleurs:

ASSEMBLEUR 6502: magazine JEDI n° 10.

ASSEMBLEUR 8080 version 1: magazine JEDI n° 14.

ASSEMBLEUR 8080 version 2: Listing d'implantation F83 pour 8080 disponible chez ANGELIKA FLESCHE, Schützenstraße 3, D-7820 TITISEE-NEUSTADT RFA (BRD).

ASSEMBLEUR 6809: pour THOMSON T07 prochainement dans JEDI.

ASSEMBLEUR 8086: pour IBM prochainement dans JEDI.

ASSEMBLEUR Z80: LE CONCEPT FORTH par Pascal COURTOIS, éditions TEST

ASSEMBLEUR 68000: pour ATARI 520 ST, FORTH niveau I. disponible chez ANGELIKA FLESCHE (voir plus haut), assembleur compatible QL.

Et si vous êtes courageux, vous pouvez bien entendu créer vous même votre propre assembleur. Dans l'ensemble, c'est moins difficile que de le faire en assembleur (il y a des fous partout... prendre cette remarque au second degré...). En fait, cela dépend des modes d'adressage de votre micro-processeur; facile et court pour le 8080 et long et malaisé pour le 6809 à cause du mode d'adressage indexé. La pratique de FORTH sur des machines très répandues dans certains milieux professionnels ont formalisés certains assembleurs. Ainsi, plutôt que de réinventer la roue, vous pouvez vous inspirer de la syntaxe d'assembleurs existants à défaut de les recopier (si vous pouvez transférer la version source). Les versions les plus utilisées sont l'assembleur 6502 sur APPLE II (P.LUTUS ou CAPTAIN FORTH) l'assembleur 8086 pour IBM FORTH, l'assembleur 68000 pour ATARI 520 ST, utilisé également sur QL et Mac INTOSH. Plus controversés sont les versions pour Z80, car les professionnels américains lui préfèrent les versions 8080, ceci à cause du CP/M. Je conseille de reprendre celui du livre de Pascal COURTOIS ou de s'inspirer de celui utilisé sur le HECTOR HRX. Il existe également une version moins répandue utilisée par le métacompileur de LMI. Enfin, pour les systèmes 6809, vous pouvez reprendre celui du GOUPI FORTH, mais il comporte quelques erreurs. Cette version admet une syntaxe conventionnelle, ce qui interdit la définition de macro-instructions. Je lui préfère la version en cours d'élaboration pour THOMSON et qui a servi à illustrer cet article (sera prochainement diffusé dans JEDI).

Le mot ;CODE permet de définir un mot de manière similaire aux mots définis par CREATE ... DOES ou <BUILDS ... DOES>, mais en faisant suivre ;CODE par une définition en assembleur. Exemple:

```
: 2CONSTANT ( d --- )
CONSTANT , ;CODE
4 ,x ldd d pshu
2 ,x ldd d pshu NEXT jmp END-CODE
```

est équivalent à:

```
: 2CONSTANT
CONSTANT , DOES DUP 2- @ @ ;
```

avec l'avantage de la vitesse d'exécution pour la première version.
A noter que l'on peut tricher un peu. Ainsi, si on définit auparavant le mot 2@, on peut réécrire 2CONSTANT:

```
CODE 2@ ( adr --- d)
  pulu 2 ,x ldd
d pshu 0 ,x ldd d pshu
NEXT jmp END-CODE

: 2CONSTANT ( d ---)
CONSTANT , ;CODE
2 ,x leax
' 2@ 2+ 2+ jmp END-CODE
```

Sont également rarement définis dans les programmes définissant un assembleur, la manière de créer et gérer une table de labels. Si on accepte de sacrifier quelques octets, on peut définir les mots suivants:

```
: FORGET-LABEL
( COMPIL ) ' NFA
( HERE 40 + FORTH ) LITERAL ASSEMBLER
OVER OVER 1+ U<
IF CR " Erreur de symbole" CR DROP DROP
ELSE OVER SWAP 2- ! PFA LFA @
CONTEXT @ !

THEN :
2 ALLOT
: LABEL ( --- <nom>)
CURRENT @ HERE DEFINITIONS
( LATEST 2 - FORTH ) LITERAL ASSEMBLER
DUP @ DP ! OVER
CONSTANT HERE SWAP ! DP ! CURRENT ! ;
HERE LATEST 2 - ! 100 ALLOT
FORTH DEFINITIONS DECIMAL
```

Le mot LABEL doit être le dernier mot défini dans le vocabulaire ASSEMBLER. Le nombre de labels est limité à la taille définie par le paramètre situé avant ALLOT dans l'avant-dernière ligne définie ci-dessus. Les mêmes noms de labels peuvent être réutilisés à condition de faire un FORGET-LABEL <nom>. Tous les labels définis après celui qui est oublié sont également oubliés. Exemple:

```
ASSEMBLER LABEL DOCOL
y pshs 2 ,x leax
LABEL NEXT ,y++ ldx
LABEL NEXT3 ,x ) jmp
CODE EXECUTE
x pulu NEXT3 bra FORGET-LABEL NEXT3 END-CODE
```

Vous avez ici le listing de la moitié de l'interpréteur interne de FORTH défini à l'aide de l'assembleur FORTH (dans la majorité des systèmes FORTH, l'interpréteur interne représente moins de 40 octets).

PRINCIPES FONDAMENTAUX DE L'ASSEMBLAGE CROISE

Si on part du principe que le code généré par l'assembleur FORTH est exécutable sur le système hôte, ce même code peut être translaté à l'aide d'une pseudo-définition de HERE qui délivre une adresse correspondant à la somme de l'adresse de compilation physique et le décalage vers la machine virtuelle ou machine cible. Ainsi est-il possible de compiler du code depuis FORTH qui s'exécutera ensuite sous BASIC (ou PASCAL ou CP/M) sans avoir besoin de FORTH.

Définition d'une cible virtuelle:

0 CONSTANT TARGET-ORIGIN

: THERE TARGET-ORIGIN + (tadr --- adr) ;

VARIABLE DP-T

: HERE-T DP-T @ ;

: ALLOT-T DP-T +! ;

: C,-T (c ---)

THERE C! ;

: !-T THERE ! ;

: C,-T HERE-T C!-T 1 ALLOT-T ;

: , -T HERE-T !-T 2 ALLOT-T ;

puis on redéfinit les mots C, et HERE:

: C, C,-T ;

: , -T ;

: HERE HERE-T ;

et on recompile son assembleur. Le programme à assembler est d'abord assemblé dans le système sans décalage, en initialisant les paramètres du méta-assembleur:

DP @ DP-T !

Si tout se passe bien, votre programme doit tourner sans problème à l'adresse où vous l'avez implanté. Maintenant, recherchez l'adresse de fonctionnement prévue sans FORTH. Ainsi, si l'exécution est prévue sous BASIC à partir de adrB, l'initialisation du méta-assembleur est faite par:

adrB DP-T ! DP @ adrB - ' TARGET-ORIGIN !

et l'assemblage génère un code objet qui sera implanté physiquement à la même adresse que précédemment, mais avec des branchements absolus décalés. On sauvegarde en binaire la partie de dictionnaire qui a été assemblée, puis on la récupère sous BASIC. Si tout s'est bien passé, votre programme doit tourner sans problème.

PRINCIPES FONDAMENTAUX DE LA META-COMPILATION

La méta-compilation est un peu plus complexe. Elle permet dans son principe la génération d'un noyau FORTH minimal (l'interpréteur interne en général), et la définition de mots similaires à ceux définis dans le dictionnaire FORTH d'origine, mais dont les codes sont connectés au noyau généré par le méta-compilateur.

Deux options sont envisageables, la génération d'un FORTH minimal, avec des définitions sans en-têtes, et autorisant seulement l'utilisation du code généré. Bien entendu, une fois le noyau créé, l'application est écrite en FORTH et non en assembleur. L'application ne nécessite plus la présence de FORTH car elle possède son propre interpréteur interne et ne s'encombre pas des définitions de gestion du dictionnaire (ALLOT, VLIST, HERE, etc...). Déjà, une version FORTH pour MSX a été générée par ce principe à partir d'un IBM PC.

La seconde option est de permettre la génération d'un système FORTH complet, avec éditeur, assembleur et ...méta-compilateur (eh...oui) à un nouveau standard. C'est par ce principe que les nouveaux listings des versions FORTH 83-STANDARD peuvent être implantés. Un méta-compilateur bien au point peut permettre la génération d'un code exécutable vers n'importe quel système à condition de disposer de

l'assembleur adéquat et du programme source du noyau minimal. Le reste de l'application est intégralement portable.

Il est tout à fait envisageable de générer des programmes exécutables sous CP/M à partir d'un ORIC, d'un T07 ou d'un IBM PC. Ceci est également vrai depuis un petit système vers un gros système. Un APPLE, IBM, THOMSON peut générer des programmes qui s'exécuteront sur un système très important et dont l'immobilisation prolongée pour maintenance serait financièrement préjudiciable (je pense à des systèmes servant de serveurs ou des systèmes de gestion).

La génération du langage FORTH vers les très petits systèmes est également intéressante. En effet, les systèmes dits familiaux, ne permettent pas un développement aisé, par manque d'outils (assembleur inexistant, espace mémoire insuffisant, absence de mémoire de masse autre que cassette ...). La génération du langage FORTH vers un ALICE, ZX81 ou autre prendrait quelques heures pour une version standard, et quelques jours (de l'ordre de une à quatre semaines) si on désire rajouter des options spécifiques au système cible.

Un paramétrage intégral du méta-compileur semble à l'heure actuelle encore utopique pour qu'il puisse s'appliquer sans quelque modification à l'ensemble des systèmes passés, présents ou à venir. Par contre, la possibilité offerte par certains méta-compileurs (celui de LMI notamment), permettant de générer du code ROMable, ferait la joie d'industriels dont les appareils de régulation et d'automatismes font de plus en plus appel à la programmation. La génération de programmes en EPROM est encore une opération coûteuse pour de très petites séries. Or, FORTH admet la génération d'applications écrites autrement qu'en assembleur.

CONCLUSION

Très prometteuses, les techniques de méta-compilation et de méta-assemblage sont assez faciles à mettre en oeuvre pour ceux qui pratiquent intensivement le langage FORTH. Cette technique peut même devenir stratégique pour les milieux industriels, car elle permet une réduction considérable des temps de développement d'applications qui en temps normal se calculent en MOIS/HOMME voire ANNEES/HOMMES de travail (pour information, le système interne du Mac INTOSH représente près de 150 ANNEES/HOMMES). Cette réduction des coûts, allant jusqu'à CINQUANTE FOIS le coût d'une application conventionnelle rattrape largement les investissements en temps passés à élaborer et maîtriser un méta-compileur. A une époque où la vitesse peut aller de pair avec la sécurité du logiciel, de tels arguments ne peuvent laisser indifférents. La France se situe au troisième rang du développement de logiciels à partir de méta-compileurs et est en passe de prendre la tête. L'apparition de matériels de plus en plus complexes risque de faire perdre cette avance par manque de temps et d'hommes capables de suivre.

BAIRSTOW

HP-1x

BAIRSTOW

T334

dans la recherche des limites de notre hp 15 ,limites idéalistes d'ailleurs, je me suis fixé un nouvel objectif après avoir lu l'article sur la recherche des racines par la methode de BAIRSTOW,dans SCIENCE ET VIE.Le but est aujourd'hui -ui atteint,et je me demande si cette hp15 n'a pas été sous estimée;d'autres défis seront là pour m'apporter une reponse.Lorsque je pense à la 41 qui peut être équipée d'un module "ADVANTAGE" offrant des fonctions équivalentes à celles du hp15..... ça fait peur, non?

Bref,la methode de BAIRSTOW permet de determiner toutes les racines réelles ou complexes d'un polynôme de degré au plus égal à 5.Elle consiste à calculer simultanément la somme S et le produit Pde deux racines complexes par la methode de NEWTON dans R2:

Soit un polynôme à coefficients réels,de degré au plus égal à 5: $a_0 X^5 + a_1 X^4 + \dots + a_n X + a_n$. On écrira $P_n(X) = Q_{n-1}(X) \cdot (X^2 - SX + P) + R(X)$. Si nous posons: $Q_{n-1}(X) = b_0 X^{n-1} + \dots + b_{n-2} X + b_{n-1}$ et $R(X) = b_{n-1}(X - S) + b_n$ les coefficients b_i s'écrivent:

$$\begin{aligned} b_0 &= a_0 \\ b_1 &= a_1 + S b_0 \\ b_2 &= a_2 + S b_1 - P b_0 \\ &\vdots \\ b_{n-1} &= a_{n-1} + S b_{n-2} - P b_{n-3} \end{aligned}$$

si S et P sont respectivement la somme et le produit de deux racines conjuguées, alors $R(X) = 0$; avec $R(X) = b_{n-1} X + (b_n - S b_{n-1})$; où les coefficients b_{n-1} et b_n dependent implicitement de S et P. Nous pouvons alors écrire: $F_1(S, P) = b_{n-1}$ et $F_2(S, P) = b_n - S b_{n-1}$. On doit alors résoudre le système $F_1(S, P) = F_2(S, P) = 0$ par la methode de NEWTON dont je vous épargnerai les calculs. On arrive enfin de compte à écrire que:

$$\begin{aligned} S_{i+1} &= S_i + dS_i / \Delta_i \\ P_{i+1} &= P_i + dP_i / \Delta_i \end{aligned}$$

on verifie alors que:

$$\begin{aligned} C_0 &= b_0 \\ C_1 &= b_1 + S C_0 \\ C_2 &= b_2 + S C_1 - P C_0 \\ &\vdots \\ C_{n-1} &= b_{n-1} + S C_{n-2} - P C_{n-3} \end{aligned}$$

L'algorithme sera donc le suivant:

En partant des valeurs initiales S_0 et P_0 on calcule une suite de valeurs S_i et P_i , puis on passe directement aux valeurs S_{i+1} et P_{i+1} .

On calcule les $b_0, \dots, b_n$ (avec les S_i et P_i) puis la suite $C_0, \dots, C_{n-1}$.

On écrit ensuite:

$$\begin{aligned} dS_i &= b_n C_{n-3} - b_{n-1} C_{n-2} \\ dP_i &= b_n C_{n-2} - b_{n-1} C_{n-1} \\ &= (C_{n-1})^2 - C_{n-1} C_{n-3} \end{aligned}$$

$$\begin{aligned} S_{i+1} &= S_i + dS_i / \Delta_i \\ P_{i+1} &= P_i + dP_i / \Delta_i \end{aligned}$$

Remarque:

Il peut être utile de se fixer des valeurs initiales pour P et S, pour accélérer la convergence.

Mode d'emploi:

1'introduction des données se fait a la main pour gagner des precieux registres

Donc, soit un polynôme de degré n au plus egal à 5: $P(X)=a_0X^n+....+a_n$

Faire: a) $n+1$ enter 1 DIM A,DIM B

b) n enter 1 DIM C

c)introduire les coefficients a_0, a_1,a_n dans la matrice -

[A]: MATRIX 1,mode USER

a_0 STO A

a_1 STO A

a_2 STO A

.

.

a_n STO A

d) faire CLREG, ^{Ret 8 DIM (1)}afficher le degré n puis lancer le programme au label A

e) Au premier arrêt vous aurez deux racines : l'une à l'affichage l'autre dans Y, si elles sont réelles.Si elles sont complexes,il n'y en aura qu'une a l'affichage,mais il suffira de changer le signe de la partie imaginaire pour avoir sa conjuguée.

Sans oublier qu'un polynome de degré n peut avoir au plus n racines,appuyer sur R/S jusqu'a ce que vous ayez enregistré n racines.

par exemple un polynome de degré 3 peut avoir :

3 racines réelles

ou 1 racine réelle et 1 racine complexe et sa conjuguée

Remarques:

La precision demandée est controlée au pas 119.

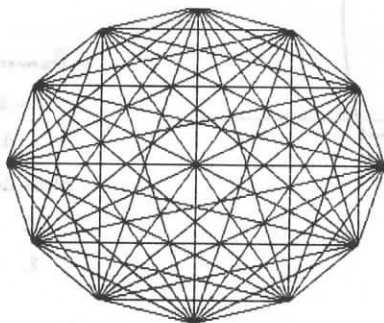
L'instruction PSE permet de suivre les variations de celle ci.Pour l'ameliorer il suffit de changer la valeur 3, qui constitue la valeur d'arret du calcul et qui commande le calcul des racines.Cependant un nombre d'iterations trop grand ou des valeurs initiales de P etS mal choisies peuvent entrainer une perte de precision.Notez que le programme prend comme valeurs initiales de P et S, une valeur nulle.

Si la precision indiquée n'est pas atteinte,il faut modifier les valeurs P etS, afin de rendre la convergence plus rapide.

Bibliographie: ALGORITHMIQUE NUMERIQUE

par:J. COUOT, J. GACHES,J. SOUHAIT.

Première partie (1981) du cours de L'ECOLE NATIONALE SUPERIEURE DE L'AERONAUTIQUE ET DE L'ESPACE.



exemples

1) Soit l'équation: $f(x) = x^3 - 3x + 2$

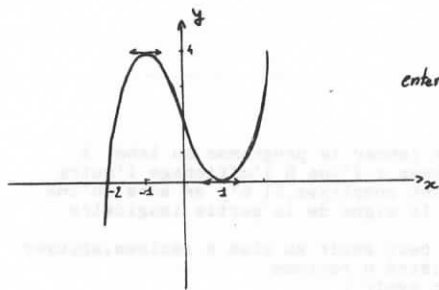
$n=3$.

Faire:

4 ET 1 DIM A
DIM B

3 ET 1 DIM C

entrer les coeff. par ☐ CLREG,
mettre 3 à l'affichage.
Lancer le programme.



Premier arrêt:

-2 dans X

1,0008 dans Y

Faire R/S

deuxième arrêt: 0,9985 à
l'affichage.

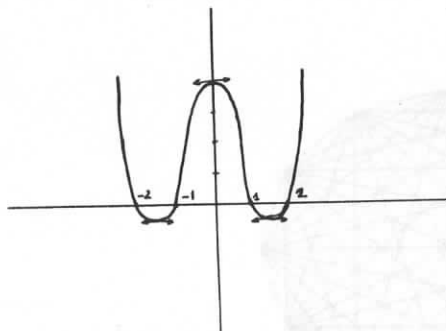
Donc il y a une racine double en $x=1$.

2) $f(x) = x^4 - 5x^2 + 4$

5 ET 1 DIM A
DIM B

4 ET 1 DIM C

entrer les coefficients, ☐ CLREG,
lancer le programme.



Premier arrêt:

-1 → X

1 → Y

R/S
deuxième arrêt:

-2 → X

2 → Y

1 LBL P	*	LASTX
1	X<>Y	R↑
+	LASTX	150 X<>Y
STO 2	*	F? 8
LBL C	R↑	I
RCL 2	RCL B	F? 8
3	30 *	GTO 8
X=Y?	-	-
GTO 9	CHS	LBL 8
10 X<>Y?	RCL B	R/S
GTO D	RCL* 6	RCL MATRIX B
FIX 4	R↑	STO MATRIX A
LBL 1	-	2
MATRIX 1	RCL 6	ST- 2
RCL A	X↑2	RCL 2
*STO B (mode user)	RCL 7	RCL 1
RCL 3	90 RCL* 5	DIM A
X<>Y	-	DIM B
*	/	-
20 RCL A	X<>Y	LASTX
LAST X	LASTX	DIM C
RDN	/	GTO C
+	RCL+ 3	170 LBL D
*STO E	X<> 3	MATRIX 1
LBL 2	X<>Y	*RCL A
RCL 3	RCL+ 4	1/X
X<>Y	100 X<> 4	RCL A
*	RCL 4	*
RCL A	-	CHS
30 R↑	ABS	RTN
LAST X	X<>Y	LBL 9
RDN	RCL 3	MATRIX 1
RCL* 4	-	180 *RCL A
-	ABS	*RCL A
+	RCL 3	X<>Y
*STO B	110 ABS	/
GTO 2	RCL 4	CHS
LBL 3	ABS	STO 3
MATRIX 1	+	RCL A
40 RCL B	/	LASTX
*STO C	PSE	/
RCL 3	EEX	STO 4
X<>Y	3	190 GTO E
*	CHS	
RCL B	X<>Y?	
LAST X	120 GTO 1	
RDN	LBL E	
+	1	
*STO C	RCL 3	
50 LBL 4	CHS	
RCL 3	RCL 4	
X<>Y	CF 8	
*	X<>Y	
RCL B	RDN	
R↑	X<>Y	
LAST X	130 /	
RDN	R↑	
RCL* 4	LASTX	
-	2	
60 +	*	
*STO C	/	
GTO 4	CHS	
RCL 2	E↑	
3	RDN	
-	X↑2	
STO 0	140 -	
*RCL C	X>0?	
STO 5	SF 8	
*RCL C	ABS	
70 STO 6	SQRT	
RCL C	R↑	
STO 7	X<>Y	
*RCL B	+	

Le "U" signifie que le mode
user doit être activé avant de taper
l'instruction.

MAGAZINE DU CLUB

ENTER↑



USER

USER er medlemsblad for PPC-Danmark. PPC - Personal Programming Center - er brugerklubben for alle der har en programmerbar lommeregner eller mikrocomputer fra Hewlett-Packard. USER, 1986, 6. Årgang, nr.:

1



PPC-Danmarks adresse (indlæg til USER sendes hertil): PPC-DANMARK, Postboks 2, 3500 Værløse. Indmelding i foreningen sker ved indbetaling af 220 kr. på foreningens postgirokonto nr. 1 37 75 23. Forespørgsler vedrørende foreningen og dens aktiviteter kan rettes til Sten Petersen, tlf.: 02-96 82 10

LES JOURNAUX

Nous n'avons pas reçu grand chose depuis deux mois. Il semble bien qu'il y ai une crise dans les clubs. Ont été reçus depuis MICROREVUE N°10 :

PPC Journal Décembre 85. V12 N12, reçue en Avril 86. Contient les rubriques habituelles. A signaler un article sur la nouvelle calculatrice Casio à écran graphique (FX-7000G) qui a un rapport qualité/prix assez remarquable. PPC, PO Box 90579, Long Beach, CA 90809-0579, USA. Montant de l'abonnement pour 12 numéros format 21x27, 32 pages, \$37.

Prisma, le journal du club Allemand CCD. C'est dans ce journal que nous avons trouvé le programme de dessin de fractals qui a servi pour illustrer le précédent numéro de MICROREVUE. Le dernier numéro reçu est Février 86. Il me semblait même avoir le numéro de Mars, mais il a disparu au cours de la réunion de bureau d'hier (! il reviendra). Dans le numéro de Février, un article sur le "HP PRINTER COMMAND LANGUAGE", la suite d'une série d'articles sur la Résistance des Matériaux, un calcul d'une transformation étoile triangle pour moteur électrique. Prisma, Postfach 2129, 6242 Kronberg 2 RFA.

JEDI marche toujours aussi FORTH, nous avons les numéros de Février et Mars, le premier traite d'un mini traitement de texte et de structures sonores en FORTH, le traitement de propositions logiques en LISP et les fonctions graphiques du Turbo-Pascal Amstrad ; le suivant démystifie APL, donne la 8ème partie d'une série sur le MUMPS, une cargaison de FORTH et même des math et du COBOL. JEDI 8 rue Poirier de Narçay, 75014 Paris.

Club Spiegel N°18 avec le HP Vectra en couverture. Petite par le format et le nombre de pages, cette revue est grande par la qualité et la régularité. HPclub (Schweiz), Postfach 8152 Glatthbrugg.

N°7 de HPuser Nieuws, PRINS, Binnenwatersloot 30, 2611 BK DELFT Pays Bas. Toujours relié par une simple agraphe, mais remarquablement imprimé au format A4 et sur 32 pages.

USER, la revue Danoise, nous avons reçu le numéro 6/85 et le numéro 1/86. Ce dernier avec en couverture la photo du nouveau terminal portable HP qu'avec un peu de chance vous trouverez dans ces pages. PPC Danmark, Postboks 2, 3500 Vaerloose.

SUR LE MARCHE

Je relève dans le catalogue de PRINS (cf User-News dans "les Journaux") une interface HP-IB/Centronic qui ne concerne pas directement les calculatrices, mais qui devrait intéresser les utilisateurs de micros HP ou Commodore. prix 425 Florins.

Mathematisches Büro Köln, Dipl.-Math. A. B. Gemein, An der Flora, 5000 Köln 60 annonce la sortie d'un lecteur d'Eproms et d'une boîte RAM pour HP-41 (de 1540 à 1300 DM pour 16K RAM et de 555 à 465 DM pour le lecteur d'Eprom). Ecrivez-leur pour les détails.

Le catalogue EDULCALC N°30, avec Einstein en couverture est une mine inégalable d'accessoires pour calculatrice HP, ainsi que de livres. Le plus remarquable est sans doute un module (CMT) de 32K de mémoire vive pour HP-71 à \$149 ! Inutile de dire que nous en avons déjà commandé un et que nous vous en parlerons. Ce module se branche à l'avant du 71 et vous pouvez en brancher 4 ! Il est même possible de les câbler à l'intérieur du 71, EDULCALC peut le faire pour vous pour environ \$90. EDULCALC mail store 27953 Cabot Road Laguna Niguel, CA 92677 USA. Carte VISA acceptée.

PORTATEK, 12021 Wilshire blvd., Suite 44, West Los Angeles, CA 90025 peut vous fournir du matériel pour calculatrices HP à prix discount. Hélas ils n'acceptent pas les cartes de crédit. Demandez-leur leurs tarifs.

Je vous ai parlé dans le dernier numéro du FORTH COMFORT de A. Pirard, 5 Pirefontaine, B-4052 Dalembeux. J'ai depuis reçu la notice de ce FORTH, en anglais, d'environ 120 pages tapées à la machine. Ce FORTH semble particulièrement complet et performant. Il peut aussi bien travailler en gérant une disquette par blocks comme les FORTH traditionnels qu'en utilisant les fichiers standard du DOS (APPLE, IBM, Commodore...) et passer de l'un à l'autre. Il possède de nombreux utilitaires, y compris une gestion des chaînes de caractères avec pile/chaîne. La présence d'un utilitaire COMGEN qui génère des fichiers exécutables à partir d'un programme FORTH est extrêmement commode. En effet, vous pouvez laisser en mémoire tous les utilitaires d'édition, de débogage, ... sans vous soucier de place mémoire, puisque COMGEN est là pour "faire le ménage" et supprimer tout ce qui n'est pas immédiatement utile. Donc mise au point avec tous les utilitaires, puis obtention automatique d'un programme exécutable optimisé. Très intéressant pour un prix réduit de \$90.

La COOP AHUNTSIC 9155 rue St Hubert, Montréal, Québec H2M 1Y8 Canada, dirigée (pour ce qui nous concerne) par Sylvain Côté est à la fois distributeur HP et amateur ; Sylvain prépare un MLDL de 8K ram et 16K Eprom qui devrait coûter \$250 canadiens (environ 1250F). Une affaire à suivre.

Un nouveau livre vient de sortir en anglais sur le HP-71 : The Basic HP-71 qui est un recueil d'astuces avancées et permet même une première approche du langage machine. Les Editions du Cagire en publieraient volontier une traduction, mais je suis à court de temps disponible. Si l'un de vous peut faire une traduction sur disquette IBM avec Wordstar, qu'il me contacte. Les droits d'auteur sont symboliques, vue la faiblesse du tirage (au plus 300 exemplaires). En attendant, il coûte \$19.95+port chez EDULCALC (#71-658). L'auteur est Richard Harvey. Richard a également publié un "Work book 71" qui est un ensemble de programmes genre visicalc, traitement de texte, gestion de base de données et travaille en mémoire virtuelle sur la cassette ou la disquette. La version que nous avons ici n'est pas exempte de bugs dont il faut espérer qu'ils soient rapidement corrigés. Chez Educalc #71-611 sur disque \$65.95, #71612 sur cassette \$69.95, manuel copieux en anglais.

NOUVELLES BREVES

Patrick Belaire, 65 rue Paul Eluard, 62210 Avion, propose ses services à ceux qui ont besoin de réaliser des EPROMS le contacter. tel 21 78 17 95.

ENFIN Le manuel du ROM PANAME, attendu depuis si longtemps est enfin arrivé. Il est volumineux, bien documenté et augmente considérablement l'intérêt de ce module. Le manuel est dénommé : "Module de fonctions Gestion de tableaux". Rappelons que ce module, avec son manuel, est disponible contre 1170F. Notice descriptive sur demande. Ceux d'entre vous qui nous ont déjà acheté ce module ont déjà reçu le manuel.

Antoine M Couëtte 37 avn Charlevoix de Villers 40000 Mont de Marsan nous a envoyé, sous une forme malheureusement non publiable, un correctif au programme LPAS du PPC-ROM. Si cela vous pose problème, écrivez-lui.

François Le Grand (T391) rue Joseph Coosemans 121 1030 Bruxelles ou rue de la Poterie - Trevignon 29128 Tregunc, a mis au point un logiciel de navigation, écrit en BASIC pour HP-71 et, d'après l'auteur, supérieur au module navigation de la HP-41. Il le vend pour 200F sur cassette, 250F sur cartes magnétiques.

André Vialaron, 11 rue Darquié, 31000 Toulouse tel 61 52 20 03, aimerait entrer en contact avec possesseur de livret d'application HP-41 "Electronic engineering" et "antennes" pour échange de points de vue sur programmes proposés.

PETITES ANNONCES

Gérald Allier 7 rue E. Vaillant 34590 Marsilargues tel 67 83 02 20 (T657) vend HP-41CX avec batterie rechargeable, 1 imprimante 82143 + 2 batteries rechargeables, + papier thermique, + lecteur de cartes, + 40 cartes magnétiques, matériel 1 an et demi, très peu servi. Prix neuf - 30%

Marco Louvert 236 rue Jean Mermoz 33320 Eysines (T253) tel 56 34 16 90 vend HP-41CX février 84 2300F, lecteur de cartes 1000F, MLDL janvier 84 1500F, X-Mémoire 400, Module math 100F + livres prix groupé 5000F.

Jean-Philippe Boucherie Le Rion de la cuve St Denis de piles 33230 Coutras (T127) vend une imprimante 82143 (3 ans) 2000F, 1 lecteur de codes barre (3ans) 650F, 1 lecteur de cartes magnétiques (3ans) 1050F (avec 100 cartes), module math 100F, livret d'application gestion/marketing/Vente 50F, différents ouvrages et revues ou l'ensemble pour 4000F.

J Witté (T735) cherche PPC ROM et module Advance avec docs. 13 av du Littoral Alenya 66200 Elne.

Jean-Pierre Fassiola 11 av des Lacs 94100 St Maur tel 48 83 61 08 vend imprimante 82162 2000F, interface vidéo HP 82163 (32 colonnes) 1000F.

Didier Renevot (T388) 208 rue Bailly de Suffren La garonne 83220 Le Pradet, vend lecteur de cartes HP-41 avec 160 cartes 1500F, imprimante 82143 2000F.

PIERRE BARTHELEMY, T-659

12, SQUARE HOPKINSON

13004 MARSEILLE

TEL : 91/34/69/77

vends 1 cassette numérique

presque pas servie : 80 FF

frais de port à ma charge

Jean-marc MORANDI

vends :

HP82162A

3200,00

HP-IL

800,00

38 rue du Paret 69000

St PRIEST

Francois LE GRAND

Rue de la Poterie

Trevignon

29128 TREGUNC

tel. 98 50 00 10

VENDS

BRADÉ :

1	HP-41 CV	1100
1	Imprimante 82143	1000
1	Lecteur de Cartes 82104	800
1	Module Time	400
1	Module Extention de Fonctions	400
2	Modules Extention de Memoire	800
1	Module Navigation 14017	200
500	Cartes Magnetiques	400
1	Clavier souple	0
1	Convertisseur HP-IL 82166 (neuf)	1500
1	Overlay kit	0
?	 (Accessoires divers)	0

Total : 6500

LE TOUT POUR : 6000 FF

de tout sauf Convertisseur IL : 4500 FF

sont 25% du prix neuf !

Sont disponibles directement au club, 77 rue du Cagire, dans le cadre de la bourse d'échange :

- 1* HP-75 1983 complet avec module 8K à 5000F
- 2* Imprimante 82162 1983, batterie à changer 2200F,
- 3* Livrets d'application pour HP-75 Math II, Math III, Statistics, Electronics à 150F l'un,
- 4* ~~Logiciel Forth sur cassette~~^{ven du}, logiciel gestion de fichiers sur cassette, Masslex+Editeur de texte+Displlex sur cassette avec doc, chacun 250F.
- 5* HP-75 1983 complet avec module 8K 4000F,
- 6* Imprimante 82162 1982, batterie à changer, 2200F
- 7* Lecteur de cassette 82161 1982 2800F,
- 8* Module HP-IL pour HP-41C 1982 600F,
- 9* 6 cassettes pour 82162 les 6 : 300F.

De préférence confirmer votre commande par téléphone (61 44 03 06) avant d'envoyer le chèque. Les prix des #5 à 9 sont donnés sous réserve de l'accord du propriétaire qui ne nous est pas encore parvenu. A ce prix j'ai presque honte de vendre des HP-75 dont le prix neuf est deux à trois fois plus élevé. N'oubliez pas que l'HP-IL et le lecteur de cartes sont inclus d'origine dans le HP-75!

REACTIONS

GOUBAULT de BRUGIERE Alain
27 avenue de Brainot
78400 CHATOU
'T441)

Chatou, le 21 janvier

Gher J.D.O.

Sa y est, j'ai acheté mon HP75 et je dois dire que cette merveille d'électronique a bien largement dépassé mes espérances. J'ai hâte de me procurer le module FORTH/Assembleur pour en utiliser à fond les formidables possibilités. Je me suis d'ailleurs dépêché de commander "HP75 c'est facile" que tu as traduit et j'ai déjà rentré les programmes fournis dans MR9 extraits de cet ouvrage ; histoire de me mettre en appétit. Vivement que je puisse le lire en entier.

J'en ai profité pour commander l'IDS n°1 en m'imaginant que le club pourrait me le procurer. Bruno Piquet m'a répondu que cet ouvrage n'était pas disponible au club et que je devais m'adresser à HP, ce que je fis :

IDS n°1	Présentation détaillée	500,00
IDS n°2	Points d'entrée	500,00
IDS n°3	Listing de la ROM	2000,00 !
IDS n°4	Schémas hardware	2000,00

total : 5000,00 et je ne compte pas l'IDS
l'HPIL ni celui du modèle FORTH. Rien cher

que la machine ! Peux-tu me dire quel ouvrage parmi
ceux là est le plus utile pour programmer en ASM ?

A ce propos, je voudrais faire part d'une certaine
déception quant à certaines informations (ou plutôt au
manque d'informations) que l'on peut extraire des articles
ou des lettres publiées dans la revue. J'ai la désagréable
impression que les bonnes informations restent l'apanage
d'une "élite" et ne sont pas diffusées comme elles
devraient l'être, je prendrais comme exemple (ce n'est
pas le seul) la lettre d'Eric Gengoux dans MR n°9 à
propos du LEX "DSIPLEX 71" de Vogt. Qu'id ce lex ?
pourquoi personne n'en a-t-il parlé avant et surtout
n'en a-t-il décrit le contenu ? Est-il envisageable et
envisagé de publier un catalogue des LEX disponibles
avec les informations nécessaires pour se les procurer ?

Jc t'envoie ci-joint une liste d'articles ou annonces
parues dans certains numéros de la revue américaine
CIME (Computers In Mechanical Engineering). Ils
concernent la HP41 qui a souvent eu une place de
hoix dans la revue. Je ne la revoie plus depuis
longtemps malheureusement et je ne peux pas te donner
d'informations plus récentes, mais il doit être possible
de les retrouver. Ces références pourront enrichir IND41.
Elles sont de 3 types :

- des articles généralement accompagnés du programme
- des annonces pour se procurer des logiciels spécialisés
(Software exchange ou New products)
- des ouvrages (Literature)

Si tu veux de plus amples enseignements, n'hésites pas
à me contacter.

Pour l'instant, j'attends avec impatience MR n°10 qui
de toute façon sera dévoté en quelques heures.

Happy Programming

Renault

PS Peux-tu m'envoyer quelques numéros récents de "Looping"
Merçi -

Les IDS les plus utiles sont certainement les N°1 et 3. Ils sont presque indispensables. Il faut dire que programmer le HP-71 en langage machine n'est pas un travail de débutant. La 41 est simple, à côté. Aucune information n'est "cachée" dans le club, si l'on excepte les très rares cas d'informations confidentielles transmises par HP, qui n'en est pas prodigue. Par contre, les grands experts en programmation... ne sont pas toujours des écrivains et ne transmettent pas toujours leur savoir, essentiellement faute de temps. Le logiciel que tu cite a été disponible pendant un an sur le catalogue des Editions du Cagire. Il n'est plus disponible aujourd'hui que d'occasion ou chez EDUCALC. Les auteurs sont libres de décider si ils donnent leurs programmes ou si ils les vendent. Le catalogue d'EDUCALC est la meilleure liste de tous ces produits. J'en reçois une quinzaine tous les deux mois, qui sont distribués dans les trois jours. Demandez-le aux USA, c'est plus facile que vous ne le pensez, surtout si vous avez une carte VISA. IND41 s'est très peu vendu. Dès que le stock sera épuisé, une nouvelle version sera disponible en photocopies. Le mieux est de transmettre ces informations à Robert Pulluart (lusterlaan 31 2665 th bleiswijk pays bas). Il y a longtemps que je n'ai pas reçu de n° de Looping.

ARGUS S.O.S.

ERRATUM

Le programme ci-dessous, commenté dans "Le HP-71 c'est facile", a été omis à la mise en page, veuillez nous en excuser.

136 HP-71 BASIC Made Easy

```
10 ! S.M.A.T.T.E.R. (C) 1984 Joseph Horn
20 DIM A$(96),X,X @ DELAY 9,0
30 DISP "Code Decode More Quit"
40 AS=KEYWAITS @ IF AS="Q" THEN 130 ELSE IF AS="M" THEN 70
50 IF AS="C" THEN K=1 ELSE IF AS="D" THEN K=-1 ELSE 40
60 OPTION ROUND NEG @ INPUT "Key:" ;X @ RANDOMIZE X
70 INPUT "Text: "AS @ IF AS="" THEN 30 ELSE PRINT "<";
80 FOR X=1 TO LEN(AS) @ IF AS(X,X)<" " OR AS(X,X)>"z" THEN 120
90 RANDOMIZE 10*(RND*100)
100 POKE "2f6fe",STR$(FLAG*(-12,RND*(RND)+1))
110 PRINT CHR$(MOD(NUM(AS(X,X))+IP(RND*92)*K-32,91)+32);
120 NEXT X @ PRINT ";" @ GOTO 30
130 OPTION ROUND NEAR @ RANDOMIZE @ DELAY 0,0 @ DISP "Done" @ END
```

Le programme "TOKENIZE" est incorrectement listé dans la livre,vous trouverez la version correcte ci-dessous.

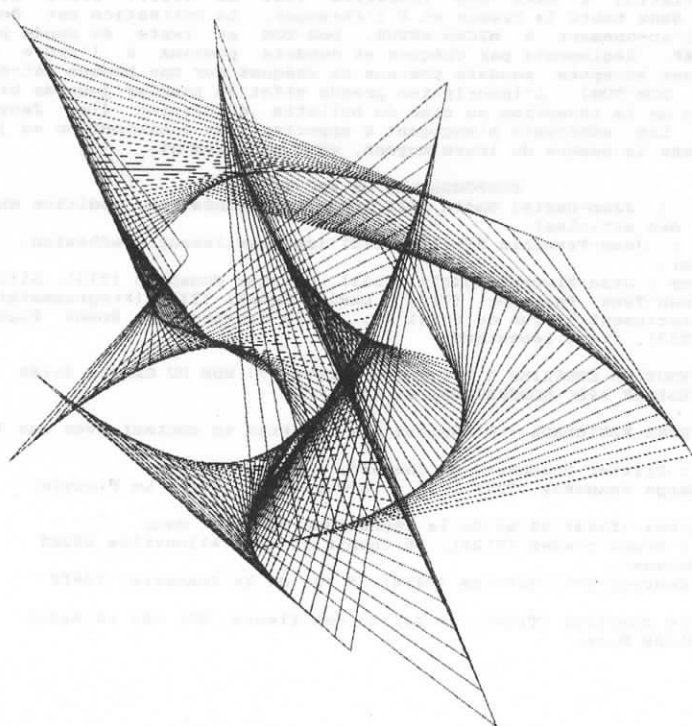
```
! Nimporde quelle lign
e ici
10 SUB A @ N=256 @ DIM A
$(N) @ AS=ADDR$(CAT$(0))
20 AS=PEEK$(DTH$(HTD$(AS)
+55),N)
30 AS=AS[1.POS(AS,"0F010
0")-1]
40 PLIST 1,9 @ DISP AS;"
(*,STR$(LEN(AS));)*"
```

ARGUS

ARGUS PPC-T Avril 86					
Référence	MATERIEL	PX NEUF	UN AN	DEUX ANS	TROIS ANS
	Taux	100%	70%	55%	40%
	HP-11	590	413	325	236
	HP-15	1174	822	646	470
	HP-16	1411	988	776	564
	HP-12	1174	822	646	470
	HP-41	-	1500	1200	1000
	HP-41CV	1980	1386	1089	792
	HP-41CX	2798	1959	1539	1119
	HP-71BF	5913	4139	3252	2365
	HP-75C	-	6000	5000	4000
	HP-75D	15304	10713	8417	6122
	Modules HP-41				
41-15042	Auto Dup	469	328	258	188
41-15043	IL-DEV	1005	704	553	402
	MEM simple	-	0	0	0
	QUAD	-	500	400	300
82180	XF	1005	704	553	402
82181	XM	1005	704	553	402
82182	TIME	1005	704	553	402
82183	IL-E/S 82183	1005	704	553	402
82184	PLOTTER	1005	704	553	402
82160	HP-IL	1675	1173	921	670
	Périph HP-41				
82104	Lect Cartes	2615	1830	1438	1046
82153	Lect Cod. Barres	1675	1173	921	670
82143	Imprimante thermique	5165	3615	2841	2066
	Accessoires HP-71				
82400	Lect Cartes	2200	1540	1210	880
82420	RAM 4K	965	676	531	386
82401	HP-IL	1622	1135	892	649
82490	Trans Pack	1677	1174	922	671
82441	FORTH/assembleur	2012	1408	1107	805
82480	Math	1273	891	700	509
82481	Annalyse circuits	1005	704	553	402
82482	Finance	1005	704	553	402
82484	Lissage	1273	891	700	509
82485	Edit texte	1005	704	553	402
82440	Utilitaire de dévelo	469	328	258	188
82479	acquisition données	2615	1830	1438	1046
82489	Statistiques ampi	1677	1174	922	671
82483	module topographie	2011	1408	1106	804
82488	pac datacom	2011	1408	1106	804
	Accessoires HP-75				
82725	Mod C. Barres 75D	1005	704	553	402
82700	MEM 8K	2615	1830	1438	1046
75-15001	I/O ROM	1273	891	700	509
75-15019	Text Formateur	1273	891	700	509
75-15015	Math	1945	1362	1070	778
75-15014	Visicalc	2615	1830	1438	1046
82718	Pod 64K	11374	7962	6256	4550

75-15035	datacom rom	2347
75-15012	Surveying rom	3956
toutes reflivrets applic		603

Accessoires HP-IL Universels				
2225	Thinkjet 2225	5869	4108	3228
9114	Disquette 9114	10424	7297	5733
82161	Lecteur cassettes	7337	5136	4035
82162	Imprimante thermique	6036	4225	3320
7470/003	Table tracante a4	14374	10062	7906
82164	IL/RS232	3930	2751	2162
82165	GPIO/IL	3957	2770	2176
82169	IL/HP-IB	5231	3662	2877
	Cassettes (piece)	127	89	70
82938	HPIL/SERIE 80	3665	2566	2016
92198b	HPIL/VIDEO 80COL	3848	2694	2116
	disquettes (pièce)	76	53	42
				30



651211

ORGANISATION DE PPC-T

Cette page sera publiée dans chaque numéro de MICRO-REVUE. Seules les indications répertoriées dans le dernier numéro du journal font foi. PPC-T étant une association sans but lucratif, elle ne fait pas de bénéfice. Elle est administrée par des bénévoles et ses produits et services sont réservés à ses membres. La marge prise sur les produits vendus est minimale et destinée à couvrir les frais de fonctionnement.

ASSOCIATION PPC-T

Association à buts non lucratifs (loi de 1901), PPC-T accepte des adhérents dans toute la France et à l'étranger. La cotisation est de 250F y compris l'abonnement à MICRO-REVUE. DOM-TOM et reste du monde par avion, ajouter 50F. Règlements par chèques et mandats postaux à l'ordre de PPC-T (Eurochèques acceptés, mandats postaux ou chèques sur une banque métropolitaine pour les DOM-TOM). L'inscription prends effet au premier jour du bimestre en cours lors de la réception au club du bulletin d'adhésion (1er Janvier, 1er Mars...). Les adhérents s'engagent à apporter leur contribution au journal et au club dans la mesure de leurs moyens, même modestement.

RESPONSABLES DE PPC-T

Président : Jean-Daniel Dodin (T1) (nouvelles adhésions, édition du journal, réception des articles).
Trésorier : Jean-François Sibille (T15) (renouvellement d'adhésion, problèmes financiers).
Secrétaires : Jean-Pierre Baudoin (T131), Olivier Monachon (T13), Gilles Barret (T22), Jean-Yves Pasquier (T18), Damien Debril (T33) (Programmathèque, lui écrire directement 38 rue du 8 Mai 45, 59190 Hazebrouk), Bruno Piguet, Eric Sergent (T33), Franc Lebastard.

ADRESSER TOUT LE COURRIER A PPC-T/MICRO-REVUE 77 RUE DU CAGIRE 31100 TOULOUSE.
TEL (A UTILISER AVEC MODERATION) 61 44 03 06

COORDINATEURS REGIONAUX : N'hésitez pas à entrer en contact avec eux !

Bretagne : Olivier Pugeon 6 BD Chanard, 56170 Quiberon.
Paris : Serge Vaudenay (T270), 62 a- Ardouin, B3, 94420 Le Plessis Tréville
Olivier Arbey (T164) 34 av de la République 94100 St Maur.
Picardie : Bruno Tredez (T120), 16 Chemin St Jean Allonville 80260 Villers Bocage.
Savoie : Georges-Noël Nicolas (T438), 5 rue de la Jonchère, 74600 Seynod.
Nice : Eric Angelini (T120), Le Vallon des fleurs, B9, 103 Av Henri Dunand, 06100 Nice.



FORTH
Interest
Group

Club d'utilisateurs d'ordinateurs de poche
Hewlett-Packard
Club d'utilisateurs du langage Forth
77 rue du Cagire 31100 Toulouse France

Nouvel adhérent
Renouvellement
Changement d'adresse
(Rayer la mention inutile)
PPC-T N°.....

BULLETIN D'ADHESION

NOM :
PRENOM :
ADRESSE :
PAYS : Téléphone :
PROFESSION :
Ecrivez clairement en majuscules, merci.

=====

La fonction de PPC-T est de favoriser les connaissances sur les ordinateurs et leurs langages, grâce aux échanges entre les adhérents, dont les adresses sont régulièrement publiées.

=====

Je demande à adhérer à PPC-T. Je joins mon règlement de 250F (300F par avion) par : Mandat Chèque CCP Espèces.

Date :

Signature :

Votre adhésion sera prise en compte à réception par PPC-T du présent bulletin convenablement rempli.

=====

Remplir le verso SVP, merci

RENSEIGNEMENTS FACULTATIFS

Sujets d'intérêt, domaine préférentiel d'utilisation de votre ordinateur de poche :

Je dispose des matériels informatiques suivants (type et modèle, date d'achat) :

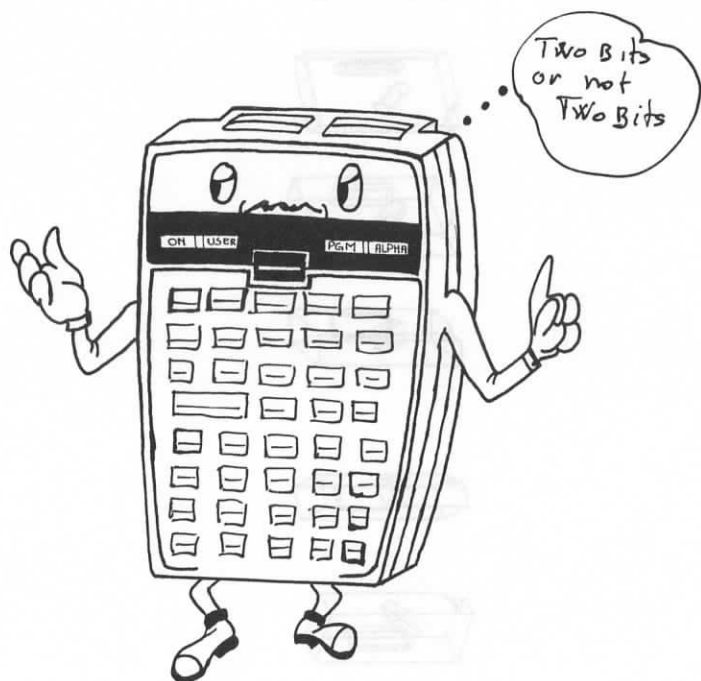
J'ai eu connaissance de PPC-T par :

N°PPCUS

N°CHHU

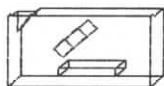
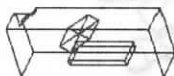
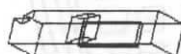
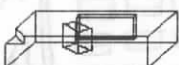
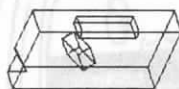
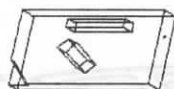
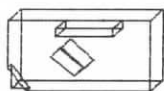
Observations :

OU B L I.



Votre dévoué T334.

ROTATIONS



Two 8-1/2
on top
Two 8-1/2

CAL

Fonctions Calendrier I du HP-12C

Le programme trouve son fondement dans l'examen comparatif des fonctions calendrier du Module TIME (du HP-41) et de celles du HP-12C, ainsi que du désir de réduire autant que possible le nombre d'appuis de touche. Le programme fonctionne aussi bien en mode M.DY qu'en mode D.MY. Pour la clarté de l'exposé, ce dernier mode est choisi.

Module TIME (HP-41)

HP-12C

DATE+

DATE

DDAYS

ΔDYS

DOW

pas de fonction DOW au clavier,
mais accès à cette fonction par la fonction DATE
avec W de 1 (pour Lundi) à 7 (pour Dimanche).

Habituellement, pour obtenir le DOW d'une date donnée, il faut taper:

(d)d,mm,yyyy [ENTER] 0 [g][DATE] -->dd.mm,yyyy W
soit, 12 (ou 13) appuis de touche DOW

Simplification avec le programme: après introduction de celui-ci, passer en mode RUN, par appui de [f][P/R], faire [f]0 (FIX 0), puis choisir un nombre N correspondant au siècle sur lequel porteront les calculs. N est stocké en R.0, par le programme.

S	N	Dates de	à
1	15	1500*	1599
2	16	1600	1699
3	17	1700	1799
4	18	1800	1899
5	19	1900	1999
25	39	3900	3999
26	40	4000	25 Novembre 4046

Les dates sont entrées sous le format:
(d)dmmyy (différent du format d'origine (d)d,mm,yyyy,
soit une économie de 3 appuis de touche par date.)

Les dates sont entrées de deux manières, au choix:

- si N n'a pas été entré auparavant, faire:
N [ENTER]
(d)dmmyy [R/S]

- si N a été entré auparavant, faire:
(d)dmmyy [ENTER] [R/S]

* En réalité, n'entrer des dates qu'à partir du 15 Octobre 1582 (début du calendrier grégorien). Le programme est valide pour 24 siècles entiers et pour 2 siècles partiels, le 1er (1500..) et le 26ème (4000..).
Toute introduction de date antérieure au 15 Octobre 1582 ou postérieure au 25 Novembre 4046 entraîne Error 8.

Nombre de Jours entre Dates.

Le calcul est fait sur les deux dernières dates entrées qui seront rendues dans le format dd.mm.yyyy W, W est le DOW.

1. [ON]
2. Entrer le N choisi [ENTER]. (Si N est déjà en machine, aller en 3 directement).
3. Entrer la première date. (Si N est déjà en machine et qu'on vient de l'allumer, entrer la première date, appuyer sur [ENTER], puis 4 et 5).
4. [R/S] 0 apparaît à l'affichage.
5. [SST] Première date avec son DOW.
6. Entrer la deuxième date, [ENTER]
7. [R/S],[SST] Deuxième date avec son DOW.
8. [g][ΔDYS] Nombre de jours entre les dates.

Exemple: Quel est le nombre de jours entre le 3 Juin 1983 et le 15 Octobre 1984. Supposer que N (19) est en machine et qu'on vient d'allumer la machine.

```
30683 [ENTER][R/S][SST]      3.06.1983 5   Vendredi
151084      [R/S][SST]      15.10.1984 1   Lundi
[g][ΔDYS]      500 Nombre de jours
[X<>Y]      492 Nombre de jours comptés
              sur un mois de 30 jours.
```

Exemple: Quel est le nombre de jours entre le 15 octobre 1582 (début du calendrier grégorien) et le 25 novembre 4046 (fin du calendrier valide)?

```
15 [ENTER]
151082 [R/S][SST]      15.10.1582 5   Vendredi
40 [ENTER]
251146 [R/S][SST]      25.11.4046 7   Dimanche
[g][ΔDYS]      899999 Nombre de jours
[X<>Y]      887080 Nombre de jours comptés
              sur un mois de 30 jours.
```

Dates passées, présentes (1900) ou futures.

1. [ON]
2. Entrer le N choisi [ENTER]. (Si N est déjà en machine, aller en 3 directement).
3. Entrer la date d'origine. (Si N est déjà en machine et qu'on vient de l'allumer, entrer la date, appuyer sur [ENTER] puis 4 et 5).
4. [R/S][SST] Date d'origine avec son DOW.
5. Entrer le nombre de jours (négativement si la date est dans le passé).
6. [g][DATE]

Exemple: Nous sommes le 14 Mai 1981. Dans 120 jours, quelle sera la date?

```
19 [ENTER]
140581 [R/S][SST]      14.05.1981 4   Jeudi
120 [g][DATE]      11.08.1981 5   Vendredi
```

REMARQUE : Il est possible d'entrer un nombre de jours qui fasse passer d'un siècle au(x) suivant(s).

Exemple: Le début du calendrier grégorien se situe le 15 Octobre 1582. Il s'écoulera 899.999 jours. Quelle sera alors la date? (fin du calendrier valide)

```
15 [ENTER]
151082 [R/S][SST]      15.10.1582 5   Vendredi
899.999 [g][DATE]      25.11.4046 7   Dimanche
```

Touches
=====

Affichage
=====

```
[g][x<y]      01- 43 34
[g][GT0]06    02-4333 06
[x<y]          03-    34
[ST0][.] 0    04-44 48 0
[x<y]          05-    34
[EEX]         06-    26
2             07-    2
[/]           08-    10
[g][INTG]     09- 43 25
[g][LSTx]    10- 43 36
[g][FRAC]     11- 43 24
[RCL][.] 0    12-45 48 0
[+]           13-    40
[EEX]         14-    26
2             15-    2
[/]           16-    10
[+]           17-    40
[EEX]         18-    26
2             19-    2
[/]           20-    10
[RCL][.] 1    21-45 48 1
[x<y]          22-    34
[ST0][.] 1    23-44 48 1
0             24-    0
[R/S]         25-    31
[g][DATE]     26- 43 16
[g][GT0]01    27-4333 01
```

L. Huthmann
PCLT-SS1



PROGRAMMATION SYNTHETIQUE, LCD, IMPRIMANTE, REGISTRES, MODULE, TOUCHES, OVER LAYS, MATRICES, MEMOIRE CONSTANCE, LECTEUR

DE CODES A BARRES, LECTEUR DE CARTES, BASIC, RPN, FORTH, LANGAGE SPECIALISE, MICROCODE, ASSEMBLEUR, MATHEMATIQUES, SOUS PROGRAMME, NOMBRE ALEATOIRE, DRAPEAUX, LUCASIEWICZ, PILE, ALPHANUMERIQUE, PARASITES, RI-DEAU, SWAP, NOTATION POLONAISE INVERSE, DUP, SHIFT, ECRAN, MOT, COMPI-
LER, EXECUTION, TESTS, BOUCLES, DO LOOP, BEGIN UNTIL, IF ELSE THEN, : ;, BOUCLES CONDITIONNELLES, LABELS, TABLEAUX, TRACE, SST, BST, R/S

FORTH
Interest
Group

Edité par PPC-T - TITRE: MICRO-REVUE

77, rue du Cagire 31100 TOULOUSE

ISSN 0754-6335

Directeur de Publication : J.-D. Dodin

Imprimé par la Société Pyrénéenne d'Impression S.A.

20 Av. du Lauragais 31400 TOULOUSE