

# L'Ordinateur de poche

ISSN 0291-5243

*Nouveau :*

*PB-700*

*de Casio*

*Des idées,  
des programmes.*

*Ah ! Si vous aviez su...*

M 1859 - 20 - 16-F

PÉRAY.

Belgique 130 FB - Canada 1.95 \$C - Suisse 5 FS

n°20 janvier-février 1984

16 F

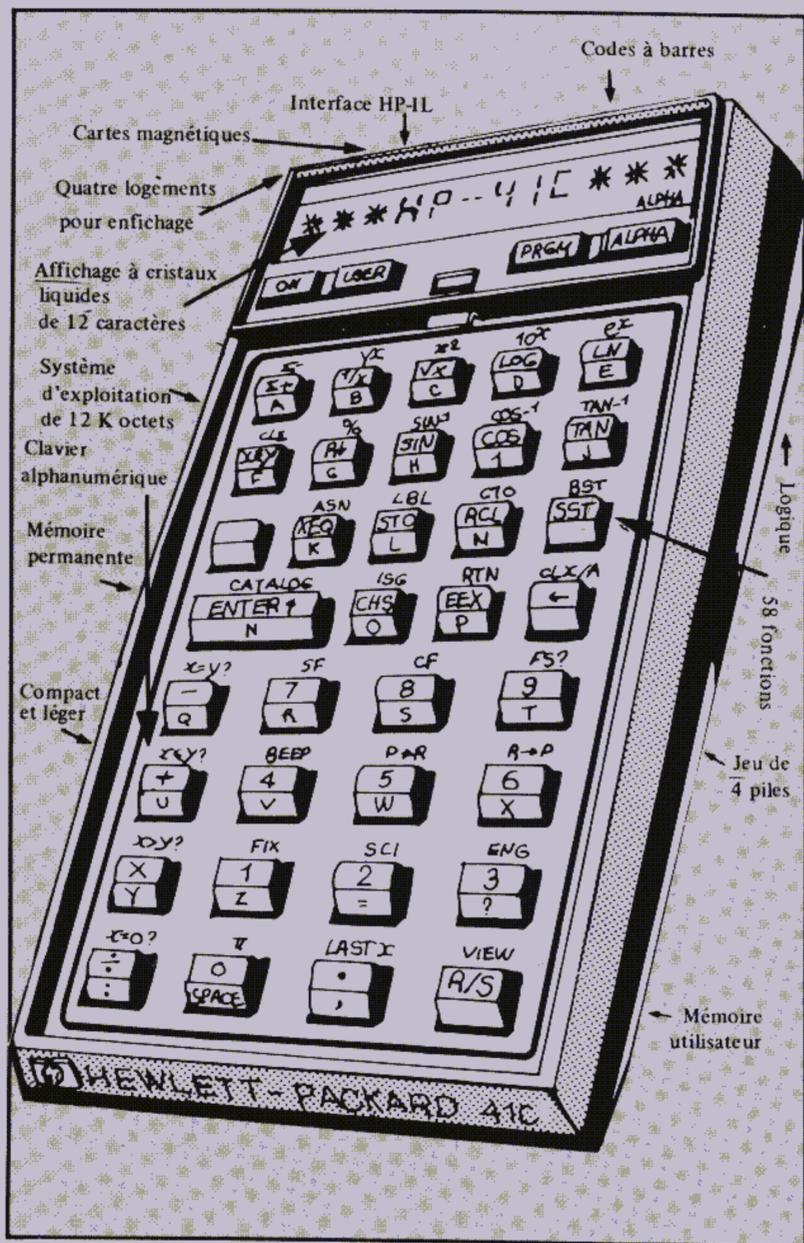


1<sup>er</sup> distributeur agréé  
des calculatrices Hewlett-Packard France

**PROMOTION RÉSERVÉE AUX ÉLÈVES DES GRANDES ÉCOLES**

## HP 41C :

le calculateur programmable alphanumérique qui, équipé d'un module quadruple, vous donne jusqu'à 2 000 lignes de programme.



### PRIX SPÉCIAL

**HP 41C**  
+  
**module quadruple HP 82170**

**: 1500 F TTC**

#### Périphériques et accessoires

• lecteur de cartes	HP 82104	1 495 F TTC
• imprimante	HP 82143	3 085 F TTC
• module multifonctions	HP 82180	500 F TTC
• module horloge	HP 82182	500 F TTC
• module mathématiques	15010	295 F TTC
		<b>PROMOTION PROMOTION</b>
• batterie HP 41	82120	360 F TTC
• chargeur HP 41	82066B	130 F TTC
• livrets d'application (28 titres)		135 F TTC l'unité
• jeu de 4 grilles d'assignation	HP 82152	119 F TTC
• clavier à membrane	HP 82200 A	235 F TTC
• port expandeur 8 entrées et sorties		1 250 F TTC

#### Livres :

• programmez	HP 41	102 F TTC
• au fond de la	HP 41	100 F TTC
• autour de la	boucle	100 F TTC
• synthetic programming on the HP 41 (en anglais)		149 F TTC
• calculateur tips et routines (en anglais)		184 F TTC

**BON DE COMMANDE : OFFRE SPÉCIALE GRANDES ÉCOLES dans la limite des stocks disponibles au 1/4/1984**

• Je commande les logiciels ou produits suivants   
TOTAL TTC : ..... Participation aux frais de port + 30 F  
Ci-joint mon règlement par :  
CB  CCP  Mandat   
qui ne sera encaissé par la RAC qu'après l'expédition.

Nom \_\_\_\_\_ Prénom \_\_\_\_\_  
Ecole \_\_\_\_\_  
Adresse \_\_\_\_\_  
Code postal \_\_\_\_\_ Ville \_\_\_\_\_

LA RÈGLE A CALCUL - 65/67, bd St-Germain 75005 Paris  
Tél. 325.68.88 - Télec 220064F ETRAV/1303 RAC

## 1 COUVERTURE

L'illustrateur Fabrice Péray décrit, à sa manière, l'activité qui règne chez les constructeurs d'ordinateurs de poche. Quelles surprises nous réserve l'année 1984 ? Monsieur de La Palisse répondrait sans hésiter que nous en aurons une excellente idée dans un an... En attendant, nous vous tiendrons informés des nouveautés.

## 11 A VOS CLAVIERS

## 14 MAGAZINE

## 18 N'EMPRUNTEZ PAS A L'AVEUGLETTE

Comment obtenir en un clin d'œil le tableau d'amortissement d'un prêt (applications sur FX-602 et 702 P).

## 21 UN NOUVEAU NUMÉRO SUR LA LIGNE

Cet utilitaire pour PC-1251 renumérote les programmes comme il se doit, mais il permet aussi quelques fantaisies...

## 23 MISEZ P'TIT : OP'TIMISEZ

Un nouveau défi à relever, et les premiers résultats de celui du n° 18.

## 25 LE MORSE : UN TRAIT, UN POINT, C'EST TOUT

Comment, sur ZX 81 ou sur TI-57, se familiariser avec l'alphabet morse.

## 27 UN BONNET ROUGE SUR LE CALENDRIER

Recherches historiques ou généalogiques, la révolution française, de nos jours encore, pose un problème de date. Vous avez dit le 15 Brumaire An VI ? Voyons, voyons... Un programme grégorien pour le PC-1211 et PC-1.

## 30 NOUVEAU : LE CASIO PB-700

Le dernier-né de Casio et ses périphériques. Un ordinateur de poche qui paraît doué pour les graphismes.

## 34 DEMANDE A PIERRE OÙ SE TROUVE PAUL

Trop souvent négligées, les opérations indirectes sont des outils de programmation très puissants.

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'Art. 41, d'une part que « les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et, d'autre part, que les analyses et les courtes citations dans un but d'exemples et d'illustrations, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants-droit ou ayants-cause est illicite » (alinéa 1<sup>er</sup> de l'Art. 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contre-façon sanctionnée par les Art. 425 et suivants du Code Pénal.



# L'Ordinateur de poche

16 F janvier-février 84

## n° 20

### RÉDACTION-RÉALISATION

Directeur de la rédaction : Bernard Savonet  
Rédacteur en chef : Jean Baptiste Comiti  
Rédaction : Anne-Sophie Dreyfus  
Secrétaire de rédaction : Éliane Gueylard  
Assistante de rédaction : Maryse Gros  
Administration : Michelle Aubry  
Ont participé à ce numéro : Olivier Arbey, Michel Arditti, Jean Claude Besse, Jean-Pierre Cayre, Joël Chacornac, Jacques Deconchat, Bernard Demandre, Bernard Elman, Pierre Flener, Olivier Freud, Pierre Ladislas Gedo, Jean-Charles Géménin, Fabrice Gigandet, Jean-Luc Gilliéron, Jean-Luc Goudet, Jean-Claude Guérout, Pedro Inigo Yanez, Jean-Christophe Krust, Xavier de La Tullaye, Bernard Lafay, Yannick Le Nagard, Raoul Lebastard, Jean-Charles Lemasson, John Luck, Jean-Louis Marx, Daniel Saada, Joël Surcouf, Laurent Tordjmann.

Illustrations : Chimulus, Alain Mangin, Alain Mirial, Fabrice Péray, Alain Prigent, Nicolas Spinga.

### ÉDITION-PUBLICITÉ-PROMOTION

Éditeur : Jean-Pierre Nizard  
Assistante d'édition : Maryse Marti  
Publicité : Jean-Daniel Belfond

### VENTES

Diffusion NMPP : Sophie Marnez  
Abonnements : Muriel Watremez  
assistée de Sylvie Trumel

L'Ordinateur de Poche est une publication du **groupe tests**

5 place du Colonel Fabien  
75491 PARIS CEDEX 10  
Téléphone : (1) 240 22 01  
Télex : LORDI 215 105 F

Directeur de la publication : Jean-Luc Verhoye

## 37 LA COURSE AUX CHIFFRES

Un jeu pour PB-100 où l'on doit compter autrement que sur ses doigts.

## 39 OTHELLO SUR PC-1500 LE PROGRAMME DU VAINQUEUR

Lors du dernier Sicob, pour remporter le tournoi international de programmes d'Othello, il « suffisait » d'avoir le programme du vainqueur ! Son auteur accepte aujourd'hui de le rendre public, mais il en prépare certainement un autre encore meilleur...

## 44 CE QUI SE CONÇOIT BIEN...

Les langages de programmation sont extrêmement rigides, et les ordinateurs se rebiffent à la première incartade. Quelques réflexions sur ces questions de vocabulaire et de syntaxe.

## 47 CHERCHEZ LA PANNE...

...ou comment un ordinateur (ici un PC-1251) peut aider à diagnostiquer ce qui ne tourne pas rond dans une voiture.

## 48 AH ! SI VOUS AVIEZ SU...

Pour en savoir plus sur les machines que vous ne connaissez pas à fond.

## 51 LE POT COMMUN

Différents programmes pour PB-100, TI-58/59, PC-1211 et PC-1, FX-702 P et ZX 81.

## 56 UN ANCÊTRE

Minuscule, mais programmable, la Cambridge de Sinclair est une calculatrice qui date de 1973. Dix ans déjà pour cette machine dont le petit boîtier était bourré d'astuces.

Ce numéro contient en encart des bulletins d'abonnement paginés 60 et 61.

### Attention !

Ce vingtième numéro de l'Op est un numéro double (daté janvier-février 84). Il est donc inutile de réclamer le n° 21 à votre marchand de journaux avant la fin du mois de février.

D'autre part, avec la nouvelle année, nous avons emménagé dans des locaux neufs. N'oubliez pas de noter sur vos tablettes notre nouvelle adresse :

L'ORDINATEUR DE POCHE  
5 place du Colonel Fabien  
75491 PARIS CEDEX 10  
Tél. (1) 240 22 01



Notre publication contrôle les publicités commerciales avant insertion pour qu'elles soient parfaitement loyales. Elle suit les recommandations du Bureau de Vérification de la Publicité. Si, malgré ces précautions, vous aviez une remarque à faire, vous nous rendriez service en écrivant au BVP, BP 4508, 75382 PARIS CEDEX 08.

**DOSSIER  
HORS SÉRIE**

# L'ORDINATEUR INDIVIDUEL

*Ce dossier présente une sélection des meilleurs programmes publiés dans l'Ordinateur individuel*

# 500 PROGRAMMES BASIC

Jeux, enseignement, professionnels, utilitaires, etc. et un tableau comparatif de tous les Basic pour pouvoir utiliser les programmes sur : Alice, Apple, Atari 400, Atom, BBC, Commodore 64, DAI, Dragon 32, Hector, Lynx, MZ 80 A, Oric, PET/CBM, TO 7, TI 99, TRS 80, Vic 20, ZX 81, ZX Spectrum.

**en vente dans tous  
les kiosques  
à partir du 10 décembre**



éditorial

## Au doigt et à l'œil

**T**out ordinateur de poche est un compromis entre deux exigences difficilement conciliables : volume réduit (plus l'appareil est petit, plus il est portatif), et confort d'utilisation (on est plus à son aise devant un grand clavier et un grand écran). La mesure ici est tout bonnement le doigt et l'œil de l'utilisateur.

Les ordinateurs « de mallette » se transportent dans un petit bagage à main : porte-documents ou attaché-case. En contrepartie de ce caractère un peu encombrant, ils offrent un afficheur de quatre ou huit lignes et un clavier grandeur nature qui en rendent l'usage agréable.

Du côté des ordinateurs de très petites dimensions, certaines techniques récentes pourraient bien apporter du neuf. C'est ainsi que les téléviseurs miniatures, à écran plat, se développent au Japon ; pour certains d'entre eux, la définition de l'image est maintenant très correcte. Avec ces écrans, on est loin des afficheurs monolignes équipant les premiers poquettes.

Dans un autre secteur de la miniaturisation, les montres numériques, l'évolution est aussi très rapide. On connaissait les montres-bracelets incorporant un jeu électronique ou une calculette, on voit maintenant apparaître des montres faisant fonction de calculatrice scientifique. « Gadgets », diront certains. Toujours est-il que l'on peut fabriquer des ordinateurs occupant le volume d'un paquet d'allumettes... Et c'est là qu'interviennent des nouveautés remarquables. Sans quitter le domaine des montres, on retiendra l'écran-clavier : on presse sur un bouton et l'écran devient le clavier sensitif d'une calculatrice. D'une pierre, deux coups. La même surface est utilisée pour recueillir les données et pour afficher les résultats. Il y a fort à parier que l'on verra des ordinateurs de poche exploitant ce procédé.

Autre innovation : la calculatrice-portefeuille. Elle s'ouvre comme un portefeuille, mais l'une de ses faces (épaisse de deux ou trois millimètres) est en réalité un clavier sensitif.

Plusieurs de ces nouveautés vous sont présentées dans ce numéro. Mais on peut encore prévoir au moins une autre façon d'augmenter le confort d'utilisation d'un ordinateur de petites dimensions : c'est le clavier sans touche où l'on dessine du bout d'un doigt les signes ou les caractères que la machine reconnaît. Une telle machine existe (ce n'est pas un ordinateur, mais un agenda-répertoire). Nous en reparlerons dès que nous en aurons une à l'essai.

Que conclure de tout cela ? Deux tendances en tout cas. La première confirme l'essor de machines relativement volumineuses que l'on emporte avec soi sous forme d'un petit bagage à main. La seconde indique clairement que des solutions originales pourraient déboucher bientôt sur une nouvelle génération d'ordinateurs de poche.

□ l'Op

# LOGICIELS POCKET SOFT®

POUR

## SHARP PC-1500® & TANDY PC-2®

Disponibles en boutique micro-informatique :

PC-CALC3 :	Feuille électronique de calcul,
PC-PLOT3 :	Utilitaire graphique de PC-CALC3,
PC-WORD :	Traitement de textes,
PC-GRAPH :	Graphiques de gestion,
PC-MACRO :	Macro-assembleur,
PC-HEX :	Moniteur hexadécimal avec TRACÉ,
PC-BANK :	Gestion de 4 comptes bancaires,
PC-MATH :	15 fonctions mathématiques, statistiques & financières,
PC-UTIL2 :	17 extensions au BASIC SHARP,
PC-DATABASE :	Gestion de fichiers.
PC-PLAN :	Planning + échéancier
PC-POLYNOMIAL :	Traitement des polynomes avec graphiques

Informations :

**Pocket Soft Ltd**

Bureau Parisien - Tél. : (1) 326.86.70 - Réf. : OP

# 1 MICRO + 1 METHODE = le Basic enfin chez vous

1 METHODE PEDAGOGIQUE SPECIFIQUE

● 1 micro-ordinateur sharp PC 1212 PC 1245 ou PC 1251

fourni (ou non si vous en possédez un). Possibilité Interface ou Imprimante.

- Notions fondamentales (si vous ne possédez pas de connaissances en Informatique)
- Un cours complet de **BASIC**, plus de 200 exercices sur machine avec corrections de nombreux sujets de composition avec contrôle des connaissances.

Disponible : un cours  
pour les possesseurs  
d'un **SINCLAIR**  
**ZX81**

APPRENDRE RAPIDEMENT - EFFICACEMENT - A SON RYTHME - PAR CORRESPONDANCE

**ECOLE UNIVERSELLE-IFOR** - 28, rue Pasteur 92551 Saint-Cloud  
Cedex. Tél. : 771.91.19

Etablissement privé d'enseignement à distance

Conseils-Documentation  
APPELEZ le 771.91.19



Étude gratuite dans le  
cadre de la formation continue après  
accord de l'employeur

Bon pour une documentation gratuite N° 202

Nom, prénom \_\_\_\_\_

Adresse \_\_\_\_\_

Tél. \_\_\_\_\_

Niveau d'études \_\_\_\_\_ Age \_\_\_\_\_

désire recevoir une documentation gratuite sur le cours Initiation/basic.

**ECOLE UNIVERSELLE - IFOR - 28 rue Pasteur 92551 Saint-Cloud Cedex.**  
Tél. 771.91.19

# 6 LIVRES DE POCHE POUR VOS POQUETTES

Les ordinateurs par leur aspect universel sont comme les boîtes à outils : on y trouve ce que l'on y met. Cette collection Mega O Poche n'a pas la prétention de livrer le secret des grandes applications informatiques, mais bien de mettre à la disposition des utilisateurs des petits programmes tout prêts qui leur permettront de résoudre de nombreux problèmes de la vie quotidienne.

Format livre de poche 35,00 FF/250,00 FB/12,20 FS :

**Boîte à outils pour Sharp PC-1251**

par Jean-Pierre Lhoir

**Boîte à outils pour Sinclair et Timex - ZX-81, ZX Spectrum, Timex 1000, 1500 et 2000**

par Marcel Henrot

**Boîte à outils pour PC-1500**

par Jean-Pierre Lhoir et Richard Poret

**Boîte à outils pour FX 702 P**

par Jean-Pierre Lhoir

**Boîte à outils pour Casio PB-100, FX-802 P et TRS-80 PC-4**

par Jean-Pierre Lhoir

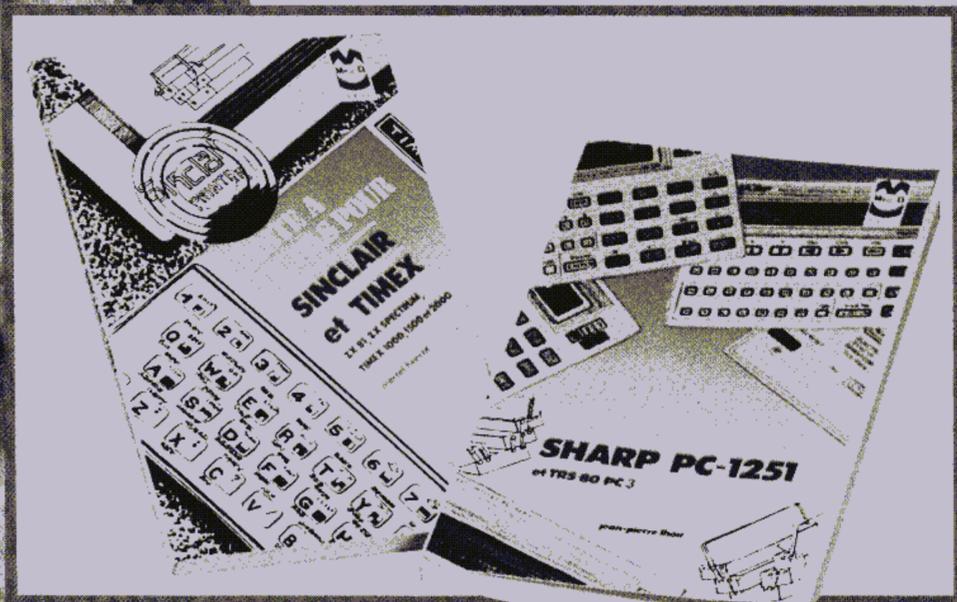
**NOUVEAU**

et aussi :

**NOUVEAU**

**Micro-compta pour Sinclair et Timex - ZX-81, ZX Spectrum, Timex 1000, 1500 et 2000**

35 PROGRAMMES DE COMPTABILITE GENERALE  
par Gaston Miclot



P.S.I. DIFFUSION  
BP 86 - 77402 Lagny-S./Marne Cedex  
FRANCE  
Téléphone (6) 006.44.35

P.S.I. BENELUX  
5, avenue de la Ferme-Rose  
1180 Bruxelles  
BELGIQUE  
Téléphone (2) 345.08.50

P.S.I. SUISSE  
Case postale  
Route Neuve 1  
1701 Fribourg  
Tél. : (037) 23.18.28

au Canada  
SCE Inc.  
65, avenue Hillside  
Montreal (Westmount)  
Quebec H3Z1W1  
Tel. : (514) 935.13.14

Envoyer ce bon accompagné de votre règlement à P.S.I. DIFFUSION ou, pour la Belgique et le Luxembourg à P.S.I. BENELUX ou pour la Suisse à P.S.I. SUISSE

DESIGNATION	NOMBRE	PRIX
		TOTAL

Frais de port : + 10 FF / 50 FB / 1,50 FS

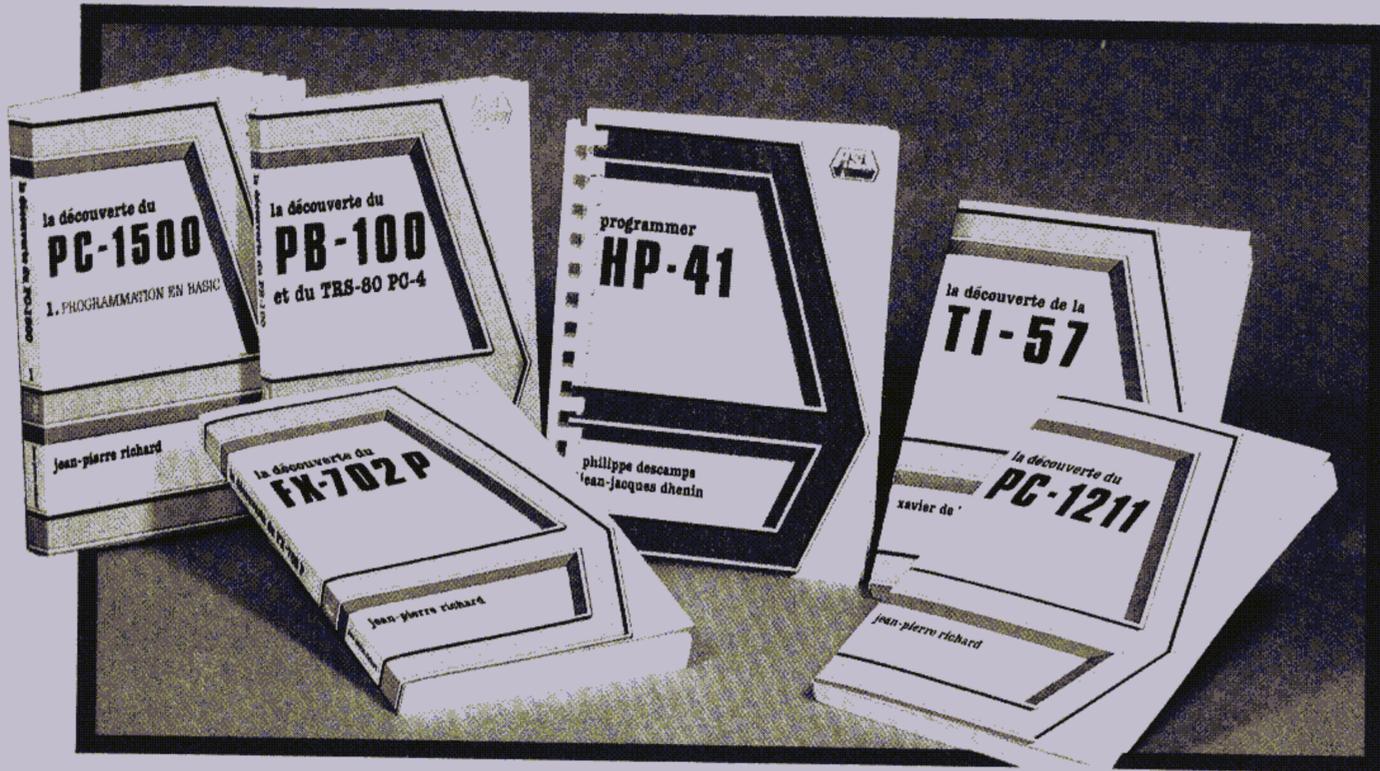
NOM ..... PRENOM .....  
 rue ..... n° .....  
 Code postal ..... Ville .....

OPG 1

AG 09/81

EN VENTE EN BOUTIQUE SPECIALISEE  
ET DANS LES LIBRAIRIES  
GENERALES DE VOTRE VILLE

# LES LIVRES POUR VOS POQUETTES...



## matériels

### La découverte du PB-100

La Pierrick Moigneau

Ce livre dévoile progressivement toutes les facettes de l'ordinateur de poche Casio PB-100 et du Basic à l'aide de nombreux exemples d'application, permettant ainsi aux novices d'entrer en douceur dans le monde de l'informatique.

Série verte - Format : 14,5 x 21  
168 pages - 90,00 FF

### La découverte du PC-1500

par Jean-Pierre Richard

Un ouvrage pour les néophytes curieux qui veulent en savoir plus sur leur ordinateur de poche PC-1500 (ou TRS-80 PC-2). Quelles instructions et commandes emploie-t-il ? Quels types de variables utilise-t-il ? Comment la mémoire est-elle structurée ? Toutes ces questions et bien d'autres trouvent leur réponse dans ce premier tome de "La découverte du PC-1500". Un manuel riche de tous les éléments nécessaires à la programmation en Basic, largement complété d'exercices, d'exemples d'application et... d'un index.

Série verte. Format : 14,5 x 21  
208 pages - 100,00 FF

### La découverte du PC-1211

par Jean-Pierre Richard

Au fil des chapitres, cet ouvrage fera découvrir au non initié instructions et commandes, variables et mémoires, fonctions périphériques. Enrichi d'exercices d'applications et d'un index, ce manuel fournit à l'utilisateur tous les éléments de base nécessaires à la programmation en langage Basic du PC-1211 (ou TRS-pocket).

Série verte - Format : 14,5 x 21  
152 pages - 90,00 FF

### La découverte de la TI-57

par Xavier de la Tullaye

S'adressant aux débutants, cet ouvrage les conduira, dans un langage clair, de l'élémentaire 2 + 2 à des programmes perfectionnés. Après une étude fonctionnelle de la calculette, la programmation est expliquée progressivement, de la conception à la réalisation en s'appuyant sur de nombreux exemples.

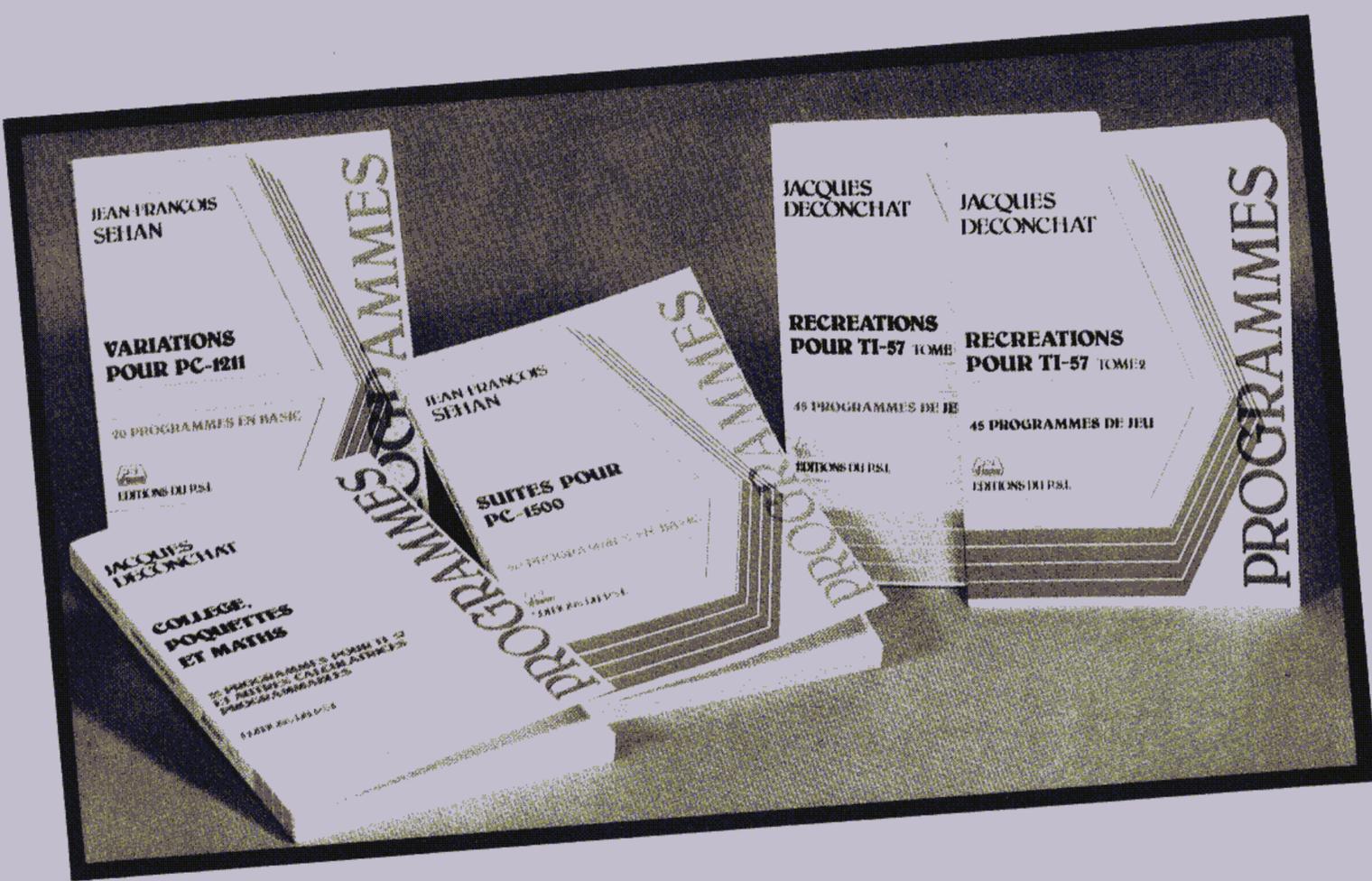
Série verte - Format : 14,5 x 21  
144 pages - 80,00 FF

### La découverte du FX-702 P

par Jean-Pierre Richard

Instructions et commandes, variables et mémoires, fonctions périphériques, cet ouvrage fournit aux débutants tous les éléments de base nécessaires à la programmation en langages Basic de la Casio FX 702 P. Il est complété de nombreux exemples et exercices d'application.

Série verte - Format : 14,5 x 21  
216 pages - 100,00 FF



# A vos claviers

## Des différences mineures mais gênantes

La séquence IF A... correspond sur mon PC-1500 à IF A <> 0... Ceci est important lorsqu'on reprend certains programmes dans votre revue car sur les PC-1500 de première génération, IF A ... correspond à IF A > 0...

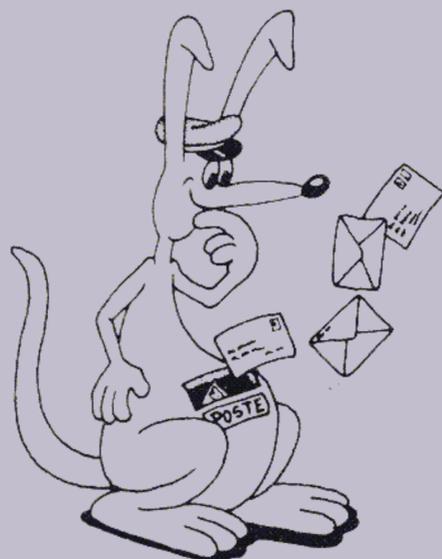
**Jean-Pierre Mégard**  
12 Rodez

Sur mon PC-1500 les instructions IF A PRINT « VRAI » affichent « VRAI » pour toute valeur non nulle (c'est à dire positive et négative) de A, alors que dans l'article « Contre toute logique » de l'Op n° 14 vous écriviez « pour toute valeur positive — exclusivement — de A ». Qu'en est-il vraiment ?

**Pierre Vignacq**  
40 Soustons

■ Il y a en effet entre IF A <> 0 et IF A > 0 une différence de taille... infinie, puisqu'elle concerne l'ensemble des nombres négatifs et nuls ! Si pour la majorité des programmes que nous publions cela importe peu, il en est (comme le « petit bonhomme » de l'Op n° 18 page 11) qui ne fonctionnent pas comme cela serait souhaitable sur tous les PC-1500. De même la boucle FOR I = 0 TO 10 : NEXT I se terminera sur certaines machines avec la valeur 10 (c'est logique) dans la variable I, et d'autres auront 11. Cela peut être gênant quand on utilise dans un calcul cette variable I. D'autres petites choses encore peuvent varier.

La raison en est que Sharp a commercialisé successivement ses PC-1500 avec quatre mémoires mortes Basic différentes. Conséquence de ces améliorations : certains logiciels fonctionnent à merveille sur un modèle et doivent être modifiés pour tourner sur les autres...



## Attention !

Le n° 21 de l'Ordinateur de poche paraîtra à la fin du mois de février.

D'autre part, nous venons d'emménager dans des locaux neufs. N'oubliez pas de noter notre nouvelle adresse :

L'ORDINATEUR DE POCHE  
5 place du Colonel Fabien  
75491 PARIS CEDEX 10

## Des fonctions « cousues main »

Utilisant un PB-100, je cherche à convertir les programmes pour FX-702 P en programmes pour PB-100. Mais, certaines fonctions du FX ne se retrouvent pas sur mon PB-100. C'est le cas de DMS et de DEG, par exemple. A quoi servent-elles et est-il possible de les remplacer ?

**Christophe Cassagnol**  
28 Epernon

■ Ces fonctions peuvent être remplacées... par quelques lignes de programme. Elles ne sont pas comprises, sous cette forme, par le PB-100, ni par le FX-802 P. Mais elles peuvent être transformées en sous-programmes qu'il faudra appeler au bon moment.

DMS transforme des degrés décimaux en degrés, minutes et secondes. Si Z représente les degrés décimaux, le résultat de DMS Z se présente sous la forme D° M' S'' où  $D = \text{INT } Z$ ; et si  $N = 60 * \text{FRAC } Z$ , alors  $M = \text{INT } N$  et  $S = 60 * \text{FRAC } N$ .

Il reste à faire imprimer le résultat. Quant à la fonction DEG, elle fait l'opération inverse. Elle transforme des degrés (D), minutes (M) et secondes (S) en un nombre Z de degrés décimaux. L'opération à effectuer est tout simplement :  $Z = D + M / 60 + S / 3600$ .

Il ne vous reste plus qu'à intégrer ces opérations dans les programmes à transformer. Le résultat apparaît aussi rapidement que sur le 702.

## Les anciennes et les modernes

Dernièrement, j'ai acheté une TI-57 LCD alors que je n'avais que des programmes pour TI-57. Je voudrais savoir si la touche SUM des TI-57 a une touche qui lui correspond sur TI-57 LCD, et si oui, laquelle ?

Merci d'avance.

**Florence Habib**  
Paris 19<sup>e</sup>

Alors que je voulais transformer l'instruction de la TI-57, SUM 7, pour la TI-57 LCD en STO + 7, ma nouvelle calculatrice n'écrivait que STO + (61.85) mais pas le 7. Pouvez-vous me donner la raison de cela ?

Longue vie à l'Op.

**Pascal Kabulo**  
Bruxelles, Belgique

■ Vous êtes nombreux à nous demander si les programmes publiés pour la TI-57 classique peuvent être repris sur la version LCD de la 57.

Si le programme initial est long, s'il utilise plusieurs registres de données, ce n'est généralement pas possible

bien que les deux machines présentent de nombreuses similitudes. La raison principale en est que la TI-57 LCD est dotée d'une mémoire vive un peu plus courte que son aînée. A cela s'ajoute parfois que l'afficheur à cristaux liquides est toujours éteint pendant le déroulement d'un programme, ce qui n'était pas le cas sur la 57. Beaucoup de programmes de jeu utilisent les caractéristiques de cet affichage à diodes.

Néanmoins, si la mémoire n'est pas débordée, un grand nombre de programmes mathématiques sont aisément transposables d'une machine à l'autre, moyennant quelques adaptations minimales. C'est ainsi que SUM 1 (sur TI-57) se traduit STO + 1 (sur TI-57 LCD), Prd 1 devient STO \* 1, etc. L'important ici est de savoir si la 57 LCD, compte tenu de la mémoire consommée par les pas de programme, garde suffisamment de registres de données...

Une autre différence à noter : la mémoire t (registre de test) est directement accessible sur la TI-57 (c'est la mémoire 7) ; sur la LCD, on ne peut l'utiliser qu'en passant par le registre d'affichage. On ne peut donc plus y effectuer des opérations arithmétiques directes. En revanche la LCD dispose de deux fonctions supplémentaires intéressantes lorsqu'elles sont utilisées de façon judicieuse : ce sont n! (factorielle) et Frac (partie fractionnaire).

Mais si,  
ça marche !

Nouvel acquéreur d'un PC-1251 et d'un ZX Spectrum, débutant en la matière, je vous prie de bien vouloir m'indiquer la façon de résoudre sur PC-1251 les deux cas ci-dessous.

```
10 INPUT " " ; B$
20 N = LEN B$
30 FOR I = 1 TO N
40 C$ = LEFT$(B$,I)
50 PAUSE C$
60 NEXT I
70 END
```

Même avec une instruction DIM, la machine refuse de prendre plus de 7 lettres. D'autre part, le PC-1251 refuse l'instruction STEP suivie d'une lettre.

Merci et salutations distinguées.

**Michel Boulmé**  
Donala, Cameroun

■ *Concernant votre première question, la réponse est simple, et c'est bien l'instruction DIM qui vous apportera la solution. Il suffit, dans votre liste d'ajouter une ligne 5 DIM B\$(1)\*24, de remplacer (aux lignes 10, 20 et 40) B\$ par B\$(0), et de remplacer (lignes 40 et 50) C\$ par B\$(1). Vous obtiendrez alors le résultat escompté.*

Au sujet de STEP suivi d'une lettre, c'est-à-dire du nom d'une variable, le PC-1251 l'accepte, avec toutefois deux restrictions. Tout d'abord, la valeur de la variable est tronquée de sa partie décimale : si B vaut - 4,999, STEP B équivaut à STEP - 4. Enfin il ne faut pas que cette valeur soit égale à zéro. Si B vaut - 0,577 ou 0,99 (etc.), le poquette affichera une erreur 9 : l'incrément d'une boucle ne peut pas être nul... C'est très probablement l'origine de vos désagréments.

## Optimiser la partie

Certains poquettes ne possèdent pas la fonction FRAC. Un moyen simple permet de la remplacer. Il évite tous les tests et économise ainsi les

# A vos claviers

pas de programme. Il suffit d'utiliser les fonctions SGN et ABS. SGN N prend la valeur + 1 si N est positif, - 1 si N est négatif et 0 si N est nul. ABS N prend la valeur absolue de N. Aussi, FRAC N peut être remplacé par l'expression suivante : (ABS N - INT ABS N) \* SGN N.

Essayez, pour être convaincu.

**Philippe Trussart**  
Bruxelles, Belgique

■ *Difficile de faire plus court.*

## Une question « indirecte »

La TI-58 C — mon ordinateur de poche — est un instrument assez remarquable. Je me suis amusé à la tester bien des fois et à créer des programmes arithmétiques dont je suis assez fier. Mais voilà : mon manuel a beau me dire que la fonction 2nd Ind augmente considérablement la puissance de ma machine, je n'arrive pas à m'en servir convenablement.

Pourriez-vous décrire l'utilisation de cette fonction dans un de vos numéros ? Je suis sûr que si vous abordez des questions de ce genre, cela comblera beaucoup de vos lecteurs. Approfondir les touches oubliées ou délicates d'emploi, voilà une mine de sujets intéressants. Donnez des exemples. Pour tout vous dire, j'ai un programme qui me trotte dans la tête depuis plusieurs mois et je pressens que cette touche doit y jouer un rôle primordial.

En clair, je veux tout savoir sur la touche 2nd Ind.

**Jean-Michel Dropsit**  
92 La Garenne-Colombes

■ *Coincidence : quand nous avons reçu votre lettre, il y a moins d'un mois, nous avons déjà inscrit au sommaire de ce n° 20 un article traitant de la question qui vous intéresse.*

*Bien sûr, cet article (pages*

*34 à 36) ne vous dira pas tout sur les fonctions indirectes, mais nous pensons qu'il devrait vous « mettre le pied à l'étrier ». Il est rare en informatique que l'on puisse vraiment faire le tour d'une question. C'est d'ailleurs ce qui fait l'un des charmes de cette discipline. Mais il suffit souvent d'être « mis sur la piste » pour entreprendre une exploration dont on n'aurait pas eu l'idée.*

## D'une machine à l'autre

Utilisant un FX-702 P, je voudrais transposer sur mon ordinateur le programme d'Othello que vous avez publié dans l'Op n° 10. Ce programme est prévu pour un PC-1211. Seulement voilà : il y a deux instructions que je ne comprends pas. A la ligne 21, je trouve AREAD R ; et à la ligne 26 IF - S.

Je voudrais aussi ajouter que je trouve votre journal super, et que je lui souhaite une longue (très longue !) vie.

**Laurent Evangelista**  
13 Bouc Bel Air

■ *Sur le PC-1211, comme sur plusieurs autres poquettes Sharp, l'instruction AREAD range dans une variable (la variable R pour reprendre votre exemple) le contenu de l'affichage. Quant au test de la ligne 26 : IF - S, il est vérifié si la variable S contient une valeur négative. On le remplacera donc par IF S < 0.*

## Méfions-nous des puissances...

Je suis assez content de mon ordinateur de poche, un PC-1500, mais il est parfois peu précis dans ses réponses. Ainsi, pour lui,  $96^2$  (soit  $96 \wedge 2$ ) vaut 9215,9999 au lieu de 9216. Il fait sûrement

d'autres erreurs. Est-ce particulier à ma machine ?

**Henri Moulins**  
37 Tours

■ *Oui et non, car si le PC-1500 fait des erreurs, il n'est pas le seul. Ne vous inquiétez pas, c'est normal. Pour certaines fonctions préprogrammées, comme justement les élévations à une puissance, les calculs internes de la machine font appel à des formules qui ne donnent parfois que des solutions approchées.*

C'est pourquoi tous les poquettes ne fournissent pas la même réponse à une opération comme  $9216 - 96^2$ , par exemple. Ainsi, pour le PC-1500, le PC-1251, le PC-1211/1212, c'est 0,00000052. Pour le PC-1401, le FX-702 P, le PB-100, la FX-602 P, la HP-41, ça fait bien 0. Pour la TI-57 et la TI-59, le résultat dépend de la manière dont l'opération a été introduite :  $9216 - (96 x^2)$  = donne bien 0 mais  $9216 - (96 y^x 2)$  = donne - 0,0000111 sur la TI-57 et 0,000000014 sur la TI-59.

Selon le rang des variables, l'opération et le poquette, la solution approchée peut différer plus ou moins de la solution réelle.

## La touche Op et quelques autres

Il y a sur la TI-58/59 certaines instructions dont je ne connais pas la signification : Pgm, Eng, Cp, CMs, Op.

Merci d'avance.

**Fabien Reygrobellet**  
63 Châtel-Guyon

Pour pouvoir traduire sur HP les programmes écrits pour TI, il serait intéressant de connaître les différentes fonctions de la touche Op.

Amicalement.

**Bruno Girard**  
21 Dijon

■ *A défaut de recopier ici le texte du manuel « Programmer soi-même » qui décrit les instructions des TI-58 et 59, voici quelques indications qui permettront d'y voir plus clair :*

- Pgm donne accès à l'un des programmes du module de mémoire morte enfiché dans la TI ;

- Eng permet d'afficher les résultats numériques sous forme d'une mantisse et d'un exposant (puissance de 10) qui est un multiple de 3, ce qui est très commode dans le système métrique (du gramme au kilo, du kilo à la tonne, par exemple, le facteur est de mille, soit  $10^3$ ) ;

- Cp a une double fonction : inscrite dans un programme, cette instruction efface le registre t de la machine, registre de test ; elle équivaut alors à  $0 \times t$  ; frappée au clavier, elle efface le programme en mémoire ;

- CMs remet à zéro le contenu des registres de données excepté le registre t.

Reste la fonction Op comme opération — et non pas comme Ordinateur de poche ! Sur les TI-58 et 59, ces opérations ne sont pas moins de quarante (la TI-58 C en présente même une en plus). La syntaxe en est 2nd Op x x où x x vaut de 00 à 39 et 40 dans le cas de la 58 C. Il est évidemment impossible de les décrire ici toutes en détail. Nous nous contenterons donc de vous en donner un aperçu.

Les neuf premières (Op 00 à 08) sont les instructions grâce auxquelles on pilote l'imprimante PC-100. Les Op 11 à 15 ont trait à des calculs statistiques. Les Op 16 et 17 permettent respectivement d'afficher et de modifier la partition de la mémoire entre pas de programme et registres de données. Les Op 20 à 39 incrémentent ou décrémentent

# A vos claviers

mentent ou décrémentent d'une unité le contenu des registres 00 à 09.

L'ensemble de ces instructions montre clairement qu'il est souvent très difficile de transposer un programme d'une machine sur une autre. Ainsi, sur TI-58/59, 2nd Op 19 lève le drapeau n° 7 si une condition d'erreur est rencontrée. Il n'existe aucun équivalent à cette fonction spécifique sur FX-602 P, sur PC-1211, etc. La traduction terme à terme du programme est, dans ce cas, impossible.

## A défaut d'imprimante

Pouvez-vous me dire ce qu'il faut faire pour vous envoyer des programmes ? J'utilise un PB-100 doté de son extension de mémoire. Certains de mes programmes sont donc longs. Mais je n'ai pas d'imprimante ni d'interface-cassettes. Je ne peux par conséquent que vous envoyer les listes manuscrites...

**Eric Villéger**

45 La Chappelle St Mesmin

■ A l'impossible, nul n'est tenu. Un bon programme, c'est-à-dire un programme à la fois original et intéressant, reste un bon programme

quelle que soit la forme sous laquelle il est transcrit.

Bien entendu, nous préférons de loin les recevoir enregistrés sur un support magnétique (carte ou cassette) et accompagnés d'une liste sortie d'une imprimante. On y gagne beaucoup de temps.

Toutefois, la liste imprimée suffit souvent. Mais nous devons alors, pour essayer le logiciel, retaper la liste au clavier d'un poquette, et cela sans y introduire de faute. Quand la liste est manuscrite, le risque augmente. Les vérifications sont plus longues. Cela n'empêche : il nous arrive souvent de publier des articles dont la version originale est intégralement rédigée à la main, ou tapée à la machine.

Dans tous les cas, il est préférable que vos propositions d'articles originaux comportent un texte explicatif aussi développé que possible (mode d'emploi du pro-

gramme, si programme il y a, exemples d'exécution, etc.). Cela nous fait gagner, à vous comme à nous, beaucoup de temps.

## Renverser l'afficheur ?

Quand j'emporte mon PC-1251 à l'école, il prend une position verticale dans ma poche. Est-ce que cette position peut l'abîmer (je pense aux cristaux liquides).

Amicalement

**Jérôme Pageot**

10 St André les Vergers

■ Non, rassurez-vous. Même si les cristaux sont liquides, il n'existe pas de risque de fuite... En tout cas, nous n'en avons jamais constaté. On peut transporter les ordinateurs de poche à plat, debout, sens dessus-dessous... Si un de nos lecteurs a connu une expérience qui apporte la preuve du contraire, qu'il n'hésite pas à nous la décrire.

## Index des annonceurs

Duriez .....	p. 57
École Universelle .....	p. 6
Éducatel .....	p. 10
JCR .....	p. 62
L'Ordinateur Individuel .....	p. 4
L'Ordinateur Personnel .....	p. 58
Pocket Soft .....	p. 6
PSI Diffusion .....	p. 7, 8 et 9
La Règle à Calcul .....	p. 2



Toute l'équipe de l'Ordinateur de poche vous souhaite une bonne année **1984**

# Magazine

## UN LIVRE



**J'apprends l'ordinateur**  
Jean-Pierre Pouget  
et Michel Lefèvre  
Editions d'Informatique  
Diffusion Scodel,  
Paris, 1982  
Broché, 128 pages  
Prix : 41 FF

■ Rédigé par deux enseignants, *J'apprends l'ordinateur* est articulé autour de deux thèmes : comprendre et faire.

La première partie (comprendre) permet de simuler le fonctionnement d'un ordinateur, sans avoir un ordinateur sous la main. La mise en scène de deux petits personnages (Util pour l'utilisateur et Ordi pour l'ordinateur) est agréable et permet de suivre pas à pas le déroulement de quelques programmes très élémentaires (addition, addition ou multiplication, somme des puissances n-ièmes des p premiers nombres entiers). Le démarrage est assez progressif et bien conçu, malgré une présentation qui manque quelquefois un peu d'ordre. Les auteurs emploient un langage très particulier, le L.P. (pour Langage Premier), qui n'est pas toujours très heureux dans ses formulations (j'avoue que l'expression

« chemise A nombre B » m'a laissé un peu perplexe).

Une deuxième partie reprend les trois exemples précédents en assurant le passage du langage L.P. en Basic et en L.S.E. (Langage Symbolique d'Enseignement). C'est lourd, mais certainement inévitable pour le lecteur dont la seule information (formation ?) repose sur la partie précédente.

Dans la troisième partie, intitulée « Faire », les auteurs proposent des programmes, en général simples, qui permettent de progresser rapidement dans la connaissance des mécanismes élémentaires de l'informatique et de s'initier aux instructions, bien mises en parallèle, du Basic et du L.S.E. Il est assez rare de trouver des ouvrages présentant ainsi des langages

en parallèle, cette initiative mérite donc d'être soulignée. Cette partie m'a paru claire et au bout du compte très réussie. Plutôt destiné aux lycéens qui se préparent à découvrir l'informatique au travers des ordinateurs mis en place dans leurs écoles, ce livre devrait leur faciliter l'accès à ce domaine.

□ JD

Une machine  
très originale

■ L'étui de plastique s'ouvre comme un petit calepin (dimensions : 84 × 140,5 × 12,1 mm) et l'œil reconnaît aussitôt une calculatrice. Mais, tiens, qu'est-ce que viennent faire les touches TEL, MEMO, FILE, EDIT, SECRET ? Et pourquoi donc cet alphabet écrit à même le plastique sur le volet de droite ?

Finalement, cela ressemble à une calculatrice programmable, mais la PF-3000 de Casio est en fait tout autre chose, à la fois agenda téléphonique, carnet de notes et feuille de calcul.

L'afficheur (cristaux liquides) donne dix signes alphanumériques ainsi que des indicateurs divers et nombreux (25 !). L'originalité la plus visible est ce clavier sensitif et souple inclus dans la partie droite de l'étui et supportant l'alphabet. De l'autre côté, sur le corps de la machine, le cla-

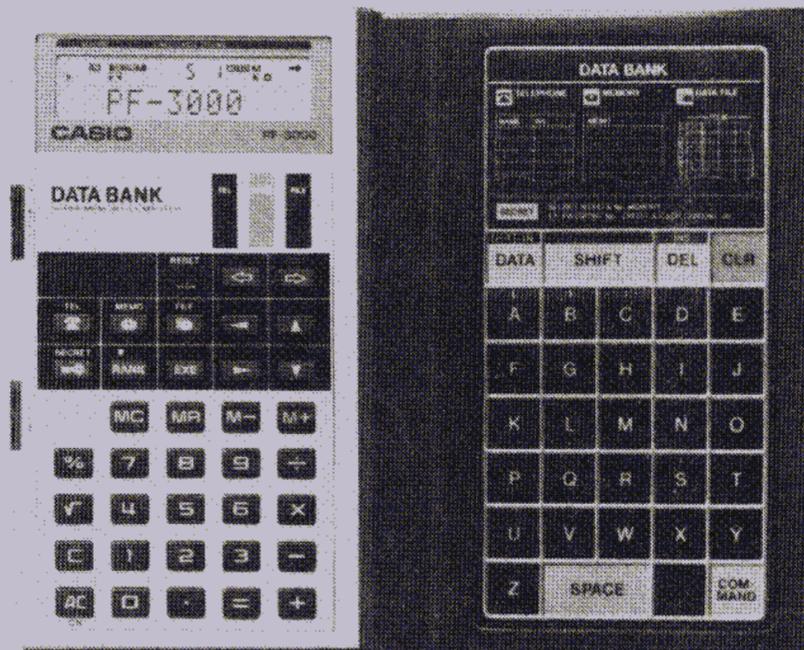
vier est plus classique avec des touches de fonctions et de calcul.

Ces dernières sont d'ailleurs peu nombreuses. La calculatrice elle-même est assez banale : quatre opérations, racine carrée et pourcentage, une seule mémoire enfin que l'on utilise au moyen des classiques MC, MR, M- et M+. Le clavier alphabétique étouffe-t-il la machine ? Apparemment non... Le bilan semble maigre : pas de priorité dans les calculs et même pas de parenthèses. Par contre, la touche % est étonnante, combinée avec les opérations et maniée avec habi-

tude, elle résout tous les problèmes de pourcentage.

Nettement plus originales, la fonction TEL mémorise des noms et les numéros correspondants de téléphone, MEMO présente un carnet de notes et FILE permet de créer des tableaux. Les procédures d'utilisation de ces trois fichiers sont similaires, ce qui facilite grandement les choses. A noter que l'on peut protéger par un mot de passe les informations confidentielles (n° de carte de crédit par exemple).

TEL fournit un tableau spécial à deux colonnes, NAME et NUMBER, avec



La PF-3000, une machine de poche d'un nouveau type

un agréable tri alphabétique immédiat. La recherche est effectuée sur un nom, le début d'un nom ou sur un numéro de téléphone.

MEMO est un carnet de notes organisé en lignes dont chacune peut compter jusqu'à soixante caractères. Ce peut être du texte ou un carnet de rendez-vous et l'on peut accéder directement à une ligne en indiquant ses premières lettres.

FILE est la plus puissante des fonctions. Version de poche des célèbres « tableurs », elle permet de créer et de manipuler des tableaux de calculs avec leurs lignes et leurs colonnes, explicitement nommées. De la même manière que sur les deux types de fichiers précédents, on s'y promène grâce aux quatre touches de déplacement, deux petits chiffres indiquant constamment la position. Comme les grands, ce tableur accepte des colonnes dont les données sont des calculs.

Exemple : si la dernière s'intitule « MOYENNE =  $12 + 13 + 14 \div 3$  », elle contiendra toujours la moyenne des trois premières colonnes. Une ligne spéciale, nommée TOTAL, intègre les sommes par colonne. Tout cela est recalculé à chaque consultation, si vous changez la valeur d'une case, vous constaterez les modifications réalisées là où il le fallait. Avec une fonction supplémentaire qui établit un classement par rang dans chaque colonne, on obtient un outil remarquable qu'il est rarissime de trouver dans une poche. Jusqu'à quatorze tableaux, identifiés par un nom, peuvent coexister, chacun d'eux pouvant contenir  $15 \times 253$  cases. Ces nombres sont théoriques car dans sa version de base, la PF-3000 ne comporte que 961 octets disponibles pour l'utilisateur, ce qui est, somme toute, peu. La machine devient beaucoup plus intéressante avec le module d'extension mémoire de 2 Ko (200 F ttc).

Son domaine d'application reste celui du quotidien, pas question d'entrer un épais carnet d'adresses ou les archives d'une société dans ses tableaux,

mais l'ensemble reste étonnant, pratique et, surtout, il faut en souligner l'originalité. Dans sa version de base, la PF-3000 vaut environ 550 FF, et elle est également commercialisée par **Olympia** sous le nom de **Manager**.

□ JLG

# Magazine

## ■ QUELQUES LIVRES

**Boite à outils pour Sinclair et Timex (ZX 81, ZX Spectrum, Timex 1000, 1500 et 2000)**  
Marcel Henrot  
Editions SAPECA  
Collection MegaO-poche  
Bruxelles, 1983  
Broché, 128 pages  
Prix : 35 FF

**Boite à outils pour Sharp PC-1251**  
Jean-Pierre Lhoir  
Editions SAPECA  
Collection MegaO-poche  
Bruxelles, 1983  
Broché, 128 pages  
Prix : 35 FF

## 50 programmes pour Casio FX-702 P et FX-801 P

Gilles Probst  
Editions Techniques et Scientifiques Françaises  
Collection  
Poche Informatique  
Paris, 1983  
Broché, 128 pages  
Prix : 32 FF

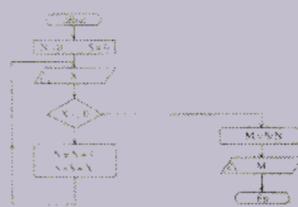
## Mathématiques sur ZX81

Michel Rousselet  
Editions Techniques et Scientifiques Françaises  
Collection  
Poche Informatique  
Paris, 1983  
Broché, 122 pages  
Prix : 32 FF

## ■ UN LIVRE

Maurice Bernadet

introduction à l'informatique  
l'algorithmique



PRESSES UNIVERSITAIRES DE LYON

## Introduction à l'informatique : l'algorithmique

Maurice Bernadet  
Presses Universitaires de Lyon  
Lyon, 1983  
Broché, 228 pages  
Prix : 70 FF

■ L'algorithmique, vous connaissez ? Non, ce n'est pas une nouvelle gymnastique, mais un ensemble d'enseignements pleins de sagesse dont le premier commandement est le suivant : réfléchis d'abord, programme ensuite...

C'est bien l'optique de ce livre : une introduction à l'informatique pour débutants intégraux sans donner un seul programme Basic.

L'informatique sans ordinateur, voilà qui prend le contre-pied de beaucoup d'ouvrages actuels qui commencent par : *Asseyez-vous devant votre machine et tapez PRINT « BONJOUR »*. Ici, c'est la théorie de la programmation qui prime.

L'organigramme est considéré comme le produit fini, tandis que la liste, la suite des instructions qui « tourneront sur l'ordinateur » n'est qu'un sous-produit, relégué au second ordre.

L'auteur travaille sur des exemples bien choisis et variés allant des petits calculs de statistique à des problèmes comptables, en passant par quelques utilitaires. J'aurais aimé un peu plus de jeux qui, à mon avis, forment le champ le plus riche pour l'apprentissage de la programmation. Le lecteur est ainsi amené progressivement à élaborer des organigrammes de plus en plus complexes et à aborder petit à petit les

notions fondamentales de la programmation.

En réalité, l'habitué reconnaîtra dans ces notions-là l'objet d'instructions Basic très précises. L'auteur explique en préface que c'est bel et bien aux futurs utilisateurs de ce langage qu'il s'adresse. Mais alors, si c'est le Basic que l'on apprend, pourquoi ne pas appeler un chat un chat, pourquoi éviter d'écrire FOR...NEXT et préférer parler d'une « séquence répétitive avec test implicite » ? A l'image de cette expression, le ton de l'ensemble est théorique et « professoral », il demande beaucoup d'attention de la part du lecteur, qui doit être un élève studieux.

A mon sens, et contrairement à ce qui semble avoir été recherché, le public intéressé n'est pas celui des tout débutants qui risquent fort d'être un peu « refroidis » par l'aspect complexe donné d'emblée à la programmation.

Le lecteur idéal me semble être le suivant : il a un ordinateur, il a appris à se servir des instructions Basic et a poussé un cri de joie quand sa machine a calculé la surface d'un cercle. Il s'est alors cru programmeur patenté, il a décidé de réaliser un jeu de 421... et il s'est empêtré dans ses lignes de programme.

Celui-là, oui, est prêt à repartir sur des bases plus saines et il comprendra clairement le message de l'auteur : le fond d'un programme est plus important que sa forme et il convient d'abord de résoudre le problème posé avant de le programmer. Il pourra appliquer immédiatement ces concepts en langage Basic, en particulier grâce aux exercices de fin de chapitre et à leurs corrigés, l'un des bons points de cet ouvrage.

□ JLG

# Magazine

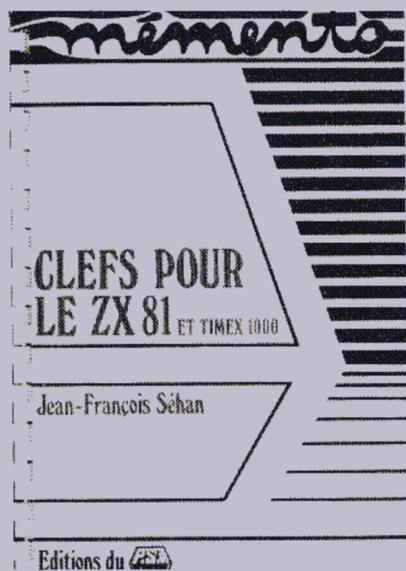
## ■ UN LIVRE

### Clefs pour le ZX 81 et Timex 1000

Jean-François Séhan  
Editions du PSI  
Lagny, 1983  
Reliure spirale, 94 pages  
Prix : 82 FF

■ Cet ouvrage fait partie d'une collection intitulée Memento qui compte pour l'instant huit titres, l'idée générale étant de condenser dans un recueil facile à consulter les renseignements dont on a besoin pour utiliser convenablement un matériel donné.

Précisons tout d'abord que ce livre ne s'adresse pas aux débutants, qu'il ne



contient aucun programme, et relativement peu d'explications. Mais les particularités du langage Basic, du langage-machine, et même les particularités technologi-

ques du ZX y sont résumées et commentées.

L'ouvrage se révèle très pratique pour qui veut utiliser souvent son appareil. On peut, grâce à lui, retrouver rapidement tel ou tel renseignement : toutes les instructions du Basic sont résumées en début d'ouvrage, avec les codes, les priorités de calcul et des indications sur le rôle de chaque instruction ; les cartes haute résolution, le générateur de caractères, etc., n'ont pas été oubliés. Les codes d'erreur sont rappelés, comme les codes de l'écran et du clavier, les adresses des routines du langage, le jeu d'instructions du Z 80, l'architecture de la machine et les circuits utilisés. Quelques trucs et astuces viennent compléter le tout en fin d'ouvrage.

C'est un livre destiné à demeurer en permanence à côté d'un ZX 81, mais il rendra avant tout service à ceux qui dépassent le stade du Basic et commencent à s'intéresser au langage-machine.

□ BE

## ■ UN LIVRE

### Mathématique élémentaire d'un point de vue algorithmique

Arthur Engel  
Adapté de l'allemand par Daniel Reisz  
Editions du Cedic  
Paris, 1979  
Broché, 320 pages  
Prix : 99 FF

■ Un algorithme, c'est comme une recette de cuisine : cela décrit le détail des opérations à effectuer pour aboutir au résultat recherché. S'il est écrit dans un langage spécialisé, l'algorithme en question devient un programme pour ordinateur (de poche ou non).

On peut prendre du goût à concevoir des algorithmes. Si cette activité vous tente, ce livre vous apportera de précieux éléments.

Le premier chapitre est consacré à la présentation du Basic, présentation

## Vers des ordinateurs-bracelets ?

■ Casio vient de commercialiser deux nouvelles montres. Au moment de la parution de cet article, les Casio CFX-200 et TC-500 seront peut-être en vente en France.

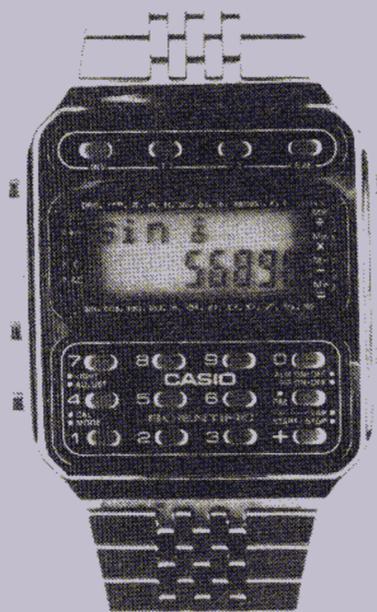
Ces deux nouveautés marquent chacune un pas en avant dans la technologie des montres, mais aussi dans la miniaturisation des calculatrices.

La CFX-200 ne possède pas moins de 24 touches pour les fonctions calculatrice, montre, alarme et chronomètre. La calculatrice dispose de fonctions scientifiques ; on peut ainsi calculer un logarithme ou un cosinus, ou encore convertir des degrés Fahrenheit en degrés Celsius. Il est également possible de faire des

La CFX-200  
montre et  
calculatrice  
scientifique

opérations avec douze niveaux de parenthèses... Par ailleurs, la CFX-200 dispose de trois mémoires permanentes de données. Elle coûte 13 500 yens (environ 470 F) à Tokyo.

La TC-500 n'a rien de particulier a priori. On trouve trois alarmes, la fonction « Dual time », un minuteur et un chronomètre. On ne voit sur cette montre que quatre touches, dont une de réglage. Il s'agit pourtant d'une montre-calculatrice. Il suffit d'appuyer sur la touche de changement de mode pour voir apparaître seize touches sur l'écran à cristaux liquides. Pour les faire



fonctionner, on effleure du bout du doigt ce minuscule clavier. Cela vous permettra de faire additions, soustractions, multiplications et divisions.

Malgré la taille réduite de l'écran, la précision est bonne et il n'est pas nécessaire d'avoir des doigts de fée. La TC-500 coûte 18 000 yens (environ 630 F). Au Japon, une course de vitesse a démarré dans la technologie des montres et des calculatrices de poche. A quand la première montre programma-

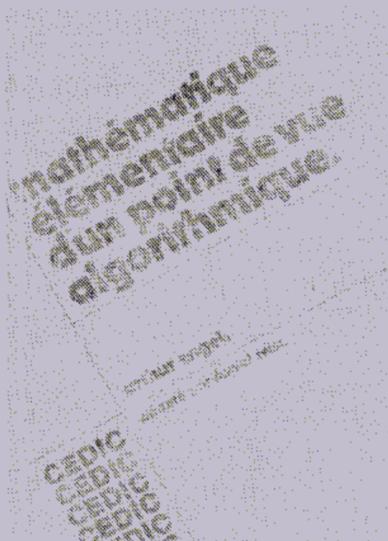
ble en Basic ? D'après certaines rumeurs, Seiko pourrait proposer une telle montre dans le courant de l'année 1984, au Japon bien entendu.

Pour la première calculatrice programmable avec un clavier intégré dans un écran à cristaux liquides, il faudra attendre un peu...

□ JLM



La TC-500 : l'écran de la montre fait aussi fonction de clavier sensible pour la calculatrice intégrée.



claire et pédagogique qui se lit facilement. L'accent a été mis délibérément sur la méthode et le raisonnement : le lecteur ne devrait pas rencontrer de difficulté pour transposer les algorithmes présentés sur son calculateur programmable (les programmes en Basic sont introduits à titre d'exemple, en plus des algorithmes).

Suit une étude des « fonctions scientifiques » que l'on rencontre habituellement sur les calculateurs électroniques. A chaque fois, l'auteur définit la signification de ces fonctions en termes mathématiques, ce qui permettra à plus d'un lecteur d'étendre ou d'approfondir ses connaissances en ce domaine.

Les chapitres 3 à 7 constituent un tour d'horizon du programme actuel des mathématiques de la classe de seconde à la terminale. Ils abordent respectivement la géométrie, l'analyse numérique, les probabilités et la simulation de processus aléatoires, les algorithmes de classement (le chapitre 8 décrivant la solution du casse-tête des « Huit Dames » sur l'échiquier).

L'auteur a bâti son livre comme un ouvrage de référence aussi bien pour l'élève ou son professeur que pour l'ingénieur curieux. Avec un minimum de « bagage mathématique », on peut donc lire les chapitres dans n'importe quel ordre, y compris dans le bon (!), sans se perdre. Chaque point fait l'objet d'explications claires et judicieusement étayées par des exemples qui, bien souvent, répondent aux questions suscitées chez le lecteur.

Rien de rigide dans la démarche : les sujets traités débouchent sur plusieurs solutions différentes, l'important n'est pas tant dans le résultat que dans la démarche adoptée pour y aboutir. Mathématique, algorithmique, programmation sont ainsi abordées ensemble, et ce mélange des genres est l'un des points forts du livre. La présentation mérite elle aussi une mention spéciale (instructions regroupées en sous-routines décalées dans les listes, caractères gras soulignant les endroits importants, etc.). Enfin, deux cent trente cinq exercices et leurs corrigés sont proposés à ceux qui veulent aller plus loin.

Ce livre devrait convenir principalement aux étudiants et aux lycéens (de terminale en particulier), mais bien des autodidactes pourront aussi le découvrir avec profit.  MA

# Magazine

## Du côté des clubs

### En Côte d'Or et en Saône et Loire

■ L'Association des Clubs d'Utilisateurs d'Ordinateurs de Poche (ACUOP) compte maintenant quatre clubs en activité. En voici la liste :

- Chalon-sur-Saône (réunions deux fois par mois). Contact : *Christian Desmigneux*, 8 rue Albert Camus, 71100 Chalon-sur-Saône

- Mâcon (une réunion par mois). Contact : *Bernard Bernigaud*, Les Gaudrioles, Sance, 71000 Mâcon

- Dijon (une réunion par mois). Contact : *Alain Rolland*, 8 rue de Lorraine, 21000 Dijon

- Le Creusot (réunions pendant les vacances scolaires). Contact : *Gabriel Vernot*, 27 rue Saint-Georges, 71200 Le Creusot

L'ACUOP se déclare prête à collaborer avec tous. Principalement consacrée aux pochettes Sharp PC-1500 et Tandy PC-2, elle n'exclut pas l'ouverture sur d'autres ordinateurs de poche. Pour tout renseignement, et notamment pour d'autres régions, écrire à : *ACUOP*, Cheilly les Maranges, 71150 Chagny

## Une version musclée

## de la HP-41 : la CX

■ En 1979, Hewlett-Packard présentait la HP-41 C qui était alors le nouveau haut de gamme de ses calculatrices de poche. Ce modèle était doté dans sa version de base de 63 registres de données, mémoire continue et entièrement négociable en pas de programme, à raison d'un registre pour 7 pas.

La HP-41 CV, apparue en 1981, présentait les mêmes caractéristiques, mais possédait d'origine, sans module d'extension de mémoire, 319 registres de données. Avec la 41 CX, la firme américaine est maintenant sur le point de commercialiser une version encore plus développée de sa machine. Poursuivant sa politique de montée en puissance de la série 41, Hewlett-Packard a incorporé à la version de base de son nouveau modèle plusieurs des options déjà existantes pour les 41 C et CV. Le module *X-fonctions*, par

exemple, est incorporé (portant la mémoire vive disponible à 3101 octets), ainsi que le module *Horloge* enrichi de cinq nouvelles fonctions.

A cela s'ajoutent un éditeur de texte traitant des lignes d'une longueur maximale de 254 caractères, et vingt nouvelles fonctions (dont trois catalogues). La CX serait également dotée d'une batterie complète de tests directs entre le registre x et l'un quelconque des registres de données. Comme on le voit, il y a là de quoi ravir les inconditionnels de la 41... et faire sans doute de nouveaux adeptes.

Si la date de commercialisation effective n'est pas encore connue, elle est imminente ; le prix de la machine devrait être légèrement supérieur à 3 000 FF ttc, à condition, bien entendu, que le cours du dollar ne vienne pas faire des siennes.  JCK

### En Charente

■ Un club de petite informatique vient de naître à Ruffec. Il rassemble principalement des personnes désirant s'initier à l'informatique et met à la disposition de ses membres le TI 99/4 de son président et le PC-1500 de son vice-président. Les machines de poche y sont les bienvenues. Contact : *Jean-Yves Ramos*, 22 rue Jean Jaurès, 16700 Ruffec

### De la TI-57 au HP-75 dans les Pyrénées Atlantiques

■ Autre naissance d'un club, Micrôles, qui s'adresse à ceux et à celles qui désirent s'initier à l'informatique de poche. Contacts : *Pierre-Alexis Lalanne*, 30 Bd des pommes, 64300 Orthez, ou *Eric Roblès*, rue Matachot, 64300 Orthez

# Les prêts à échéances fixes : intérêts et principal...

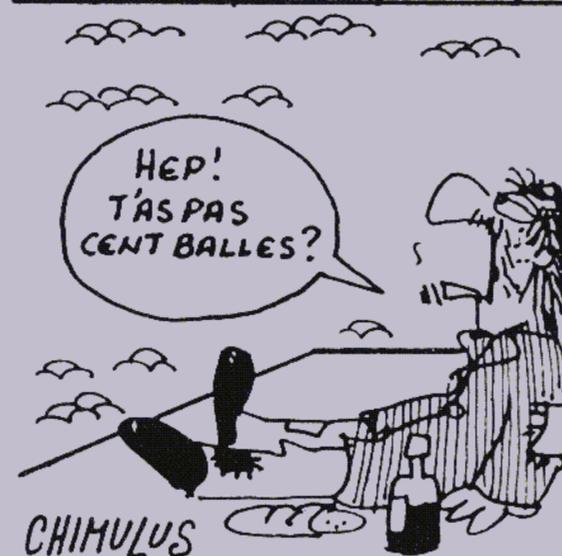
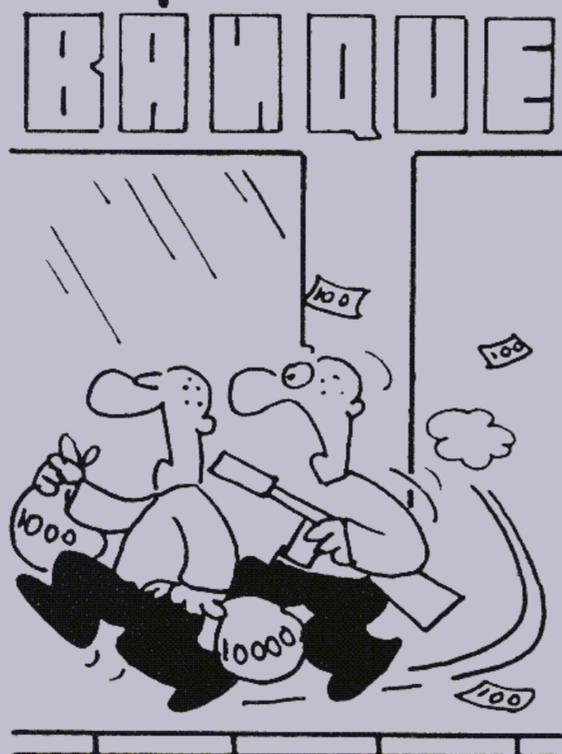
Avant d'accepter un prêt, il est bon de savoir exactement à quoi l'on s'engage, et par la suite de savoir où l'on en est dans ses remboursements. D'où l'utilité des tableaux d'amortissement... que les FX-602 et 702 P peuvent très bien tracer pour vous.

■ Il y a longtemps que l'informatique a fait son entrée dans les banques pour la gestion totale de la comptabilité et les échanges entre banques. Par ailleurs, certains des contacts traditionnels des clients avec leur banquier ont été transformés : guichets automatiques, cartes de crédit...

Mais, au niveau des relations personnelles entre client et banquier, l'utilisation de l'ordinateur, si l'on exclut la gestion informatisée des comptes, est encore assez rare. En effet, dans la plupart des cas, lorsqu'il est question d'un prêt par exemple, les employés de banque sont le plus souvent dans l'incapacité d'en fournir le tableau d'amortissement dans la journée, tributaires qu'ils sont de sources informatiques trop éloignées d'eux.

Si vous disposez d'un FX-702 ou 602 P avec imprimante FP-10, la réponse peut vous être fournie très rapidement. Bien sûr, la largeur du papier de l'imprimante nous obligera à recourir à des collages et, si possible, à une photocopie finale, si nous voulons obtenir un tableau vraiment présentable. Une restriction cependant : cette méthode ne s'applique qu'aux prêts à échéances fixes.

Ces tableaux d'amortissement présentent l'avantage de réunir les éléments indispensables du prêt (le



nombre d'échéances et leur montant) et de reprendre, pour chaque échéance, le montant du capital amorti et celui des intérêts payés.

Tout versement comprend deux parties : la première paie les intérêts, la seconde rembourse la somme avancée par la banque. La formule pour calculer le montant de chaque mensualité est la suivante :  $(\text{Montant de l'emprunt}) \times (i / [1 - (1+i)^{-n}])$  où  $i$  représente le taux d'intérêt ramené au nombre d'échéances par année, et  $n$  le nombre total de versements. A partir de cela, il est possible de savoir, pour chaque mensualité, quelle est la part des intérêts payés et celle du capital remboursé.

Les six  
« données »  
du prêt...

Les intérêts sont égaux à la centième partie du capital restant dû multiplié par  $i$ , et le capital amorti est égal au montant de l'échéance moins les intérêts remboursés. Quant au capital restant encore dû après le paiement de la mensualité considérée, on l'obtient en soustrayant le capital amorti du capital initial.

Sur la FX-602 P, la partie P4 du programme se charge du calcul du montant de l'échéance après que la machine ait demandé les informations nécessaires à l'établissement du tableau, à savoir :

- le montant du capital prêté,
- le taux d'intérêt (entrer le taux annuel rapporté au nombre d'échéances de l'année ; ainsi pour un prêt de 10 000 F sur deux ans et à 18 %, si les versements sont mensuels, le taux est égal à 18/12),
- le nombre d'échéances,
- la périodicité (entrer le nombre de mois séparant chaque versement, soit 1 pour un paiement mensuel, 2 s'il est bimestriel, 3 trimestriel, etc., jusqu'à 12),

- la date de la première échéance sous la forme JJMMAA, ce qui permettra de faire apparaître les dates sur le tableau,
- l'assurance enfin ; en général, les prêts sont assortis d'une assurance décès-invalidité calculée sur le mon-

tant du prêt et que l'on paie à chaque échéance.

Une fois ces éléments entrés, la 602 affiche le montant du versement mensuel (ou bimestriel, etc.) et attend que l'on appuie sur EXE pour établir le tableau. Elle fait alors appel

au programme P7 pour établir les dates. J'ai prévu une totalisation annuelle des intérêts et du capital, puis une totalisation finale. Vous remarquerez sans doute des écarts allant d'un centime à un franc (sur les grands tableaux). Ils proviennent des problèmes d'arrondi inévitables en raison de la taille mémoire de la 602 P. Les calculs du tableau sont ensuite assurés par le programme P8 dont le début calcule et stocke dans les mémoires 7, 8 et 9 le capital amorti, les intérêts et le capital restant dû.

La suite de P8 sert à afficher (grâce au sous-programme P6) la mémoire pointée par M02 ; c'est donc ce seul programme qui sert à l'affichage de tout le tableau. Il suffit de mettre dans M02 la mémoire à afficher (programme P0) et d'appeler P8.

L'ordre dans lequel vont apparaître les colonnes est arbitraire et vous pouvez le changer en sachant :

- que P9 sert à l'impression (SAVE INV EXE) ; si vous n'utilisez pas la FP-10, mettez PAUSE dans P9,
- que P5 affiche le cadre,

Tableau d'amortissement d'un prêt Programme pour FX-602 P Auteur Pedro Inigo Yanez Copyright l'Ordinateur de poche et l'auteur	
M00-19,F-1F 512steps	GSBP9
*** P0	GSBP5
GSBP5	9 Min02
"CAPITAL"	GSBP8
GSBP9	GSBP5
HLT Min11	"*ASSURANCE"
GSBP9	GSBP9
"TAUX"	"*"
GSBP9	GSBP9
HLT Min13	GSBP5
GSBP9	14 Min02
"NBRE ECHEAN"	GSBP8
GSBP9	GSBP5
HLT Min17	"*MONTANT"
GSBP9	GSBP9
"PERIODICITE"	"*ECHEANCE"
GSBP9	GSBP9
HLT Min15	GSBP5
GSBP9	MR12 + MR14 = Min05
"DATE JJMMAA"	5 Min02
GSBP9	GSBP8
HLT Min16	...245steps
GSBP9	*** P4
"ASSURANCE"	MR13 ÷ 100 = Min03 ÷
GSBP9	( 1 - ( 1 + MR03 )
HLT Min14	x <sup>y</sup> MR17 +/- ) = x
GSBP9	MR11 = FIX2 Min12
GSBP4	...028steps
"MT ECHEANCE"	*** P5
GSBP9	"*****"
MR12 + MR14 =	GSBP9
GSBP9	...015steps
HLT	*** P6
GSBP7	"#####*"
GSBP5	GSBP9
"*"	...015steps
GSBP9	*** P7
"*INTERETS"	GSBP5
GSBP9	"*DATE"
GSBP5	GSBP9
8 Min02	"*ECHEANCE"
GSBP8	GSBP9
GSBP5	GSBP5
"CAPITAL"	MR17 Min00
GSBP9	13 MinF
"*AMORTI"	MR16 ÷ 4 10 <sup>x</sup> = Min03
GSBP9	INT M-03 Min01
GSBP5	MR03 x 2 10 <sup>x</sup> = Min03
"*CAPITAL"	INT Min02 Min16 M-03
GSBP9	MR03 x 2 10 <sup>x</sup> = Min03
"*RESTANT DU"	LBL1
	MR02
	" * AR01 .##. AR03 *
	"
	GSBP9
	DSZ GOT02
	GOT04
	LBL2
	MR15 M+02
	MR02 x≥F GOT03
	GOT01
	LBL3
	"*TOTAL AR03 "
	GSBP9
	1 M+03
	12 +/- M+02 GOT01
	LBL4
	"*TOTAL AR03 "
	GSBP9
	...108steps
	*** P8
	MR17 Min00
	MR11 Min09
	MR16 Min01
	AC Min03 Min04
	13 MinF
	LBL0
	MR09 x MR13 % = FIX2
	Min08 +/- + MR12 =
	Min07 M-09
	IND MR02 M+03 M+04
	FIX2
	GSBP6
	DSZ GOT01
	GOT05
	LBL1
	MR15 M+01
	MR01 x≥F GOT02
	GOT00
	LBL2
	MR02 - 9 = x=0 GOT03
	MR03 M-03 FIX2
	GSBP6
	GOT04
	LBL3
	" "
	GSBP9
	LBL4
	12 +/- M+01 GOT00
	LBL5
	MR02 - 9 = x=0 GOT06
	MR03 FIX2
	GSBP6
	MR04
	"#"
	GSBP9
	LBL6
	...006steps
	*** P9
	SAVE invEXE
	...003steps

### Sur la FX-602 P

**Utilisation des mémoires :**  
M00 : adressage indirect, pointeur  
M01 : calcul du mois  
M02 : adressage indirect  
M03 : totalisation annuelle  
M04 : totalisation globale  
M07 : capital amorti  
M08 : intérêts  
M09 : capital restant dû  
M11 : capital initial  
M12 : échéance (montant)  
M13 : taux  
M14 : assurance  
M15 : périodicité  
M16 : date première échéance  
M17 : nombre d'échéances

**Utilisation du programme :**  
Lancer le programme P0, entrer le capital (EXE), le taux d'intérêt divisé par le nombre de versements en une année (EXE), le nombre d'échéances au total (EXE), le nombre de mois entre deux paiements, de 1 à 12 (EXE), la date de la première échéance entrée sous la forme « jour, mois, année » : JJMMAA (attention au passage d'un siècle à l'autre, de 1999 à l'an 2000) (EXE), enfin, entrer le montant de l'assurance versé à chaque fois (EXE). Le programme s'arrête sur le montant de l'échéance ; relancer alors avec EXE et les données s'imprimeront au fur et à mesure, suivies par le tableau d'amortissement. Il ne reste plus qu'à découper, coller et photocopier.

**Exemple d'exécution sur FX-602 P**

```
*****
CAPITAL      6000.
TAUX         0.095833333
NBRE ECHEAN 12.
PERIODICITE  1.
DATE JJMMAA 251282.
ASSURANCE    1.23
MT ECHEANCE  530.82
```

On emprunte 6 000 F à un taux de 10.75 % que l'on rembourse en 12 mensualités.

*DATE	*INTERETS	*CAPITAL	*CAPITAL	*ASSURANCE	*MONTANT
*ECHEANCE	*INTERETS	*AMORTI	*RESTANT DU	*	*ECHEANCE
*25.12.82*	53.75*	475.84*	5524.16*	1.23*	530.82*
*TOTAL 82	53.75*	475.84*		1.23*	530.82*
*25.1.83*	49.49*	480.10*	5044.06*	1.23*	530.82*
*25.2.83*	45.19*	484.40*	4559.66*	1.23*	530.82*
*25.3.83*	40.85*	488.74*	4070.92*	1.23*	530.82*
*25.4.83*	36.47*	493.12*	3577.90*	1.23*	530.82*
*25.5.83*	32.05*	497.54*	3080.26*	1.23*	530.82*
*25.6.83*	27.59*	502.00*	2578.26*	1.23*	530.82*
*25.7.83*	23.10*	506.49*	2071.77*	1.23*	530.82*
*25.8.83*	18.56*	511.03*	1560.74*	1.23*	530.82*
*25.9.83*	13.98*	515.61*	1045.13*	1.23*	530.82*
*25.10.83*	9.36*	520.23*	524.90*	1.23*	530.82*
*25.11.83*	4.70*	524.89*	0.01*	1.23*	530.82*
*TOTAL 83	301.34*	5524.15*		13.53*	5839.82*
	355.09	5999.99		14.76	6369.84

• et qu'il faut mettre dans M02 la variable à afficher (bis repetita).

Après quoi, vous pourrez faire votre tableau dans l'ordre que vous voulez. Pour soigner sa présentation, j'ai dû utiliser beaucoup de mémoire pour les textes. Il reste maintenant une dizaine de pas.

Vous devriez désormais, grâce à ce programme, être en mesure de concurrencer votre banquier, de dis-

cuter avec lui des modalités du prêt qui vous concerne, et de mieux tenir vos comptes. Vous saurez par exemple quand vous pourrez rembourser le prêt par anticipation. Si, inversement, vous travaillez dans une banque, ce programme vous sera utile pour satisfaire vos clients. Dans tous les cas, informez-vous des taux, de l'assurance, de la durée... Ces renseignements devraient toujours figurer sur une offre préalable de prêt.

**Prenons les ciseaux**

Nous avons maintenant une longue liste de chiffres qui doit finalement composer un tableau. Pour ce bricolage, il nous faudra : du ruban adhésif (invisible de préférence), du papier, des ciseaux et, accessoirement, une photocopieuse. Coupez la partie droite de votre liste aluminisée au ras des petites étoiles qui forment le cadre du tableau, puis les petites bandelettes à partir des titres.

Il ne reste plus qu'à coller les bandes en les faisant se chevaucher pour qu'elles arrivent au bord du trait du tableau.

Après avoir réuni toutes les pièces du puzzle, vous pourrez photocopier l'ensemble et vous obtiendrez un petit tableau qui aura l'air d'être sorti d'une imprimante grande largeur.

□ Pedro Inigo Yanez

**Tableau d'amortissement d'un prêt**

Programme pour FX-702 P

Auteur Pedro Inigo Yanez

Copyright l'Ordinateur de poche et l'auteur

VAR: 36 PRG: 1600

P0: 792 STEPS

```
1 MODE 7:WAIT 0:I
  NP "CAPITAL",K,
  "TAUX",T,"NBRE
  ECHEANCES",N
2 INP "PERIODICIT
  E",P,"DATE JJMM
  AA",D,"ASSURANC
  E",A0:T/100
3 A1=RND((T/(1-(1
  +T)^-N)*K),-3):
  A5=A1+A0:PRT "E
  CHEANCE",A5:STO
  P
4 GSB 25:PRT "*DA
  TE", "*ECHEANCE"
  :GSB 25:X=0
5 J=INT (D/1E4):M
  =INT (D/100)-(J
  *100):A=D-(J*1E
  4)-(M*100):Y=M
6 PRT J:":":##:Y:
  ":":A:":":X=X+1
```

```
:IF X=N THEN 9
7 Y=Y+P:IF Y<12 T
  HEN 6
8 Y=Y-12:PRT "TOT
  AL":A:A=A+1:GOT
  0 6
9 PRT "TOTAL":A:B
  =25:C=24:D=15
10 GSB B:PRT "*INT
  ERETS", "*":GSB
  B:X=2:GSB D
11 GSB B:PRT "*CAP
  ITAL", "*AMORTI"
  :GSB B:X=4:GSB
  D
12 GSB B:PRT "*CAP
  ITAL", "*RESTANT
  DU":GSB B:X=3:
  GSB D
13 GSB B:PRT "*MON
  TANT", "*ECHEANC
  E":GSB B:X=5:GS
  B D
```

```
14 GSB B:PRT "*MON
  TANT", "*ASSURAN
  CE":GSB B:X=0:G
  SB D:MODE 8:END
```

```
15 Y=0:Z=0:V=0:A3=
  K:W=M
16 A2=RND((A3*T),-
  3):A4=A1-A2:A3=
  A3-A4:Y=Y+A(X):
  Z=Z+A(X)
17 A=A(X):GSB C:V=
  V+1:IF V=N THEN
  23
18 W=W+P:IF W<12 T
  HEN 22
19 IF X=3:PRT "
  *":GOTO
  22
20 A=Y:GSB C:Y=0
22 W=W-12:GOTO 16
23 IF X=3:A=Y:GSB
  C:A=Z:GSB C:RET
  URE
24 PRT "#####.##:
  A:":":RET
25 PRT "*****
  *":RET
```

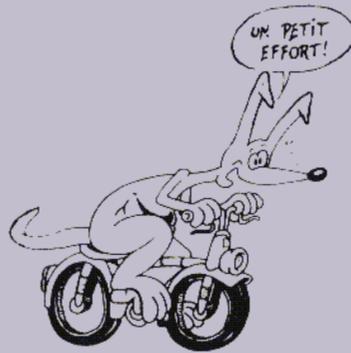
Avant d'utiliser le programme, faire DEFM 1.

**Sur la FX-702 P**

**Principales variables utilisées :**

- A à D : utilisées pour les appels de sous-programmes
- K : capital initial
- J : jour
- M : mois
- N : année
- P : périodicité
- T : taux d'intérêt
- A0 : assurance
- A1 : montant de l'échéance
- A2 : Intérêts
- A3 : capital restant dû
- A4 : capital amorti
- A5 : montant de l'échéance + assurance

L'utilisation du programme est la même que pour la 602 P : lancer le programme par F1 P0. Lorsque le montant de l'échéance est affiché, presser sur la touche CONT pour établir le tableau.



# Renumérotation et PC-1251 : quelques surprises

Quelques lignes de Basic qui « piquent » et « poquent » aux bons endroits, sept ou huit touches à presser, un peu de patience, et voilà : c'est fait, vous pouvez lister votre programme avec ses nouveaux numéros de ligne.

## Renumérotation

Programme pour PC-1251  
Auteur Bernard Demandre  
Copyright l'Ordinateur de poche et l'auteur.

```

900: "N"X=32817: INPUT "L
      IGNE ORIGINE ? ";L,"
      INCREMENT ? ";I:D=L
910:C=224:E= INT (L/100)
      :C=C+E: POKE X,C:U=L
      -E*E2: IF (U)=0) AND
      (U<10) LET H=U: GOTO
      930
920:R$= STR$ U:H=16* ASC
      R$+ ASC MID$ (R$,2,1
      )-816
930:X=X+1: POKE X,H
940:X=X+1: IF PEEK X<>0
      GOTO 940
950:X=X+1:L=L+I: IF PEEK
      X<233 GOTO 910
960:BEEP 2: PRINT "TERMI
      NE": GOTO D
  
```

■ Si l'on est loin de tout connaître sur le fonctionnement interne du PC-1251, on en sait assez pour réaliser de petites applications utiles (1). Voici un programme de 7 lignes (226 octets) qui crée une routine de renumérotation automatique des lignes, fonction inexistante sur le PC-1251.

Rappelons rapidement comment sont codées les lignes d'un programme. On trouve d'abord deux octets représentant le numéro de la ligne :

- le premier octet, dit de « poids fort », vaut 224 si le numéro de la ligne est compris entre 1 et 99 inclus, il vaut 225 pour les numéros compris entre 100 et 199, 226 pour les numéros 200 à 299, et ainsi de suite jusqu'à la valeur 233 (numéros 900 à 999) ;

- dans le deuxième octet, dit de « poids faible », les unités et les dizaines du numéro de ligne sont codées en décimal à partir du nombre interprété comme hexadécimal (10 s'écrit 16, 11 s'écrit 17, etc.).

Viennent ensuite les différents codes des instructions composant la ligne, puis un dernier octet, dont la valeur est 0, signale la fin de la ligne. Immédiatement après, on trouve les deux octets représentant le numéro de la ligne suivante.

Le principe du programme de renumérotation est donc simple : il consiste à introduire dans les deux premiers octets de chaque ligne le code convenable à l'aide de l'ins-

truction POKE. L'octet à zéro qui marque la fin d'une ligne permet de passer au numéro suivant par incrémentation de l'adresse mémoire.

———— Choisissez ————  
———— vos numéros ————

Après avoir entré le programme au clavier, si l'on dispose du combiné-magnétophone, on l'enregistrera pour le réutiliser par la suite avec la commande MERGE. Dans le cas contraire, le plus commode est sans doute de le conserver aux lignes où il se retrouve. On lance son exécution par DEF N. Au message « LIGNE ORIGINE ? », on répond par le numéro que l'on veut voir attribuer à la première ligne. Le poquette demande alors « INCREMENT ? », et l'on indique la valeur qui sera ajoutée au numéro de la première ligne pour obtenir le deuxième numéro, puis ajoutée au

## Un Basic un peu chamboulé

Le programme de renumérotation a été conçu pour des applications sérieuses, mais il permet de s'éloigner un peu de la programmation classique. Trois courts programmes d'essai en apportent la preuve page suivante.

(1) Voir l'Op n° 12 p. 37 et n° 13 p. 55.

# Renumérotation et PC-1251 : quelques surprises

## Programme de départ :

```

10:"X" WAIT 15: GOTO "A"
"
20:"A" PRINT "A": GOTO
"3"
30:"B" PRINT "B": GOTO
"C"
40:"C" PRINT "C": GOTO
"D"
50:"D" PRINT "D": GOTO
"E"
60:"E" PRINT "E": GOTO
"F"
70:"F" PRINT "F": GOTO
"G"
80:"G" END

```

le même programme renuméroté depuis la ligne 0 avec un incrément nul (!) :

```

0:"X" WAIT 15: GOTO "A"
"
0:"A" PRINT "A": GOTO
"3"
0:"B" PRINT "B": GOTO
"C"
0:"C" PRINT "C": GOTO
"D"
0:"D" PRINT "D": GOTO
"E"
0:"E" PRINT "E": GOTO
"F"
0:"F" PRINT "F": GOTO
"G"
0:"G" END

```

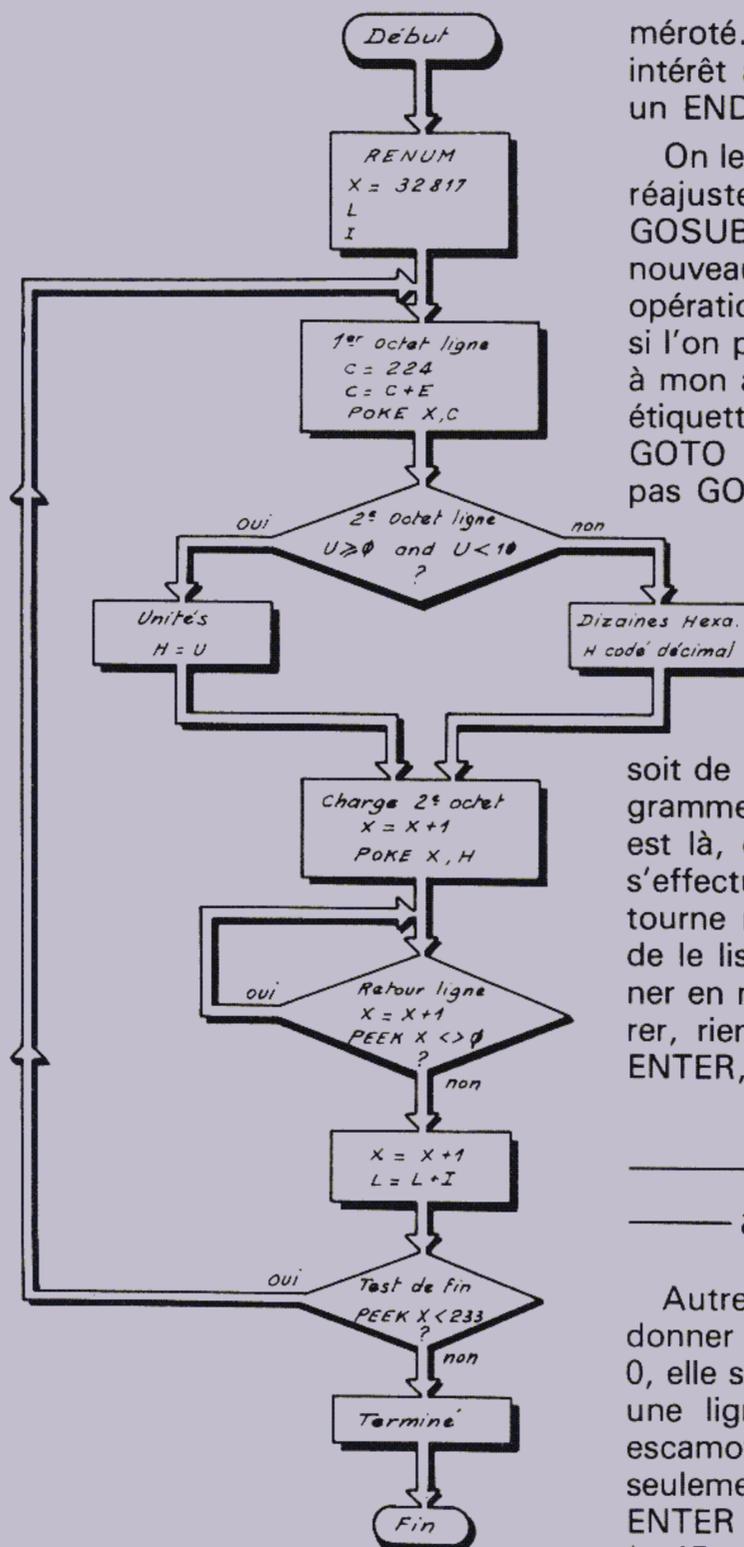
Cette fois-ci, le numéro de départ est 7 et l'incrément vaut -1 :

```

7:"X" WAIT 15: GOTO "A"
"
6:"A" PRINT "A": GOTO
"3"
5:"B" PRINT "B": GOTO
"C"
4:"C" PRINT "C": GOTO
"D"
3:"D" PRINT "D": GOTO
"E"
2:"E" PRINT "E": GOTO
"F"
1:"F" PRINT "F": GOTO
"G"
0:"G" END

```

Les trois versions fonctionnent aussi bien les unes que les autres.



méroté. On aura, dans certains cas, intérêt à remplacer le GOTO D, par un END.

On le fera en particulier si l'on doit réajuster la valeur des GOTO, GOSUB, THEN en fonction des nouveaux numéros de ligne. Cette opération sera faite en un clin d'œil si l'on prend l'habitude — excellente à mon avis — d'utiliser les sauts sur étiquettes (GOTO « A », et non pas GOTO 100, GOSUB « F », et non pas GOSUB 250, etc.).

Voyons maintenant quelques tours de passe-passe. Demandons par exemple que la première ligne ait pour numéro -100 et que l'incrément

soit de 1. Que se passe-t-il ? Le programme s'évapore. Et pourtant, il est là, et si tous ses branchements s'effectuent vers des étiquettes, il tourne rond. Essayez, par curiosité, de le lister, et surtout faites-le tourner en mode TRON. Pour le récupérer, rien de plus facile : DEF N, 10 ENTER, 10 ENTER, et il réapparaît.

## Des listes à rebrousse-poil

Autre fantaisie : vous pouvez donner à la première ligne le numéro 0, elle sera listée et exécutée comme une ligne normale. On peut aussi escamoter le début du programme seulement : DEF N -15 ENTER 5 ENTER créera trois lignes invisibles (-15, -10 et -5), une ligne 0, puis les lignes 5, 10, 15 etc.

Plus bizarre encore : rien n'empêche de donner à toutes les lignes d'un programme le même numéro, il suffit de donner la valeur 0 à l'incrément. Et si l'on donne à l'incrément une valeur négative, les numéros de ligne vont par ordre décroissant. Dans ces deux derniers cas, bien sûr, les programmes ordinaires ne tournent pas. Si l'on veut tout de même pouvoir les exécuter, il faut que chaque ligne soit identifiée par une étiquette et se termine toujours par un branchement incondionnel vers une autre étiquette.

□ Bernard Demandre

————— Quand on —————  
————— escamote —————  
————— un programme —————

Comme on peut le voir, la fin de la ligne 960 renvoie automatiquement au début du programme renuméroté.

# Misez p'tit : Op'timisez !

Si jongler avec la pile opérationnelle de votre HP-41 C, traquer la milliseconde perdue et rogner le moindre octet est votre pain quotidien... Ou si, à l'inverse, vous échappe parfois un peu de la subtile recherche des programmes en Notation Polonaise Inverse... Alors, voici qui doit vous intéresser !



En matière de programmation, est-on jamais certain d'avoir fait aussi bien que possible ? Dans cette rubrique, les défis se succèdent (faire plus court ou plus rapide) et les records tombent.

## Programme Size n° 1

Programme pour HP-41 C

Auteur Jean-Claude Guérout

Copyright l'Ordinateur de poche et l'auteur

```

01 *LBL "SZ1
"
02 -1
03 512
04 2
05 *LBL 00
06 /
07 ST+ Y
08 SF 25
09 VIEW IND
Y
10 FC?C 25
11 ST- Y
12 LASTX
13 X<=Y?
14 GTO 00
15 RDN
16 +
17 CLD
18 .END.

```

■ Jean-Claude Guérout a lancé dans ces colonnes, au mois de novembre, le défi suivant : « trouver le SIZE, autrement dit la taille nnn de la mémoire des données (instruction SIZE nnn), à l'aide du plus petit programme possible. N'utiliser que des fonctions classiques de la HP-41 C et trouver le résultat en moins de 4 secondes quelle que soit la configuration ».

Le programme réalisé par Jean-Claude Guérout, qu'il fallait battre de vitesse, occupe 27 octets, trouve un SIZE 023 en 3 secondes 23/100èmes en n'utilisant que de la pile opérationnelle. Il est reproduit ci-contre.

\_\_\_\_\_Une méthode\_\_\_\_\_

\_\_\_\_\_simple :\_\_\_\_\_

\_\_\_\_\_la dichotomie\_\_\_\_\_

La méthode suivie pour trouver le SIZE est celle de la dichotomie, très employée en informatique puisqu'elle permet de trouver à coup sûr un nombre inconnu compris entre deux valeurs connues avec une efficacité optimale. Elle mérite sans doute d'être expliquée.

Pour l'illustrer, jouons un peu. Choisissez un nombre compris entre 1 et 512. Ne me le montrez pas (!), je vais le deviner en 8 essais au maximum. D'ailleurs, pour prouver que je ne triche pas, vous énoncerez pour moi mes réponses...

Si, en questionnant : « est-ce le nombre x ? », je trouve, c'est fini. Si

non, indiquez-moi seulement si mon essai avec le nombre x est trop élevé ou trop bas. On y va ?

« Est-ce le nombre 256 ? » (si oui, choisissez-en vite un autre, c'est trop facile !). Si 256 est **trop haut**, retranchez 256/2, c'est-à-dire 128 ; mon nouvel essai sera 128. Si 256 est **trop bas**, ajoutez 256/2, mon nouvel essai est donc 384.

## Programme Size n° 2

Programme pour HP-41 C

Auteur Laurent Tordjmann

Copyright l'Ordinateur de poche et l'auteur

```

01 *LBL "SZ2
"
02 ,
03 25
04 XEQ 01
05 LASTX
06 -
07 7
08 XEQ 01
09 LASTX
10 -
11 1
12 *LBL 01
13 SF 25
14 ABS
15 *LBL 00
16 X<> L
17 +
18 RCL IND
X
19 FS? 25
20 GTO 00
21 .END.

```

Misez p'tit :  
Op'timisez !  
HP-41

La logique est simple : ajouter une valeur si l'essai est trop bas, en retrancher une si l'essai est trop haut. Le problème est : quelle valeur ? Il s'agira toujours de diviser en deux l'espace d'incertitude : le nombre est compris entre a et b, la question sera « *est-il supérieur ou inférieur au nombre situé juste au milieu de l'intervalle de a à b ?* », et ainsi de suite jusqu'à la solution. Cette valeur, dans le cas présent, est déterminée par la petite formule suivante :  $256 / (2 \wedge n^\circ \text{ de l'essai})$  où le signe  $\wedge$  signifie « à la puissance ». Les valeurs consécutives à ajouter ou retrancher sont donc : 128, 64, 32, 16, 8, 4, 2 et 1. Continuez le jeu sans moi, de toute façon, le nombre inconnu sera trouvé en 8 essais, au maximum, comme promis.

Encore  
plus vite

Jean-Claude Guérout a donc employé cette méthode car « *trouver le SIZE* » n'est rien d'autre que trouver un nombre compris entre 0 et 319. La question est posée à la HP-41 C sous cette forme : « *est-ce que le registre  $n^\circ x$  existe ?* » et s'écrit VIEW IND x. Une erreur se produit si le registre Rx n'existe pas, signifiant que le SIZE est inférieur à x. Cette erreur est décelée par la mise à zéro du flag 25. Si le registre existe, c'est que le SIZE lui est supé-

## Qui dit mieux ?

■ Il m'est arrivé, au cours d'un exercice de mathématiques touchant aux intégrales, d'avoir à calculer la fonction suivante : pour tout nombre x compris entre 0 et 1, f(x) lui associe le même nombre dont on a seulement permuté les deux premières décimales.

Mais si, c'est simple ! Par exemple, si x vaut 0, 14 159 il faut trouver f(x) qui vaut 0, 41 159, de même  $f(0, 77 19) = 0, 77 19$ ,  $f(0, 39 1) = 0, 93 1$ , etc...

Saisissant aussitôt ma calculatrice préférée, j'ai programmé cette fonction en 26 octets et 19 pas de programme (sans compter le LBL de tête ni le END final). Le résultat avec le nombre 0,141592654 est trouvé en 73/100èmes de seconde : à vous de mieux faire. A titre de curiosité, l'intégrale de cette fonction entre 0 et 1 vaut 1/2.

□ Olivier Arbey

rieur (d'une logique imparable, non ?) et le flag 25 reste armé. Dans le premier cas on ajoute, dans le second on retranche, et le tour est... programmé.

Notez enfin les premiers pas du programme : on économise deux fois ENTER entre -1, 512 et 2. Ceci est obtenu en programmant -1 (SST) 512 (SST) et 2, gain : 2 octets.

Mais Laurent Tordjmann a lui aussi développé une méthode rapide de détermination du SIZE : 19 pas, 30 octets et trouve SIZE 023 en 1

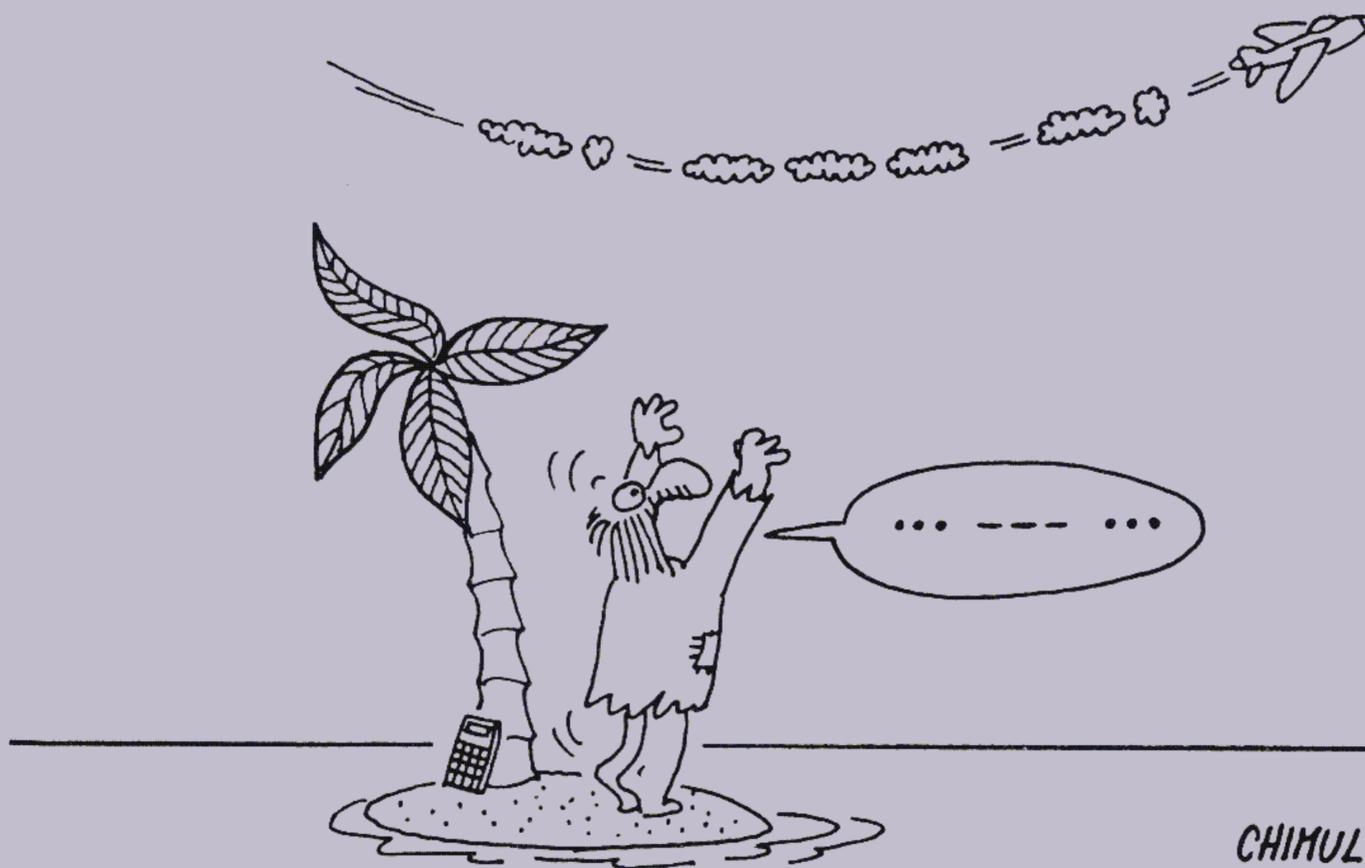
seconde 54/100èmes, ce qui signifie un gain de temps très appréciable (1 sec. 27/100èmes) mais une perte de 3 octets.

La méthode est très simple et utilise elle aussi le flag n° 25 de contrôle d'erreur avec l'instruction RCL IND x. La recherche se fait de 25 registres en 25 registres, puis par pas de 7, puis de 1. On change de pas d'incréméntation à chaque erreur rencontrée. Enfin, le SIZE est déterminé par le n°x du dernier registre testé avec erreur.

□ Jean-Christophe Krust



# Un trait, un point, c'est tout



## TI-57 et ZX 81

Inventé au siècle dernier par Samuel Morse pour transmettre des textes à distance, le morse est le plus souvent utilisé sous sa forme sonore. Voici deux courts programmes qui peuvent vous donner le goût d'apprendre cet alphabet.

■ Les moyens modernes de communication, beaucoup plus élaborés que le morse, n'ont pas réussi à le faire disparaître, et une bonne pratique de ce code reste indispensable dans un certain nombre de professions. Les deux programmes proposés font appel à la même méthode de codage et permettent un affichage en morse visuel. S'ils sont

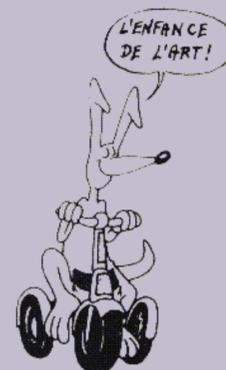
différents l'un de l'autre, c'est avant tout parce que la TI-57 ne permet que l'affichage de chiffres, alors que le ZX 81 peut afficher des points, des traits et des barres, symbolisant les divers signaux. De plus, la TI-57 traduira (de façon d'ailleurs très « visuelle ») des mots de cinq lettres au plus, alors que le ZX 81 accepte des phrases beaucoup plus longues, il est vrai sans ponctuation ni chiffres, mais ce sont des améliorations que vous pourrez apporter aisément.

\_\_\_\_\_Quel code\_\_\_\_\_  
\_\_\_\_\_utiliser ?\_\_\_\_\_

En morse, chaque lettre est codée à l'aide de 2 symboles (le point . et le trait -) sur au plus quatre positions. Ainsi, le A est codé .- (point et trait) sur deux positions. Le B est codé -... (un trait et trois points, quatre positions), etc. Pour la commodité de la transposition, nous

conviendrons de coder chaque lettre sur quatre positions, en ajoutant des zéros si nécessaire. De plus, nous remplacerons chaque point par un 1 et chaque trait par un 2.

Partant de ces principes, nous obtiendrons pour chaque caractère un code en base 3 (en effet, seuls les chiffres 0, 1 et 2 sont utilisés) que l'on inversera d'ailleurs pour faciliter la restitution ultérieure du message. Le décodage étant effectué par des divisions successives, ce sont les symboles de droite qui apparaîtront en premier. Ainsi A (en



**Morse**  
Programme pour ZX 81  
Auteur Jacques Deconchat  
Copyright l'Ordinateur de poche et l'auteur.

```

10 LET A$="0741501401401740047
92343000526527116130222672560774
4"
20 PRINT "VOTRE TEXTE"
30 INPUT A$
40 FOR I=1 TO LEN A$
50 LET P$=A$(I)
60 IF P$<"A" OR P$>"Z" THEN GO
TO 150
70 LET C=CODE P$-37

```

```

80 LET C=VAL A$(2+C-1 TO 2+C)
90 LET J=C-3*INT (C/3)
100 LET C=INT (C/3)
110 PRINT "." AND J=1; "-" AND J
=2;
120 IF C<>0 THEN GOTO 80
130 PRINT "/";
140 NEXT I
150 PRINT "///";
160 NEXT I

```

**Exemple**

VOTRE TEXTE  
.....  
.....  
.....  
.....  
.....

**Morse**  
Programme pour TI-57  
Auteur Jacques Deconchat  
Copyright l'Ordinateur de poche et l'auteur

00	15		CLR
01	36	2nd	Pause
02	33	4	RCL 4
03	66	2nd	x=t
04	81		R/S
05	45		÷
06	33	5	RCL 5
07	65		-
08	49	2nd	Int
09	32	4	STO 4
10	85		=
11	55		×
12	33	5	RCL 5
13	85		=
14	86	8	Lbl 8
15	32	1	STO 1
16	33	1	RCL 1
17	45		÷
18	03		3
19	85		=
20	49	2nd	Int
21	32	2	STO 2
22	55		×
23	03		3
24	65		-
25	33	1	RCL 1
26	75		+
27	01		1
28	85		=
29	84		+/-
30	-76	2nd	INV x>=t
31	71		RST
32	-66	2nd	INV x=t
33	03		3
34	34	0	SUM 0
35	33	6	RCL 6
36	86	5	Lbl 5
37	56	2nd	Dsz
38	51	5	GTO 5
39	33	2	RCL 2
40	51	8	GTO 8
41	86	0	Lbl 0
42	55		×
43	33	0	RCL 0
44	-18	2nd	INV Log
45	85		=
46	34	4	SUM 4
47	02		2
48	34	0	SUM 0
49	81		R/S

morse . - ) sera écrit 1200 (codage en base 3 sur 4 positions), puis inversé 0021, soit en pratique 21. De même, B (en morse - ...) sera écrit 2111, puis inversé 1112. Et ainsi de suite. Chacun des nombres ainsi obtenu sera transposé en base 10 pour être compacté, toute lettre étant alors codée sur deux chiffres seulement.

On utilise à cet effet une grille de passage de base 3 en base 10.

3 <sup>3</sup>	3 <sup>2</sup>	3 <sup>1</sup>	3 <sup>0</sup>
27	9	3	1
0	0	2	1

On calcule donc  $0 \times 27 + 0 \times 9 + 2 \times 3 + 1 \times 1 = 7$ . La lettre A sera introduite en mémoire sous la forme 7 (ou 07 pour le ZX 81). Même système pour la lettre B :

3 <sup>3</sup>	3 <sup>2</sup>	3 <sup>1</sup>	3 <sup>0</sup>
27	9	3	1
1	1	1	2

Cette fois-ci, le résultat est 41, car  $1 \times 27 + 1 \times 9 + 1 \times 3 + 2 = 41$ . La lettre B sera donc introduite sous la forme 41, etc.

Pour vous épargner la suite des calculs, un peu fastidieuse à vrai dire, je vous livre ci-dessous la liste des codes ainsi obtenus.

Sur le ZX 81, les codes sont déjà inscrits dans le programme. Il suffit donc d'entrer — en clair — le texte à coder. Sur la même machine, chaque lettre est séparée de la suivante par une barre oblique (/), et les espaces entre deux mots sont représentés par quatre barres (/////).

A (..)	7	B (....)	41	C (....)	50	D (....)	14
E (.)	1	F (....)	49	G (....)	17	H (....)	40
I (..)	4	J(....)	79	K (....)	23	L (....)	43
M (..)	8	N (..)	5	O (....)	26	P (....)	52
Q(....)	71	R (....)	16	S (....)	13	T (..)	2
U (....)	22	V (....)	67	W (....)	25	X (....)	68
Y(....)	77	Z (....)	44				

**Utilisation des programmes**

Sur la TI-57, on initialise avec 100 STO 5, 11111111 STO 6 et 2nd fix 0. Pour obtenir une traduction en morse du mot "alpha", on fait donc 7 SBR 0, 43 SBR 0, 52 SBR 0, 40 SBR 0, et 7 SBR 0. On tape ensuite 0 STO 0, RST, et on lance l'exécution par R/S. Il ne reste plus alors qu'à observer l'affichage. Une série de 1 apparaissant fugitivement représente un point. La même série de 1, affichée plus longuement représente un trait. Pour la transcription d'un autre mot, SOS par exemple, il suffit de faire 13 SBR 0, 26 SBR 0, 13 SBR 0, puis 0 STO 0, RST et R/S.

Sur le ZX 81, on tape RUN et, à l'apparition du message "VOTRE TEXTE", on introduit le message à transcrire et l'on presse la touche NEWLINE pour l'affichage du résultat.

Avec la TI-57, en revanche, on devra se référer à la liste des codes pour entrer les mots. Ainsi, le mot "alpha" sera introduit en tapant successivement 7, 43, 52, 40 et 7. La séparation visuelle des lettres apparaîtra à l'affichage sous la forme d'un 0 suivi d'une pause.

Peut-être ces petits programmes vous donneront-ils envie de vous mettre au morse sérieusement. Bonne idée, je crois, car l'apprentissage de ce code n'est pas aussi futile qu'il y paraît. De nos jours encore, beaucoup de messages en morse voyagent sur les ondes courtes d'un bout à l'autre de la planète...

□ Jacques Deconchat



# Quand le calendrier était sans-culotte

## Le calendrier révolutionnaire

■ Très schématiquement, et sans vouloir faire un cours d'histoire, voyons comment se présente le calendrier révolutionnaire.

Les hommes de la Révolution française cherchèrent à remplacer l'année grégorienne, alors en usage, par un nouveau mode de datation, une autre notion du temps, plus scientifique et plus rationnelle. Ils voulurent soumettre le temps lui-même au système de la numération décimale, qu'ils venaient d'appliquer avec succès aux distances, aux volumes, aux poids et aux surfaces (1).

Pourtant, c'est seulement le 5 octobre 1793, plus de quatre ans après la prise de la Bastille, qu'un décret de la Convention fixa au 22 septembre 1792 le point de départ de l'an I de l'« ère des Français ». Cette date était celle de la proclamation de la République : le hasard veut qu'elle coïncide également avec l'équinoxe d'automne.

L'année républicaine était réglée sur l'année solaire : elle comptait 365 jours et se trouvait divisée en 12 mois égaux de 30 jours, se répartissant eux-mêmes en 3 décades de 10 jours.

A la fin de chaque année, par conséquent entre le 30 fructidor et le 1<sup>er</sup> vendémiaire, on plaça 5 « jours complémentaires » (appelés aussi « sans-culottides »), n'appartenant à aucun mois.

Enfin, tous les quatre ans (on disait chaque « sextile ») était intercalé un sixième jour complémentaire, dit « jour de la Révolution » ; dans la brève période qui nous occupe, ce sixième jour apparut trois fois : à la fin de l'an III, de l'an VII et de l'an XI.

Pour éviter une chronologie trop abstraite des événements (tel jour aurait été daté : « 3<sup>e</sup> jour de la 3<sup>e</sup> décade du 3<sup>e</sup> mois de la 3<sup>e</sup> année de la République » !), le poète Fabre d'Eglantine imagina et fit adopter (par décret du 4 frimaire an II (24 novembre 1793)) des noms concrets et plus faciles à retenir : *vendémiaire*, *brumaire* et *frimaire* pour les mois d'automne, *nivôse*, *pluviôse* et *ventôse* pour les mois d'hiver, *germinal*, *floréal* et *prairial* pour les mois de printemps, enfin *messidor*, *thermidor* et *fructidor* pour les mois d'été.

Mais ce système avait de très graves inconvénients (2) et la force des traditions fit que l'ancien calendrier grégorien continua en fait à être utilisé dans la vie courante. Si bien qu'à la fin de la période révolutionnaire, le calendrier républicain était tombé à peu près en désuétude dans la pratique et n'était plus guère utilisé que dans les documents officiels, notamment dans les actes d'état civil.

Napoléon préféra le supprimer : un sénatusconsulte du 22 fructidor an XIII (9 septembre 1805) abolit officiellement le calendrier républicain à dater du 11 nivôse an XIV (1<sup>er</sup> janvier 1806).

(1) Cette volonté d'appliquer au temps le système décimal allait très loin : le jour était divisé en 10 heures, l'heure en 100 minutes et la minute en 100 secondes décimales !

(2) Ainsi, chaque année, après que des astronomes aient calculé le jour de l'équinoxe d'automne, un décret devait fixer le début de l'année, qui pouvait donc commencer le 22, le 23 ou même le 24 septembre. A cette incertitude s'ajoutait l'embarras des calculateurs lorsque l'équinoxe devait se produire vers minuit : selon leur décision, l'année pouvait avoir un jour de plus ou de moins !

□ JS

Il y a presque deux siècles fut adopté un calendrier républicain. Il devait rester en vigueur pendant moins de treize ans, mais il continue à compliquer les recherches historiques ou généalogiques.

Voici un programme pour PC-1211/PC-1 qui règle le problème.

■ Pendant près de treize ans, de 1793 à 1806, les Français ont compté les jours sur un calendrier tout à fait original et très différent du calendrier grégorien que nous connaissons aujourd'hui.

Treize ans, c'est peu. Mais il se trouve que cette période est une des plus mouvementées de notre histoire. Tous ceux qui, professionnellement ou pour occuper leurs loisirs, sont amenés à travailler sur les documents de cette période, toujours datés selon le calendrier républicain, sont confrontés à la gênante nécessité d'établir la concordance avec notre calendrier grégorien. En particulier, ceux qui cherchent à retrouver des traces de leurs aïeux abordent tôt ou tard la période révolutionnaire.

———Poisson de———  
———germinal !———

Même les lycéens trouvent dans leurs livres d'histoire des dates qu'ils ne peuvent pas reconnaître. Les uns et les autres se heurtent alors à d'étranges « 13 pluviôse an VIII » ou « 2<sup>e</sup> jour complémentaire de l'an XIII » qui, traduits dans l'actuel calendrier, leur sembleraient bien plus clairs.

Il existe naturellement des tableaux de concordance entre calendrier républicain et calendrier grégorien, mais ces tableaux sont le



plus souvent partiels (ils ne fournissent la concordance que pour certains jours donnés, par exemple le premier de chaque mois : au chercheur de calculer lui-même le jour qui l'intéresse) et sont isolés dans des ouvrages spécialisés. Il n'est pas toujours possible d'avoir ces livres à sa disposition, et de toute façon il est nécessaire de les ouvrir, de les feuilleter pour y retrouver les tableaux en question, d'y repérer les colonnes voulues, et enfin d'effectuer les soustractions ou les additions nécessaires : calculs qui n'ont rien de savant mais dont le caractère répétitif devient vite fastidieux, ralentit les recherches et engendre souvent des erreurs.

Le programme proposé ici demande à peu près 4 secondes pour fournir la date du calendrier

#### Les années républicaines

An I	: 22sept 1792 au 16sept 1793
An II	: 22sept 1793 au 16sept 1794
An III	: 22sept 1794 au 16sept 1795
An IV	: 23sept 1795 au 16sept 1796
An V	: 22sept 1796 au 16sept 1797
An VI	: 22sept 1797 au 16sept 1798
An VII	: 22sept 1798 au 16sept 1799
An VIII	: 23sept 1799 au 17sept 1800
An IX	: 23sept 1800 au 17sept 1801
An X	: 23sept 1801 au 17sept 1802
An XI	: 23sept 1802 au 17sept 1803
An XII	: 24sept 1803 au 17sept 1804
An XIII	: 23sept 1804 au 17sept 1805
An XIV	: 23sept 1805 au 31déc 1805

Les jours compris entre deux années successives (par exemple, du 17 au 21 septembre 1793, entre la fin de l'An I et le début de l'An II) sont les « jours complémentaires ».

#### Les mois républicains

vendémiaire	: 22-23 sept au 21-23 oct
brumaire	: 22-24 oct au 20-22 nov
frimaire	: 21-23 nov au 20-22 déc
nivôse	: 21-23 déc au 19-21 janv
pluviôse	: 20-22 janv au 18-20 fév
ventôse	: 19-21 fév au 20-21 mars
germinal	: 21-22 mars au 19-20 avril
floréal	: 20-21 avril au 19-20 mai
prairial	: 20-21 mai au 18-19 juin
messidor	: 19-20 juin au 18-19 juill
thermidor	: 19-20 juill au 17-18 août
fructidor	: 18-19 août au 16-17 sept

#### Calendrier républicain

Programme pour PC-1211/1212 et PC-1  
Auteur Joël Surcouf  
Copyright l'Ordinateur de poche et l'auteur

```

25: "Z"INPUT "JD      GOTO 150
    UR, MOIS, ANNE   85: IF R#="FRUC"
    E ?";X,R#,Z      LET Y=12:
30: IF R#="VEND"      GOTO 150
    LET Y=1:GOTO     90: IF R#="JOUR"
    110              THEN 300
35: IF R#="BRUM"     110: IF (Z<4)+(Z
    LET Y=2:GOTO     <8)LET M=0:N
    110              =0:GOTO 130
40: IF R#="FRIM"     115: IF (Z=4)+(Z>
    LET Y=3:GOTO     7)*(Z<12)
    110              LET N=-1:N=1
45: IF R#="NIVÔ"     :GOTO 130
    LET Y=4:GOTO     120: M=-2:N=2
    110              130: IF Y<3LET L=
50: IF R#="PLUV"     Y+8+M:GOTO 2
    LET Y=5:GOTO     00
    110              135: IF (Y>2)*(Y<
55: IF R#="VENT"     6)LET L=Y+7+
    LET Y=6:GOTO     M:GOTO 200
    150              150: IF Z<8LET M=
60: IF R#="GERM"     0:N=0
    LET Y=7:GOTO     155: IF (Z=4)*(Y=
    150              6)LET N=1:
65: IF R#="FLOR"     GOTO 170
    LET Y=8:GOTO     160: IF Z>=8LET M
    150              =-1:N=1
70: IF R#="PRAI"     165: IF (Z=12)*(Y
    LET Y=9:GOTO     =6)LET N=2
    150              170: IF (Y=6)+(Y=
75: IF R#="MESS"     7)LET L=Y+4+
    LET Y=10:        M:GOTO 200
    GOTO 150         175: IF (Y=8)+(Y=
80: IF R#="THER"     9)LET L=Y+3+
    LET Y=11:        M:GOTO 200

```

```

180: IF Y>9LET L=
    Y+2+M
200: IF X<=LLET B
    =Y+8:GOTO 23
    0
205: B=Y+9
230: IF B>12LET B
    =B-12
250: IF X>LLET A=
    X-L:GOTO 310
260: IF Y<3LET A=
    X+21+N:GOTO
    310
265: IF (Y=3)+(Y=
    7)LET A=X+20
    +N:GOTO 310
270: IF Y=4LET A=
    X+20+N:C=Z+1
    791:GOTO 320
275: IF (Y=5)+(Y=
    8)+(Y=9)LET
    A=X+19+N:
    GOTO 310
280: IF (Y=6)+(Y=
    9)*(Y<12)LET
    A=X+18+N:
    GOTO 310
285: A=X+17+N:
    GOTO 310
300: Y=13: B=9: IF
    Z<=7LET A=X+
    16:GOTO 310
305: A=X+17
310: IF Y>3LET C=
    Z+1792:GOTO
    320
315: C=Z+1791
320: BEEP .1:PRINT
    A;B;C:GOTO "
    Z"

```

actuel correspondant à une date républicaine.

En mode DEF, après avoir lancé le programme par SHFT Z, on répond à la question « Jour, Mois, Année ? » en introduisant le jour suivi de ENTER, les 4 premières lettres du mois (ENTER) et l'année républicaine (en chiffres arabes, bien sûr : taper « 8 » et non « VIII » !). Par exemple, 11 (ENTER) THER (ENTER) 8 (ENTER). Quelques secondes plus tard, la réponse s'affiche : « 30.7.1800 », le 30 juillet 1800.

Pour désigner les mois républicains, on doit utiliser les quatre pre-

#### Variables utilisées

- A : jour grégorien
- B : mois grégorien
- C : année grégorienne
- L : valeur du jour républicain tombant le dernier jour du mois grégorien
- M : valeur à retrancher à la variable L
- N : valeur à ajouter à la variable A
- R# : nom du mois républicain
- X : jour républicain
- Y : mois républicain
- Z : année républicaine

mières lettres : VEND pour vendémiaire, VENT pour ventôse, etc. Pour rentrer un jour complémentaire (qui n'appartient à aucun mois), il faut taper JOUR.

Les lignes 110 à 180 constituent la partie principale du programme ; elles sont destinées à calculer les variables L, M et N, d'une part pour les cinq premiers mois de l'année (lignes 110 à 135), d'autre part pour les sept mois suivants (lignes 150 à 180).

En effet, l'établissement de la concordance des deux calendriers est fondée sur la valeur du jour révolutionnaire tombant le dernier jour du mois grégorien. C'est la variable L qui reçoit cette valeur (lignes 130-135 et 170-180). Par exemple, le 31 octobre 1793 correspondant à un 10 brumaire,  $L=10$  ; le 30 juin 1799 correspondant à un 12 messidor,  $L=12$ .

Suivant les mois et les années, cette valeur est comprise entre 7 et 14 ; pour un même mois républicain, elle ne varie pas de plus de 2 jours. La variable M, qui prend les

**Petite bibliographie**

**Manuel de diplomatique**, A. Giry, Hachette, Paris, 1894.

**Le Maître-calendrier**, Daniel Langlois-Berthelot, Imprimeries réunies, Chambéry, 1975.

**Le calendrier**, Paul Couderc, collection « Que sais-je ? », n° 203, Presses universitaires de France, Paris.

**Le calendrier républicain. Tables complètes de concordances du calendrier grégorien**, Fierens Gevaert, Editions administratives, U.G.A., Heule (Belgique), 1965.

valeurs 0, -1 ou -2, tient compte de cette variation et permet d'obtenir la valeur exacte de L pour le mois et l'année considérés. Cette variable M est calculée aux lignes 110-120 ou (selon que le mois est ou non supérieur à 5) aux lignes 150-160.

Quant à la variable N, qui reçoit sa valeur (0, 1 ou 2 selon le cas) aux lignes 110-120 ou 150-165, elle n'est utilisée qu'aux lignes 260-285 pour calculer la variable A : elle permet en effet de corriger le chiffre qu'il

faut ajouter au jour républicain pour obtenir le jour grégorien.

Une fois les valeurs de L, M et N déterminées, le mois B est calculé (lignes 200-230), puis le jour A (lignes 250-285).

## — Les giboulées — — de ventôse —

Après avoir traité à part le cas des jours complémentaires (lignes 300 et 305), il ne reste plus qu'à préciser l'année C (lignes 310 et 315) et à afficher la date de l'année grégorienne.

On pourra raffiner en ajoutant au programme des tests de détection d'erreurs destinés à s'assurer que la date d'entrée est au moins plausible. Il peut arriver en effet que le chercheur ait mal déchiffré la date d'un document ou — c'était assez fréquent — que l'auteur du document lui-même, peu familiarisé avec le nouveau calendrier, ait commis une erreur de datation.

On fera donc vérifier cette date par l'ordinateur : le jour ne devra pas être supérieur à 30, le jour complémentaire à 6, le mois à 13, l'année à 14 ; pour l'an XIV, si le mois est égal à 4, le jour ne devra pas être supérieur à 10 (c'est en effet le lendemain que le calendrier grégorien a été remis en vigueur). En ce qui concerne le sixième jour complémentaire (donc si  $Y=13$ ), on pourra également contrôler que l'année est bien 3, 7 ou 11.

Tous ces tests pourront aboutir à une ligne ainsi libellée : BEEP 2 : PRINT « DATE D'ENTRÉE ERROREE » : GOTO « Z ».

Le faible encombrement du PC-1211 et son autonomie permettent de l'emporter avec soi dans les mairies, les archives et les bibliothèques. Si telle est votre intention, n'oubliez pas alors de supprimer l'instruction BEEP de la ligne 320 : il ne faudrait pas, en effet, que le poquette ne trouble le silence studieux des salles de lecture...

□ Joël Surcouf



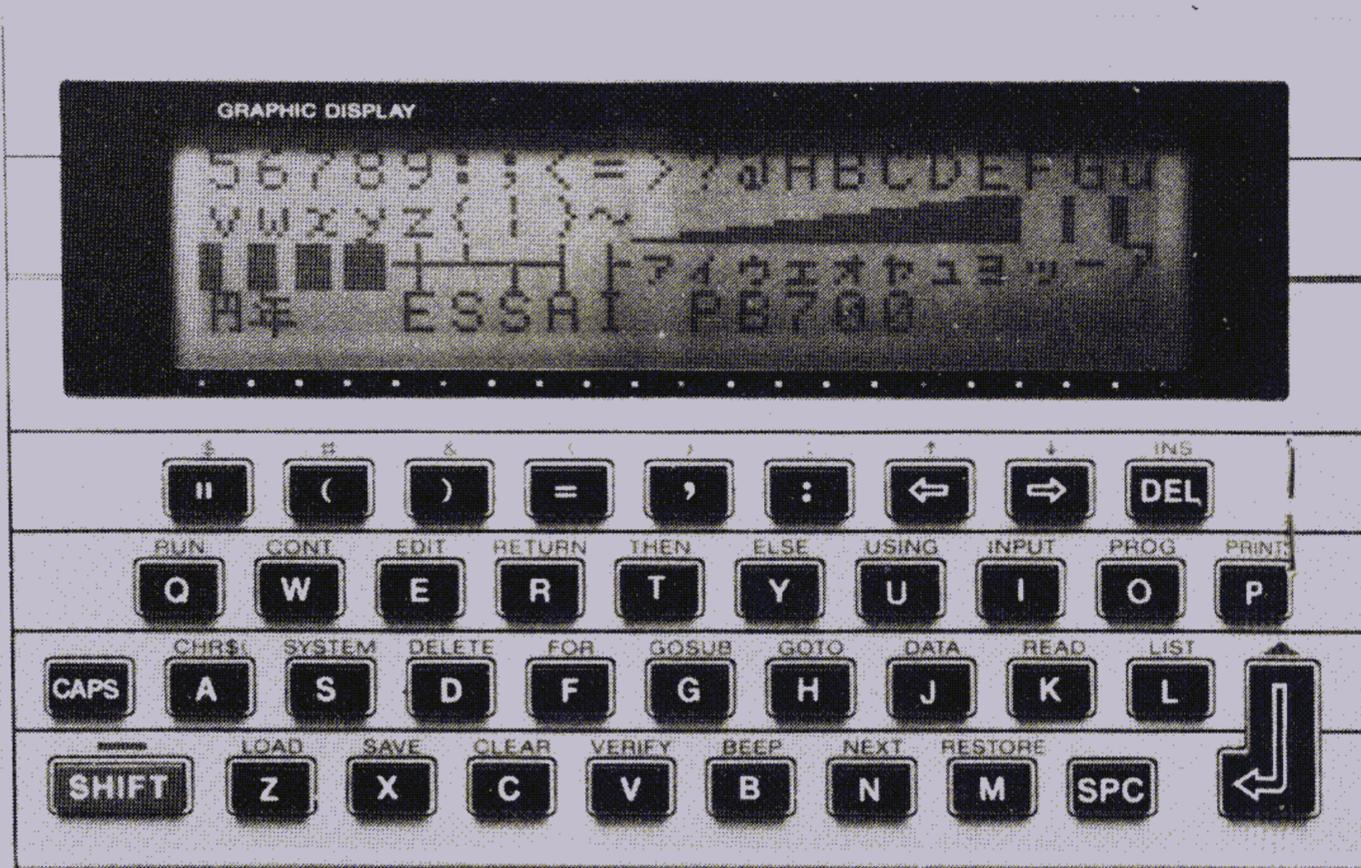
coup d'œil sur...

# Le PB-700 de Casio

Avec quatre lignes d'affichage, de nombreuses fonctions graphiques, avec une table traçante (en option), le PB-700 est un bon dessinateur.

Mais cela ne l'empêche pas de résoudre les problèmes classiques soumis aux ordinateurs de poche.

Le prix de vente de la version de base devrait se situer aux alentours de 1 700 FF ttc.



## Signe particulier : des graphismes de poche

■ Le marché des ordinateurs de poche continue à s'agrandir, et chaque nouveau venu se distingue des autres sur un point ou sur un autre. Sous un format réduit (19,8 × 8,7 × 2,4 cm), le PB-700 présente d'intéressantes possibilités graphiques. Ce poquette possède un écran à cristaux liquides affichant quatre lignes de vingt caractères. Mais cet écran est en réalité composé de 5 120 points formant une matrice ininterrompue de 32 × 160.

— Deux touches —  
— à ne pas confondre —

Le Basic de la machine permet d'adresser individuellement chacun de ces points, c'est-à-dire d'allumer ou d'éteindre à volonté l'un quelconque d'entre eux. Si ce n'est pas vraiment de la haute définition, cela

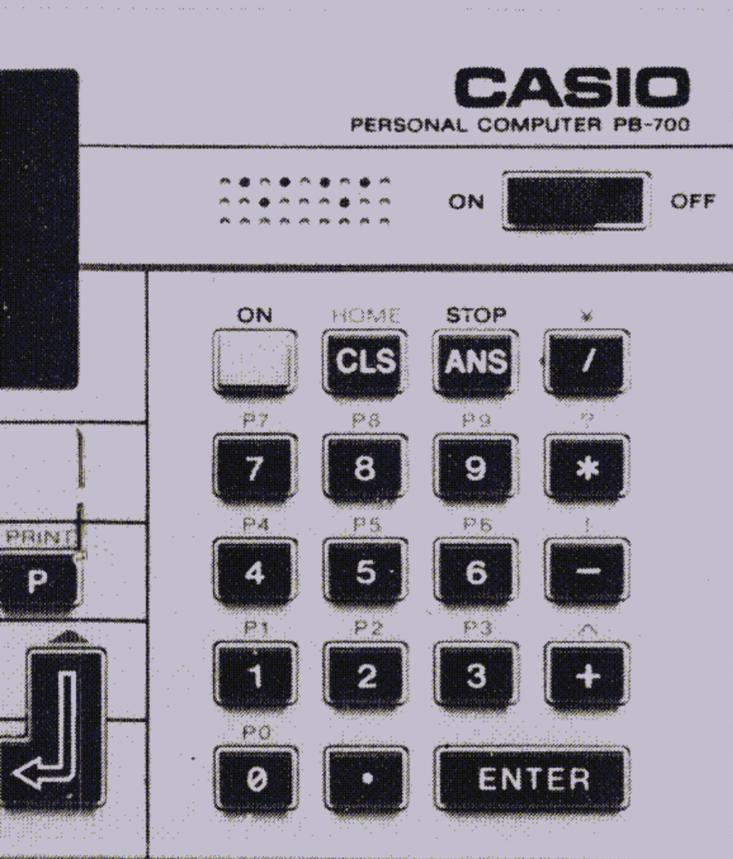
suffit pour obtenir des effets graphiques très variés.

Le clavier est séparé en deux parties : à gauche un secteur alphabétique « qwerty », et un pavé numérique à droite. Chacune de ces deux zones possède sa propre touche « fin de ligne » ; touche  $\downarrow$  du côté alphabétique et ENTER du côté numérique. On peut, à première vue s'en étonner. A la réflexion, c'est une solution qui paraît astucieuse, et ces deux touches ne font pas double emploi : la première valide tout ce qui concerne la programmation (exécution d'un ordre Basic, écriture, déroulement des programmes...), la seconde valide les calculs manuels. L'utilisateur évitera ainsi de créer de nouvelles lignes sans le vouloir, car il n'existe pas, sur le PB-700, de distinction entre mode *programme* et mode *calcul*.

Le clavier se révèle agréable à l'usage, même si l'on retrouve cer-

taines dispositions (courantes sur les matériels de table) qui ne se justifient pas vraiment sur une machine de format réduit. C'est ainsi que la touche SHIFT fonctionne exactement comme sur un « gros » ordinateur : elle doit être enfoncée en même temps que la touche dont elle modifie l'action. Sur le PB-700, SHIFT donne accès à vingt-six des instructions les plus usitées du Basic. Dans la majorité des cas, on retrouve ces instructions placées sur leur initiale (EDIT sur E, RETURN sur R, PRINT sur P, etc.). C'est excellent.

La touche SHIFT étant réservée à l'obtention des instructions Basic, il en fallait une autre pour passer de majuscules en minuscules. C'est le rôle de la touche CAPS, qui, elle aussi, n'est active que si elle est enfoncée conjointement avec une autre touche : pour obtenir un « n » minuscule, on doit appuyer sur CAPS et sur N (il est donc impossi-



**Le PB-700  
grandeur  
nature**

ble ici aussi de procéder avec un seul doigt). Cela rappelle évidemment le fonctionnement d'un clavier de machine à écrire standard, mais le clavier n'est pas assez large pour une véritable frappe à deux mains, et, de toutes les façons, les touches CAPS et SHIFT ne se retrouvent pas à main droite. La dactylographie qu'impose le PB-700 contraindra donc ses utilisateurs à changer d'habitudes... s'ils en avaient déjà, bien entendu !

Citons encore les quatre touches fléchées (↑, ↓, ← et →) grâce auxquelles on déplace le curseur sur la ligne en cours d'écriture, INS et DEL (insertion ou suppression de caractères), CLS qui efface l'écran en replaçant le curseur sur la première position de l'afficheur, et ANS (précieux héritage du FX-702 P) qui rappelle le résultat du dernier calcul effectué.

A gauche de la machine, un

potentiomètre variable règle le contraste de l'afficheur, et l'on trouve, sur la face arrière, le connecteur (15 broches) destiné au branchement avec l'imprimante/interface-cassette FA-10.

L'alimentation de l'unité centrale est assurée par quatre piles bâton de 1,5 V et une pile au lithium évitant la perte d'informations au cours du changement de piles. L'autonomie est d'une centaine d'heures d'utilisation continue (la consommation est alors d'un dixième de watt). Quant à la pile-pastille au lithium, en l'absence d'autre source d'énergie, elle suffit à elle seule pour conserver le contenu de la mémoire vive pendant dix mois dans la version de base de l'ordinateur, et deux mois et demi avec 16 Ko, c'est-à-dire trois modules d'extension.

— Le bip est là, —  
— bien là —

Dans sa version de base, le PB-700 offre en effet 4 Ko de mémoire vive dont, apparemment, 2,8 seulement sont utilisables pour la programmation. Mais il est possible d'étendre cette mémoire jusqu'à 16 Ko par adjonction de modules de 4 Ko, les OR-4 dont l'enfichage dans une trappe située au dos de l'appareil paraît fiable (prix de chaque module, environ 450 FF ttc.).

Nouveauté sur un ordinateur de poche Casio, le bip est là, et bien là : BEEP 0 ou 1 donne le choix entre deux tonalités assez bruyantes, il est vrai, mais un bip est fait pour s'entendre, non ?

Les calculs se font sur douze chiffres, dont dix sont affichés (10 + 2 pour l'exposant en notation scientifique). Cela étant dit, dans un tableau, les variables peuvent être déclarées en demi-précision : les calculs se font alors sur cinq chiffres

significatifs et la variable n'occupe que quatre octets (au lieu de huit pour un nombre de dix chiffres). A condition qu'ils ne reprennent pas les mots réservés du Basic tels que IF ou TO, les identificateurs des variables simples peuvent compter deux caractères.

On peut définir des tableaux à une ou deux dimensions : variables numériques, ou alphanumériques comprenant d'un à soixante-dix-neuf caractères. C'est l'instruction DIM qui déclare ces tableaux, et elle est complétée par ERASE qui les efface au besoin en tout ou partie.

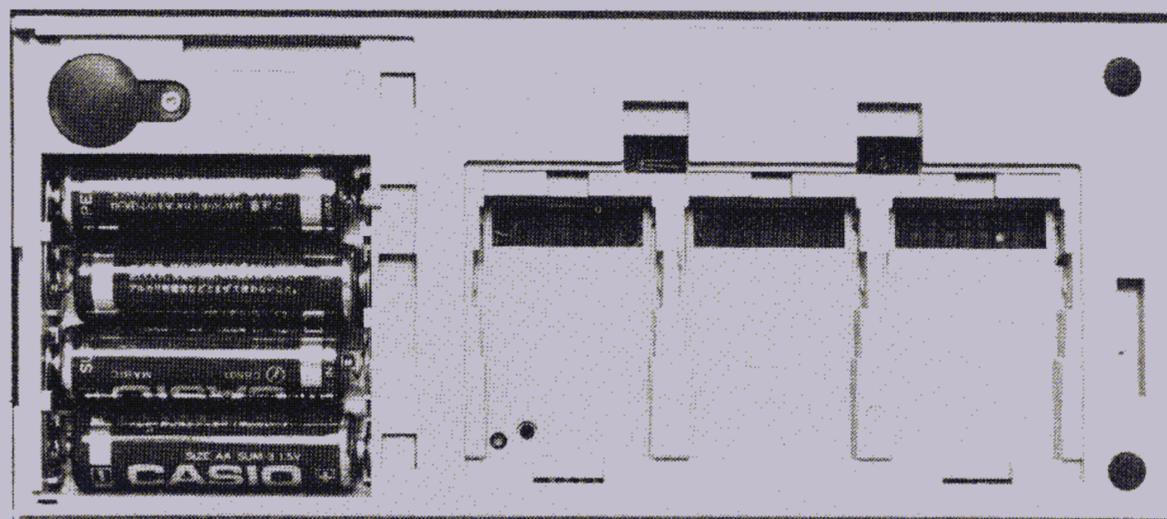
Comme sur pratiquement tout ordinateur de poche Basic, les calculs s'effectuent selon le système de la notation algébrique parenthésée, et l'on obtient l'affichage du résultat en pressant sur la touche ENTER.

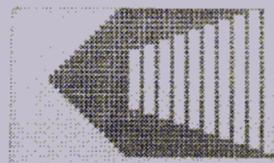
Les messages d'erreurs (il y en a vingt et un au total) sont affichés de façon assez parlante. Un exemple : « OV error » signale un résultat dépassant la capacité de l'ordinateur (*overflow error* en anglais) ; on s'y habituera beaucoup plus facilement qu'aux messages du type « error 12 ».

Au chapitre des mathématiques, si le PB-700 n'est pas préprogrammé pour les calculs de statistique, il comporte bon nombre de fonctions classiques : trigonométrie, logarithmes décimaux et naturels, exponentielles, valeur absolue, MOD (qui retourne le reste de la division euclidienne de deux nombres), RND (nombres pseudo-aléatoires compris entre 0 et 1), etc.

Une fonction SYSTEM indique la mémoire vive disponible, les zones

**Au dos de l'appareil, le compartiment des piles et celui des extensions de mémoire vive.**





de programme libres et le mode angulaire : 0 pour degré, 1 pour grade, 2 pour radian. Pourquoi ne pas avoir indiqué ces modes en toutes lettres ?

Si les opérateurs booléens AND, OR et NOT sont absents, les fonctions de traitement de chaînes en revanche n'ont pas été oubliées : RIGHT\$, LEFT\$, MID\$, LEN, INKEY\$, CHR\$, VAL...

### — Des graphismes — — point par point —

On retrouve, comme sur les précédents poquettes Casio, les dix zones de programme (P0 à P9) qui constituent un système pratique, GOTO PROG et GOSUB PROG effectuant les branchements d'une zone à une autre. Le Basic, inscrit en mémoire morte, est riche. C'est ainsi, par exemple, qu'IF (condition) et THEN (test vérifié) sont accompagnés d'ELSE (test non vérifié). Pour charger rapidement des données dans des variables ou pour en changer le contenu, on dispose de READ-DATA-RESTORE. Le PB-700 offre douze niveaux de sous-programmes et six niveaux de boucles FOR-NEXT. Existe également un mode trace (TRON/TROFF) grâce auquel on peut suivre l'exécution d'un programme pas à pas.

En ce qui concerne la gestion de l'écran et les graphismes, l'un des points forts de la nouvelle machine, relevons entre autres LOCATE, CHR\$, DRAW, POINT... Ainsi, LOCATE X,Y place le curseur sur la matrice de coordonnées X,Y (X variant entre 0 et 19 et Y entre 0 et 3). CHR\$ permet d'obtenir des symboles inaccessibles au clavier, caractères japonais, caractères graphiques... Avec DRAW et DRAWC, on éteint ou l'on allume l'un des 5 120 points de l'écran ou l'on trace des droites entre les points dont on précise les coordonnées. La fonction POINT, par ailleurs, teste si l'un de ces points est allumé (elle retourne la valeur 1) ou éteint (valeur 0). Le Basic du PB-700 permet ainsi de réaliser des dessins, des courbes, des graphiques sur les 5 120 points de l'écran : il y a de quoi faire !

### A gauche, quelques-uns des caractères disponibles; à droite, un exemple d'affichage graphique.

La commande PRINT USING permet de formater les sorties ; c'est ainsi par exemple que PRINT USING « ###.##### » donnera un résultat avec trois chiffres avant et cinq chiffres après le point décimal.

Une ligne FOR I = 1 TO 1000 : NEXT I est parcourue en 11 secondes (7 secondes sur le PB-100), ce qui est assez raisonnable, mais pourra tout de même se révéler un peu lent pour certaines des applications interactives que permet l'écran. Le langage-machine, ici, aurait été le bienvenu.

Les lignes de Basic, numérotées de 1 à 9999, s'introduisent dans la zone de programme choisie au moyen de la commande PROG : PROG 5 donne accès à la zone P5 de l'ordinateur. La commande LIST par ailleurs donne la liste des programmes ou des variables en tout ou partie, selon la demande, comme LLIST qui est réservée à l'imprimante. Détail important, ces deux commandes sont entièrement programmables.

Les modifications de programme se font au moyen d'EDIT suivi du numéro de la ligne à corriger. Ce dispositif est efficace : on peut, par exemple, passer d'une ligne à la suivante ou remonter dans la liste. Pour quitter ce mode, il suffit de presser simplement sur BRK. Enfin, on peut très commodément supprimer une ou plusieurs lignes, avec la commande DELETE.

Bien utile aussi dans ce domaine, l'existence d'une instruction REM permettant de documenter les programmes. L'ensemble de ce système, combiné avec le mode « trace », rend aisées la vérification, la modification et la correction des programmes.

On regrettera parfois que le constructeur n'ait pas prévu un témoin d'exécution des programmes. En effet, si le programme ne se mani-

feste pas à l'écran, il n'est pas facile de savoir qu'il tourne. Font défaut aussi les instructions PAUSE et WAIT permettant de régler la durée d'un affichage, mais on les remplace facilement au moyen d'une boucle de temporisation.

L'exécution d'un programme peut être suspendue par STOP et reprise par CONT, ou totalement arrêtée avec BRK, dont la fonction essentielle est de sortir du mode EDIT ou LIST.

Chaque zone de programme peut être protégée par un mot de passe, ce qui empêchera l'utilisateur de modifier involontairement ses programmes. Cela est bien vu. En effet, comme il n'y a pas de mode programme, il suffit de faire des calculs au clavier et d'enfoncer par mégarde la touche ↓ au lieu d'ENTER pour valider sans le vouloir une nouvelle ligne de programme. Une ligne peut contenir jusqu'à 75 caractères, soit exactement le contenu de l'écran (80 caractères) moins la place réservée à l'affichage du numéro de ligne.

### — Magnétophone — — intégré ou — — extérieur, au choix —

D'un point de vue esthétique, le PB-700 est une réussite et il donne une bonne impression de solidité. Il en va de même pour son périphérique, la FA-10 qui combine, tout en un, une imprimante et une interface-cassette (prix, environ 2 400 FF ttc). Cet accessoire, dans lequel vient s'encastrier l'unité centrale est au format A4 (21 × 29,7 cm) ; il est épais de 4,5 cm et pèse 1,340 kg. Dans la mallette de transport avec laquelle il est fourni, on trouve également un chargeur pour batteries, un câble de raccordement pour magnétophone extérieur, un jeu de stylos, des rouleaux de papier et un manuel d'utilisation. La consommation de la FA-10 peut atteindre 15 W et l'autonomie de l'imprimante est de 1 500 lignes (20 caractères/ligne) si les accus sont complètement rechargés.

Un logement a été prévu pour recevoir un magnétophone à micro-cassettes, le CM-1, connectable à

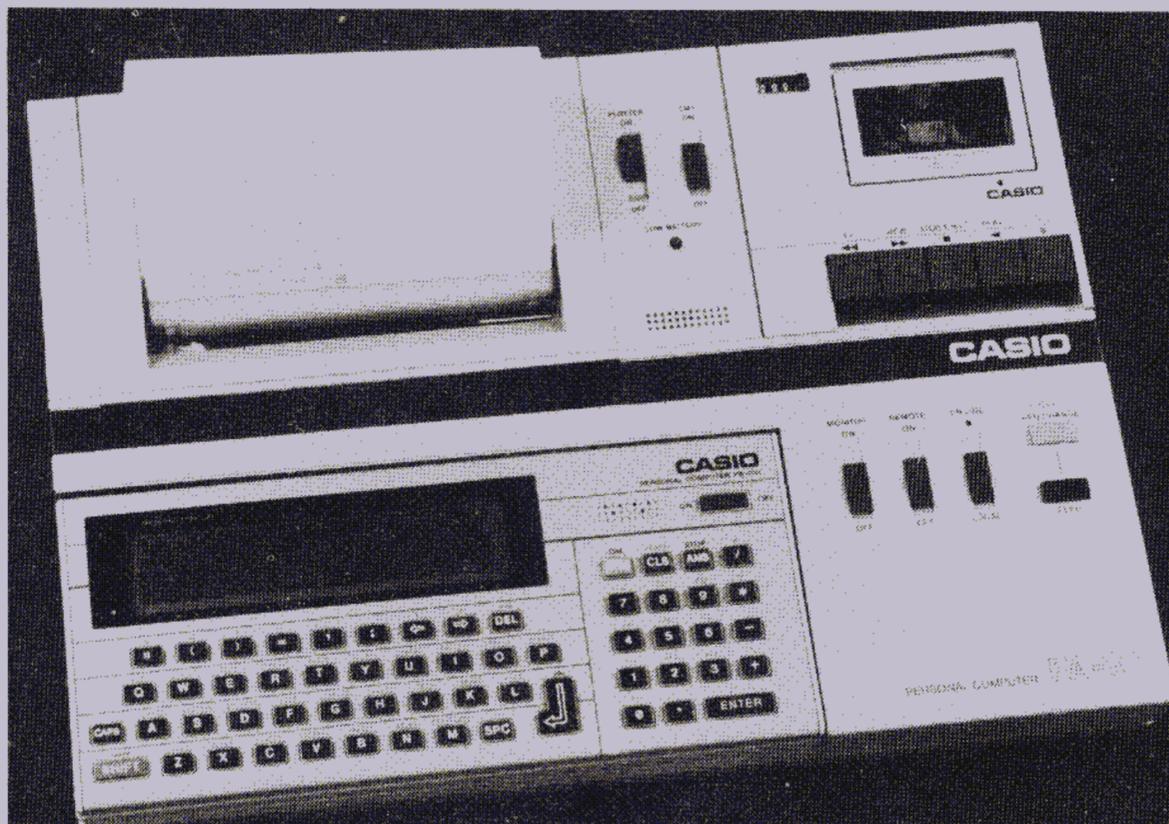
demeure. Si l'on fait le choix de ce périphérique (environ 850 FF ttc), on dispose ainsi d'un ensemble monobloc. Cela dit, il est aussi possible de connecter un magnétophone extérieur. Un petit haut-parleur intégré, que l'on peut éventuellement débrancher, permet un contrôle auditif des transferts de signaux lors des opérations d'enregistrement ou de lecture. Il n'opère toutefois qu'avec le CM-1.

Chose peu courante sur du matériel de poche, deux PB-700 peuvent échanger directement programmes ou données par l'intermédiaire de deux FA-10 reliées par câble. La transmission est assez rapide et les transferts sur les micro-cassettes du CM-1 sont fiables. Les instructions de sauvegarde et de chargement des informations sont les classiques SAVE, LOAD, VERIFY, CHAIN pour les programmes, PUT et GET pour les données.

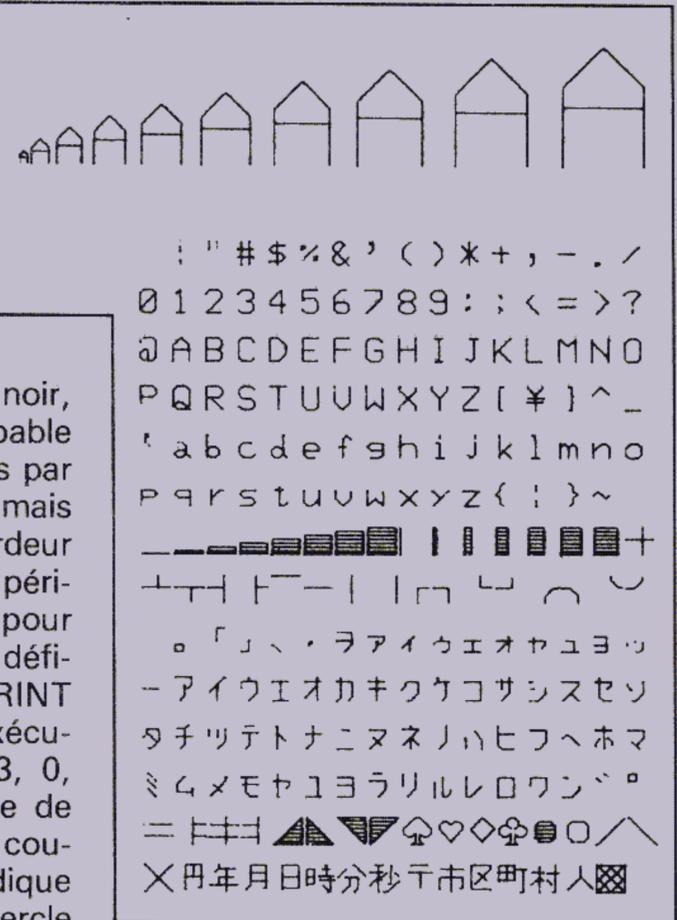
— La table traçante —  
— est un peu lourde —  
— ... à programmer —

L'imprimante fonctionne selon le même principe que celles qui équipent les PC-1500, TPC-8300, etc., mais la largeur du papier est de 114 mm. C'est une table traçante dotée d'un barillet mobile contenant

**Le PB-700 et le magnétophone à microcassettes encastrés dans la FA-10 : un aspect très sérieux.**



**Le jeu de caractères de la table traçante. En haut, la lettre A a été imprimée dans les dix tailles possibles.**



quatre petites pointes-billes (noir, bleu, vert et rouge) et capable d'imprimer jusqu'à 80 caractères par ligne. Tout cela est excellent, mais on regrettera une certaine lourdeur dans la programmation de ce périphérique. A titre d'exemple, pour tracer un cercle, il faut d'abord définir le mode graphique par LPRINT CHR\$(28); CHR\$(37), puis exécuter la commande LPRINT « C3, 0, 360 » : on trace alors un cercle de rayon 3 à partir de la position courante des stylos (0), et 360 indique l'amplitude de l'angle, ici un cercle complet. Une fonction CIRCLE x,y,z aurait été bien plus simple pour l'utilisateur...

Cela étant, l'éventail des fonctions graphiques est très large : positionnement de l'origine, tracé de droites brisées (ou non), de carrés, de rectangles, de systèmes d'axes, de pointillés, etc. On peut également choisir la taille des caractères à imprimer, la couleur, l'orientation (écriture horizontale ou verticale), la distance séparant les signes ou les lignes. Le principe de la programmation reste le même que pour le tracé d'un cercle : on change seulement le nom de la commande à l'intérieur des paramètres de LPRINT. Et ces commandes sont au nombre de 23, ce qui offre beaucoup de possibili-

tés. Ainsi, à la place du C, on utilise D pour tracer des droites, G pour hachurer l'intérieur d'un rectangle, L pour déterminer un type de pointillé...

— Un manuel —  
— très bien conçu —

Un auto-contrôle (obtenu en plaçant le commutateur PLOTTER sur la position ON et en appuyant sur FEED) permet de vérifier le bon état de la FA-10. Ce périphérique est donc un outil puissant mais assez complexe à piloter. Il faudra le pratiquer assez longtemps pour que, l'habitude aidant, il paraisse simple d'emploi !

La documentation fournie avec le PB-700 comprend un bon, un très bon manuel de référence (330 pages) qui décrit en détail les différentes instructions et comporte les tables et les index utiles. Également fourni avec la machine, un aide-mémoire récapitulant l'essentiel. Pour cet essai, nous n'avons disposé que de la version anglaise de ces documents. Espérons que la traduction vaudra l'original. Car le manuel est complet, agréable à consulter et fort bien conçu.

Le PB-700, en version de base, devrait être vendu à un prix voisin de 1 700 FF ttc.

Olivier Arbey  
Anne-Sophie Dreyfus  
□ Jean-Charles Lemasson



# L'adressage indirect vaut le détour

Adressage indirect, gestion indirecte des registres... Voilà un jargon incompréhensible. Essayons de clarifier les choses en prenant des exemples sur quelques calculatrices programmables en langage-machine spécialisé.

■ Les opérations indirectes font en général partie des dernières astuces de programmation que l'utilisateur apprend en consultant le manuel de sa machine. Les explications qu'il y trouve sont parfois un peu courtes. L'emploi de ces fonctions, par ailleurs, n'est pas évident ; de là, trop souvent, l'incompréhension partielle ou totale des opérations indirectes qui se trouvent alors délaissées. Et

c'est dommage car ces opérations offrent des possibilités très intéressantes. Quand on les a découvertes, on ne peut généralement plus s'en passer.

Nous nous intéresserons à quelques ordinateurs de poche programmables en LMS, langage-machine spécialisé (1). Sur ce type de matériel, le principe de base régissant les opérations indirectes est le suivant : vous utilisez une mémoire (ou un registre) non pas pour exploiter l'information qu'elle contient, mais pour savoir où se trouve l'information recherchée. Tout se passe en somme comme si vous disiez à une personne « va demander à Pierre où se trouve Jean ». Si Pierre sait où est Jean, la personne que vous interrogez saura aussitôt où se trouve Jean, alors qu'elle aurait pu

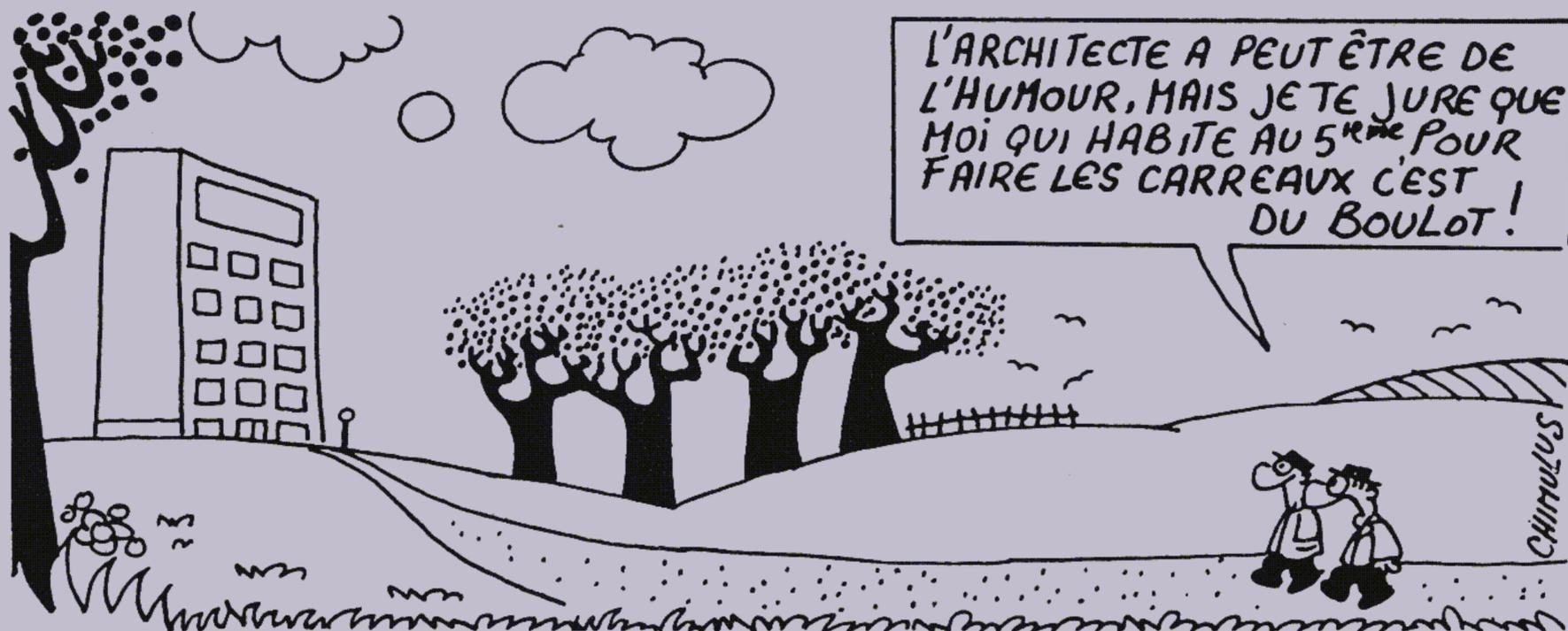
perdre des heures à le chercher elle-même. Dans certains cas, il est même impossible, sans l'aide de Pierre, de retrouver Jean...

Dans le cas des machines LMS, Pierre sera toujours un registre et Jean le contenu de ce registre, c'est-à-dire un nombre qui désignera soit un numéro de pas, soit le numéro d'un autre registre, ou, d'une façon plus générale, l'argument numérique d'une fonction (DSE, ISG, FIX, SF, CF, F?, STO, RCL, adresse de GTO, GSB, etc...).

Bien que le principe soit le même, la méthode mise en œuvre par les deux principaux constructeurs, HP et TI, diffère quelque peu.

Commençons par la série des HP 11, 15, 16, 34, 67 et 97 (nous laisserons de côté le cas particulier de la 41). Sur ces HP, Pierre est appelé « I » et Jean « i ». Seulement « i » peut changer de nom (il s'appellera aussi bien Paul, René, etc.), mais jamais « I ». Il faudra donc toujours passer par Pierre, registre I, qui est

(1) Le sujet concerne aussi, bien entendu, la famille sans cesse croissante des poquettes Basic, mais ces derniers ne seront pas traités dans le cadre de cet article, pas plus que la HP-41 dont le système est original.



supposé tout savoir sur les autres et constitue en ce sens la véritable commère de la machine. Sur ces HP, le registre I est toujours un registre à part ; il n'est jamais touché par les modifications de la partition mémoire.

### — Une bonne façon — — de gagner des pas —

La séquence STO I donne à Pierre l'adresse où trouver Jean ;  $x \Rightarrow I$  effectue la même opération, mais rappelle à l'affichage l'ancien contenu de I. Bien qu'il soit le seul registre d'adressage indirect existant, I peut, par ailleurs, être utilisé comme les autres mémoires de la machine. Voici un court programme de démonstration :

01	LBL A	10	LBL B
02	9	11	9
03	STO I	12	STO I
04	LBL 0	13	LBL 1
05	R/S	14	RCL (i)
06	STO (i)	15	PSE
07	DSE	16	DSE
08	GTO 0	17	GTO 1
09	R/S	18	R/S

L'utilisateur doit introduire neuf nombres de son choix (STO (i) du pas 06), nombres qui sont ensuite réaffichés au moyen de l'instruction pause (RCL (i) et PSE des pas 14 et 15). Les nombres sont stockés dans les registres 1 à 9, et l'appui sur les touches A et B permet respectivement de les modifier ou de les réafficher.

Sans utiliser d'opérations indirectes, on aurait programmé, par exemple :

01	LBL A	20	LBL B
02	STO 9	21	RCL 9
03	R/S	22	PSE
04	STO 8	23	RCL 8
05	R/S	24	PSE
06	STO 7	25	RCL 7
07	R/S	26	PSE
08	STO 6	27	RCL 6
09	R/S	28	PSE
10	STO 5	29	RCL 5
11	R/S	30	PSE
12	STO 4	31	RCL 4
13	R/S	32	PSE
14	STO 3	33	RCL 3
15	R/S	34	PSE
16	STO 2	35	RCL 2
17	R/S	36	PSE
18	STO 1	37	RCL 1
19	R/S	38	PSE

Pas besoin d'être expert pour se rendre compte que cette dernière méthode consomme beaucoup de pas. Le premier programme, sans

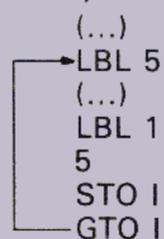
doute plus difficile à déchiffrer, effectue le même travail en 18 pas seulement : 20 pas d'économie donc, et cela grâce à l'adressage indirect. Le fonctionnement en est simple : sous l'étiquette A du premier programme (LBL A), après avoir placé 9 dans le registre I, la calculatrice s'arrête au pas 05 dans l'attente de la première donnée, laquelle sera stockée dans le registre 9 grâce à l'instruction STO (i).

Si nous reprenons notre image, nous dirons que Pierre, autrement dit le registre I, connaît l'adresse de Jean : 9. Avec l'instruction DSE du pas suivant, le pointeur retourne à l'étiquette 0 ; I vaut alors 8 et la seconde donnée est donc stockée (STO (i)) dans le registre 8, etc. Ici, DSE agit implicitement sur I, mais sur certaines machines, il faudra préciser DSE I.

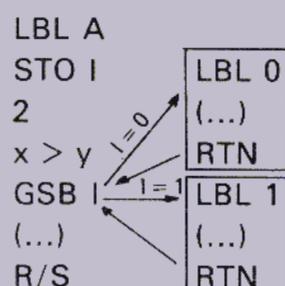
Sous l'étiquette B (pas 10 à 18), nous retrouvons le même principe, à cette différence près que Pierre (le registre I) va rechercher (RCL (i)) les données stockées dans les mémoires 9 à 1. Si vous avez saisi ce principe, vous avez pratiquement tout compris des opérations indirectes.

### — Branchements — — et tests —

Le deuxième grand domaine d'application de ces opérations concerne les branchements. Le registre I contient alors le numéro de l'étiquette ou du pas vers lequel le pointeur doit se rendre. Ainsi, quand I contient 5, GTO I ou GSB I conduit à l'étiquette 5 (LBL 5). Exemple :



Un autre exemple, un peu plus compliqué, est schématisé ci-dessous.



Il permet de voir que les opérations indirectes peuvent aussi être employées après un test. Si l'utilisa-

teur introduit 0 ou 1 en A, il se produira un transfert indirect au sous-programme 0 ou 1, selon la valeur introduite, mais le GSB I ne sera pas exécuté si la valeur introduite en A est supérieure ou égale à 2. A remarquer que pour ces adressages, la machine utilise toujours la partie entière du contenu de I.

### — Pierre et — — sa famille —

Sur tous les matériels cités (exception faite de la HP-16), on peut combiner arithmétique directe en mémoire et adressage indirect, ce qui conduit à des formes du type STO + (i), STO - (i), STO x (i), STO ÷ (i) et STO + I, STO - I, etc. Par ailleurs, la fonction  $x \Rightarrow (i)$  effectue un échange indirect du contenu de l'affichage et du registre dont le numéro est stocké en I.

Par rapport aux HP-11, 16 et 34 qui se valent à peu près de ce point de vue, les HP-15, 67 et 97 offrent un léger « plus » ; on peut, entre autres, faire agir ISG et DSE sur des registres autres que I et même (HP-67), mettre en œuvre des opérations du type DSE (i) ou ISG (i) : incrémentation ou décrémentation indirecte d'un registre.

Sur ces machines, les opérations indirectes sont, dans l'ensemble, faciles à maîtriser. Mais l'existence d'un seul et unique registre I est une sérieuse limitation. Il est par exemple difficile, sans acrobaties, de gérer indirectement deux tableaux de variables à une dimension.

Sur les TI-58, 58 C et 59, Pierre n'est pas seul : sa famille comporte jusqu'à 60 membres (TI-58) et même 100 (TI-59). Une version, adaptée aux TI, du programme de stockage et de rappel des nombres (écrit plus haut sur HP) est donnée page suivante.

Comme nous pouvons le constater, c'est ici le registre 00 qui joue le rôle de Pierre. L'astérisque (c'est sous cette forme que l'imprimante des TI-58/59 liste, le plus souvent, la fonction Ind) indique le caractère indirect de l'opération à effectuer. On lance le programme en tapant 9 A et la mémoire 00 prend respectivement les valeurs 9, 8, 7, etc. jusqu'à 1, rangeant ainsi dans les registres 9, 8, 7... les valeurs introduites par l'utilisateur. En lançant le programme avec 15 A, on obtient le même résultat avec les registres 15 à 1. Les opérations

## L'adressage indirect vaut le détour

d'arithmétique directe en mémoire adressée indirectement sont également possibles à l'aide de SUM Ind xx, INV SUM ind xx, etc., ainsi que l'échange indirect du contenu d'un registre et de celui de l'affichage. Les GTO et SBR indirects sont disponibles sous la forme GTO Ind xx et

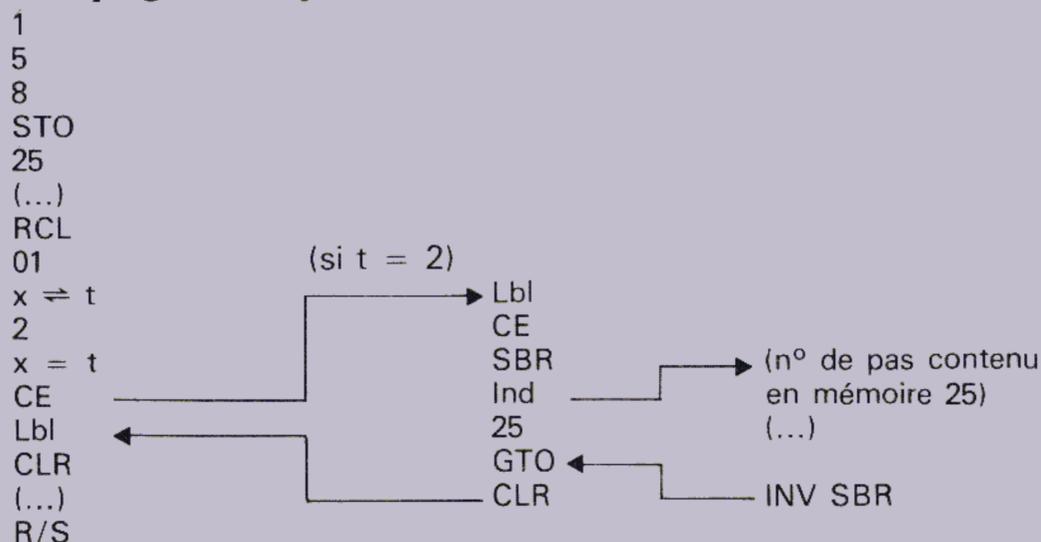
### Programme de démonstration sur TI-58/59

```

000 76 LBL
001 11 A
002 42 STO
003 00 00
004 76 LBL
005 24 CE
006 91 R/S
007 72 ST#
008 00 00
009 97 DSZ
010 00 00
011 24 CE
012 91 R/S
013 76 LBL
014 12 B
015 42 STO
016 00 00
017 76 LBL
018 25 CLR
019 73 RC#
020 00 00
021 66 PAU
022 97 DSZ
023 00 00
024 25 CLR
025 91 R/S

```

### L'une des façons, sur TI-58/59, d'appeler indirectement un sous-programme après un test.



SBR Ind xx ; mais le contenu du registre xx devra obligatoirement représenter un numéro de pas, ce qui oblige à utiliser ce type d'adressage de préférence sur la version définitive du programme. Le principe, ici aussi, est simple : si le registre 25, par exemple, contient la valeur 158, GTO Ind 25 conduit le pointeur au pas 158.

———— Il faut ————  
———— parfois jongler ————

S'il n'est pas possible d'adresser indirectement une étiquette, l'adressage numérique indirect, en revanche, offre de vastes possibilités au prix d'une mise au point délicate des programmes.

Le même principe se retrouve sur les tests. Cela dit, étant donné les spécificités de programmation de la TI-58/59, la seule opération indirecte

possible après un test conditionnel est un branchement (du type GTO) vers un numéro de pas. Autrement dit, pas d'appel de sous-programme (du type SBR). Si l'on veut tout de même y parvenir, il faut passer par un artifice tel que celui qui est illustré dans la liste ci-dessus. La méthode n'est pas vraiment souple, mais on arrive tout de même à ses fins. Dans l'exemple retenu, si le registre 01 contient bien la valeur 2, le pointeur ira se brancher sur l'étiquette CE : appel indirect du sous-programme par l'intermédiaire du registre 25 (SBR Ind 25), puis retour à l'étiquette CLR.

Bien sûr, il y aurait encore beaucoup à dire, car l'adressage indirect offre énormément de possibilités. Je souhaite seulement avoir donné à certains l'idée d'explorer cette voie trop souvent négligée.

□ Jean-Charles Lemasson



```

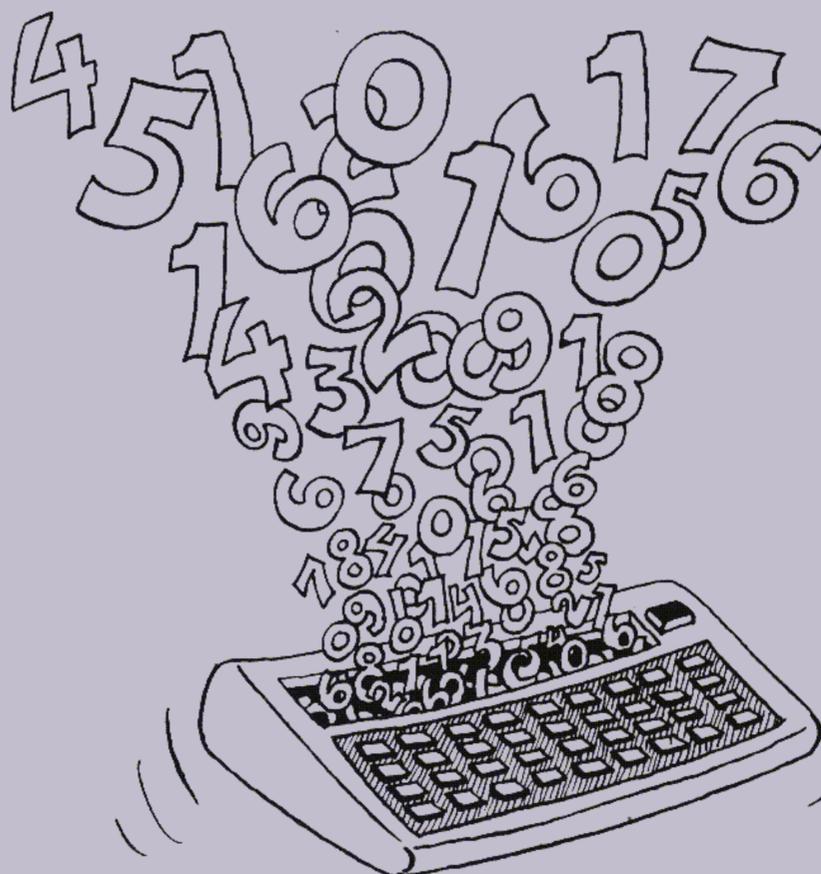
684pas
14variables
DEFM 14
10 D=0:G=0:L=0:K=0
12 FOR A=14 TO 38:AC
A)=INT (RAN#*9)+1:NE
XT A:B=14
15 FOR H=B TO B+4:PR
INT A(H):NEXT H
18 FOR A=0 TO 1000:P
RINT :NEXT A:B=B+5:I
F B>39 THEN 15
40 INPUT "QUI DEBUTE
",C$:IF C$="PB-100"
THEN 60
42 FOR A=14 TO 38:60
SUB 500:NEXT A:A=E
50 M=1:G=6+INT A(A):
A(A)=A(A)/10
51 FOR H=14 TO 34 ST
EP 5:IF A>H:IF A<H+4
THEN 54
52 M=M+1:NEXT H
54 I=(M-1)*5:N=A-I-1
3
55 D=1:PRINT "#COL:"
:M:" LIG:"M:"#":GOS
UB 558
60 INPUT "COL? LIGNE
",J$:IF J$="*" THEN
598
61 J=VAL(J$):INPUT F
:I=(F-1)*5
62 IF N*J:IF M*F:IF D*0:
PRINT "COUP IMPOSSIB
LE!+ ":PRINT :GOTO
60
65 B=I+J+13:K=0:A=B:
L=L+INT A(A):A(A)=A(
A)/10
71 M=5-J:H=1:GOSUB 5
47:H=-H:M=1-J:GOSUB
547
75 H=5:M=5*(5-F):GOS
UB 547:H=-H:M=-5*(F-
1):GOSUB 547:A=E
102 IF A(A)<1 THEN 5
98
105 GOTO 50
500 IF A(A)>K:K=A(A)
:E=A
509 RETURN
547 FOR A=B TO B+M S
TEP H:GOSUB 500:NEXT
A:RETURN
558 PRINT "JOUEUR:"
L:" /PB-100:"G:" "
:PRINT :RETURN
598 IF G>L:PRINT "PE
RDU!+ ":
600 IF G<L:PRINT "GA
GNE!+ ":
605 IF G=L:PRINT "EG
ALITE!+ ":
610 GOSUB 558:END

```



Avant de jouer, faire DEFM 14.

# Une course aux chiffres pour PB-100



Avec son module d'extension, le PB-100 organise une course aux chiffres. Si vous voulez le défier, vous devrez seulement vous munir d'une feuille de papier et d'un crayon. Voilà qui permet même de faire semblant de travailler en s'amusant !

■ Les ordinateurs sont généralement de bons compagnons de jeu : discrets, patients, jamais en colère. Ils répondent vite, mais en attendant leur tour. Attention toutefois : le PB-100 ne vous laisse que sept

minutes de réflexion ; passé ce délai, il s'éteint de lui-même. Il ne faut donc pas trop « lambiner » avant de répliquer au programme dans cette course aux chiffres.

Au début de chaque partie, le PB-100 remplit une grille de 25 cases (cinq sur cinq) avec des chiffres tirés au hasard entre 1 et 9 compris. Puis, chacun à son tour se pose sur une case et s'attribue le chiffre correspondant. Le but du jeu est d'obtenir le maximum de points en ajoutant au fur et à mesure les chiffres des cases jouées. Quant à la règle, elle est simple. La case jouée doit être choisie dans le prolongement (même ligne ou même colonne) de celle qui vient d'être occupée par l'adversaire. Ainsi, si le poquette a choisi la case de la colonne 2, ligne 4 (voir schéma ci-après), vous ne pouvez jouer que sur une des cases de la colonne 2 ou de la ligne 4. Sinon, vous aurez droit à un message indiquant : « COUP IMPOSSIBLE ! ». De plus, il est inutile de jouer plusieurs fois la même case : une case déjà jouée a une valeur nulle.

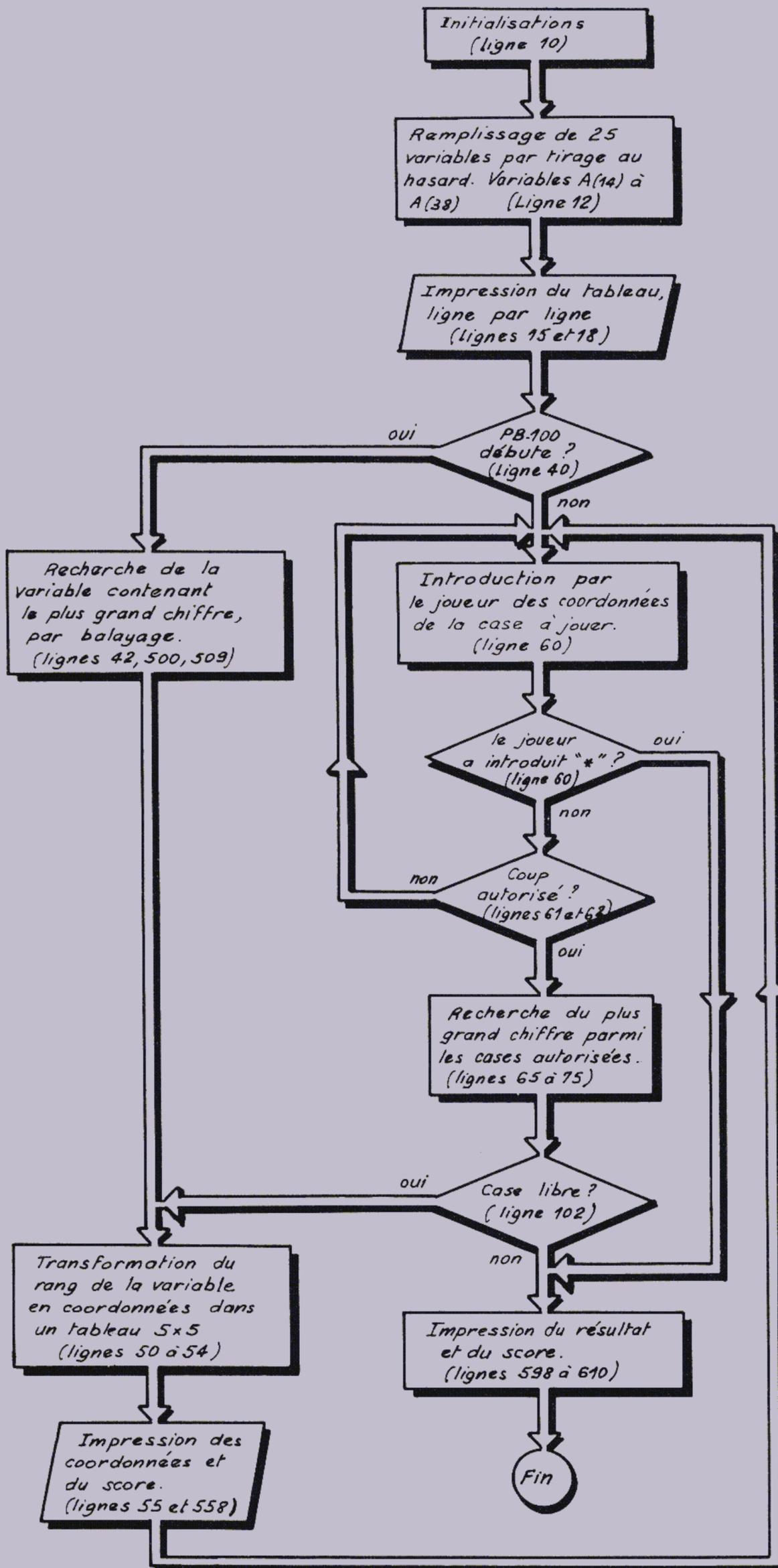
Avant d'entamer la partie, le programme tire au hasard les valeurs de chaque case et il les affiche ligne par

## Course aux chiffres

Programme pour PB-100

Auteur Raoul Lebastard

Copyright l'Ordinateur de poche et l'auteur



	1	2	3	4	5
1	5	4	9	8	7
2	3	2	1	4	6
3	8	7	2	2	4
4	5	X	3	8	1
5	6	1	5	7	2

Si le PB-100 a joué en COL 2, LIGNE 4 (case marquée X), vous devrez jouer une des cases grisées. Par exemple, COL 4 et LIGNE 4.

ligne (on les recopie alors sur une feuille). Il demande ensuite « qui débute ? ». Si c'est lui, il faut répondre « PB-100 », sans faute. Sinon, vous commencerez et la question qui s'affichera sera « COL ? LIGNE ? ». Vous devrez alors entrer dans l'ordre le chiffre indiquant la colonne (entre 1 et 5) et la ligne (entre 1 et 5 également).

Ainsi, pour la case de la colonne 4, ligne 4 : 4 EXE 4 EXE. A son tour, le PB-100 affichera sa position puis, quand vous aurez pressé sur la touche EXE, il donnera les scores. Pour ne rien perdre de la partie, notez bien chaque coup joué.

— Perdre d'abord —  
— pour —  
— gagner ensuite —

Si vous ne pouvez plus jouer (il n'y a plus de cases libres dans le prolongement de la dernière case jouée), faites \* EXE pour prendre connaissance du résultat définitif de la partie (gagné, perdu ou égalité) ainsi que des scores. Si c'est le programme qui ne peut plus jouer, il donne immédiatement le résultat.

N'allez pas trop vite : le coup qui rapporte le plus de points dans l'instant n'est pas toujours le meilleur. Si vous jouez une case qui vaut trois points, le PB-100 peut en trouver une autre qui lui donne neuf points. Il aurait mieux valu, alors, jouer une case à deux points qui ne laissait qu'un point au PB-100.

□ Raoul Lebastard

# Othello-Reversi damier 6 x 6

## le programme du champion (PC-1500)

Le cinquième tournoi international de programmes d'Othello-Reversi, organisé par notre confrère l'Ordinateur Individuel, s'est déroulé en septembre dernier à l'occasion du Sicob.

Le vainqueur dans la catégorie ordinateur de poche utilisait un PC-1500.

Il nous livre son programme.

■ Le programme de jeu d'Othello fonctionne sur un Sharp PC-1500 ou Tandy PC-2 équipé d'un module d'extension mémoire de 8 Ko exclusivement (pas de 16 Ko). Le programme occupe 5773 octets de mémoire vive. Ajoutez 2482 octets de tableaux et variables divers, un soupçon (489 octets) de langage-machine, et vous obtenez un cocktail de 8744 octets qui joue à Othello comme vous et moi, plutôt mieux même. Les possesseurs d'un module d'extension mémoire de 16 Koctets auront beaucoup de difficulté à reloger la routine en langage-machine, mais on ne peut tout avoir...

— D'abord le travail —  
— ensuite le jeu —

Pour programmer, exécuter tout d'abord en mode PRO un NEW&3AAE, puis introduire un à un les 489 codes qui composent la routine en langage-machine. Pour ce faire vous pouvez utiliser le petit programme Basic de la page 43 qui traite dans un premier temps, les codes, et les visualise aux fins de contrôle et correction dans un second temps : ENTER si le code est juste ; et POKE I, & bon code ENTER ENTER si le code était faux (bien faire précéder un code hexadécimal du signe &).

Ensuite, il ne vous reste plus qu'à programmer en Basic. Rappelez-vous seulement 5773 petits octets.

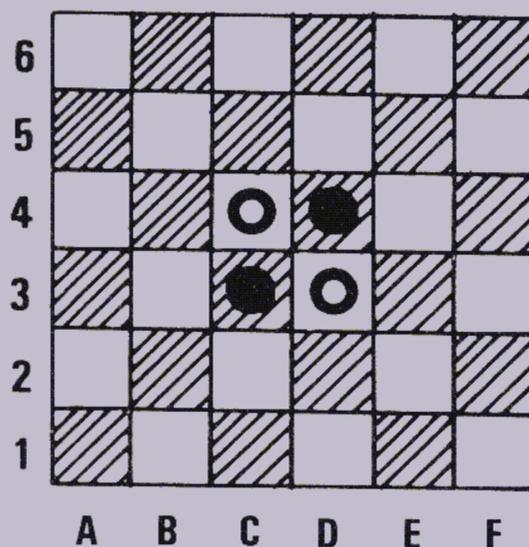
Une fois la fastidieuse épreuve de l'introduction du programme termi-

### Les règles du jeu

■ Le jeu d'Othello, adapté aux ordinateurs de poche, se dispute sur échiquier de 36 cases (6x6) à l'aide de pions qui ressemblent à ceux des dames, à cette différence près qu'ils sont blancs côté face et noirs côté pile.

Au début de chaque partie, quatre pions sont disposés au centre de l'échiquier. Comme on le voit sur la figure, deux présentent leur face noire (ils sont en C3 et D4), et deux leur face blanche (en C4 et D3). On tire au hasard lequel des deux adversaires utilisera la couleur noire. Par convention, c'est lui qui commence la partie en posant un pion, côté noir visible, sur une case inoccupée et en respectant les règles suivantes qui valent pour tout le déroulement de la partie :

- lorsqu'en posant un pion, noir par exemple, on enferme entre deux pions noirs un pion blanc ou un alignement ininterrompu de pions blancs, le ou les pions blancs qui se trouvent ainsi prisonniers entre deux noirs deviennent noirs : on les retourne simplement, et cela vaut aussi bien pour les lignes horizontales et verticales que pour les diagonales ;
- la pose d'un seul pion (blanc ou noir) peut conduire à retourner des



pions de l'adversaire dans plusieurs directions à la fois ;

- s'il ne trouve aucune case où le joueur peut, en posant un pion à sa couleur, retourner un pion adverse au moins, il passe son tour.

La partie s'achève lorsque toutes les cases de l'échiquier sont occupées ou lorsque les deux joueurs passent à tour de rôle. Dans les deux cas, le gagnant est celui qui a sur l'échiquier le plus grand nombre de pions à sa couleur, sauf partie nulle. □

# Pockethello

Programme pour PC-1500, PC-2 (avec 8 Ko)

Auteur Jean-Pierre Cayre

Copyright l'Ordinateur de poche et l'auteur.

```

20:REM ***** RETURN
   INITIALISATION 149: "*****
   *****
30:WAIT 0:DJM G(5 *****
   6),N$(56)*3,L( *****
   31),B(188):E=1 *****
   E6 *****
40:FOR I=0TO 7:G( *****
   I)=5:G(56-I)=5 *****
   :G(I*7)=5:NEXT *****
   I *****
50:FOR I=0TO 31: *****
   READ L(I):NEXT *****
   I *****
60:FOR I=8TO 28: *****
   READ N$(I):N$( *****
   56-I)=N$(I): *****
   NEXT I *****
65:FOR I=0TO 188: *****
   READ B(I):NEXT *****
   I *****
70:C1=0:C2=7:L1=6 *****
   4:L2=1:Y=4:CLS *****
   :INPUT "Je jou *****
   e avec les noi *****
   rs?";A$ *****
80:IF LEFT$(A$,1) *****
   )="O"LET G(24) *****
   =1:G(25)=-1:G( *****
   31)=1:G(32)=1: *****
   G(38)=1:C=38: *****
   GOTO 600 *****
90:IF LEFT$(A$,1) *****
   )="N"LET G(24) *****
   =-1:G(25)=1:G( *****
   31)=-1:G(32)=- *****
   1:G(38)=-1:C=2 *****
   3:GOTO 750 *****
95:IF LEFT$(A$,1) *****
   )="N"LET G(24) *****
   =-1:G(25)=1:G( *****
   32)=-1:GOTO 10 *****
   0 *****
99:"***** *****
   SOUS-PROGRAMME *****
   ***** *****
100:Q=V:R=0:IF G(Q) *****
   )=0RETURN *****
110:IF G(Q+Z)=G(Q) *****
   LET S=S+8*G(Q) *****
   :RETURN *****
120:R=R+1:Q=Q+Z:IF *****
   G(Q)=0LET S=S+ *****
   6*G(U)*R: *****
   RETURN *****
130:IF G(Q)<>G(U) *****
   IF G(Q)<>5THEN *****
   120 *****
140:S=S-8*G(U)*R: *****
   IF N=0NEXT K: 410:POKE 29081,232
   GOTO 440 :POKE 29080,
280:POKE 29078, *****
   PEEK 29079:O=1 *****
   :CALL 14859:
290:FOR L=0TO 31: *****
   NEXT L *****
   IF G(L(L))NEXT 420:POKE 14885,L(K
   L:GOTO 420 ) :IF B>ALET A=
300:N=0:O=-1:P=1: *****
   B:IF A>GLET K= *****
   POKE 14666,L(L) *****
   ):POKE 29075,2 430:POKE 29081,208
32:CALL 14661: *****
   IF N=0NEXT L: *****
   GOTO 420 *****
   PEEK 29078:O=- *****
   1:CALL 14859:
310:POKE 29079, *****
   PEEK 29075:S=0 440:POKE 14885,L(J
   :IF RIGHT$(N$ *****
   (L(J)),1)<>". *****
   THEN 350 *****
320:Z=7:IF L(L)<13 *****
   OR L(L)>43LET *****
   Z=1 *****
330:IF G(L(L)-Z)<> *****
   G(L(L)+Z)IF G( *****
   L(L)-Z)=1OR G( *****
   L(L)+Z)=1LET S *****
   =S+4*ABS VAL N *****
   $(L(L)) *****
340:IF G(L(L)-Z)=1 *****
   IF G(L(L)+Z)=1 *****
   LET S=S-3*ABS *****
   VAL N$(L(L)) *****
350:V=8:Z=1:GOSUB *****
   100:Z=7:GOSUB *****
   100:V=13:GOSUB *****
   100:Z=-1:GOSUB *****
   100:V=48:GOSUB *****
   100:Z=-7:GOSUB *****
   100:V=43 *****
360:GOSUB 100:Z=1: *****
   GOSUB 100 *****
370:IF G(8)LET S=S *****
   +10*G(16) *****
375:IF G(13)LET S= *****
   S+10*G(19) *****
380:IF G(43)LET S= *****
   S+10*G(37) *****
385:IF G(48)LET S= *****
   S+10*G(40) *****
390:CALL 14906:M= *****
   PEEK 29082-5: *****
   IF M>127LET M= *****
   M-256 *****
395:S=S+PEEK 29083 *****
   -PEEK 29084:IF *****
   Y>31LET S=0 *****
400:F=INT (S*EXP ( *****
   -Y*.042)+M-M* *****
   EXP (-Y*.042)+ *****
   1E3) *****
408:POKE 14885,L(L *****
   ):IF F<BLET B= *****
   F:IF B<ALET L= *****
   31 *****
150:POKE 29081,184
   :POKE 29080,
   PEEK 29077:O=1
   :CALL 14859:
   NEXT J
460:IF G>HLET H=G+
   L(I)/100
470:POKE 14885,L(I
   ):POKE 29081,1
60:POKE 29080,
   PEEK 29076:O=-
   1:CALL 14859:
   NEXT I
500:IF D=0CLS :
   PRINT "Je pass
   e";GOTO 150
510:C=(H-INT H)*10
   0:O=1:P=-1:
   POKE 14666,C:
   POKE 29075,160
   :CALL 14661
513:T=INT (C/C2)+C
   1:U=L1+((C1-T)
   *C2+C)/L2
516:IF ABS C2=1LET
   U=INT (ABS (C/
   L2))*SGN (C/L2
   )+L1:T=C1+((L1
   -U)*L2+C)/C2
520:C$=CHR$ U+STR$
   T:Y=Y+1:CLS :
   PRINT "Je joue
   en ";C$;GOTO
   150
599:"*****
   BIBLIOTHEQUE
   D OUVERTURE
   *****"
600:GOSUB 705:IF C
   =37THEN 200
610:IF C=39THEN 66
   0
620:C=18:GOSUB 700
   :IF C=12LET W=
   0:X=1:C=22:
   GOTO 690

```

```

630: IF C=26LET W=2      71: L2=-7: GOTO      1050: W=136: X=140:      , "0", "-3.", "
      : X=8: C=34: GOTO 780      C=27: GOTO 13      -9", "1", "1",
690      690      00      "9", "-3.", "
640: IF C=39LET W=9      760: IF C$="D2"LET      1100: C=45: GOSUB 7      0"
      : X=15: C=46:      C1=8: C2=-7: L2= 00: IF C=16      2030: DATA "3.", "1
      GOTO 690      -1: GOTO 780      LET W=141: X=      ", "0", "0", "1
645: IF C=40LET W=1      765: IF C$="E3"LET      144: C=26:      ", "3.", "0"
      6: X=19: C=22:      C2=1: L2=7: GOTO  GOTO 1300      2040: DATA 3029, 39
      GOTO 690      780      1110: IF C=22LET W      26, 1122, 1222
650: W=20: X=24: C=22      770: GOTO 750      =145: X=148: C      , 2720, 3722, 3
      : GOTO 690      780: GOSUB 700: C=C-  =33: GOTO 130      946, 4022, 452
660: C=26: GOSUB 700      0, 900, 1000, 110 0      1120: IF C=30LET W      2, 1222, 1917,
      : IF C=18LET W= 800: C=45: GOSUB 700      =149: X=152: C      2634, 3322, 40
      25: X=32: C=10:      : IF C=16LET W=  =22: GOTO 130      22
      GOTO 690      45: X=50: C=17:      0      2050: DATA 4522, 47
670: IF C=23LET W=3      GOTO 1300      1130: IF C=37LET W      48, 1126, 1617
      3: X=39: C=22:      810: IF C=18LET W=5  =153: X=158: C      , 3039, 4445, 1
      GOTO 690      1: X=54: C=33:      =39: GOTO 130      508, 1711, 192
680: W=40: X=44: C=43      GOTO 1300      0      6, 2639, 3339,
690: GOSUB 700: FOR      820: IF C=26LET W=5  1140: W=159: X=163:      1112, 1617, 20
      J=WTO X: IF C=      5: X=61: C=17:      C=43: GOTO 13      46
      INT (B(J)/100)      GOTO 1300      00      2060: DATA 2322, 27
      LET W=J: J=X      830: IF C=30LET W=6  1200: C=45: GOSUB 7      , 4446, 1508, 1
695: NEXT J: H=B(W)/      2: X=68: C=39:      00: IF C=16      710, 1810, 193
      100-INT (B(W)/      GOTO 1300      LET W=164: X=      0, 3346, 3733,
      100): Y=10: GOTO 840: IF C=39LET W=6      168: C=13:      4530, 1744, 18
      510      9: X=73: C=46:      GOTO 1300      44
700: O=1: P=-1: POKE      GOTO 1300      1210: IF C=22LET W      2070: DATA 2033, 23
      14666, C: POKE 2 850: W=74: X=79: C=43      =169: X=173: C      22, 3629, 1108
      9075, 160: CALL      : GOTO 1300      =13: GOTO 130      , 1808, 2629, 3
      14661      900: C=39: GOSUB 700 0      1220: IF C=30LET W      008, 3908, 444
705: T=INT (C/C2)+C      : IF C=22LET W= 1220: IF C=30LET W      3, 3029, 3711,
      1: U=L1+((C1-T)      80: X=85: C=8:      =174: X=179: C      3930, 4111, 92
      *C2+C)/L2      GOTO 1300      =26: GOTO 130      7, 1127
707: IF ABS C2=1LET      910: IF C=26LET W=8  1230: IF C=37LET W      2080: DATA 1629, 30
      U=INT (ABS (C/      6: X=91: C=27:      =180: X=183: C      27, 3727, 3927
      L2))*SGN (C/L2  GOTO 1300      =26: GOTO 130      , 4443, 1729, 1
      )+L1: T=C1+((L1  920: IF C=30LET W=9  =26: GOTO 130      829, 1929, 262
      -U)*L2+C)/C2      2: X=98: C=8:      0      9, 3327, 4629,
710: C$=CHR$ U+STR$      GOTO 1300      1240: W=184: X=188:      4729, 1710, 19
      T: CLS : PRINT "      930: W=99: X=104: C=3  C=43      30
      Je joue en "; C      3: GOTO 1300      1300: GOSUB 700:      2090: DATA 2617, 33
      $;      1000: C=11: GOSUB 7  FOR J=WTO X:      27, 4748, 1622
720: BEEP 4: INPUT "      00: IF C=10      IF C=INT (B(      , 1722, 1822, 2
      A toi: "; C$: C=      LET W=105: X=      J)/100)LET W      633, 3022, 392
      (VAL RIGHT$ (C      110: C=39:      =J: J=X      2, 910, 1015, 1
      $, 1)-C1)*C2+(      GOTO 1300      1310: NEXT J: H=B(W  715, 2615, 404
      ASC LEFT$ (C$,      1010: IF C=15LET W  )/100-INT (B      8, 4745
      1)-L1)*L2: IF C      =111: X=116: C      (W)/100): Y=1      2100: DATA 1711, 18
      <8OR C>48THEN      =45: GOTO 130      1: GOTO 510      08, 2013, 3045
      720      0      2000: DATA 8, 13, 43      , 4048, 4633, 9
730: N=0: O=-1: P=1:      1020: IF C=18LET W  , 48, 10, 11, 22      15, 1715, 1811
      POKE 14666, C:      =117: X=122: C      , 29, 27, 34, 45      , 2615, 3315, 4
      POKE 29075, 160  =45: GOTO 130      , 46, 17, 18, 23      048, 4745, 154
      : CALL 14661: IF      0      , 30, 26, 33, 38      8, 1826
      N=0THEN 720      1030: IF C=19LET W  , 39      2110: DATA 2248, 26
740: RETURN      =123: X=128: C      2010: DATA 9, 12, 15      27, 2748, 3048
750: CLS : BEEP 4:      =26: GOTO 130      , 20, 36, 41, 44      , 1216, 2245, 2
      INPUT "A TOI: "      0      , 47, 16, 19, 37      627, 2945, 334
      ; C$: IF C$="C5" 1040: IF C=22LET W      =40      5, 4033, 1639,
      THEN 780      =129: X=134: C      2020: DATA "12", "-      1829, 2629, 30
755: IF C$="B4"LET      =45: GOTO 130      3.", "3.", "3.      29
      C1=7: C2=-1: L1=      0      ", "-3.", "12" 2120: DATA 3946, 44

```

# Othello-Reversi

## Le programme du champion

43, 933, 1633,  
2233, 3022, 37  
39, 4443, 1013  
, 1513, 1613, 1  
813, 2013, 331  
3, 1633

2130: DATA 1833, 26  
33, 3039, 3946  
, 4443, 1039, 1  
529, 1645, 184  
5, 1945, 2013,  
3008, 3408, 39  
08

2140: DATA 4017, 37  
11, 3930, 4017  
, 4111, 1508, 2  
926, 3626, 444  
3, 1643, 2243,  
3011, 4048, 44  
43

2150: DATA 4643, 16  
26, 1710, 2233  
, 3026, 3726, 1  
826, 2226, 262  
0, 3022, 3739,  
1826, 2620, 30  
26

2160: DATA 3926, 44  
43, 1513, 1608  
, 1713, 1813, 3  
313, 4613, 151  
3, 1713, 3313,  
3913, 1613, 17  
13

2170: DATA 2213, 30  
36, 3713

Sur les PC-1500 de pre-  
mière génération, des lignes  
telles que la ligne 99 :  
"\*\*\*SOUS-PROGRAMME  
\*\*\*", c'est-à-dire où se  
trouvent *seules* program-  
mées des étiquettes, provo-  
quent une erreur : ERROR  
21 IN 99.

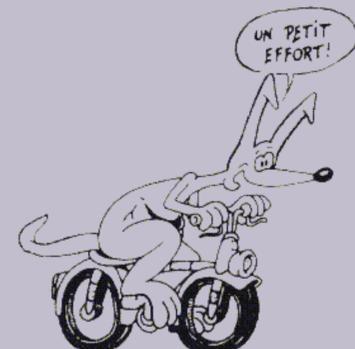
Si votre PC-1500 (ou  
PC-2) n'accepte pas ce type  
de lignes, vous pouvez y  
remédier de deux manières :

- supprimer purement et simplement la ligne fautive,
- ajouter, comme aux lignes 10 et 20 l'instruction REM (si vous supprimez deux signes \*, vous conserverez une longueur égale, en octets, du programme).

### Routine LM pour Pockethello

*X0	A5	71	92	AE	200	38	EF	28	EF	400	48	05	B7	2D
**4	71	90	A5	71	204	39	0B	D8	BE	404	8B	29	25	B7
**8	91	28	EF	71	208	38	C5	EF	38	408	00	8B	05	EF
*12	90	C0	83	02	212	D2	08	EF	38	412	71	9A	FE	54
*16	FD	62	A5	71	216	EF	08	EF	39	416	EF	71	9A	01
*20	90	2A	A5	7D	220	0B	F8	BE	38	420	05	B7	33	89
*24	7A	27	8B	01	224	C5	EF	38	D2	424	05	5F	03	BA
*28	9A	FD	28	46	228	08	EF	38	EF	428	3A	9E	B7	30
*32	A5	7D	79	07	232	08	EF	39	0B	432	8B	27	44	05
*36	8B	01	9A	EF	236	F8	BE	38	C5	436	B7	32	89	02
*40	71	90	C0	83	240	B5	F8	AE	38	440	5F	0B	5F	01
*44	02	FD	62	A5	244	D2	AE	38	EF	444	BA	3A	9F	25
*48	71	90	2A	A5	248	B5	08	AE	39	448	B7	00	8B	05
*52	7D	7A	27	8B	252	0B	EF	38	D3	452	EF	71	9A	FE
*56	01	9A	FD	28	256	02	EF	38	F0	456	56	EF	71	9A
*60	46	A5	7D	79	260	02	EF	39	0C	460	01	54	44	F9
*64	07	9B	1C	EF	264	FE	EF	38	D6	464	05	B3	D0	AE
*68	71	90	40	81	268	02	EF	38	F3	468	3A	9C	5F	03
*72	02	FD	60	A5	272	02	EF	39	0F	472	46	44	44	44
*76	71	90	2A	A7	276	FE	BE	38	C5	476	B5	09	FD	EA
*80	71	92	89	06	280	EF	38	D2	D8	480	6E	C2	99	5C
*84	A4	A7	71	91	284	EF	38	EF	D8	484	6C	5F	99	60
*88	8B	1B	B5	10	288	EF	39	0B	28	488	9A			
*92	2E	FD	28	46	292	BE	38	C5	EF					
*96	A5	7D	71	0E	296	38	D2	F8	EF					
100	48	71	A5	71	300	38	EF	F8	EF					
104	93	0A	A4	41	304	39	0B	08	BE					
108	24	0E	EF	71	308	38	C5	EF	38					
112	93	02	BA	39	312	D2	F8	EF	38					
116	08	B5	10	AE	316	EF	F8	EF	39					
120	7D	6A	63	A5	320	0B	08	BE	38					
124	7D	71	2E	9A	324	C5	9A	48	71					
128	4A	5E	B5	3A	328	A5	71	99	0A					
132	5A	15	50	52	332	45	28	45	2A					
136	8B	08	B3	07	336	66	A5	7D	71					
140	81	01	40	52	340	2E	04	A7	71					
144	99	08	AE	71	344	98	99	0F	4A					
148	92	04	AE	71	348	5E	B5	3A	5A					
152	91	B5	08	AE	352	0F	50	52	8B					
156	38	D2	AE	38	356	08	B3	07	81					
160	EF	B5	F8	AE	360	01	40	52	99					
164	39	0B	B5	81	364	08	2A	04	28					
168	AE	38	D3	AE	368	B5	00	63	2E					
172	38	F0	B5	83	372	9A	B5	00	AE					
176	AE	39	0C	B5	376	71	9A	AE	71					
180	60	AE	38	D6	380	9B	AE	71	9C					
184	AE	38	F3	B5	384	68	5E	6A	7A					
188	62	AE	39	0F	388	48	5D	4A	9E					
192	BE	38	C5	EF	392	58	71	5A	9B					
196	38	D2	28	EF	396	67	B7	00	8B					

La 1<sup>re</sup> colonne de chiffres (avec des \*) indique le n° d'ordre dans la routine du 1<sup>er</sup> code de la ligne (de 4 en 4). Bien sûr ce n° ne doit pas être introduit dans la mémoire. Les colonnes de chiffres 2 à 5 sont les 489 codes du programme. Ils sont en hexadécimal. Aussi est-il nécessaire de les faire précéder du signe & quand on les introduit (taper successivement &A5, &71, &92, &AE, &71, etc.).



### Le programme

Les lignes 30 à 90 du programme effectuent les initialisations. De 100 à 140 se trouve la partie d'analyse des bords de l'échiquier. De 150 à 190 on saisit le coup du joueur et de 200 à 520 le programme joue.

La bibliothèque d'ouvertures se trouve aux lignes 600 à 695 pour le cas où l'ordinateur commencerait le jeu, et de 750 à 1310 lorsque le joueur commence.

Les "DATA", de 2000 à 2170, contiennent jusqu'en 2010 la liste préférentielle (tableau L()), jusqu'en 2030 les valeurs des cases (tableau N\$()), et enfin la bibliothèque d'ouvertures (tableau B()).

La partie en langage-machine se compose en fait de quatre sous-pro-

grammes. De &3945 à &3A0A, un programme d'étude dans les huit directions pour le retournement des pions appelle huit fois le deuxième sous-programme (de &38C5 à &3944) qui effectue l'étude proprement dite dans la direction spécifiée. Le troisième sous-programme (&3A0B à &3A39) remet la grille en l'état initial après chaque étude. Enfin le dernier sous-programme (&3A3A à &3AAD) calcule les pondérations M et S de la fonction d'évaluation. La stratégie mise en œuvre par le programme fera l'objet d'un article à venir.

Ces routines ne sont pas des modèles du genre (n'allez surtout pas suivre leur exemple !) mais elles ont le mérite de fonctionner !

### Programme chargeur-contrôleur de codes LM

```
1:FOR I=PEEK &78
  63*256+197TO I
  +489
2:BEEP 1,1:INPUT
  "CODE";A:POKE
  I,A
3:NEXT I
4:FOR I=PEEK &78
  63*256+197TO I
  +489
5:N=PEEK I:M=INT
  (N/16):N=N-16*
  M:A$=CHR$ (M+4
  8+7*(M>9))+
  CHR$ (N+48+7*(
  N>9)):PRINT A$
:NEXT I:END
```

née, vous pouvez jouer. RUN lance l'exécution. L'ordinateur vous demande s'il joue avec les pions noirs ou blancs ; souvenez-vous que les noirs commencent.

La notation utilisée pour indiquer les coups joués est illustrée dans l'encadré "les règles du jeu" (page 39). C'est la notation classique.

Le programme refuse un coup illé-

gal (case déjà jouée ou coup ne provoquant aucun retournement) et vous rappelle à l'ordre par 4 bips. Un coup introduit est validé par ENTER.

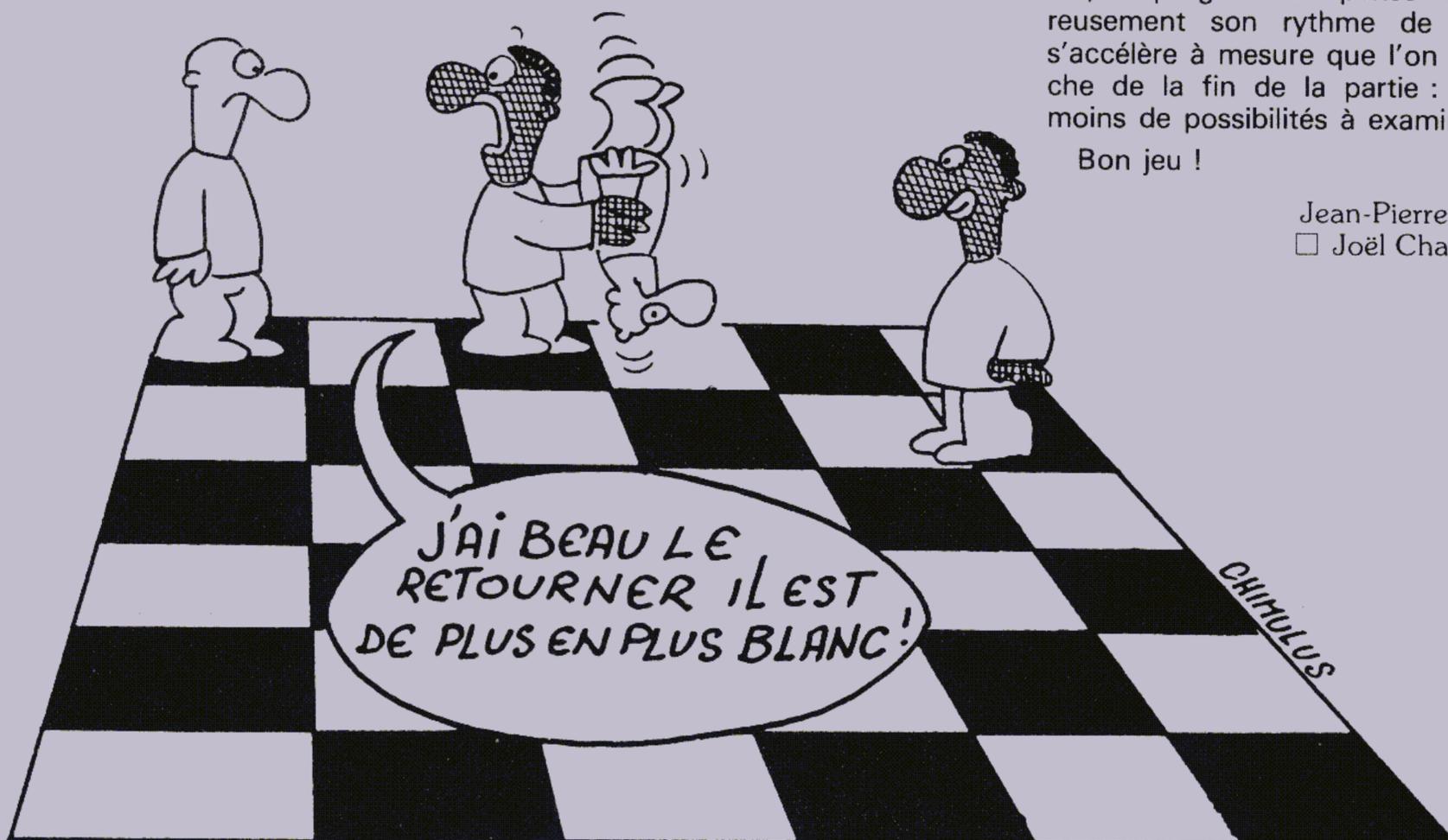
Si vous ne pouvez jouer aucun coup valide, il faut passer. Appuyez alors sur 0 et ENTER. La fin de la partie se réalise toujours par un "Je passe" de l'ordinateur et de vous-

même. Le PC-1500, par modestie (?), ne vous donnera pas le score final et ne fera aucun commentaire.

Le programme joue sa partie en 35 à 45 minutes. Les quatre premiers coups sont instantanés grâce à une bibliothèque d'ouvertures sur 8 demi-coups. En revanche, les 2 ou 3 coups suivants sont très lents (parfois plus de 8 minutes). Patientez, le programme "pense" ! Heureusement son rythme de travail s'accélère à mesure que l'on approche de la fin de la partie : il y a moins de possibilités à examiner.

Bon jeu !

Jean-Pierre Cayre  
□ Joël Chacornac



# Orthographe et syntaxe : une question de langage

Sur tout ordinateur, on utilise un langage de programmation qui possède son propre vocabulaire et sa propre grammaire. Les règles à respecter sont d'ailleurs extrêmement rigides. Voici une série de réflexions sur le sujet.

■ L'écriture d'un programme quel qu'il soit n'est jamais très compliquée. Si la préparation a été bien conduite, la rédaction des instructions ne pose guère de problèmes. Il suffit de connaître un minimum du langage informatique utilisé par l'ordinateur. Ça se passe comme avec une langue étrangère. Lorsque vous vous adressez à un étranger dans sa langue, vous avez en tête une idée que vous formulez en utilisant du vocabulaire et en respectant des règles de syntaxe.

La communication avec un ordinateur obéit, tout compte fait, au même schéma. L'idée que vous devez faire passer, c'est l'algorithme du programme. Pour la transmettre, vous utiliserez une syntaxe et un vocabulaire compréhensibles par l'ordinateur.

En réalité, les mots que vous utilisez pour « adresser la parole » à l'ordinateur ne sont pas directement compréhensibles par le microprocesseur, le cerveau du système. Un microprocesseur tout seul, si l'on me permet cette comparaison, c'est un peu comme un enfant nouveau-né : il réagit à la voix de ses parents, mais sans rien comprendre au sens des phrases qui lui sont adressées. C'est l'apprentissage, l'expérience qui lui permettront petit à petit de progresser dans ce domaine.

Pour le microprocesseur, l'apprentissage est apporté en une fois par un circuit de mémoire morte contenant un interpréteur. Le rôle de

l'interpréteur est d'indiquer au cerveau de l'ordinateur comment réagir en face d'une situation donnée. Les mots (anglais) frappés au clavier seront convertis en suites de 0 et 1 qui sont le seul langage compréhensible par l'ordinateur.

Au début, les ancêtres des actuels poquettes n'avaient pas d'autres mots à leur vocabulaire que ces suites de 1 et de 0. Et les programmeurs devaient leur adresser la parole sous cette forme. Réaliser une simple multiplication était une véritable aventure. Pour écrire un programme, il fallait alors connaître parfaitement la structure interne du processeur et se lancer dans un travail préparatoire complexe en suivant un cheminement très strict. Ce mode de programmation n'est plus guère employé aujourd'hui. Des gens ingénieux ont en effet trouvé que les 0 et 1 étaient bien rébarbatifs et qu'il y avait intérêt à faciliter le dialogue par l'usage d'ordres compréhensibles, traduisibles ensuite en états binaires. C'est ainsi que sont nés les assembleurs, beaucoup plus commodes à utiliser, mais obligeant toujours à respecter le fonctionnement de tel microprocesseur particulier. L'assembleur se contente en effet de traduire les instructions du programme-source, écrites dans un jargon plus ou moins clair, en mots binaires constituant le programme-objet.

Un grand pas avait été accompli. Mais il y avait mieux à faire et, en particulier, s'affranchir complètement de la structure interne de l'ordinateur. C'était le seul moyen de rendre les programmes facilement adaptables sur différentes machines. L'idéal, évidemment, aurait été de pouvoir s'adresser à la machine comme à un individu. On n'est pas près d'y parvenir. Mais on a fait certains progrès dans cette direction en utilisant, pour programmer, des mots d'un vocabulaire courant, faciles à manipuler.

C'est ainsi qu'apparurent les langages évolués, Fortran, Cobol, APL, Basic, Pascal... qui convertissent

des ordres clairs, le plus souvent exprimés par des mots anglais, en instructions machine.

La traduction peut être faite en une seule fois, avant l'exécution du programme. On dit alors que le langage est compilateur. Ce système assure une grande rapidité d'exécution, mais l'édition, la mise au point du programme est assez compliquée.

La traduction peut aussi se faire « mot à mot », pendant l'exécution du programme. Le langage est alors qualifié d'interpréteur. Si ce système est sensiblement plus lent, il est beaucoup plus simple à utiliser. La correction est rapide, sur une liste de programme. Actuellement, les ordinateurs de poche fonctionnent tous avec un langage interprété, en général, le Basic.

Il subsiste encore quelques machines qui s'expriment dans un langage un peu plus rudimentaire, appelé « langage machine spécialisé ». Ce sont les calculatrices programmables. Sur celles-ci, c'est la fonction calculatrice qui est primordiale. Et l'interpréteur est demeuré le plus près possible du déroulement d'un calcul au clavier. Les structures évoluées de la programmation, répétitives (boucles) et conditionnelles (tests), y sont moins immédiatement accessibles qu'en Basic. Mais la traduction d'une séquence de calcul en programme est quasi-immédiate.

— Un dictionnaire —  
— pour se faire —  
— comprendre —

Quel que soit le langage utilisé pour « discuter » avec un ordinateur, il faut se servir de mots pour s'exprimer. Le vocabulaire informatique varie selon les langages. Pour un même langage, il existe même des dialectes différents, des nuances propres à un constructeur ou un concepteur de logiciels. Et cela est si vrai que le terme de Basic, par exemple, désigne plus une famille de langages qu'un langage unique.

Ce qui importe, c'est de savoir ce que l'on veut dire. On est parfois contraint à utiliser des périphrases pour pallier un manque de vocabulaire. Sur l'ordinateur, ce manque peut provenir de l'interpréteur, ou d'une connaissance imparfaite du langage de la part de l'utilisateur. Dans un cas comme dans l'autre, la solution consiste à emprunter des

chemins détournés. Par exemple, certains Basics possèdent un groupe d'instructions capable de réaliser directement des structures itératives vraies (répéter une opération jusqu'à ce qu'un résultat soit atteint). Ils utilisent pour cela REPEAT... UNTIL (« répète jusqu'à... »).

Votre poquette n'a pas cette possibilité ? Ce n'est pas grave. Il suffira de fabriquer cette fonction avec un test et un branchement vers le début de la boucle tant que la condition ne sera pas remplie. Autre exemple, les opérateurs logiques qui permettent de combiner les tests : IF A = B AND C < D THEN... Si votre ordinateur ne connaît pas ce langage, il n'y a qu'à programmer ces tests en série.

Dans presque tous les cas, il y a moyen de combler les lacunes de l'interpréteur (ou du programmeur). Il suffit pour cela de trouver dans la panoplie des instructions disponibles celles qui permettront de traduire l'algorithme sous forme de programme. Quelquefois, il faudra aller jusqu'à modifier l'algorithme.

On se sortira de ces situations délicates en établissant clairement la relation existant entre les structures de programmation et les mots (les instructions) qui permettent de les mettre en œuvre sur l'ordinateur. Il faut connaître le vocabulaire, certes, mais aussi savoir sa signification

exacte, pour l'employer à bon escient.

Pour préciser les choses, j'ai essayé de regrouper les différents mots utilisés par un interpréteur Basic, selon leur rôle. Cela conduit à distinguer deux classes de termes. Il y a d'abord le vocabulaire de communication de l'algorithme et ensuite le vocabulaire de commande. Le premier comprend les mots qui sont habituellement écrits dans les lignes de programme, opérateurs, instructions et fonctions. Le second est utilisé pour l'exécution, la sauvegarde et la mise au point des programmes.

Le tableau ci-dessous détaille ces deux types de vocabulaire en indiquant le rôle des mots qui les composent. La liste n'est, bien sûr, pas exhaustive, mais recense l'essentiel des mots du Basic. En fait, ce que je retiens dans ce tableau, c'est surtout le classement, je trouve qu'un langage informatique, vu de la sorte, paraît beaucoup moins impressionnant.

———— La syntaxe : ————  
 ———— comment ————  
 ——— combiner les mots ———

Pour parler une langue étrangère, il ne suffit pas d'avoir un dictionnaire dans la tête. Encore faut-il savoir agencer les mots de façon à

former des phrases cohérentes. En informatique aussi, il existe des règles de syntaxe. Elles sont même draconiennes. On ne s'adresse pas à un ordinateur sans y mettre les formes : la machine est en effet parfaitement têtue et bornée. Tout ce qui ne correspond pas exactement aux règles strictes de son langage est impitoyablement rejeté. Je trouve d'ailleurs que le terme d'interpréteur peut prêter à confusion. L'ordinateur n'interprète rien ; si une expression est un tant soit peu erronée, il la rejette. On devrait plutôt qualifier un langage évolué de traducteur.

La forme syntaxique courante (c'est vrai pour le Basic, mais pas pour tous les langages) combine essentiellement un numéro de ligne, des fonctions et instructions disposées selon certaines règles, et divers signes de ponctuation. Le numéro de ligne indique à l'ordinateur dans quel ordre exécuter un programme. Il sert également à repérer les branchements. La valeur maximum que prend ce numéro dépend des interpréteurs. Certains ne dépassent pas 999, d'autres 32767, d'autres encore 65535.

Ces chiffres ne sont pas le fait du hasard. Ils découlent du nombre de bits utilisés pour le codage ; 999 est codé en BCD (de l'anglais *Binary Coded Decimal* : décimal codé en binaire) sur 12 bits ; 32767 et 65535 sont codés directement en binaire respectivement sur 15 et 16 bits.

Les règles de disposition des instructions sont définies selon une logique caractéristique du langage (quelquefois d'un interpréteur donné). Ainsi, le plus souvent, pour affecter une valeur à une variable, on doit écrire l'identification de la variable, à gauche, le signe égal, puis un nombre, une autre variable ou une expression plus complexe. Chaque fonction ou instruction a ses règles à respecter impérativement sous peine de « syntax error ».

Dernier constituant de la syntaxe d'un langage informatique, la ponctuation. Elle comprend en Basic les virgules, points-virgules, deux points, points (décimaux), guillemets et parenthèses. On pourrait ajouter également le point d'exclamation et les caractères %, #, \$ qui servent de spécificateurs de type pour les variables. Un cas particulier à signaler aussi, le point d'interrogation. Ce n'est pas en Basic un signe de ponctuation, mais souvent une abréviation de l'instruction PRINT.

Vocabulaire de communication de l'algorithme	
<b>• Opérateurs</b>	
Arithmétiques	: +, -, *, /
Logiques	: AND, OR, NOT, XOR, EQU
Concaténation (addition de chaînes de caractères)	: +, ;
<b>• Fonctions</b>	
Mathématiques	: LN, EXP, LOG, SQR, ↑ (puissances), ABS, INT, FRAC, SGN, RND
Trigonométriques	: SIN, COS, TAN, ASN, ACS, ATN
Traitement de chaînes	: LEFT\$, RIGHT\$, MID\$, LEN, CHR\$, ASC, STR\$, VAL
<b>• Instructions</b>	
de structure	{ Branchements : GOTO, GOSUB, RETURN, CALL
	{ Tests : IF, THEN, ELSE
	{ Boucles : FOR, STEP, NEXT, REPEAT, UNTIL
d'entrées-sorties	: INPUT, PRINT, LPRINT, INPUT#, PRINT#, PUT, GET
d'affectation	: LET, INKEY\$, READ, DATA, RESTORE
de déclaration	: DIM, OPEN
d'attente	: PAUSE, WAIT
d'arrêt	: STOP, END
Vocabulaire de commande	
Exécution	: RUN
Visualisation édition	: LIST, LLIST, DIR, ↑, ↓, →, ←, INS, DEL
Sauvegarde	: SAVE, CSAVE, LOAD, CLOAD, VERIFY, CLOAD ?
Partition	: DEFM
Effacement	: NEW, CLR, CLEAR, VAC

Revenons aux premiers symboles cités.

La virgule et le point-virgule servent de séparateurs pour les instructions d'entrées-sorties. Leur effet varie légèrement d'un ordinateur à l'autre. Les deux points permettent de placer des instructions indépendantes sur une même ligne de programme. Leur abus est très souvent une cause d'illisibilité dans une liste. S'il y a de la place, il vaut mieux les éviter. Et il faut faire particulièrement attention lorsqu'ils sont placés sur une ligne comprenant un test.

Les guillemets sont les délimiteurs de chaînes alphanumériques. Vous connaissez sans doute la différence entre PRINT A (afficher le contenu de la variable numérique A) et PRINT « A » (afficher le caractère A). Les parenthèses sont employées pour modifier les priorités de la hiérarchie algébrique dans les calculs, mais elles servent aussi comme délimiteurs d'arguments (tableaux, traitements de chaîne).

Dans l'écriture d'un programme, le vocabulaire et la syntaxe peuvent être sources d'erreurs. Dans le premier cas, ce sont des fautes de frappe : une instruction, une fonction ou une commande s'est trouvée mal dactylographiée. Cela se produit fréquemment quand on recopie dans l'ordinateur des listes de programmes. La réponse de l'ordinateur sera en général le même « syntax error » que celui provoqué par un non-respect des règles d'écriture. Dans les deux cas, la correction se fera par une relecture attentive du programme introduit.

Plus insidieuses sont les erreurs dites « de logique ». Celles-ci proviennent d'une mauvaise communication de l'algorithme à la machine. Une idée peut parfaitement être claire dans votre esprit et même sur votre organigramme, mais vous devez toujours penser que l'ordinateur a une latitude de raisonnement complètement nulle. Il ne saura pas s'adapter à une communication imprécise, puisqu'il n'a aucune intelligence. C'est pour cela qu'une bonne méthode de préparation des programmes doit être appliquée soigneusement. Une erreur de logique est plus facile à déceler dans un organigramme que dans une liste d'instructions.

S'il vous arrive de commettre ce genre de fautes, il existe tout de même des moyens de les corriger. La première méthode consiste à uti-



liser les commandes TRACE ON et TRACE OFF (ou leurs équivalents), si elles existent sur votre ordinateur. Elles déroulent l'exécution du programme en indiquant les numéros de ligne. Il est ainsi relativement aisé de repérer les endroits où « ça se plante ». Vous constaterez très souvent que les erreurs proviennent de défauts de branchements : trop de GOTO fusant dans tous les sens.

Autre méthode pour déceler les erreurs « logiques », l'impression provisoire de variables. Nous avons vu dans l'article du mois dernier comment utiliser ce procédé. Il consiste à ajouter, pendant la mise au point, des lignes de programme contenant des ordres PRINT suivis des noms de variables « stratégiques ». Un remède souverain !

### Quel style adopter

Cette petite étude de la méthode d'écriture des programmes serait très incomplète si je ne disais un mot sur le style.

Nous avons vu qu'un programme est la traduction d'un algorithme dans un langage informatique. Bétien comme il est, l'ordinateur se moque complètement que vous fassiez des efforts de style. Donc vous pouvez lui assener des lignes pleines à ras bords et des branchements dans toutes les directions. Du moment que le tout tourne rond et que les résultats sont justes, c'est l'essentiel.

Tout cela serait parfait si votre ordinateur était le seul à lire vos pro-

grammes. Mais il peut vous arriver de vouloir relire une liste après quelque temps. Ou même, vous pouvez avoir envie de la communiquer à vos amis (ou à l'Op). C'est là qu'il devient important d'avoir fait des efforts de style. Sur un ordinateur de poche, ce n'est pas toujours très facile. L'instruction REM (remarque) fait parfois défaut. L'écriture décalée des boucles, quand elle est possible, donne de piètres résultats sur un écran monoligne ou sur un papier d'imprimante large de 3,5 cm.

Les seuls moyens qui restent sont d'écrire des lignes courtes et de donner au programme un plan du type « encyclopédie » (voir l'Op 16, page 33). Ce genre de construction consiste à réaliser le programme comme une succession de sous-programmes appelés à partir d'une racine. Le programme de chasse au sous-marin du précédent numéro était écrit de la sorte. Un inconvénient à ce procédé, il consomme de la mémoire.

Alors, pour satisfaire à la fois les exigences de l'utilisateur (la lisibilité) et de la machine (la compacité), il existe une astuce. Elle consiste à mettre au point et à archiver le programme dans une version délayée. Lorsque tout fonctionne bien et que cette version a été sauvegardée et listée, on s'attaque à l'optimisation. On supprime le superflu, on condense la structure du programme pour qu'il ne reste qu'une nourriture extrêmement concentrée que l'on donne en pâture à l'ordinateur.

□ Xavier de La Tullaye

# Cherchons la panne

La plupart des pannes automobiles sont faciles à identifier. Avant d'appeler un mécanicien à la rescousse, essayez de dépister d'où elles proviennent et d'y remédier vous-même (programme pour PC-1251).



■ L'informatique au service de l'automobile, c'est ce que vous propose ce petit logiciel. Celui-ci chargé, le PC-1251 aura en mémoire les pannes les plus courantes pouvant survenir sur une voiture, ainsi que les principaux composants du moteur. On a retenu les incidents les plus fréquents que l'on peut souvent dépanner soi-même avec du petit matériel.

Le but de ce programme n'est pas, bien entendu, d'expliquer comment réparer un véhicule, il ne remplace pas non plus le mécanicien. Il indique seulement quels organes du moteur doivent être examinés. Parmi eux se trouve (en principe) le

responsable de la panne. L'ordre dans lequel apparaissent à l'affichage les différents composants « suspects » du moteur n'a pas été laissé au hasard : ceux qui ont le plus de chances d'être en cause apparaissent en premier.

Pour utiliser ce programme, on tape DEF D (comme dépannage), puis après un petit générique, vient le libellé d'une panne, en l'occurrence : "MOTEUR REFUSE DE PARTIR". Le programme fonctionne selon le principe du menu exposé dans *l'Op* n° 10, page 45. Si c'est la panne qui vous intéresse, appuyez sur la touche O (ou sur n'importe quelle autre touche alpha-

numérique) suivie de ENTER. Sinon, appuyez directement sur ENTER jusqu'à l'apparition à l'écran de la panne qui vous concerne.

Après avoir tapé O ENTER, l'ordinateur vous invite à regarder sous le capot un premier organe du moteur pouvant être à l'origine de la panne. Si celui-ci n'est pas défectueux, appuyez sur ENTER pour prendre connaissance d'un autre organe à examiner, et ainsi de suite.

En espérant que ce programme vous servira le moins souvent possible et que vous n'irez pas renverser de l'essence sur votre PC !

□ Bernard Lafay

## Cherchons la panne

Programme pour PC-1251

Auteur Bernard Lafay

Copyright l'Ordinateur de poche et l'auteur

```

10:"1" DATA "RESERVOIR
D/ESSENCE
20:"2" DATA "BOUCLEUR PR
INCIPAL
30:"3" DATA "BOUCLEUR RA
LENTI+VIS REGL
40:"4" DATA "STARTER
50:"5" DATA "FILTRE A E
SSENCE
60:"6" DATA "FILTRE A A
IR
70:"7" DATA "FLOTTEUR O
U POINTEAU
80:"8" DATA "FUITES D/E
SSENCE
90:"9" DATA "POMPE A ES
SENCE
100:"0" DATA "BOBINE
110:"W" DATA "DISTRIBUTE
UR
120:"E" DATA "CONDENSATE
UR
130:"R" DATA "DYNAMO ET
REGULATEUR
140:"T" DATA "BATTERIE
150:"Y" DATA "BOUGIES
160:"U" DATA "CALAGE DU
DISTRIBUTEUR
170:"I" DATA "COURROIE D
E DYNAMO
180:"P" DATA "SOUPAPES
190:"/" DATA "COMPRESSION
200:"- " DATA "POMPE A HU
ILE
210:"+ " DATA "JAUZE A HU
ILE
220:"*" DATA "MANOMETRE
D/HUILE
230:"D" CLEAR : WAIT 120

```

```

240:DIM P$(0)*11,B$(0)*2
4
250:PRINT "***CHERCHONS
LA PANNE***
260:WAIT 200: PRINT "REP
ONDRE " ; CHR$ 34;"0"
; CHR$ 34;" OU <ENTE
R>
270:WAIT : USING "###"
280:INPUT "MOTEUR REFUSE
DE PARTIR",R$:P$(0)
="194538TYWEQ": GOTO
"R1
290:INPUT "MOTEUR CALE A
U RALENTI",R$:P$(0)=
"34U": GOTO "R1
300:INPUT "MOTEUR ETOUFF
.ACCELERAT.",R$:P$(0)
="2956": GOTO "R1
310:INPUT "RATES AU MOTE
UR",R$:P$(0)="26YWE"
: GOTO "R1
320:INPUT "LE MOTEUR NE
TIRE PAS",R$:P$(0)="
290UEW6": GOTO "R1
330:INPUT "LE MOTEUR COG
NE",R$:P$(0)="UY/":
GOTO "R1
340:INPUT "MARCHE IRREG.
REGIM.NORM.",R$:P$(0)
)="Y/PW": GOTO "R1
350:INPUT "CONSOMMATION
EXAGEREE",R$:P$(0)="
84UW27P": GOTO "R1
360:INPUT "PAS DE PRESSI
ON D/HUILE",R$:P$(0)
="+-": GOTO "R1
370:INPUT "DYNAMO NE CHA
RGE PAS",R$:P$(0)="I
RT": GOTO "R1
380:GOTO "D
390:"R1" FOR W=1 TO LEN
P$(0)
400:PAUSE W;" ) REGARDER.
..
410:RESTORE MID$(P$(0),
W,1)
420:READ B$(0): PRINT B$
(0)
430:NEXT W: END

```

# Ah !

## si vous aviez su...

Vous ne connaissez pas votre machine à fond, et moins encore les autres machines... Ces quelques "ficelles" vous montreront comment on peut toujours en tirer un peu plus.

### Un cas d'amnésie sur HP-41 C

■ Si comme moi, vous laissez sonner votre 41 et son module *horloge* sans réagir le dimanche matin (au lieu de vous lever comme en semaine), lisez plutôt ce qui suit.

Lorsque l'alarme cesse de sonner d'elle-même, elle devient « périmée + activée » nous dit HP. C'est pourquoi la 41 vous rappellera le message manqué si d'aventure vous la réveillez quelques heures plus tard. Méfiance ! Si vous essayez alors d'interrompre le carillon avant le troisième bip, le message à afficher devient (hélas !) « MEMORY LOST ».

Bien sûr, vous venez de perdre toutes les données emmagasinées dans la machine sauf une : l'heure qu'il est. J'avance immédiatement une interprétation : lorsque vous appuyez sur ON, la 41 est encore en état d'« apathie » (ou de « léthargie » si vous préférez). Elle demande au TIME de vérifier s'il y a une alarme à rappeler tandis qu'elle fait la « toilette » de ses registres internes. Si vous pressez « ← » peu après, elle effectuera le « ← ON » qui est l'ordre de « MEMORY LOST ». Alors respectez dorénavant

le lever musical de votre calculatrice...

□ Michel Arditti

### Réanimation d'un programme sur PC-1500

■ Qui n'a jamais déploré un NEW accidentel, prématuré, destructeur ? Ce NEW, qui a parfois effacé le produit de dizaines d'heures de labeur acharné, ne frappera plus ! En effet, NEW n'efface rien. En fait, le premier octet de la mémoire prend la valeur 255 et les pointeurs sont remis à zéro : plus de programme, du moins apparemment !

Il suffit de changer ce code en un autre, 0 par exemple, pour faire sauter la barrière : POKE (STATUS 2 - 1), 0. Reste alors à réajuster les pointeurs : effacer une ligne de votre ancien programme (d'où l'intérêt de toujours avoir une ligne 1 : REM...) en écrivant son numéro suivi de ENTER. Le PC-1500 réajuste aussitôt ses pointeurs, récupérant ainsi le programme « perdu » amputé seulement de la ligne effacée. Un inconvénient : d'autres lignes sont aussi réapparues, parfois incohérentes. Le ménage n'est pas toujours aisé, mais dans tous les cas, cela vaut mieux que la perte totale du programme...

□ Jean-Luc Gilliéron

### De nouveaux tests pour la FX-602 P

■ Sur la FX-602 P, quatre tests sont disponibles au clavier. Ils comparent le contenu du registre X (registre d'affichage) à la valeur zéro

ou à la valeur du registre F. Ces tests ne vont que dans deux sens : supérieur ou égal ( $x \geq 0$ ,  $x \geq F$ ) ou strictement égal ( $x = 0$ ,  $x = F$ ).

Si un test est positif, le programme prend en compte toute instruction qui le suit immédiatement. Toute instruction, **même un autre test**. C'est alors que les résultats deviennent intéressants. En mettant à la suite l'un de l'autre deux de ces tests disponibles au clavier, il est possible d'obtenir un test original. Il fait des comparaisons d'infériorité, stricte ou large, ou d'inégalité.

Par exemple, la paire de tests  $x \geq F$   $x = F$  a exactement les mêmes effets qu'un test  $x \leq F$  qui n'existe pas au clavier. Et ce résultat est logique si l'on considère comment procède la machine : si un test est positif, l'instruction qui le suit est exécutée ; s'il est négatif, l'instruction qui le suit est « sautée » et c'est seulement celle qui vient après qui est exécutée.

Soit A l'instruction qui suit immédiatement la paire de tests  $x \geq F$   $x = F$  et soit B l'instruction qui suit A. La partie de programme considérée se présente sous la forme :  $x \geq F$   $x = F$  A B. L'instruction A est exécutée si la réponse aux deux tests  $x \geq F$  et  $x = F$  est positive (ce qui n'est possible que s'il y a égalité entre les contenus de X et de F) ou si la réponse au premier test est négative (ce qui arrive lorsque le contenu de X est inférieur strictement au contenu de F). En effet, dans ce cas, le programme « saute » l'instruction qui suit le test  $x \geq F$ , à savoir  $x = F$ , et il passe directement à A.

Quant à l'instruction B, elle est exécutée directement (sans passer par A), dans le cas où le test  $x \geq F$  est positif et le test  $x = F$  est négatif (c'est-à-dire lorsque le contenu de X est supérieur sans être égal à celui de F). On obtient bien les

mêmes résultats qu'avec un test unique  $x \leq F$  (la partie de programme considérée serait de la forme  $x \leq F A B$ ) : l'instruction A est exécutée si le contenu de X est inférieur ou égal à celui de F ; s'il est supérieur à F, c'est l'instruction B qui est exécutée.

D'autres tests peuvent être créés de la sorte. Il faut cependant faire attention, dans certains cas, aux valeurs du registre F (voir l'exemple ci-dessous). Ces tests supplémentaires permettent parfois d'économiser un pas de programme et même une étiquette. C'est vrai, en particulier, lorsque le test simple est suivi de deux instructions GOTO renvoyant chacune à une étiquette, la seconde introduisant une partie de programme qui devra être exécutée dans tous les cas.

Dans l'exemple proposé (dont le seul but est d'illustrer le procédé), la liste 1 peut être remplacée par la liste 2. Mais le registre F doit, ici, avoir un contenu différent de zéro pour que la paire  $x = 0 \ x = F$  soit bien équivalente à un test unique  $x \neq 0$ . C'est pourquoi au départ, il reçoit la valeur 2. Après avoir lancé le programme, on introduit un nombre. S'il est nul, la 602 l'annonce en toutes lettres avant de l'afficher mathématiquement. Sinon, elle l'affiche directement sous forme mathématique.

**Liste 1  
Un test simple**

```

*** P0
2 MinF
0
"un nb ?"
HLT x=0 GOTO1
GOTO2
LBL1
"X est nul"
HLT
LBL2
"X=#"
HLT
...037steps

```

**Liste 2  
Une paire de tests**

```

*** P0
2 MinF
0
"un nb ?"
HLT x=0 x=F GOTO1
"X est nul"
HLT
LBL1
"X=#"
HLT
...036steps

```

Les listes 1 et 2 donnent donc un même résultat, la première en utilisant le test  $x = 0$ , la seconde en utilisant l'équivalent du test  $x \neq 0$ . Mais alors que la liste 1 occupait 37 pas de programme et utilisait deux étiquettes, la liste 2 compte 36 pas et ne comporte qu'une étiquette...

□ Olivier Freud

**Un INPUT amélioré pour le PC-1251**

■ L'entrée des données dans un PC-1251 se fait le plus souvent par l'intermédiaire de l'instruction INPUT. Cette instruction accepte un « guide-opérateur » qui permet de rappeler sous la forme d'un court message quelle est la donnée demandée. On peut ainsi programmer 10 : INPUT "VITESSE ?";V.

Mais un petit problème se pose quand il faut entrer une liste de données, dans un tableau par exemple : il n'est pas possible d'utiliser une variable dans ce guide-opérateur pour rappeler le numéro d'ordre de la donnée qui va être entrée.

Une solution consiste à utiliser PRINT :

```

10 : PRINT "DONNEE NO:"; N
20 : INPUT D(N)

```

Dans ce cas, des inconvénients surgissent sournoisement :

- soit on est amené à presser deux fois sur ENTER (après PRINT et après la donnée ; c'est lassant) ;
- soit, si l'on a placé un WAIT, on doit rester vigilant et ne pas perdre de vue l'afficheur pour savoir où l'on en est.

Le programme final permet d'afficher le numéro d'ordre de la donnée à entrer dans le guide-opérateur de l'INPUT, d'où l'absence de manipulation supplémentaire et la référence toujours sous les yeux quand on frappe la donnée.

En premier lieu, nous entrons le programme suivant :

```

10:INPUT " " ;I2
900:FOR I=0 TO 10
910:A= PEEK (47152+I)
920:LPRINT I,A
930:NEXT I
940:END

```

Après avoir fait RUN 900, nous obtenons une liste de 11 valeurs : 255, 224, 16 (...) 106 et 0. Elle a été reproduite dans le tableau ci-dessous et correspond en fait à la manière dont a été codée la ligne 10 par le PC-1251 (1).

I	A	Adresse	Commentaires
0	255	47152	début du programme
1	224	47153	numéro de ligne :
2	16	47154	10
3	194	47155	INPUT
4	18	47156	"
5	17	47157	espace
6	17	47158	espace
7	18	47159	"
8	28	47160	:
9	106	47161	Z
10	0	47162	fin de ligne

La ligne 10, octet par octet : on voit que le guide-opérateur est situé aux adresses 47157 et 47158.

On voit que le guide-opérateur, ici deux espaces entre guillemets, commence à l'adresse 47157 (soit & B835 en hexadécimal). La solution du problème sera donc de paker à partir de cette adresse les codes des informations que l'on désire faire figurer dans le guide-opérateur. C'est ce que réalise le programme suivant :

```

10:INPUT "NO.00: ";D(I)
15:RETURN
20:"D" CLEAR : DIM D(20)
30:FOR I=0 TO 20
40:A= INT (I/10):B=10*(I/10-A)
50:POKE 47160,(A+64):
POKE 47161,(B+64)
60:GOSUB 10
70:NEXT I

```

A la ligne 40, on décompose l'indice I en unités (B) et en dizaines (A). A la ligne suivante, on pake dans la mémoire programme la valeur de A et de B codée sous une forme assimilable par le PC-1251.

On lance le programme par DEF "D", ou RUN 20, et l'affichage demande en clair l'introduction de la donnée 00, puis 01, puis 02, etc. On économise ainsi à chaque fois une pression sur ENTER.

□ Jean Claude Besse

(1) Voir notamment à ce sujet l'Op n° 12 page 37 et n° 18 page 32.

## Ah ! si vous aviez su...

### Les codes du FX-702 P

■ Certains des codes du FX-702 ont permis de découvrir des caractères spéciaux (1). Suivons maintenant la démarche inverse, et cherchons le code des instructions et des fonctions préprogrammées du FX-702 P.



Voilà quelle est la procédure à suivre :

- vider le 702 par CLR ALL ;
- se mettre en mode 1, en zone P0 ;
- introduire la ligne : 10 TOTO... TO (EXE) en la saturant de T et de O frappés en toutes lettres et sans utiliser la touche qui donne l'instruction TO ;
- éteindre la machine et la rallumer ;
- se mettre en mode 1, en zone P1 et introduire la ligne : 99 A (EXE) ;
- revenir en zone P0 et taper : 9999 « instruction » A (EXE) où « instruction » représente la fonction ou l'instruction dont on souhaite découvrir le code ;
- il ne reste plus qu'à frapper LIST # 1 (EXE) pour voir apparaître le code cherché.

Ce code doit être lu dans les deux premiers caractères de la ligne affichée. Par exemple, si IF a été mis à la place d'« instruction », la ligne 9 π 99 A apparaît à la fin des

(1) Dans l'Op. n° 13 page 58, on a vu que certains codes numériques permettaient d'obtenir des caractères spéciaux sur l'imprimante ( $\mu$ ,  $\Sigma$ , @,...) et certains aussi sur l'écran.

opérations décrites. Ce qui permet de conclure que 9π est le code de IF. Si l'un de ces deux caractères est blanc (c'est le cas avec LRA, PASS, RUN, par exemple), l'imprimante — si elle est branchée — révélera un D, un L ou un - (tiret).

Pour continuer la recherche de ces codes, il suffit de retourner en P0, de taper une nouvelle ligne 9999 « instruction » A (EXE) et de la faire suivre de LIST # 1 (EXE). Cette méthode permet aussi de découvrir les codes des caractères disponibles au clavier.

□ Jean-Charles Lemasson

### N'y touchez pas... (PC-1500)

■ En général, on aime bien que personne ne touche à son ordinateur quand on a le dos tourné. Quelquefois l'intervention maladroite d'une tierce personne suffit à vous faire perdre de longues heures de travail (destruction de programmes ou de données).

Aussi ai-je réalisé un tout petit programme utilitaire qui interdit à toute personne non habilitée l'usage de mon PC-1500.

```
1: ARUN : POKE# &F
  00D, 128: POKE#
  &F007 66: WAIT
  0: PRINT "PASS
  ?"
2: IF INKEY$ <>"J
  "GOTO 2
3: POKE# &F007, 0:
  POKE# &F00D, 0:
  END
```

Une fois le programme « MOT DE PASSE » introduit, éteignez le PC-1500. Si quelqu'un tentait de l'allumer (essayez), le mot PASS ? s'afficherait accompagné d'un bip strident et illimité. La touche BREAK est inopérante. La solution consiste à introduire la lettre « de passe » (ici

la lettre J) qui seule interrompt le bip et redonne la « main » à l'opérateur.

Si vous souhaitez changer de lettre de passe, modifiez simplement la ligne 2. Le POKE# & F00D, 128 inhibe la touche Break, la rendant inopérante. Le POKE# & F007, 66 produit le son strident qui persiste même lorsque le programme « tourne » (ce qui n'est pas le cas de l'instruction BEEP). Enfin, la ligne 3 rétablit la fonction Break et stoppe le bip.

Bien entendu, on peut éviter d'avoir à répondre à la lettre de passe (utile quand on l'a oubliée) : en allumant la machine touche Break pressée (on a le temps si le CE-150 est connecté) ou bien en pressant la touche All Reset située au dos du PC-1500, puis CL (au message NEW 0 : CHECK). Mais il faut le savoir...

□ Jean-Charles Gémin

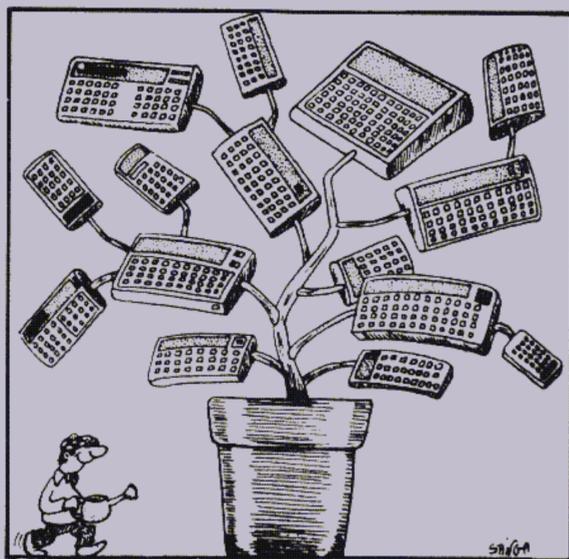
### Trucs et astuces sur ZX 81

■ Utilisez-vous pleinement la puissance et la souplesse des fonctions logiques de votre ZX ? Ces fonctions sont une véritable mine. Prenons par exemple le classique IF... THEN... ELSE..., dont le troisième terme est absent sur le ZX. On pourra très souvent le remplacer par un LET utilisant les fonctions logiques : 100 IF A = 2 THEN X = 5 ELSE X = 7 (impossible) pourra s'écrire 100 LET X = (5 AND A = 2) + (7 AND A <> 2) ou encore 100 LET X = 5 x (A = 2) + 7 x (A <> 2).

De même les deux lignes : 100 IF A = 2 THEN PRINT « BONJOUR MONSIEUR » et, 110 IF A = 3 THEN PRINT « BONJOUR MADAME » pourront avantageusement être remplacées par : 100 PRINT « BONJOUR MONSIEUR » AND A = 2; « BONJOUR MADAME » AND A = 3

□ Jacques Deconchat

# Un pot commun pour toutes les machines



## Le repas du caméléon pour PB-100

■ Les caméléons peuvent facilement se camoufler. Mais ils doivent aussi se nourrir. Avec "le repas du caméléon" (1), ici programmé pour PB-100, vous devez aider un caméléon (il est représenté par = <) à attraper sa nourriture. A gauche de l'écran, il guette sa proie (représentée par \*) : c'est un moustique. Dès que l'insecte apparaît, le caméléon doit tirer la langue (---<) pour l'attraper.

Après avoir lancé le programme, le message "FORCE 1 → 6 ?" permet de choisir entre 1 et 6 selon que vous voulez chasser 10, 20, ...ou 60 moustiques. Le caméléon reste toujours à gauche de l'écran. Seul le moustique change de place. C'est cette place qu'il faut bien repérer. Elle varie entre 1 et 9 (1 représente la position la plus proche du caméléon, et 9 la plus éloignée, à droite de l'écran). Vous devez alors presser sans attendre la touche numérique (entre 1 et 9) qui correspond à cette position pour permettre à la langue du caméléon d'atteindre le moustique. Si elle tombe juste sur ce der-

(1) Ce jeu a été proposé pour FX-702 P dans le n°8 de l'Op, page 69, et pour PC-1251 dans le n°17 page 69.

**Le repas du caméléon**  
Programme pour PB-100  
Auteur Fabrice Gigandet  
Copyright l'Ordinateur de poche  
et l'auteur

```

2 VAC
3 INPUT "FORCE 1+6",
6
5 G=6*10
10 Y=INT (RAN#*9)+1
15 PRINT "=";CSR (Y
+2);"*":
20 FOR I=1 TO 200:NE
XT I:A$=KEY
25 IF A$="9":PRINT C
SR 2:"----->":
30 IF A$="8":PRINT C
SR 2:"----->":
35 IF A$="7":PRINT C
SR 2:"----->":
40 IF A$="6":PRINT C
SR 2:"----->":
45 IF A$="5":PRINT C
SR 2:"----->":
50 IF A$="4":PRINT C
SR 2:"----->":
55 IF A$="3":PRINT C
SR 2:"----->":
60 IF A$="2":PRINT C
SR 2:"----->":
65 IF A$="1":PRINT C
SR 2:"----->":
70 GOSUB 200
75 F=F+1
80 IF A$="":A$="100"
85 IF VAL(A$)=Y:P=P+
Y*5:D=D+1:PRINT CSR
3:Y*5:" PTS":
90 GOSUB 200
95 IF F=6:PRINT CSR 2
:P:" PTS SOIT":RND(D
*20/F,-2):"/20":GOTO
2
100 GOTO 10
200 FOR I=1 TO 80:NE
XT I:PRINT :RETURN
    
```

nier, il est avalé. Vous marquez alors des points. Dans le cas contraire, il s'échappe et un autre moustique (ou peut-être est-ce le même ?) apparaît. Lorsque la chasse est terminée, le score est donné en nombre de points et sous forme d'une note sur 20.

Le programme ne présente aucune difficulté et peut être rentré dans un PB-100 privé de module d'extension : il occupe 447 pas. L'essentiel est de croire au caméléon, mais surtout de ne pas s'y attacher...

□ Fabrice Gigandet

## Notouane pour TI-58/59

■ Le Notouane est un jeu qui s'adapte assez bien aux ordinateurs de poche et il l'a prouvé (1). Il est court, rapide, ne nécessite aucun graphisme particulier et ne fait pas de bruit. Malgré la part importante du hasard, le joueur doit intervenir et prendre des décisions. Cette fois, c'est la TI-58/59 qui devient votre adversaire à ce jeu.

Les règles n'ont bien entendu pas changé. Chaque joueur lance deux dés (remplacés ici par la génération de nombres aléatoires). Seule la somme des chiffres des faces supérieures est retenue. A chaque tour, le joueur peut lancer jusqu'à dix fois les dés à condition de ne pas refaire le premier total du tour. Il doit donc savoir s'arrêter. S'il refait ce total, il perd en effet tous les points du tour. Et pour gagner, il faut recueillir plus de points que l'adversaire.

(1) Le Notouane a été adapté au PC-1211 dans l'Op n° 2 page 50 ; au FX-702 P dans l'Op n°7 page 65 ; au ZX 81 dans l'Op n°14 page 66 ; au PB-100 enfin dans l'Op n°19 page 64.

## Un pot commun pour toutes les machines

Avant de jouer contre la TI-58/59, vous devez introduire un nombre positif dans le registre t (par la touche  $x \Rightarrow t$ ) pour initialiser. Puis, vous pouvez choisir de commencer, en mettant 1 dans A (taper 1 puis A), ou de laisser la machine jouer d'abord, en mettant 2 dans A.

Lorsque c'est votre tour, vous

voyez apparaître le premier lancer (les dés sont toujours représentés sous la forme X.Y) puis la somme des faces supérieures (ici X+Y). Si vous voulez vous arrêter là, vous appuyez sur C. Au contraire, pour relancer les dés, vous devez appuyer sur B ; la TI-58/59 présente alors un nouveau lancer puis le total du tour.

Si vous refaites la première somme, vous perdez tous les points de ce tour. C'est à la machine de jouer alors. Elle n'a besoin d'aucune intervention extérieure. Il vous suffit de suivre ses lancers et les cumuls intermédiaires, à l'écran.

Pour signaler ou confirmer un changement de joueur, la TI-58/59

Notouane														
Programme pour TI-58/59														
Auteur Yannick Le Nagard														
Copyright l'Ordinateur de poche et l'auteur.														
000	76	LBL	036	71	SBR	072	01	1	112	01	1	152	05	05
001	11	A	037	01	01	073	03	3	113	95	=	153	91	R/S
002	47	CMS	038	54	54	074	67	EQ	114	59	INT	154	36	PGM
003	42	STD	039	32	X:T	075	00	00	115	42	STD	155	15	15
004	11	11	040	43	RCL	076	84	84	116	11	11	156	71	SBR
005	86	STF	041	04	04	077	97	DSZ	117	86	STF	157	88	DMS
006	40	IND	042	67	EQ	078	06	06	118	40	IND	158	65	*
007	11	11	043	01	01	079	12	B	119	11	11	159	06	6
008	32	X:T	044	01	01	080	13	C	120	69	DP	160	85	+
009	42	STD	045	01	1	081	03	3	121	28	28	161	01	1
010	09	09	046	00	0	082	09	9	122	43	RCL	162	95	=
011	00	0	047	32	X:T	083	85	+	123	08	08	163	59	INT
012	42	STD	048	44	SUM	084	43	RCL	124	32	X:T	164	42	STD
013	03	03	049	05	05	085	02	02	125	01	1	165	12	12
014	71	SBR	050	69	DP	086	85	+	126	04	4	166	85	+
015	01	01	051	23	23	087	43	RCL	127	22	INV	167	53	(
016	54	54	052	43	RCL	088	05	05	128	67	EQ	168	43	RCL
017	42	STD	053	03	03	089	95	=	129	00	00	169	09	09
018	04	04	054	77	GE	090	32	X:T	130	11	11	170	22	INV
019	42	STD	055	13	C	091	43	RCL	131	00	0	171	59	INT
020	05	05	056	87	IFF	092	01	01	132	35	1/X	172	65	*
021	87	IFF	057	01	01	093	77	GE	133	43	RCL	173	06	6
022	01	01	058	01	01	094	12	B	134	01	01	174	85	+
023	01	01	059	51	51	095	76	LBL	135	85	+	175	01	1
024	51	51	060	43	RCL	096	13	C	136	43	RCL	176	54	)
025	75	-	061	05	05	097	43	RCL	137	02	02	177	59	INT
026	07	7	062	66	PAU	098	05	05	138	55	÷	178	44	SUM
027	95	=	063	66	PAU	099	74	SM*	139	03	3	179	12	12
028	50	I×I	064	43	RCL	100	11	11	140	22	INV	180	55	÷
029	85	+	065	08	08	101	71	SBR	141	28	LOG	181	01	1
030	04	4	066	32	X:T	102	01	01	142	95	=	182	00	0
031	95	=	067	01	1	103	33	33	143	58	FIX	183	95	=
032	42	STD	068	02	2	104	22	INV	144	03	03	184	66	PAU
033	06	06	069	67	EQ	105	86	STF	145	66	PAU	185	66	PAU
034	76	LBL	070	00	00	106	40	IND	146	66	PAU	186	43	RCL
035	12	B	071	81	81	107	11	11	147	66	PAU	187	12	12
						108	43	RCL	148	22	INV	188	92	RTN
						109	11	11	149	58	FIX	189	00	0
						110	35	1/X	150	92	RTN	190	00	0
						111	85	+	151	43	RCL	191	00	0

**Tableau d'équivalence instructions/séquences de touches.**

Listé par la PC-100	Code num.	Séquences de touches
LNX	23	Inx
PAU	66	2nd Pause
EQ	67	2nd x=t
GE	77	2nd x>t
STF	86	2nd St flg
IFF	87	2nd If flg
LST	90	2nd List
RTN	92	INV SBR
PD*	64	2nd Prd 2nd Ind
ST*	72	STO 2nd Ind
RC*	73	RCL 2nd Ind
SM*	74	SUM 2nd Ind

affiche les résultats sous la forme "abc. def" où "abc" est votre score et "def" est celui de la machine. A la fin de la partie, ce résultat clignote. Inutile de préciser que si "abc" est plus petit que "def", vous avez perdu !

En recopiant la liste, n'oubliez pas que l'imprimante PC-100 n'inscrit pas tous les codes en clair. Le tableau d'équivalence ci-dessus rappelle les séquences de touches à presser pour obtenir les instructions qui prêtent à confusion.

□ Yannick Le Nagard

**Le test de Fermat en double précision sur PC-1211/PC-1**

■ L'un des théorèmes de Fermat permet d'affirmer très rapidement que certains nombres sont composés. D'après ce théorème, « si p est un nombre premier et a un nombre quelconque non divisible par p, alors  $a^{p-1} - 1$  est un multiple de p ».

En remplaçant a par 2 et p par le nombre N à tester, on peut affirmer que si le reste de la division de  $2^{N-1}$  par N n'est pas égal à 1, N n'est pas premier. De là, certains programmes, déjà publiés (1), donnent le reste de la division de  $2^{N-1}$  par N.

Mais ce « test de Fermat » — c'est-à-dire l'étude du caractère premier ou non d'un nombre en application du théorème de Fermat — se

heurte à une limitation gênante : les calculs intermédiaires s'effectuent sur des produits de deux nombres du même ordre de grandeur que le nombre N à étudier. Par exemple, pour un nombre N de 5 chiffres, les calculs intermédiaires se font sur des nombres de 10 chiffres. Si l'ordinateur travaille au maximum avec 10 chiffres significatifs, il ne peut pas appliquer correctement le test de Fermat à un nombre supérieur à 99 999. C'est pourquoi les programmes déjà publiés ne pouvaient pas recevoir plus de quatre chiffres pour la TI-57 et cinq pour la HP-41.

Sur le PC-1211/PC-1, il est possible de repousser cette limitation en s'abstenant de mettre en mémoire les résultats intermédiaires. On peut, en respectant cette condition, traiter des nombres de 12 chiffres, donc appliquer sans risques le test de Fermat à des nombres de 6 chiffres.

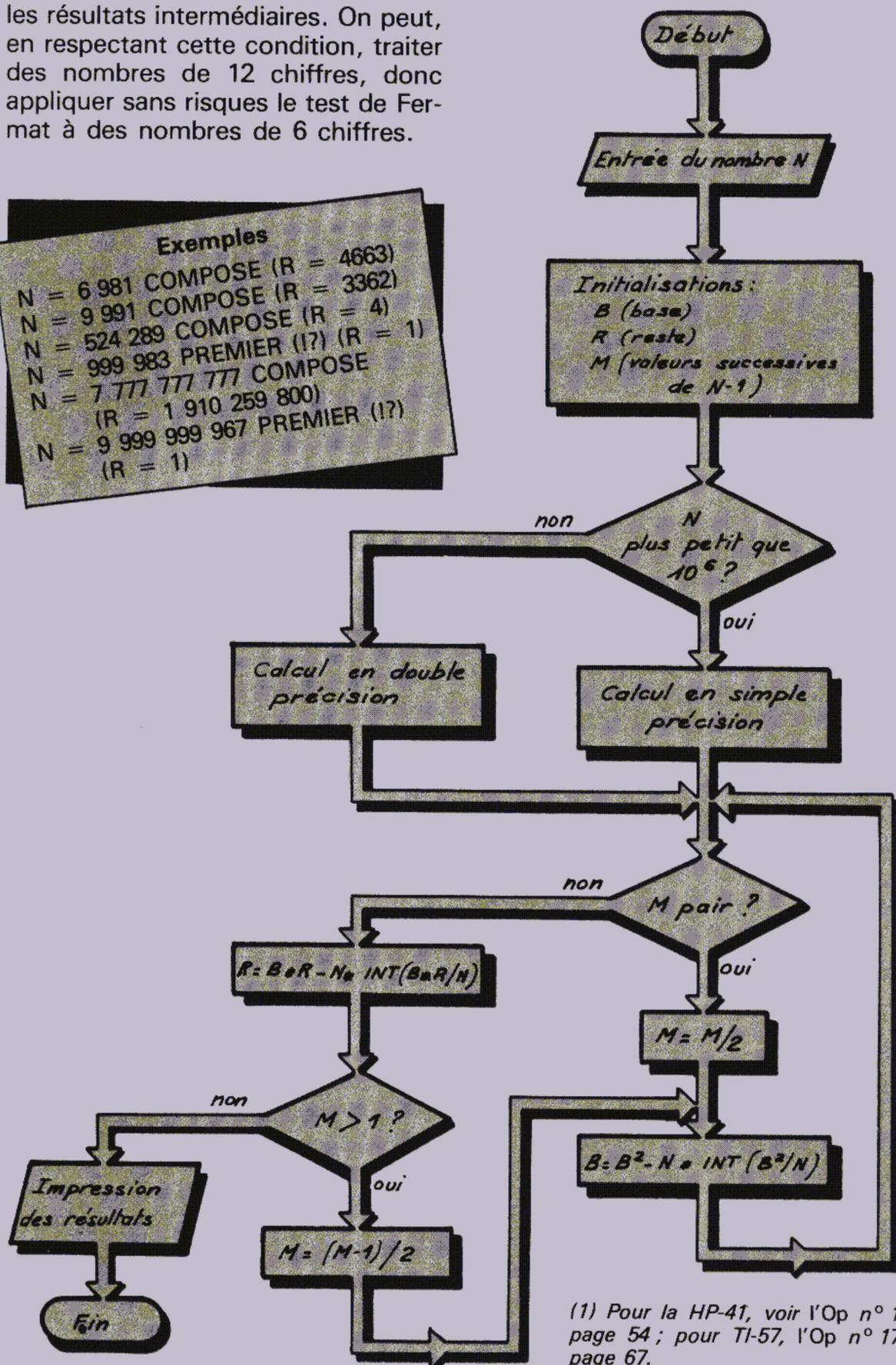
**Exemples**

N = 6 981 COMPOSE (R = 4663)  
 N = 9 991 COMPOSE (R = 3362)  
 N = 524 289 COMPOSE (R = 4)  
 N = 999 983 PREMIER (!?) (R = 1)  
 N = 7 777 777 777 COMPOSE (R = 1 910 259 800)  
 N = 9 999 999 967 PREMIER (!?) (R = 1)

Le programme présenté page suivante utilise cette propriété, combinée à un algorithme de déroulement très rapide, ne comportant en réalité que 5 lignes (lignes 60 à 100).

Mais ce n'est pas tout. Le nombre introduit peut même compter plus de 6 chiffres (sans toutefois dépasser 10 chiffres). Le programme recherche automatiquement une filière de double précision pour les calculs internes.

Le mode d'emploi du programme est très simple. Après RUN, introduire N. L'afficheur (ou l'imprimante) indique ensuite le résultat : soit « N COMPOSÉ », soit « N PREMIER (!?) ». Le symbole (!?) indique



(1) Pour la HP-41, voir l'Op n° 1 page 54 ; pour TI-57, l'Op n° 17 page 67.

**Test de Fermat  
(double précision)**  
Programme pour PC-1211/PC-1  
Auteur Pierre Ladislav Gedo  
Copyright l'Ordinateur de poche  
et l'auteur

```

10: INPUT " N =
    "; N
20: IF (N>1)*(N<
    E10)*(N=INT
    N)=0GOTO 10
30: B=2:R=1:M=N-
    1
40: IF N>=E6GOTO
    200
50: GOTO 70
60: B=B*B-N*INT
    (B*B/N)
70: IF M/2=INT (
    M/2)LET M=M/
    2:GOTO 60
80: R=B*R-N*INT
    (B*R/N)
90: IF M>1LET M=
    (M-1)/2:GOTO
    60
100: IF R=1PRINT
    N;" PREMIER(
    !?)":END
110: PRINT N;" CO
    MPOSE":END
200: C=E-5:D=INT
    (C*N):E=C*N-
    D:GOTO 220
210: X=B:GOSUB 50
    O:B=P
220: IF M/2=INT (
    M/2)LET M=M/
    2:GOTO 210
230: X=R:GOSUB 50
    O:R=P
240: IF M>1LET M=
    (M-1)/2:GOTO
    210
250: GOTO 100
500: F=INT (B+C):
    G=B+C-F:H=
    INT (C*X):I=
    C*X-H
510: J=INT (B*X/N
    ):K=INT (C*J
    ):L=C*J-K
520: P=E10*(F*H-D
    *K+F*I+G*H-D
    *L-E*K+G*I-E*
    L)
530: IF P<0LET P=
    P+N
540: IF P>=NLET P
    =E10*(F*H-D*
    K+F*I+G*H-D*
    L-E*K+G*I-E*
    L)-N
550: RETURN

```

## Un pot commun pour toutes les machines

qu'il n'est pas tout à fait certain que N soit premier, mais que c'est très probable. En effet, il existe des contre-exemples, dits nombres pseudo-premiers, qui sont heureusement très rares (22 parmi les 10 000 premiers nombres). La lutte contre ces parasites offre un joli champ de recherche.

Le programme ne comporte pas l'affichage du résultat numérique du test, c'est-à-dire R (reste de la division de  $2^{N-1}$  par N), dont la connaissance en soi est superflue. Si  $R \neq 1$ , N est composé ; si  $R = 1$ , N est

### Organisation du programme

Lignes 10 à 110 : calcul en simple précision  
Lignes 200 à 250 : calcul en double précision  
Lignes 500 à 550 : sous-programme de multiplication en double précision

### Variables

N : nombre à étudier  
M : valeurs successives de N-1  
B : base  
R : résultat numérique du test  
C à L : variables de calcul  
P : produit de deux nombres modulo N (en double précision) dont le premier terme est B  
X : index de choix du second terme de la multiplication précédente,  
Si  $X = B$ ,  $P \equiv B*B \pmod{N}$   
Si  $X = R$ ,  $P \equiv B*R \pmod{N}$

vraisemblablement premier ; pour connaître sa valeur, il suffit de rappeler la mémoire R.

Naturellement, le temps d'exécution dépend étroitement de la filière suivie : de l'ordre de 10 à 30 secondes pour les nombres de 6 chiffres ou moins, et de l'ordre de 1 à 3 minutes pour les nombres comportant 7 à 10 chiffres.

Le programme est écrit dans un langage basic standard, le plus neutre possible. Il sera donc possible de l'adapter sans trop de difficulté sur une autre machine programmable en Basic.

□ Pierre Ladislav Gedo

## Toujours Fermat en double précision sur FX-702 P

■ C'est encore le théorème de Fermat (voir l'article précédent) qui permettra de repérer rapidement la plupart des nombres composés à l'aide d'un FX-702 P. En effet, si le reste de la division de  $2^{N-1}$  par N est différent de 1, on peut conclure que N n'est pas premier. Mais, si ce reste est égal à 1, aucune certitude n'est autorisée.

Ce programme permet, lui aussi, de traiter des nombres entiers de 10 chiffres. Il occupe peu de place en mémoire.

Une fois qu'il est introduit, il suffit d'entrer N (dès qu'apparaît « N = ? », à l'écran), d'attendre, et le résultat s'affiche sous la forme : « RESTE = ». Si ce reste n'est pas égal à 1, N n'est pas premier. Si ce reste est 1, on ne peut rien affirmer, et il faut avoir recours à des méthodes plus laborieuses — et beaucoup plus longues — pour savoir si N est ou non premier.

□ Daniel Saada

### Test de Fermat double précision Programme pour FX-702 P

Auteur Daniel Saada  
Copyright l'Ordinateur de poche et l'auteur

```

10 INP "N=",N:M=N-
1:R=INT (SQR N+
0.5):D=R^2-N
20 P=INT (LOG M/LO
G 2):P=2^P:Z=1
30 IF P<=M:M=M-P:Z=
Z+Z-N*INT ((Z+Z
)/N)
40 P=P/2:IF P<1:PR
T "RESTE=";Z
50 A=INT (Z/R):Z=Z
-A*R
60 B=INT (A^2/R):C
=A^2-B*R
70 A=2*A*Z+B*D:B=I
NT (A/R):A=A-B*
R
80 Z=Z^2+A*R+D*(B+
C)
90 Z=Z-N*INT (Z/N)
:GOTO 30

```

## Quand les reines s'évitent...

### ZX 81 version de base

■ Parmi les problèmes classiques qui ont pour origine le jeu d'échecs, celui des reines est un des plus connus. Le mathématicien Gauss, en son temps, lui apporta ses lettres de noblesse en essayant de le résoudre.

Rappelons l'énoncé du problème : il s'agit de placer sur un échiquier un nombre maximum de reines de telle manière qu'aucune d'entre elles ne puisse en prendre une autre, c'est-à-dire ne se trouve sur la même rangée, la même colonne ou la même diagonale qu'une autre (1).

Le programme ci-dessous, qui tourne sur un ZX 81 sans extension de mémoire, donne toutes les solutions possibles avec une représentation graphique de l'échiquier et le résultat sous une forme codée. Il fonctionne pour tout échiquier entre deux et dix cases de côté. On pourrait même aller au-delà à condition de renoncer à la représentation gra-

(1) On trouvera un programme traitant le même problème, mais sur TI-58/59, dans l'Op n° 7 (juillet-août 82), pages 49 à 52.

phique, grosse consommatrice de mémoire sur le ZX 81 de base.

L'algorithme retenu consiste à placer les dames colonne par colonne en partant de celle de gauche. Dans un second temps, on cherche quelle est la première ligne où l'on peut placer une dame. Ainsi la première dame posée sera toujours en première colonne, première ligne. Partant de là, la deuxième dame ne pourra être placée ni en première ligne, ni en seconde. On la placera donc en troisième ligne (premier emplacement possible).

On continue ensuite l'exploration selon le même schéma jusqu'à ce qu'il ne soit plus possible de placer une dame. Si le cas se produit, on revient alors d'une colonne en arrière pour rechercher si la dame occupant cette colonne peut être déplacée. Si c'est possible, on reprend l'exploration de la colonne suivante. Dans le cas contraire, on remonte d'une colonne encore et l'on reprend le processus.

Cette méthode présente l'intérêt évident de permettre à l'ordinateur de travailler sans avoir besoin d'aucun tableau. Il suffit en fait de noter les positions successives des dames dans chaque colonne, ces positions étant modifiées quand on redescend l'arbre d'exploration. En contrepartie, le procédé peut sembler un peu lent, et il est prudent de

ne pas laisser le ZX imprimer toutes les solutions (certaines d'entre elles se déduisent des autres par symétrie).

Pour lancer le programme, on tape RUN et NEWLINE puis un nombre compris entre 2 et 10 qui sélectionne la taille de l'échiquier ; NEWLINE de nouveau, et il ne reste plus qu'à attendre... ces dames.

□ Jacques Deconchat

#### Le problème des reines

Programme pour ZX 81 (1 Ko)

Auteur Jacques Deconchat

Copyright l'ordinateur de poche et l'auteur

```

10 INPUT S
20 DIM D(S)
30 FOR I=SGN PI TO S
40 LET D(I)=SGN PI
50 FOR J=SGN PI TO I-SGN PI
60 IF D(I)=D(J) OR ABS (D(I)-D
(J))=I-J THEN GOTO CODE "0"
70 NEXT J
80 NEXT I
90 CLS
100 FOR F=SGN PI TO S
110 PRINT AT NOT PI,F+F;F;AT F+
F,NOT PI;F
115 FOR T=SGN PI TO S
120 PRINT AT F+F,T+T;" "
125 NEXT T
130 PRINT AT F+F,D(F)+D(F);"■"
135 PRINT AT CODE "=",F;D(F)
140 NEXT F
160 PAUSE 4E4
170 LET I=I-SGN PI
180 LET D(I)=D(I)+SGN PI
190 IF D(I)<=S THEN GOTO CODE "
M"
200 LET I=I-SGN PI
210 IF I THEN GOTO CODE "0"

```

#### Trois des solutions sur un échiquier 8 x 8

	1	2	3	4	5	6	7	8
1	.	■	.	.	.	.	.	.
2	.	.	.	.	■	.	.	.
3	.	.	.	.	.	.	■	.
4	■	.	.	.	.	.	.	.
5	.	.	■	.	.	.	.	.
6	.	.	.	.	.	.	.	■
7	.	.	.	.	.	■	.	.
8	.	.	.	■	.	.	.	.

25713864

	1	2	3	4	5	6	7	8
1	■	.	.	.	.	.	.	.
2	.	.	.	.	.	■	.	.
3	.	.	.	.	.	.	.	■
4	.	.	■	.	.	.	.	.
5	.	.	.	.	.	.	■	.
6	.	.	.	■	.	.	.	.
7	.	■	.	.	.	.	.	.
8	.	.	.	.	■	.	.	.

16837425

	1	2	3	4	5	6	7	8
1	■	.	.	.	.	.	.	.
2	.	.	.	.	■	.	.	.
3	.	.	.	.	.	.	.	■
4	.	.	.	.	.	■	.	.
5	.	.	■	.	.	.	.	.
6	.	.	.	.	.	.	■	.
7	.	■	.	.	.	.	.	.
8	.	.	.	■	.	.	.	.

15863724

# la Cambridge programmable

Depuis quand connaissez-vous Sir Sinclair ? Depuis le ZX 80, n'est-ce pas ? Mais bien longtemps avant, en 1973 déjà, il proposait une des toutes premières calculatrices programmables.

■ Je me souviens très bien. C'est au niveau du douzième trou que mon compagnon de golf m'annonça que Clive Sinclair avait mis sur le marché une calculatrice minuscule qui pouvait se programmer. « Elle n'est guère plus grosse qu'une balle de golf et pourrait fort bien entrer dans le trou mieux que votre balle ». Incroyable en vérité, n'est-ce pas ? Je n'y crois pas un seul instant. « Pensez-vous réellement ainsi ? » rétorquai-je.

Et pourtant c'était vrai ! Aussi blanche que son arrière-petit-fils le ZX 80, la « Cambridge programmable » était programmable et vraiment minuscule. Je n'avais jamais vu aussi petit, my God :  $4,3 \times 2 \times 0,8$  pouces ( $110 \times 50 \times 20$  mm) pour 1390 grains (90 grammes). Et pourtant, l'afficheur à diodes compte neuf positions, fournissant huit chiffres avec leur signe. Et pourtant, entre cet afficheur et le clavier, reste encore une large esplanade blanche sur laquelle s'étale le sigle Sinclair. Et les 19 touches n'offrent pas moins de 51 fonctions. L'une d'entre elles, que d'autres appelleront plus tard « Shift », « 2nd » ou « f », donne une deuxième fonction à chaque touche et même, pressée deux fois, une troisième. Vous saurez toujours où vous en êtes puisque la première pression affiche un F et la seconde un G. Presque de l'alphanumérique...

Quant à la touche notée « ./EE/- »,

Touches	Codes	Pas	Boucle de 100 à 0	
#	3	00	Entrée de chiffres	
1	1	01		
0	0	02		
0	0	03		
-	F	04		
#	3	05		
1	1	06		
=	-	07		
V	A	08		Shift bas
GOTO IF NEG	1	09		Saut en 16 si
1	1	10		le résultat est
6	6	11		négatif
V	A	12		sinon, retour en
GOTO	2	13		04
0	0	14		
4	4	15		Temps total :
STOP	0	16	environ 12 secondes	



## La Cambridge programmable grandeur nature.

il faut la presser trois fois pour parvenir au bout de ses possibilités.

Dernière astuce, les opérations directement suivies de « = » ont une autre fonction : ainsi «  $n \times =$  » donne  $n$  au carré. Les calculs se font sans sophistication, dans l'ordre où ils sont écrits, sans qu'intervienne la moindre priorité :  $4 + 2 \times 3$  donne 18 et non pas 10 comme il se doit. Les parenthèses sont là pour remettre les choses à leur place mais ne s'imbriquent pas. On a même droit à toute une mémoire avec STO, RCL et MEX qui en échange le contenu avec le registre x. Une originalité : alors que la notation scientifique ne fournit que cinq chiffres plus l'exposant, ChN remplace temporairement ce dernier par les trois manquants. Félicitations.

La panoplie de fonctions est scientifique : calculs trigonométriques, logarithmiques et conversions degrés/radians.

Et la programmation ? Très victorieuse, la Cambridge n'incite pas à la débauche : 36 pas de programme en tout et pour tout. Elle se révèle en plus très gourmande car tout appui sur une touche coûte un pas. On peut relire et modifier son programme, mais ni insérer ni supprimer un pas. Il est vrai que, même si l'on doit tout réécrire, cela ne peut être long... En plus des fonctions de calcul, on dispose d'un « GOTO » et d'un test. Il est unique mais a l'immense mérite d'exister : « GOTO IF NEG » exécutera le saut si le registre x est négatif. C'est maigre certes, mais en 1973, c'était magique : une machine plus petite qu'un paquet de cigarettes prenait des décisions.

Le codage des fonctions est très simple en vérité. Le mode programmation provoque une sorte de « Shift lock », donc les chiffres ne donnent plus de chiffres. Pour en avoir, il suffit d'appuyer sur 3 qui devrait donner ChN, mais en fait, on obtient « # » et, effectivement, le prochain chiffre en sera vraiment un. Enfantin, n'est-ce pas ? Pour les codes, aucun problème, chaque touche a le sien. Bien sûr, comme il y a trois fonctions par touche, RCL, 5 et MEX par exemple ont le même code et seront différenciées... par le contexte. Et puis, comme il y a plus de dix fonctions, certaines sont codées A, E, F ou G. Quant à =, il devient - (qui, lui, est codé F) et  $\times$  est représenté par un point, tout comme l'est d'ailleurs  $x^2$ .

Un ami français prétend ne rien y entendre. En fait, c'est un peu plus compliqué que notre système de poids et mesures, mais bien plus simple que le cricket (et les Français ne comprennent rien ni à l'un ni à l'autre). Tout cela, il faut le reconnaître ne semble pas aujourd'hui très orthodoxe. Mais comment voulez-vous qu'un pionnier soit orthodoxe ? En copiant sur ce qui se fera par la suite ? Allons !

En fait, le nombre de fonctions étant faible, le principe est vite assimilé. La puissance de calcul est suffisante pour nombre de situations et, sur cette machine qui tient dans le creux de la main, on peut même programmer des jeux (des jeux de paume comme le dit mon ami français).

Alors, pas de médisance, somme toute, et compte tenu de sa date de naissance, cette machine, par ses possibilités et sa taille est tout à fait exceptionnelle. Bravo Sir !

□ John Luck



# Le guide des nouvelles solutions informatiques



**L'ORDINATEUR PERSONNEL**

**Indispensables :**  
10 mots-clé pour faire la synthèse de l'année  
Banques : la stratégie du cuirasse

**Surprenantes :**  
les premières éliminatoires de notre coupe 83/84  
Bancs d'essai : la crème des logiciels (2)  
Des révélations sur Atari  
Tout, tout, tout sur les disquettes  
Sévère mais juste la sélection du Père Noël

**N° 3**

M - 1714 - 3 - 20 F

N° 3 - NOVEMBRE 1983 - 20 F

**N° 3**

**20 FF chez votre marchand de journaux**

**Vous  
avez  
aimé**

# L'Ordinateur de poche

**abonnez-vous  
complétez votre  
collection !**



Sommaire  
des anciens numéros

**N° 4.** Basic et alphanumérique • Essai : Casio FX-702 P • L'OP d'un médecin • Apprendre à compter avec la TI 57 • Étiquette ou adressage numérique sur TI 58/59 • Connaître les chiffres que votre TI 57 n'affiche pas • Extraire les racines d'une fonction • Votre HP 41C : une horloge • Jeu : trouver le bon mot • La fonction CLOAD 1 sur les PC-1211 et TRS 80 poquette • Le destin des nombres dans votre OP • Bibliothèque-système des TI 58/59 • La TI 57 à cœur ouvert (II) • Bricoler un pupitre pour PC-1211.

**N° 5.** Essai : Sanyo PHC 8000 ; Interface HP-IL ; Sharp PC-1500 ; Casio FX-602 P • Quelques trucs de programmation • Comment se sortir des boucles ? Décorez votre TV avec le ZX 81 • Extraire des racines carrées • Calcul mental en changeant de base • Calcul des factorielles • Jeux : attention aux platanes ; du tac au tac • Les microchips au Japon • Bricoler un « dérouleur de bande ».

**N° 6.** Nouveau : TI 57 LCD et TI 88 ; tablette et imprimante du PC 1500 • Module "x fonctions" pour HP 41 C • Les codes-barre de la HP 41 C • Introduction au langage machine du ZX 81 (1) • Vos calculs avec des indices • Dépouiller les QCM sur les TI 57/58 • Dactylo miniature • Des idées de programmes • Jeux : carrés magiques ; slalom numérique ; gare au crocodile ; chasse aux chiffres • Leçon d'anatomie : le TI 57.

**N° 7.** Nouveau : module horloge HP 41 • Les fonctions logiques du ZX 81 • Leçon d'anatomie : TI 58 et 59 • Introduction au langage machine du ZX 81 (2) • Les guillemets du Basic Sharp • Programmer des sous-programmes • Le prix d'un coup de fil • Un Op sur un bateau • Bien arrondir les résultats • Des idées de programmes • Jeux : le pendu ; combat dans les étoiles ; exercice de mémoire ; sauvetage spatial : TI 58 aux échecs ; kaléidoscope pour ZX 81.

**N° 8.** Panorama des OP • Des nouvelles du Japon • Les chiffres romains du PC-1211 • x fonctions de la HP 41 C : un indicateur de chemin de fer • Tracé de courbes avec la PC 1500 • Les drapeaux de l'affichage sur HP 41 • Réciter les tables de multiplication à une TI 57 • Navigation de plaisance avec TI 58/59 et FX 702 P • Cadran solaire pour ZX 81 • Orthographe des nombres sur FX 702 P • Compteur de bande de programme • Jeux : le repas du caméléon ; alunissage avec la TI 57.

**N° 9.** Nouveau : HP 75 C ; HP 15 C ; PC 1251 ; CE 125 • Programmer ses jeux • Basic PC-1211 contre Basic FX 702 P • Des statistiques sur HP 41 • Racines d'un trinôme sur PC-1211 • Les histogrammes sur ZX 81 • Navigation de plaisance avec TI 59 et FX 702 P • Les additions vues par le ZX 81 • Musique sur PC 1500 • Les cristaux liquides du FX 702 P • Dessins animés sur PC-1211 • La FP 10, imprimante graphique • Jeux : les petits poids • Représentation des nombres dans votre OP • Le lecteur de carte des TI 59 à cœur ouvert.

**N° 10.** Nouveau : Casio PB-100 et son interface FA-3 ; HP 10 C ; Interface vidéo pour HP 41 • Deux utilitaires pour le PC 1500 • Afficher le menu sur OP • "Haute résolution" sur PC-100 • Êtes-vous un expert en HP 41 C ? • Se repérer sur le soleil avec TI 59 et FX-702 P • ZX 81 et récursivité • Jeux : deux points sur un damier pour TI 57 ; Othello, le programme gagnant du tournoi de l'OI.

**N° 11.** Nouveau : TI 57 LCD • A l'intérieur d'une imprimante • La PC 1500 s'autoprogramme • Traitement de texte sur FX 702 P • Classement sur ZX 81 • Intégration de Gauss sur HP 41 et PC 1500 • Se repérer sur le soleil (suite) • Transposer de la musique avec PC-1211 et TRS 80 poquette • Loterie arithmétique sur TI 57 • Jeux : aux confins de la galaxie ; FX 702 P cruciverbiste • Les dessous de la TI 57 • Première découverte sur PC 1251 • Fonctions incompatibles sur TI 58/59 • Accès au compteur hexadécimal des PC-1211 et TRS 80 poquette.

**N° 12.** Nouveau jeu : le Neiscat • En démontant une HP 34 C • Table des codes du PC-1500 • Faites l'appoint avec votre TI 59 • Transposition de TI 57 sur TI 58/59 • Index pour le manuel du PC-1500 • Se repérer sur les planètes avec TI 59 et FX 702 P • Améliorer la fonction Gamma • Équations de 3<sup>e</sup> et 4<sup>e</sup> degrés sur TI 57 • Les relationnels dans la pile de la HP 41 C • Utilitaire pour MERGE sur PC-1500 • Jeux : le pot-aux-roses ; damier électronique pour Othello ; générez des nombres aléatoires ; Black-Jack ; Trio.

**N° 13.** Nouveau : CC-40 • La HP 41 C démontée • Tenue de compte (FX 702 P) • Faire le point (TI 59 et FX 702 P) • Négocier un virage (PC-1211 et PC-1) • ZX 81, calculatrice grand écran • Deux utilitaires pour PC 1500 • Exploration des mémoires du PC-1251 • Hiéroglyphes (HP 41 C) • Inventer des mots nouveaux (PC-1251) • Jeux : rallye-auto ; chasse aux canards ; labyrinthe ; jacksot.

**N° 14.** Nouveau : HHPC de Sanco • Réglez vos comptes (PC 1211) • Évitez les météorites à bord de votre FX 702 P • Débutants, avant de programmer, déblayez le terrain • Dessiner une salle de spectacle (PC 1211-1251) • Louvoyer contre le vent (TI 59 et FX 702 P) • Affichages tous formats (ZX 81) • Autoprogrammation et catalogue, 2 utilitaires (PC 1251) • Langage-machine et bruits divers (PC 1500) • Des courbes en trois dimensions (PC 1500) • Jeux : le puzzle de Nicomaque (TI 57) ; Casino de poche (HP 41 C) ; stand de tir (TI 57) • Un programme caché dans les profondeurs des TI 58/59.

**N° 15.** Les nouvelles du Japon • Optimiser avec la pile opérationnelle de la HP 41 C • A l'intérieur de la PC 1212 • Nouveau : Interface CE-158 du PC 1500 • Pour programmer, ne mettez pas la charrue avant les bœufs • PC-Calc, feuille électronique de calcul pour la PC 1500 • Pour construire un escalier (PC 1211) • Pour se repérer sur les radiobalises (TI 59 et FX 702 P) • Tracé de courbes sans imprimante (HP 41 C) • Changer de formule sans changer de programme (ZX 81) • Le puzzle de Nicomaque (TI 57) • Changez de base pour vos calculs (FX 702 P) • Améliorez l'affichage pour les jeux (TI 57) • Chaînes de caractères et applications numériques (PC 1251) • Jeux : le fou du volant (TI 58/59) ; Taïaut ! Tayaut ! (PC 1500).

**N° 16.** Nouveau : la TI 66 • A l'intérieur du FX-702 P • Le cavalier fou (ZX 81) • Optimisation des piles opérationnelles, suite (HP 41 C) • Comment construire un organigramme • Polygones et flocons de neige (PC 1500) • Renumérotation des programmes (PC 1500) • Intervenir sans arrêter le programme (TI 58/59) • Virgule flottante (ZX 81) • Accès au compteur hexadécimal (PC 1211) • Accord parfait (PC 1211) • Pour utiliser les radiophares circulaires (TI 59 et FX-702 P) • Jeux : presse-bouton (PB-100) ; jeu de massacre (HP 41 C).

**N° 17. Panorama des ordinateurs de poche :** Casio, Hewlett-Packard, Panasonic, Sanco, Sharp, Tandy, Texas Instruments • Le petit piano du PC 1500 • Un Baccara de poche (FX 602/702 P, TI 58/59) • La table des codes du PB 100 • Planifier avant de programmer • Calculer la bonne pente (PC 1211/1251) • TI 57/57 LCD, compteurs de vitesse • Des trucs, le pot commun pour toutes les machines...

**58 RUE N.D. DE LORETTE**

**LE 1<sup>er</sup> LIBRE-SERVICE**

**DU LOGICIEL**

**NOUVEAU!**



J.C.R. lance le premier libre-service du logiciel ! J.C.R., c'est pouvoir enfin choisir en toute tranquillité parmi les 400 logiciels de jeux éducatifs, scientifiques ou utilitaires.

J.C.R., c'est pouvoir aussi équiper son micro-ordinateur : des centaines d'accessoires sélectionnés vous sont proposés sur stock.

J.C.R., c'est également pouvoir compléter sa bibliothèque en consultant la librairie spécialisée J.C.R. : les derniers et meilleurs titres y sont. Et tout cela tranquillement, en fonction de ses goûts et de ses besoins.

Le libre-service, un nouveau service J.C.R. pour vous.

**56 RUE N.D. DE LORETTE**

**J.C.R. BOUTIQUE**

**TOUS LES MICRO-ORDINATEURS**

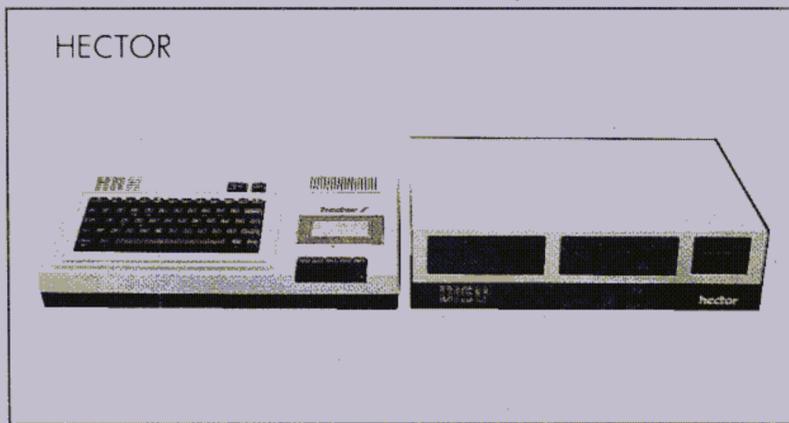
**APPLE • HECTOR • SINCLAIR • ORIC • SEIKO • CASIO • CANON • VICTOR • COMMODORE • SHARP • EPSON**



APPLE II - APPLE III  
COMMODORE 64 version SECAM 3800 F  
COMMODORE 64 version PAL 2950 F



ORIC I 48 K + cordon péritel 2180 F  
CASIO FP 200 3800 F  
SINCLAIR ZX 81 580 F



HECTOR  
48 K HR Graphique Haute Résolution 4390 F  
HRX 4950 F  
Disque 1 Drive pour HECTOR HRX 6500 F

**Catalogue JCR gratuit sur demande.**



56-58 rue N.-D.-de-Lorette 75009 PARIS Tél. : 282.19.80 - Télex : 290 350 F  
59 rue du Docteur-Escat 13006 MARSEILLE Tél. : (91) 37.62.33  
313 rue Garibaldi 69007 LYON Tél. : (71) 861.16.39  
2 rue de la Merci 34000 MONTPELLIER Tél. : (67) 58.84.37 / 58.78.36