

LO...QUES D'AUJOURD'HUI

**hors série**

# Lead

# MICRO

**APPRENDRE** • La programmation, cours de Claude Polgar • Les fortiches : gestion de fichiers, assemblage et systèmes. **MAGAZINE** • La 4<sup>e</sup> génération informatique • La contre-mesure de l'Olivetti M 21 • Le progiciel dBase III à l'essai..

DE L'INITIATION A LA PRATIQUE DE L'INFORMATIQUE

COURS  
**N°20**



**VPC 15**

ISSN 0757-6889

# UN PREMIER LEXIQUE ANGLAIS-FRANÇAIS VRAIMENT PRATIQUE ET TRÈS COMPLET + de **1500** termes !

- Index français-anglais
- Lexique des termes anglais et américains avec explication en français
- Tables de conversion

JEAN HIRAGA

## lexique de l'électronique anglais-français

Pour la première fois en électronique, un lexique anglais-français présenté sous forme pratique avec en plus des explications techniques succinctes mais précises.

**112 pages**  
PRIX : 65 F

En vente  
chez votre  
libraire  
et aux  
Editions  
Fréquences

### BON DE COMMANDE

Je désire recevoir le livre  
«le lexique de l'électronique  
anglais-français» au prix de  
**72 F** (65 F + 7 F de port).  
Adresser ce bon aux EDITIONS  
FREQUENCES 1, bd Ney,  
75018 Paris.

Nom .....

Prénom .....

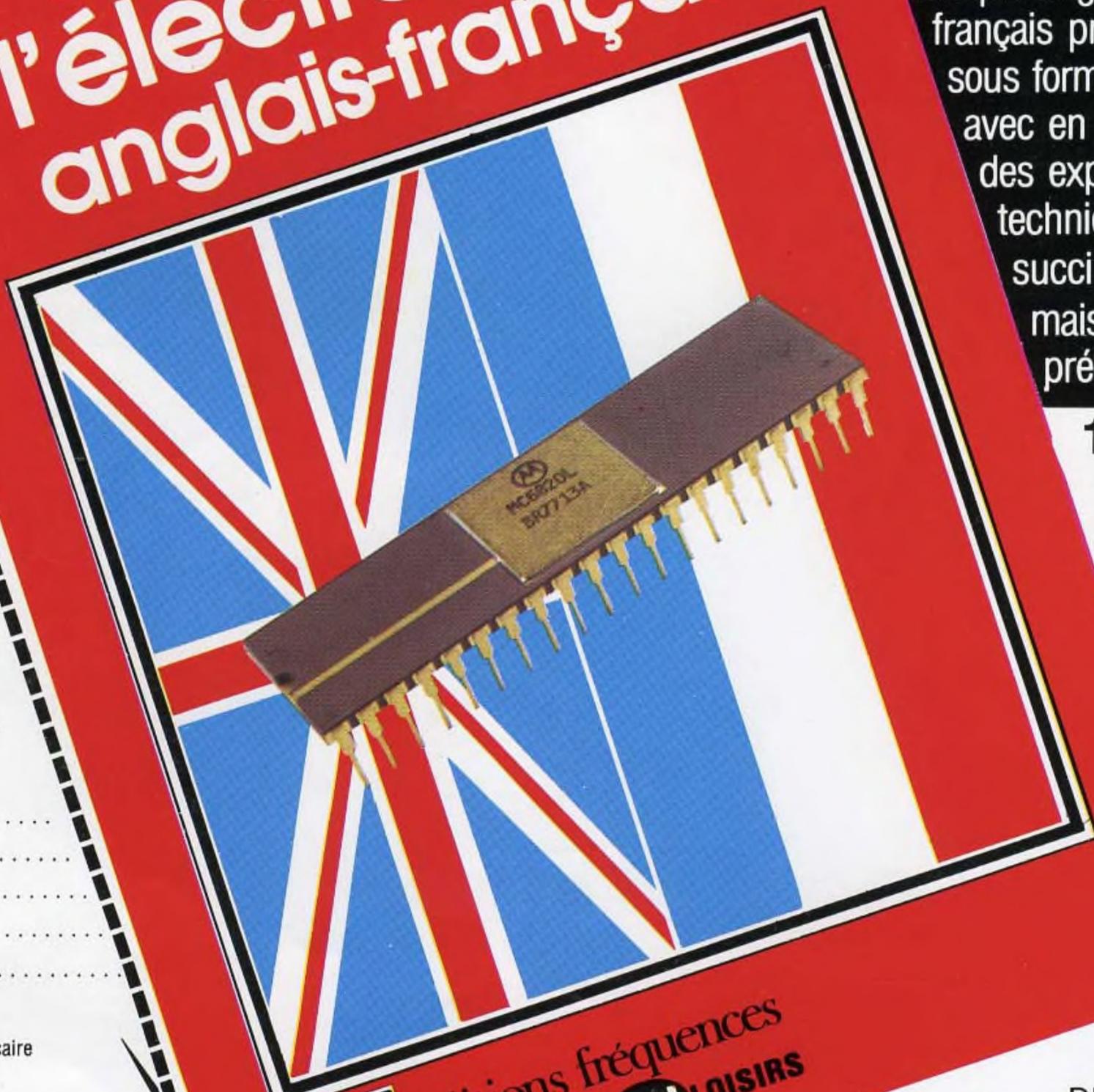
Adresse .....

Code postal .....

Règlement effectué

par CCP  par chèque bancaire

par mandat



éditions fréquences  
COLLECTION **Led** LOISIRS

DÉJÀ PARUS  
DANS LA MÊME COLLECTION

«Filtres actifs et passifs  
pour enceintes acoustiques»  
au prix de **85 F** + 7 F de port.

«Les lecteurs de compact-discs  
au prix de **130 F** + 10 F de port.

«Conseils et tour de main  
en électronique»  
au prix de **68 F** + 7 F de port.

LO... M... D'AUJOURD'HUI

**hors série**

# LED

# MICRO

Nous signalons à nos lecteurs que les Editions Fréquences seront fermées du 1<sup>er</sup> au 31 juillet 85 pour congés annuels.

MAI 85

**Société éditrice :**  
**Editions Fréquences**  
 Siège social :  
 1, bd Ney, 75018 Paris  
 Tél. : (1) 607.01.97 +  
 SA au capital de 1 000 000 F  
 Président-Directeur Général :  
 Edouard Pastor

**LED MICRO**  
 Mensuel : 16 F  
 Commission paritaire : 64949  
 Directeur de la publication :  
 Edouard Pastor

Tous droits de reproduction réservés  
 textes et photos pour tous pays  
 LED MICRO est  
 une marque déposée ISSN 0757-6889

**Services Rédaction-Publicité-  
 Abonnements :**  
 1, bd Ney, 75018 Paris  
 Tél. : (1) 607.01.97  
 Lignes groupées

**Rédaction**  
**Rubriques pédagogiques**  
**Chef de rubriques :**  
 Claude Polgar,  
 A collaboré à ce numéro :  
 Bruno Lilamand  
**Rubriques Magazine**  
**Chef de rubrique :**  
 Claude-Hélène Roze  
 Ont collaboré à ce numéro :  
 Charles-Henry Delaleu, Philippe  
 Faugeras

Secrétaire de Rédaction  
 Chantal Cauchois

**Publicité, à la revue**  
 Tél. : 607.01.97  
 Secrétaire responsable  
 Annie Perbal

**Abonnements**  
 10 numéros par an  
 France : 140 F  
 Etranger : 210 F

**Réalisation**  
 Composition-Photogravure  
 Edi'Systèmes

Impression  
 Berger-Levrault - Nancy

## RUBRIQUES PEDAGOGIQUES



**5**  
**COURS DE PROGRAMMATION  
 EN BASIC**  
 Initiation progressive à l'informati-  
 que  
 par **Claude Polgar**

**28**  
**LE COURRIER DES LECTEURS**  
 Claude Polgar répond

**30**  
**LE COIN DES FORTICHES**  
 Gestion de fichiers  
 par **Bruno Lilamand**  
 Assemblage et systèmes  
 par **Bruno Lilamand**

**46**  
**LIBRES PROPOS**  
 Réflexions sur la micro-informatique

## RUBRIQUES MAGAZINE

**48**  
**LANGAGE INFORMATIQUE**  
 Une méconnue : la 4<sup>e</sup> génération  
 informatique

**52**  
**LES CONTRE-MESURES**  
 Le Guépard (2<sup>e</sup> partie)  
 L'Olivetti M 21



**58**  
**PROGICIELS A L'ESSAI**  
 dBase III

**61**  
**LES INFOS ET LES PRODUITS**

**64**  
**BIBLIOGRAPHIE**  
 A lire.

**NOTRE COUVERTURE :** Le VPC 15 de Victor Technologies, version disque dur 15 Mo, configuration complète avec clavier et écran monochrome.

**Vous découvrez Led-Micro avec ce n°20**  
**La partie cours vous intéresse et vous désirez**  
**l'ensemble des numéros parus (depuis le n°1)**  
**Voici ce que nous vous proposons :**



**VOUS BENEFICIEZ  
D'UNE REMISE DE 30%**



**les 10 premiers  
numéros en vrac**  
**130 F les dix**  
 PORT COMPRIS

**les 10 premiers  
numéros et la  
reliure 180 F**  
 PORT COMPRIS

**SOMMAIRE DES COURS**

**N°1** Introduction générale - Vocabulaire et notions de base - L'emploi des ordinateurs • Fonctions de base  
**N°2** Configuration d'un système - L'unité centrale et ses interfaces - Ecran, clavier, imprimante • Opérateurs de base  
**N°3** Disquettes et cassettes - Machine à dessiner, numériseur, photostyle, souris • Opérateurs de base  
**N°4** Langages compilés et interprétés - Les systèmes d'exploitation - Les pro-

giciels - Classification et choix d'un micro • Opérateurs de base  
**N°5** Choisir, installer, brancher - La pratique du clavier - Mise en route • Arithmétique binaire  
**N°6** Premier programme en Basic - Ponctuation dans le Print • Arithmétique binaire  
**N°7** Déroulement d'un programme - Représentation des nombres • Les bascules  
**N°8** Calculs en BASIC - Les registres - Les compteurs

**N°9** Notion de format - Le NEWDOS • Architecture d'un système à microprocesseur  
**N°10** Le NEWDOS (fin) - Le CP/M80 - Les registres du Z80 - Déroulement d'un programme - L'U.A.L.  
**N°11** Utilisation d'un fichier enregistré en MBASIC (sous CP/M) - Le formatage • Le hardware du MPF-1B  
**N°12** L'affectation - Variables chaînes booléennes - Le langage du Z80<sup>R</sup>  
**N°13** L'affectation (fin) : INPUT - La sélection (1<sup>re</sup> partie : Sélection simple).

**N°14** La sélection - Le langage Z80<sup>R</sup> (suite).  
**N°15** Complément sur les chaînes de caractères - Le langage du Z80<sup>R</sup> (suite).  
**N°16** La sélection multiple - Les modes d'adressage.  
**N°17** La structure boucle - Contrôle des CPU et E/S.  
**N°18** La méthode béquille - Le graphisme haute résolution.  
**N°19** Les boucles - Conception de CGraph.

**Vous désirez un ou plusieurs numéros qui vous manquent (de 1 à 19) :**  
**18 F par numéro PORT COMPRIS**

**BON DE COMMANDE**

à retourner aux EDITIONS FRÉQUENCES 1, boulevard Ney - 75018 Paris

Je désire : 1 collection complète des 10 premiers numéros **en vrac**

Je désire : 1 collection complète des 10 premiers numéros **et la reliure**

Je désire le n°                      (cocher le ou les nos désirés)

Je joins à la présente commande le montant de ..... F par CCP  ch. bancaire  mandat

Nom : ..... prénom : .....

Adresse : .....

Ville ..... Code postal .....

# COURS DE PROGRAMMATION(20)

## DES EXERCICES SUR LES BOUCLES

Dans LED MICRO n° 18, nous vous avons exposé notre «méthode béquille» qui devrait vous aider à résoudre les exercices «difficiles» de boucle. A ce propos, notons que certains de nos lecteurs se sont «plantés» dans l'exercice R26 (tracé rapide du cercle). L'emploi de la méthode béquille aurait été souhaitable : nous en reparlerons dans le prochain «Coin des Matheux».

Heureusement, 95 pour 100 des boucles que l'on rencontre dans la pratique de la programmation sont des boucles FOR... NEXT... toutes bêtes. Il faut que, dès maintenant, la rédaction de telles boucles soit pour vous une seconde nature. Pour ce faire, nous paierons le prix qu'il faut : consacrer deux cours entiers de LED MICRO (numéros 20 et 21) à vous proposer des exercices sur ce thème : faites-les (même s'il vous semblent trop faciles !)

## D'UNE PIERRE... TROIS COUPS

Nous allons profiter de ces deux numéros de pause du point de vue «acquisition de notions théoriques» pour compléter notre stock d'instructions concernant les chaînes de caractères et l'affichage sur écran.

En effet, nous n'avons pas pu vous apprendre en une seule fois toutes les instructions chaînes de caractères. Vous en connaissez déjà huit (ASC, CHR\$, LEFT\$, MID\$, RIGHT\$, STR\$, VAL). C'est beaucoup. Vous en donner plus lors de la première couche, vous aurait noyés.

Maintenant que nous sommes un peu plus grands, nous allons vous donner une nouvelle liste d'instructions de la même catégorie (SPACE\$, STRING\$, POS, CSRLIN, SCREEN, INSTR) et quelques instructions d'affichage (TAB, SPC, PRINT USING, LOCATE). Les INKEY\$, GET A\$ et autres LINE INPUT viendront ultérieurement.

Ainsi, en même temps que vous vous exercerez à rédiger des boucles, vous complétez votre stock d'instructions et lorsque nous aborderons l'étude du GOSUB, nous pourrons vous donner comme exercice des petits sous-programmes d'utilité réelle (que vous pourrez utiliser comme «routines de servitude» dans vos programmes complets).

## SI VOTRE ORDINATEUR N'EST PAS UN COMPATIBLE PC

Si votre ordinateur n'est pas un compatible PC, il y a beaucoup de chances pour que vous ne retrouviez pas les instructions que nous vous décrivons. Nous sommes sortis du «noyau de base» du BASIC et entrons dans le domaine des dialectes des BASICs où tous les coups sont permis... même les plus sournois.

L'un des coups les plus sournois est celui de POS, qui ici renvoie la position horizontale du curseur, mais qui dans d'autres BASICs est l'équivalent du STRING\$ de Microsoft.

Ne vous laissez pas désarçonner : cherchez les équivalences et rédigez la liste des formats de vos instructions, puis essayez de résoudre les exercices que nous vous proposons avec votre stock. Attention : l'équivalence est rarement parfaite.

Et si vous êtes un fortiche : rédigez de petits sous-programmes permettant d'obtenir les mêmes effets que les instructions «exotiques» qui manquent à votre système.

## EN ROUTE VERS LA STRUCTURATION

A partir du numéro 22, nous pourrons (enfin !) attaquer l'étude du GOSUB et de la notion de sous-programme. C'est à partir de ce moment que nous pourrons faire de «vrais» programmes, bien structurés.

## SOMMAIRE DES N°S PRECEDENTS

Chapitres		LM n°s
1	Introduction à l'informatique	1
2.1 à 2.10	Notion de matériel -U.C. et périphériques	2 et 3
2.11 à 2 14	Notions générales sur le logiciel Le choix d'un ordinateur	4
3.1 à 3.5	De l'installation au caractère d'attente	5
3.6 à 3.10	Structure du BASIC - Le PRINT	6

Chapitres		LM n°s
3.11	Nombres et calculs	7 et 8
3.12 3.13	Notions élémentaires sur les fichiers et les SED	10 à 12
3.14 3.15	L'affectation - Notion de variables	12 à 14
3.16	La sélection	14 à 16
3.17	Les boucles	17 à 19
3.18	Compléments sur l'affichage	20 et 21
3.19	Sous-programmes et structuration	22 à...

## 3.18. Compléments sur l'affichage

### 3.18.1. RAPPEL DE NOTIONS DIVERSES

#### A. Rappel sur le code ASCII

Pour effectuer avec aisance les exercices du chapitre 18, il est nécessaire que vous ayez à l'esprit les notions sur le code ASCII que nous vous avons enseignées par petites doses en particulier dans LED MICRO n° 1 (§1.3.3 pages 22 et 23) et dans LED MICRO n° 15 (§3.16.10.E pages 20 et 21).

Rappelez-vous en particulier que :

- + Le code ASCII établit une correspondance entre un nombre et un caractère.
- + Le code ASCII est un code à 7 bits (à 7 bits) mais on représente souvent un caractère par un octet (8 bits).
- + Le code ASCII est respecté de façon assez libre par les constructeurs.
- + Le 8<sup>e</sup> bit «des» codes ASCII peut être
  - soit systématiquement un 0
  - soit un bit de parité
  - soit utilisé pour définir des caractères spéciaux (souvent du semi-graphisme).
- + Dans ce qui suit, nous prendrons comme référence le code ASCII utilisé par le BASIC de l'IBM-PC.
- + On peut distinguer :
  - les caractères **éditables** : qui sont des lettres, chiffres ou symbole, dont chacun peut se matérialiser par un dessin intérieur à la grille «caractère». Leur code ASCII est constitué par un nombre plus grand que 31 ;
  - les caractères **non éditables** : qui se traduisent par des «commandes» telles que «effacer l'écran», «revenir au début de la ligne», etc. Si on les envoie sur un autre périphérique (que l'écran), ils peuvent jouer des rôles divers. Le code ASCII de ces caractères est compris entre 1 et 31.

#### B. Rappel de vocabulaire et de notations

**Curseur** : On appelle curseur le petit signe (parfois un trait de soulignement, parfois un pavé clignotant...) qui indique la position de l'écran où va s'afficher le prochain caractère. Parfois cette prochaine position d'affichage n'est matérialisée sur l'écran par aucun signe particulier. On continue cependant à parler de la position de ce curseur invisible.

Au lieu de dire (par exemple) : «... et fait commencer l'affichage en colonne 32», on dit souvent «... et positionne le curseur en colonne 32».

**Mot-clé** : Suite de caractères utilisés par le système, comme instruction ou commande. Exemple : PRINT, SAVE, REM...

**Argument - séparateur** : Dans l'instruction BASIC

```
110 PRINT "A=" ; A , "ET B=" ; B
```

PRINT est un mot clé

"A=" A "ET B=" B sont des arguments

; , sont des séparateurs d'arguments

Pour définir les formats des instructions :  
certains auteurs notent les arguments <arg 1> <arg 2>  
d'autres auteurs différencient les arguments des mots-clés en utilisant des majuscules pour les mots-clés et des minuscules pour les arguments.  
Il ne faut pas se laisser troubler pour si peu de choses !

**Lignes et colonnes :** En «mode texte», l'écran de l'IBM-PC (par exemple) est divisé en 25 lignes et 80 colonnes soit  $25 \times 80 = 2\,000$  «cases». Chacune de ces cases peut afficher un caractère.

Lorsqu'on affiche des tableaux de valeurs numériques (par exemple), on pourra diviser verticalement l'écran en 8 tranches verticales de 10 colonnes, ou 4 tranches verticales de 20 colonnes.

Lorsque cela sera nécessaire pour éviter une confusion, on appellera «colonne caractère» chacune des 80 colonnes contenant 1 case horizontale «colonne-mot» les tranches verticales contenant 4, 6, 8, 10... caractères. On se contentera généralement d'utiliser le mot «colonne» : le contexte précisera de quoi il s'agit.

### C. Rappel sur la ponctuation dans le PRINT

Nous savons que :		voir (ou revoir) :
10 PRINT <arg>	Trois effets : 1° affiche <arg> 2° positionne le curseur en colonne n°1 (= CR) 3° puis descend le curseur d'une ligne (= LF)	LM 6 §3.6.3
20 PRINT	Effets (2) et (3) ci-dessus, c'est-à-dire saute une ligne	LM 6 §3.6.3
30 PRINT <arg 1> ; <arg 2>	affiche <arg 1> et <arg 2> à la suite l'un de l'autre puis passe à la ligne suivante	LM 6 §3.8.2
40 PRINT <arg 1> ; 50 PRINT <arg 2>	Ont le même effet que l'instruction 30. Le ; en fin de ligne supprime les effets (2) et (3) (le CR et le LF)	LM 6 §3.8.3
60 PRINT <arg 1> , <arg 2>	Affiche <arg 1> et <arg 2> dans 2 colonnes-mots successives	LM 6 §3.8.2
70 PRINT <arg 1> , 80 PRINT <arg 2>	Ont le même effet que l'instruction 60	LM 6 §3.8.3

### D. Rappel de quelques fonctions chaînes de caractères

Instruction :	Objet :	voir (ou revoir) :
ASC(X\$)	Renvoie le code ASCII du premier caractère de la chaîne X\$	LM 15 §3.16.10.E
CHR\$(X)	Renvoie le caractère dont le code ASCII est X (exprimé en décimal)	LM 15 §3.16.10.E
LEFT\$(X\$,N)	Renvoie les N caractères les plus à gauche de X	LM 15 §3.16.10.F
LEN(X\$)	Renvoie le nombre de caractères contenus dans X	LM 15 §3.16.10.C
MID\$(X\$,N[,M])	Renvoie une chaîne de M caractères extraite de X\$ et commençant par le N <sup>e</sup> caractère. Si M est omis, renvoie une chaîne extraite de X\$ et contenant tous les caractères à partir du N <sup>e</sup> .	LM 15 §3.16.10.F
RIGHT\$(X\$,N)	Renvoie une chaîne composée des N caractères les plus à droite de X\$	LM 15 §3.16.10.F
STR\$(X)	Convertit la valeur de X en chaîne de caractères	LM 13 §3.15.7 LM 15 §3.16.10.D
VAL(X\$)	Renvoie la valeur numérique de la chaîne X\$	LM 13 §3.15.7 LM 15 §3.16.10.D

### 3.18.2. QUELQUES NOUVELLES FONCTIONS

#### A. SPACE\$

##### Définition

SPACE\$(N) renvoie une chaîne de N espaces.

##### Exercice d'application A56

L'ordinateur demande à l'utilisateur d'entrer un titre :

```
20 INPUT "QUEL EST LE TITRE" ; T$
```

L'utilisateur entre un titre quelconque. Par exemple :

```
T$ = "VIVE NOUS"
```

L'ordinateur affiche ce titre en haut de l'écran, centré au milieu de ses 80 colonnes. N'utilisez que les instructions LEN et SPACE\$.

##### Solution de A56

Pas de difficulté : voir le programme CENTRE21 page ci-contre :

A la ligne 50, on calcule le nombre d'espaces restant dans la ligne soit L1.

A la ligne 60, on divise ce nombre par 2. Soit L2. Si L1 est impair, L2 est plus petit que la moitié de L1 (grâce à INT) et le titre est un peu décalé vers la gauche. C'est n'est pas grave.

##### Exercice d'application A57

Ecrivez un programme qui affiche sur l'écran le dessin représenté par la figure 2 (page ci-contre). Utilisez une boucle et des instructions SPACE\$.

##### Solution de A57

Sur la figure 2, nous avons compté le nombre de cases de chacune des tranches de la ligne N.

La seule (petite) difficulté consiste à compter le nombre d'espaces du milieu (compris entre les 2 étoiles). Soit A ce nombre d'espaces. On a :

$$(N - 1) + (1) + (A) + (1) + (N - 1) = 16$$

soit :  $2N + A = 16$

d'où :  $A = 16 - 2N$

De ce fait, la ligne courante de la boucle s'écrira :

```
50 PRINT SPACE$(N - 1) ; " * " ; SPACE$(16 - 2 * N) ; " * "
```

D'où le programme VICTOI21 (figure 3 page ci-contre).

##### Exercice d'application A58

Ecrivez un programme qui permet d'afficher sur l'écran un nombre quelconque (Z) de Vés, chacun des Vés ayant une hauteur quelconque, comme le représente la figure 4.

##### Solution de A58

Voir le programme VICTOI22 (figure 5 page ci-contre).

#### B. STRING\$

##### Définition

L'instruction STRING\$ peut se présenter sous deux formes.

**Forme 1 :** STRING\$(N,M\$)

Renvoie une chaîne composée de N fois le premier caractère de M\$.

**Forme 2 :** STRING(N,M)

Renvoie une chaîne composée de N fois le caractère dont le code ASCII est M.

##### Exercice d'application A59

Résolvez le même problème que A56, mais en utilisant STRING\$ au lieu de SPACE\$.

##### Solution de A59

Il suffit de remplacer l'instruction 70 du programme CENTRE21 soit par :

```
70 PRINT STRING$(L2," ") ; T$
```

soit par :

```
70 PRINT STRING$(L2,32) ; T$
```

(car 32 est le code ASCII du caractère «espace».)



## **C. POS et CSRLIN**

### **Définition de POS**

$H = \text{POS}(N)$

La fonction POS fournit la position horizontale actuelle du curseur (numéro de colonne-caractère).

N est un argument fictif. On peut le remplacer systématiquement par le chiffre 0.

### **Définition de CSRLIN**

CSRLIN est une abréviation de CURSOR LINE (ligne du curseur).

$V = \text{CSRLIN}$

CSRLIN indique sur quelle ligne de l'écran se trouve actuellement le curseur.

### **Pourquoi ?**

Pourquoi, diable, POS comporte un argument fictif (qui ne sert à rien) et CSRLIN n'en comporte pas ? Autre question du même genre : « Pourquoi chariot s'écrit avec 1 seul r et charrette avec 2 ? ». La seule réponse est « parce que c'est comme ça ! »

### **Utilité de POS et de CSRLIN**

POS et CSRLIN permettent d'effectuer très simplement diverses présentations agréables de textes sur un écran. Nous en verrons des exemples plus tard.

Pour le moment, utilisons POS et CSRLIN pour réviser la ponctuation du PRINT grâce à l'exercice d'application qui suit.

### **Exercice d'application A60**

Le programme CURSEUR21 (figure 1 page ci-contre) conserve en mémoire la position du curseur à différentes étapes de son déroulement dans des variables (L1,C1), (L2,C2), (L3,C3)...

Essayez de deviner ce que fournit l'exécution de ce programme et comparez votre solution avec l'exécution réelle : figure 2.

## **E. SCREEN**

### **Définition**

L'instruction SCREEN est un monde qui permet (par exemple) de passer au mode graphique haute résolution, de revenir au mode texte, de passer au mode graphique couleur, etc.

Nous ne nous intéresserons ici qu'à une forme simple de SCREEN utilisée en mode texte et dont le format est :

$V = \text{SCREEN}(L,C)$

Cette fonction fournit le code ASCII du caractère affiché sur l'écran dans la colonne L et la ligne C.

### **Exercice d'application A61**

Ecrire un programme qui compte le nombre de E majuscules affichés sur l'écran. (Rappel : le code ASCII de E est 69). Ceci constituera un exercice (facile) sur les boucles imbriquées.

Comparez votre solution avec celle que nous proposons page ci-contre (programme COMPTE21, figure 3).

```

10 REM CURSEUR21
20 CLS
30 L1 = CSRLIN : C1 = CSRLIN
40 PRINT "SALUT";
50 L2 = CSRLIN : C2 = POS(0)
60 PRINT "MONSIEUR";
70 L3 = CSRLIN : C3 = POS(0)
80 PRINT "LE PRESIDENT",;
90 L4 = CSRLIN : C4 = POS(0)
100 PRINT "ET CHER CAMARADE";
110 L5 = CSRLIN : C5 = POS(0)
120 PRINT
130 L6 = CSRLIN : C6 = POS(0)
140 FOR I = 1 TO 3 : PRINT : NEXT I
150 L7 = CSRLIN : C7 = POS(0)
160 PRINT "POSITION DU CURSEUR "
170 PRINT "APRES 20 :", L1,C1
180 PRINT "APRES 40 :", L2,C2
190 PRINT "APRES 60 :", L3,C3
200 PRINT "APRES 80 :", L4,C4
210 PRINT "APRES 100 :", L5,C5
220 PRINT "APRES 120 :", L6,C6
230 PRINT "APRES 140 :", L7,C7

```

- 1 Le CLS ramène le curseur en haut et à gauche ligne 1 et colonne 1.
- 2 Le ; laisse le curseur en place, juste après le T : ligne 1 et colonne 6.
- 3 Comme il n'y a ni , ni ; à la fin des arguments du PRINT, le curseur passe au début de la ligne suivante, donc ligne 2 colonne 1.
- 4 On conserve la ligne 2.
- 5
- 6 Ce PRINT tout nu fait passer à la ligne suivante (ligne 3) à son début : colonne 1.

Figure 1

```

SALUT          MONSIEUR
LE PRESIDENT          ET CHER CAMARADE

```

```

POSITION DU CURSEUR
APRES 20 :      1      1
APRES 40 :      1      6
APRES 60 :      2      1
APRES 80 :      2     29
APRES 100 :     2     45
APRES 120 :     3      1
APRES 140 :     6      1
OK

```

Figure 2

```

10 REM COMPTE21
20 Z = 0
30 FOR LIG = 1 TO 25
40   FOR COL = 1 TO 80
50     X = SCREEN(LIG, COL)
60     IF X = 69 THEN Z = Z + 1
70   NEXT COL
80 NEXT LIG
90 PRINT "NOMBRE DE E MAJUSCULES SUR L'ECRAN = "; Z

```

Figure 3

## F. INSTR

### Définition «réduite»

Format :  $V = INSTR(X\$,Y\$)$

Effet : INSTR :

- 1°) explore la chaîne de caractères X\$ et y recherche la première apparition de la chaîne de caractères Y\$.
- 2°) puis renvoie la position à laquelle cette correspondance est trouvée.

### Exercice d'application immédiate A62

Soit A\$ la chaîne de caractères contenant les 26 lettres de l'alphabet rangées dans l'ordre alphabétique.

Ecrire le (petit) programme qui permet de connaître l'ordre de l'une de ces lettres en utilisant une seule instruction INSTR.

Il n'y a aucune recherche à faire : il ne s'agit que de s'assurer que vous avez compris la définition ci-dessus. C'est un entraînement à la lecture des notices constructeurs ne comportant que des listes de format. Solution : programme PASBO21 figure 1 page ci-contre.

### Exercice d'application A63

Reprenez l'exercice de récapitulation R14 (les menus) dont l'énoncé se trouve dans LED MICRO n° 16 page 24 et la solution dans LED MICRO n° 19 page 21 (programme MENU1) mais :

1°) au lieu de 1, 2, 3... placez des sigles mnémoniques CAT, UDE, CMA, etc.

2°) utilisez une instruction INSTR pour explorer la liste de ces sigles.

Cherchez et comparez votre solution avec celle que nous vous proposons.

### Définition «complète»

Le format complet de INSTR est :

$$V = INSTR([N,]X\$,Y\$)$$

N représente une expression numérique dont la valeur (le «décalage») fixe la position du début.

### Exercice d'application A64. Microtraitement de texte

#### Énoncé

Rédigez un programme qui permet d'utiliser votre ordinateur comme une machine à écrire perfectionnée effectuant une justification automatique. Lorsque l'utilisateur tapera un texte quelconque (mais ne comportant ni , ni ; ni :), ce texte s'affichera normalement sur l'écran au fur et à mesure de la frappe.

Lorsque l'opérateur veut changer de ligne, il appuiera sur la touche ENTER et alors, de deux choses l'une :

- s'il appuie sur ENTER avant que l'on ait atteint le 70<sup>e</sup> caractère de la ligne, l'ordinateur supposera qu'il s'agit d'une fin d'alinéa et il imprimera cette ligne telle quelle.
- s'il appuie sur ENTER entre le 71<sup>e</sup> et le 80<sup>e</sup> caractère de la ligne, l'ordinateur imprimera également la ligne mais en la «justifiant», c'est-à-dire en augmentant les espaces entre les mots de façon à ce que la ligne fasse exactement 80 caractères.

Appelez A\$ la ligne que vous taperez et qui s'imprimera. Dès que la ligne A\$ sera imprimée, la chaîne A\$ sera réinitialisée en «chaîne vide» et vous pourrez taper une autre ligne A\$.

Pour indiquer que votre texte est fini, tapez une ligne composée seulement du caractère \$.

**N.B.** Dans LM n° 15 page 17, j'avais proposé cet exercice sous le numéro R10 comme applications des fonctions CHR\$ et LEN. Mais d'une part, c'était trop tôt : vous n'aviez pas encore assez d'expérience sur les boucles et d'autre part j'avais oublié de préciser qu'il fallait s'abstenir d'introduire des , et des ; tant que nous ne connaissions pas l'instruction LINE INPUT (comme me l'a fait remarquer M. J.-C B. de 38210 Tullins).

```

10 REM PASB021
20 A# = "ABCDEFGHJKLMNOPQRSTU"
30 CLS
40 INPUT "VOUS VOULEZ CONNAITRE L'ORDRE DE QUELLE LETTRE"; N#
50 Z = INSTR(A#,N#)
60 PRINT "C'EST LA LETTRE NUMERO "; Z

```

```

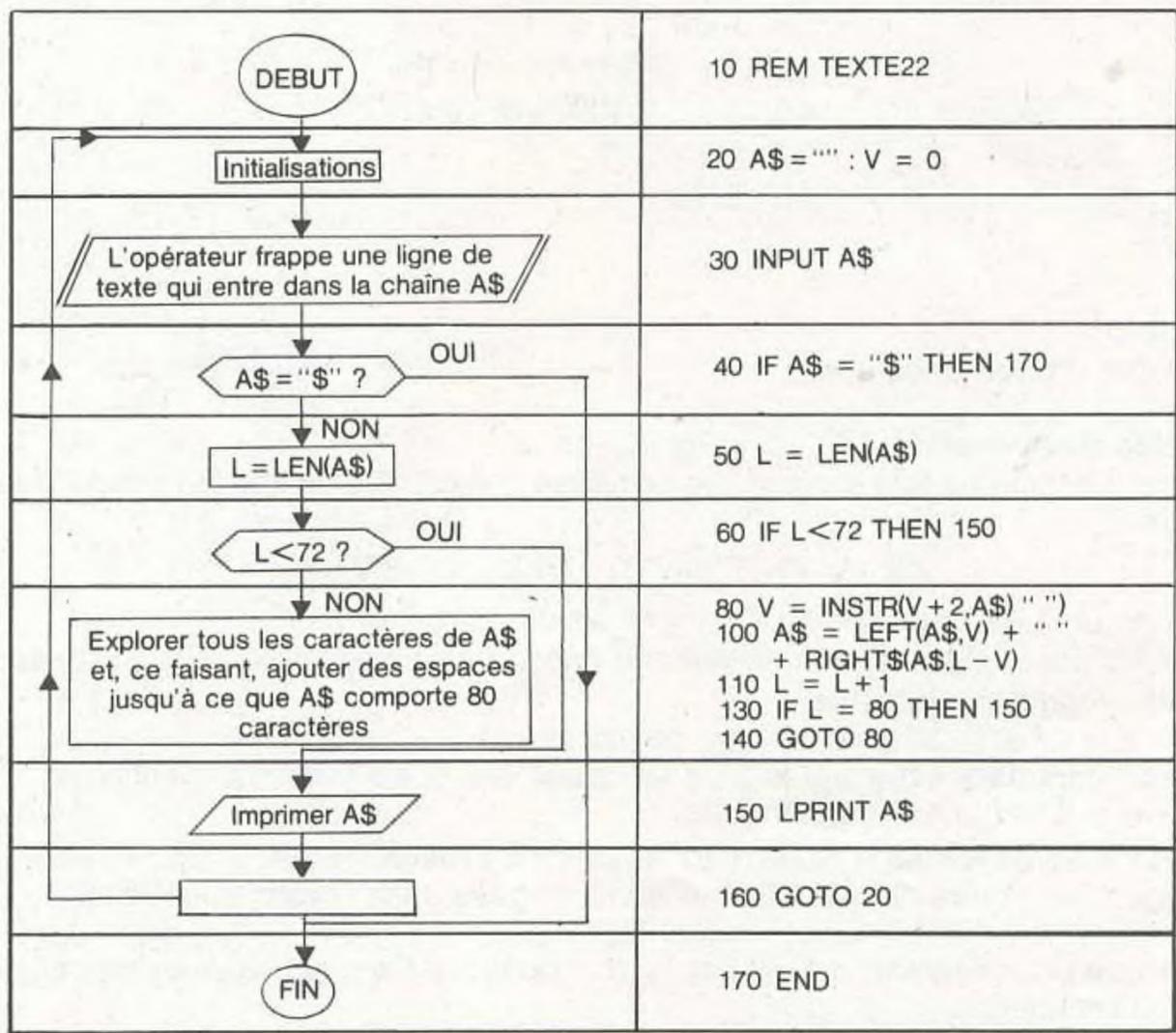
VOUS VOULEZ CONNAITRE L'ORDRE DE QUELLE LETTRE? G
C'EST LA LETTRE NUMERO 7
OK#

```

```

10 REM MENU25
20 CLS
30 T# = ""
40 PRINT T#
50 T# = ""
60 PRINT "POUR :                                     TAPEZ : "
70 PRINT " obtenir le CATalogue des fichiers .....CAT"
80 PRINT " Utiliser un DEssin.....UDE"
90 PRINT " Créer un MANnequin (taille standard ou cliente).....CMA"
100 PRINT " Utiliser les mensuration d'un MANnequin .....UTI"
110 PRINT " Changer les conditions d'EXecution .....CEX"
120 PRINT " executer le dessin TEL quel .....TEL"
130 PRINT " ARREter de travailler .....ARR"
140 INPUT R#
150 A# = "CAT-UDE-CMA-UTI-CEX-TEL-ARR"
160 K = 0
170 K = INSTR(A#,R#)
180 IF K = 0 THEN T# = "JE NE COMPRENDS PAS JE REPETE" :CLS: GOTO 40
190 N = INT (K/4) + 1
200 ON N GOTO 1000,2000,3000,4000,5000,6000,7000

```



### 3.18.3. EXERCICES SUR LES CARACTÈRES ÉDITABLES

#### A. Exercice d'application A65 : tout en majuscules

##### Énoncé

L'utilisateur du programme tape un mot (comportant un nombre quelconque de lettres) en utilisant soit des minuscules soit des majuscules soit un mélange de majuscules et de minuscules. L'ordinateur affiche ce mot en utilisant uniquement des majuscules. Réfléchissez avant de regarder la solution ci-dessous.

##### Pour vous mettre sur la voie

Le code ASCII d'une lettre minuscule est égal au code ASCII de la lettre majuscule correspondante moins 32.

Exemples :    Code ASCII de A = 65                      Code ASCII de a = 97  
                  Code ASCII de B = 66                      Code ASCII de b = 98

Il suffira d'explorer l'une après l'autre chacune des lettres du mot et de remplacer éventuellement le code ASCII d'une lettre minuscule par le code augmenté de 32 : voir programme figure 1 page ci-contre.

#### B. SWAP

##### Définition

L'instruction :

SWAP X\$,Y\$

a pour effet d'échanger le contenu des variables chaînes de caractères X\$ et Y\$,  
L'instruction :

SWAP X,Y

a pour effet d'échanger le contenu des variables numériques X et Y.

##### Exercice d'application immédiate A66

Que donnera l'exécution du programme ECHANG22 ci-dessous :

```
10 REM ECHANG22
20 A$ = " BONNET "
25 B$ = " BLANC "
26 FOR I = 1 TO 4
30     PRINT A$;B$
40     SWAP A$,B$
50 NEXT I
60 END
```

##### Solution :

Voir figure 2 page ci-contre.

##### Exercice d'application A67 : tri alphabétique

L'opérateur entre un mot quelconque composé uniquement de lettres majuscules. Par exemple :

A\$ = "ANTICONSTITUTIONNELLEMENT"

L'ordinateur lit les deux premiers caractères de ce mot et :

– si le 1<sup>er</sup> caractère est avant le second dans l'ordre alphabétique, il les laisse tels quels, sinon il les échange.

Puis, il lit le 2<sup>e</sup> et le 3<sup>e</sup> caractère de ce mot et

– si le 2<sup>e</sup> caractère est avant le 3<sup>e</sup>, il les laisse tels quels sinon il les échange.

Puis, il lit le 3<sup>e</sup> et le 4<sup>e</sup> caractère, etc.

Quand il a fini de lire les N caractères de ce mot, il recommence, N fois. Au bout de N passages, les lettres du mot A\$ sont donc rangées dans l'ordre alphabétique.

**Remarque :** Ce programme n'est pas du tout optimisé. Ce n'est qu'un exercice sur les boucles imbriquées.

### Solution de A65

```
10 REM MAJUSC21
20 CLS
30 INPUT "ENTREZ VOTRE MOT"; E$
40 L = LEN(E$)
50 FOR I = 1 TO L
60   K$ = MID$(E$,I,1)
70   K = ASC(K$)
80   IF K > 96 THEN K = K - 32
90   PRINT CHR$(K);
100 NEXT I
110 END
```

Figure 1

```
ENTREZ VOTRE MOT? cLaUde pULgaR
CLAUDE POLGAR
OK
```

### Solution de A66

```
RUN
BONNET BLANC
BLANC BONNET
BONNET BLANC
BLANC BONNET
OK
```

Figure 2

### Solution de A67

```
10 REM CLASS21
20 CLS
30 INPUT "ENTREZ UN MOT"; M$
40 L = LEN(M$)
50 FOR Z = 1 TO L-1
60   FOR J = 1 TO L-1
70     A$ = MID$(M$,J,1)
80     B$ = MID$(M$,J+1,1)
90     IF B$ < A$ THEN SWAP A$ , B$
100    M$ = LEFT$(M$, J-1) + A$ + B$ + RIGHT$(M$,L- J-1)
110   NEXT J
120 NEXT Z
130 PRINT M$
140 END
```

Figure 3

```
ENTREZ UN MOT? ANTICONSTITUTIONNELLEMENT
ACEEEIILLMNNNNNOOSTTTTU
OK
```

Figure 4

### 3.18.4. Mouvements du curseur avec des CHR\$

#### A. Code ASCII du mouvement du curseur

Le tableau de la figure 1 page ci-contre représente le code ASCII des commandes «mouvement du curseur» selon le BASIC du PC d'IBM.

Ne confondez pas le déplacement du curseur vers la droite (code ASCII 28) avec le caractère espace (code ASCII 32).

Si (voir figure 2) le curseur se trouve sous la lettre B, le «Cursor Right» (ASCII 28) déplace le curseur sous la lettre suivante, mais le «Space» (ASCII 32) déplace le curseur et «écrase» la lettre suivante.

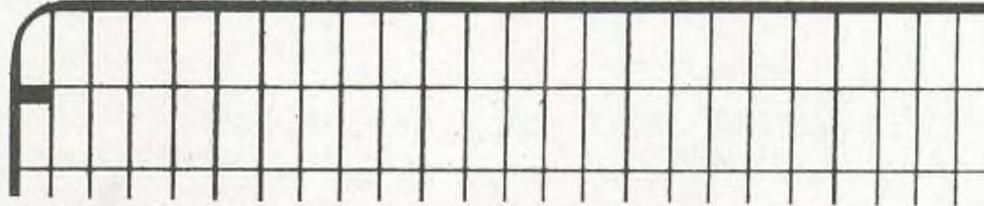
#### B. Exercice d'application A68

##### Enoncé

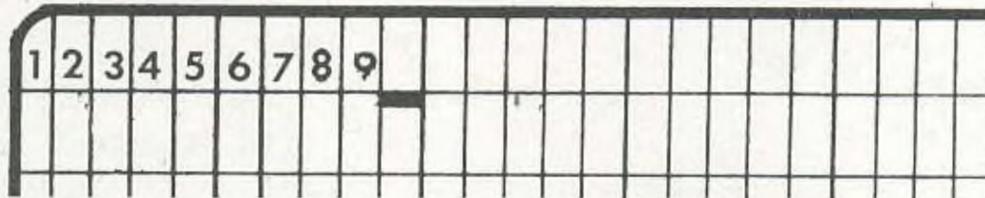
Essayez de deviner ce que donnera l'exécution du programme CURSEU21 (figure 3 page ci-contre).

##### Solution

1°) Après l'exécution de l'instruction CLS, l'écran est effacé et le curseur se trouve en position «HOME».



2°) Après l'exécution de la ligne 20, l'écran se présente ainsi :



Le curseur se trouve dans la case suivant le chiffre 9. Du fait que cette ligne se termine par un ; le curseur reste à cette place. S'il n'y avait pas de ; il passerait au début de la ligne suivant.

3°) Chaque PRINT CHR\$(29) ; fait reculer le curseur d'un pas.

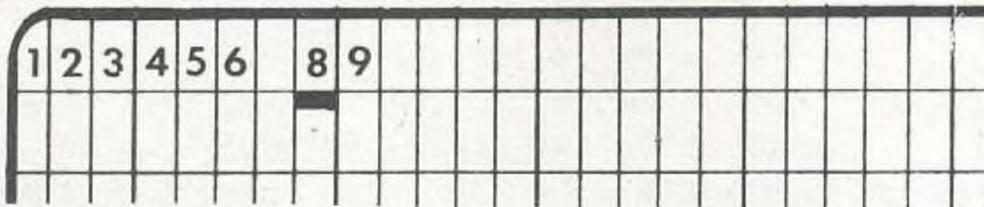
Remarquer le ; . S'il n'y avait pas ce ; le curseur passerait ensuite au début de la ligne qui suit et l'effet de recul du CHR\$(29) serait annulé.

La boucle a pour effet de faire reculer le curseur de 3 pas. A la fin de l'exécution de l'instruction 50, l'écran se présentera ainsi :



4°) L'exécution de l'instruction 70 a pour effet d'écraser le 7 par le caractère espace, et de faire avancer le curseur.

A la fin de l'exécution de l'instruction 70, l'écran se présentera ainsi :



#### C. Exercice d'application A69

##### Enoncé

Essayez de deviner ce que donnera l'exécution du programme CURSEU22 (figure 4 page ci-contre).

##### Solution

L'exécution de la ligne 60 de ce programme réalise les mêmes opérations que le programme CURSEU21 et dans le même ordre. Le résultat est donc le même.

Cet exercice a pour objet essentiel de vous montrer que l'on peut constituer des chaînes de caractères composées de caractères non éditables.

## Caractères «Commande de mouvement du curseur» du BASIC Microsoft

Valeur ASCII (en décimal)	Signification		
	Sigle habituel	en anglais :	en français :
10	LF	Line Feed	Passage du curseur au début de la ligne suivante
11	VT	Home	Retour du curseur en position «départ», c'est-à-dire en haut et à gauche de l'écran
13	CR	Carriage Return	Retour du curseur en début de ligne
28	FS	Cursor Right	Déplacement du curseur d'un pas vers la droite (avancer)
29	GS	Cursor Left	Déplacement du curseur d'un pas vers la gauche (reculer)
30	RS	Cursor Up	Déplacement du curseur d'un pas vers le haut (monter)
31	US	Cursor Down	Déplacement du curseur d'un pas vers le bas (descendre)

Figure 1

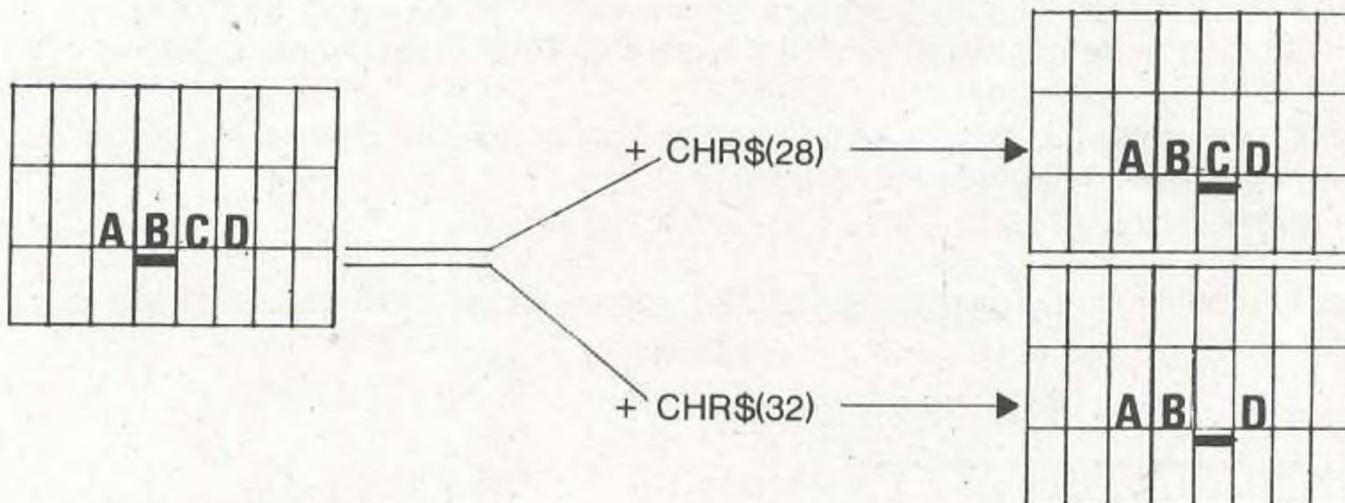


Figure 2

```

10 REM CURSEU21
20 CLS
30 PRINT "1 2 3 4 5 6 7 8 9" ;
40 FOR I=1 TO 3
50     PRINT CHR$(29) ;
60 NEXT I
70 PRINT " " ;
    
```

Figure 3

```

10 REM CURSEU22
20 CLS
30 A$="1 2 3 4 5 6 7 8 9"
40 B$=CHR$(29)+CHR$(29)+CHR$(29)
50 C$=CHR$(32)
60 X$=A$+B$+C$
70 PRINT X$ ;
    
```

Figure 4

#### D. Exercice d'application A70

##### Énoncé

Rédigez un programme dont l'exécution fasse afficher sur l'écran le motif représenté par la figure 1 page ci-contre.

Bien sûr, ce programme devra afficher successivement les lettres A, B, C, D... à des endroits que vous positionnerez avec des CHR\$ portant sur des caractères «mouvement de curseur». Cherchez avant de comparer votre solution avec celle que nous proposons.

##### Une solution commentée

Le programme VAGUE21 (page ci-contre) constitue une solution possible. Il commence par définir deux chaînes de caractères qui ne sont que des assemblages de commandes de mouvement du curseur :

- la chaîne DES\$ qui commandera la descente en diagonale ;
- la chaîne MON\$ qui commandera la montée en diagonale.

Il s'agit d'obtenir :

DES\$ pour I = 0, 1, 2, 3, 4

MON\$ pour I = 5, 6, 7, 8, 9

DES\$ pour U = 10, 11, 12, 13, 14

etc.

On pourrait aiguiller vers DES\$ ou vers MON\$ par une opération «MODULO». Comme ce chapitre est consacré à des exercices sur les chaînes de caractères, nous avons utilisé ces instructions pour extraire le chiffre le plus à droite de I.

Exemple :

I = 13

I\$ = STR(I) = "13"

K\$ = RIGHT\$(I\$, 1) = "3"

K = VAL(K\$) = 3

#### E. Exercice d'application A71

On considère la chaîne de caractères suivante :

CLE\$ = "WQAXSZCDEVFRBGTHYJUKILOMP"

Cette chaîne comporte les 26 lettres majuscules de l'alphabet sans omission ni répétition mais dans le plus parfait désordre. Elle va nous servir de «clé» pour transmettre des messages secrets (pas trop difficiles à déchiffrer d'ailleurs. Nous verrons plus loin comment notre ordinateur peut nous aider à déchiffrer des codes secrets pas trop élaborés).

Nous utiliserons CLE\$ de la façon suivante :

La lettre A (1<sup>re</sup> lettre de l'alphabet) sera transformée en X (1<sup>er</sup> caractère de CLE\$)

La lettre B (2<sup>e</sup> lettre de l'alphabet) sera transformée en G (2<sup>e</sup> caractère de CLE\$)

etc.

Le caractère espace n'est pas modifié : il ne faut pas trop compliquer le travail du décrypteur !

Ecrivez un programme qui effectue ce codage.

#### F. Exercice d'application A72

##### Énoncé

Que donnera l'exécution du programme RECUL22 (page ci-contre). Réfléchissez : il y a un petit piège.

##### Solution

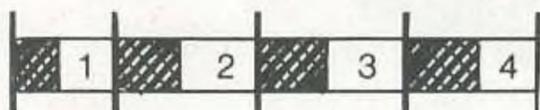
L'exécution du programme RECUL22 fera afficher



Pourquoi ?

On sait que l'ordinateur affiche un espace devant les nombres positifs (et un signe - à cette place devant les chiffres négatifs).

Donc, chaque fois qu'on affiche un nombre, le curseur avance de deux pas.



Les PRINT A\$ ; font reculer le curseur de sorte que chaque «doublet» recouvre en partie le doublet précédent, l'espace du doublet effaçant le chiffre qui vient d'être affiché.

A B C D E F G H I J K L M N O P Q R S T U V W X Y  
 OK

Figure 1

**Une solution de A70**

```

10 REM VAGUE21
20 CLS
30 DES$ = CHR$(28) + CHR$(31)
40 MON$ = CHR$(28) + CHR$(30)
50 FOR I = 1 TO 25
60   I$ = STR$(I): K$ = RIGHT$(I$,1) : K = VAL(K$)
70   IF K < 5 THEN S$ = DES$ ELSE S$ = MON$
80   PRINT CHR$(64+I); S$;
90 NEXT I
100 END

```

Figure 2

**Une solution de A71**

```

10 REM SECRET21
20 CLE$ = "WQAXSZCDEVFRBGTHYJUKILOMP"
30 CLS
40 PRINT "ENTREZ VOTRE PHRASE EN CLAIR"
50 INPUT E$
60 CLS
70 PRINT "VOICI LE MESSAGE SECRET"
80 FOR I = 1 TO LEN(E$)
90   A$ = MID$(E$,I,1)
100  IF A$ = " " THEN S$ = " " : GOTO 130
110  A = ASC(A$) - 64
120  S$ = MID$(CLE$,A,1)
130  PRINT S$;
140 NEXT I
150 END

```

Figure 3

**Une solution de A72**

```

10 REM RECU22
20 R$ = CHR$(29) + CHR$(29)
30 FOR A = 1 TO 10
40   PRINT A;R$;
50 NEXT A

```

Figure 4

### G. Exercice d'application A73

#### Énoncé

Le programme :

```
10 INPUT "NOM - pas plus de 13 caractères" ; NOM$
```

demande à l'utilisateur d'entrer un nom de moins de 13 caractères, de façon très simple... mais peu «conviviale». Il serait préférable que l'ordinateur :

1°) tout d'abord affiche

```
NOM ? _ _ _ _ _
```

2°) Au fur et à mesure que l'utilisateur tape son nom, celui-ci s'inscrit en écrasant les petits tirets :

```
NOM ? J U L E S _ C E _ _ _ _ _
```

3°) Dès que l'utilisateur a fini de taper son nom, ce nom reste apparent seul, sans point d'interrogation ni tiret :

```
NOM JULES CESAR
```

Ecrivez le programme qui résoud ce problème.

Et comparez votre solution avec celle que nous proposons page ci-contre (programme NOMLIM21).

### H. Exercice d'application A74

En vous inspirant :

- d'une part de l'exercice A69 (concaténation de chaînes de caractères éditables et de caractères «mouvement de curseur»);
  - d'autre part de l'exercice A70 (réalisation de dessins en mode texte);
- rédigez un programme constitué de petits modules fournissant :
- dans la chaîne A\$, le dessin de la lettre A composée de X dans une grille de 8 colonnes et 10 lignes (figure 6);
  - dans la chaîne B\$, le dessin de la lettre B dans la même grille;
  - .....
  - dans la chaîne SP\$, le dessin du caractère «espace»;
  - dans la chaîne CF\$, l'équivalent d'un CR + LF (retour au début de la ligne suivante).

L'exécution de ce programme devra permettre à l'utilisateur d'afficher sur son écran 2 lignes de 10 très gros caractères du texte qu'il tapera. Si votre ordinateur comporte moins de 24 lignes, et 80 colonnes, réduisez la taille de la grille pour obtenir le même résultat.

### I. Attention !

Les codes ASCII des commandes de l'écran que nous avons utilisés tout au long de ce chapitre 3.18.4 correspondent à ceux utilisés par le BASIC Microsoft de l'IBM-PC.

Si vous disposez d'un autre ordinateur, il y a beaucoup de chances pour que ce code ne soit pas respecté. Le tableau ci-dessous vous donne une idée des codes utilisés par divers micro-ordinateurs familiaux ou de bureau :

	IBM-PC	Apple II	Commo- dore 64	DA I	Dragon 32	Oric	T07	TRS 80	ZX Spec- trum
Passage à la ligne suivante	10								
Curseur en position HOME	11						30	28	
Efface l'écran				12	12	12	12	12	31
Retour du curseur en début de ligne	13	10	13		13	13			
Curseur un pas en avant	28	21		20	9	9	10	25	9
Curseur un pas en arrière	29	8		19	8	8	11	24	8
Curseur un pas en haut	30	11		17		11	8	27	11
Curseur un pas en bas	31	10		18		10	9	26	10

**Solution de A74**  
**1) LE PROGRAMME**

```

10 REM NOMLIM21
20 CLS
30 NMAX = 15
40 PRINT "NOM = ";
50 PRINT STRING$(NMAX,45);
60 FOR I = 1 TO NMAX
70   PRINT CHR$(29);
80 NEXT I
90 INPUT NOM$
100 FOR I = 1 TO 6
110   PRINT CHR$(28);
120 NEXT I
130 PRINT CHR$(30);
140 L = LEN(NOM$)
150 Z$ = NOM$ + SPACE$(15-L)
160 PRINT Z$

```

Affiche une succession de 13 petits tirets.

45 = le code ASCII du tiret

Fait reculer le curseur jusqu'à le placer devant les petits tirets.

Affichage d'un point d'interrogation et attente de la frappe de l'utilisateur.

Une fois que l'utilisateur a appuyé sur ENTER, le curseur passe en colonne 1 ligne 2.

On replace donc le curseur sur le ? en l'avançant de 6 colonnes et en le remontant de 1 ligne.

Ecrase le point d'interrogation et affiche le nom - seul - en supprimant les tirets.

**2) ASPECT DE L'ECRAN**

Après l'exécution de 90 et avant la frappe de ENTER

NOM = ? JULES CESAR--

En fin de programme :

NOM = JULES CESAR  
 OK

**3) POSITION DU CURSEUR**

Après 40 :	N O M =	█
Après 50 :	N O M =	-----█
Après 80 :	N O M =	█-----
Après 90 :	N O M =	? J U L E S C E S A R -- █
Après 120 :	N O M =	? J U L E S C E S A R -- █
Après 130 :	N O M =	█? J U L E S C E S A R --
Après 160 :	N O M =	J U L E S C E S A R

## Corrigé de l'exercice R22 : le code Morse

La plupart des réponses que j'ai reçues à cet exercice utilisent des DATA pour stocker la suite des • et des – de l'alphabet Morse. Plusieurs solutions utilisent les possibilités sonores de leur ordinateur favori. Nous nous contenterons ici de vous proposer une solution ne faisant appel qu'à ce que nous avons appris. (Voir LED MICRO n° 19 page 29, bas de la 3<sup>e</sup> colonne).

Au lieu de fournir le listing d'une solution et de la commenter, je vais essayer d'imaginer le «processus de réflexion» qui permet d'aboutir à ce petit programme.

Il faut tout d'abord apprendre le code Morse à l'ordinateur. Pour ce faire, on va entrer la suite des lettres de l'alphabet Morse dans une chaîne de caractères M\$ d'une façon telle que l'on puisse y retrouver facilement le 3<sup>e</sup> caractère ou le 7<sup>e</sup> caractère ou le...

On peut imaginer ceci de plusieurs façons :

### 1<sup>re</sup> façon :

On peut entrer directement la suite des caractères de l'alphabet Morse en les séparant par des espaces :

M\$ = ".- -... -.-. -. . .-- etc...

Pour aller chercher le caractère correspondant à la 3<sup>e</sup> lettre de l'alphabet, il suffira d'extraire les points et traits compris entre le 3<sup>e</sup> et le 4<sup>e</sup> caractère espace.

### 2<sup>e</sup> façon :

On peut également entrer le code Morse dans M\$ en complétant chacune des lettres Morse par des espaces de façon à ce que chaque lettre occupe systématiquement 4 caractères :

M\$ = ".- -... -.-. -.-. -.-. etc..."

A B C D E

Pour aller chercher le 7<sup>e</sup> caractère (par exemple), il suffira de prendre les caractères compris entre les places  $(7 - 1) / 4$  et  $7 * 4$  de M\$. C'est très facile : on recherche la position de la chaîne de points et de traits et on divise ce nombre par 4 pour obtenir le rang alphabétique.

### 3<sup>e</sup> façon :

On peut séparer les caractères Morse en les faisant précéder de leur caractère écrit en clair :

M\$ = "A.- B-... C-.-. D-. E. ..."

Pour aller chercher le caractère correspondant à la lettre K (par exemple), il suffira de récupérer les points et traits qui suivent le caractère K dans M\$.

### 4<sup>e</sup> façon :

Le Samideano René Sipra rentre les caractères de l'alphabet Morse en remplaçant les points par le chiffre 1, les traits par le chiffre 2. Il complète chacun des caractères du Morse par des chiffres 0. Il obtient ainsi une chaîne M\$ de structure identique à celle de notre «2<sup>e</sup> façon». Il lui est ainsi facile d'éliminer les caractères supplémentaires (les 0 pour lui) : il suffit de faire un VAL.

Toutes ces méthodes sont équivalentes. Il n'y a qu'à choisir celle dont la résolution est la plus facile.

Ceci est très général : l'analyse d'un problème réel consiste beaucoup plus à chercher non pas la solution à un problème précis, mais à chercher le problème qui permet d'aboutir à une solution simple.

Nous vous proposons une solution sur la page ci-contre. Elle est de la «2<sup>e</sup> façon» : les codes Morse sont complétés pour entrer dans un cadre de 4 caractères. On a fait précéder chaque code de la lettre en clair. On n'utilise pas cette lettre pour le décodage mais simplement pour rendre le programme plus lisible.

```

10 REM MORSE21
20 A$ = "A .-B-...C-.-.D -..E .F...G --.H....I .."
30 A$ = A$ + "J.---K --L-..M --N -.O ---P.---Q---R .-."
40 A$ = A$ + "S ...T -U ..-V...-W .--X-...-Y-.-Z---."
50 CLS
60 INPUT "ENTREZ VOTRE MOT";E$
70 FOR L = 1 TO LEN(E$)
80   K$ = MID$(E$,L,1)
90   K = ASC(K$) - 64
100  S$ = MID$(A$,5*(K-1)+2,4)
110  PRINT S$;" ";
120 NEXT L

```

Au lieu d'écrire tout le contenu de A\$ sur une seule ligne, ce qui serait peu lisible, on obtient le même résultat par deux concaténations.

On va travailler sur chacune des lettres de E\$, l'une après l'autre.

On prend une lettre de ce mot E\$

On détermine son rang dans l'alphabet : pour A, K = 1 - pour B, K = 2, etc.

On va chercher dans A\$ le code Morse correspondant à ce rang.

On affiche les caractères Morse l'un après l'autre en les séparant par deux espaces.

```

ENTREZ VOTRE MOT? BONJOUR
-...  ---  - .  .---  ---
OK

```

### Exercice de récapitulation R30 : trouver la phrase

Il s'agit de réaliser un programme reproduisant un jeu télévisé dirigé par un acteur de cinéma que j'aime bien. L'arbitre entre dans l'ordinateur un dicton quelconque que l'ordinateur appellera A\$.

Par exemple :

A\$ = "QUI AIME BIEN CHATIE BIEN"

Ce mot s'efface aussitôt et est remplacé par

- - - - -

c'est-à-dire une succession de tirets et d'espaces. Chaque tiret remplace une lettre et chaque espace représente l'espace réel séparant les mots.

Le premier joueur choisit une lettre. S'il choisit la lettre E (par exemple), la ligne de tirets est remplacée par :

- - - - - E - - E - - - - - E - - E -

où les E s'affichent à leur place.

Si cela suffit pour que le premier joueur devine quel est le dicton, il a gagné.

Sinon, c'est au deuxième joueur de proposer une lettre. S'il propose la lettre I, la ligne précédente, est remplacée par :

- - I - I - E - I E - - - - I E - I E -

Si cela suffit pour que le deuxième joueur devine le dicton, il a gagné, sinon, c'est au tour du premier joueur de proposer une autre lettre.

Etc.

### Exercice de récapitulation R31 : écrivez en grand

En vous inspirant :

- d'une part de l'exercice A69 (concaténation de chaînes de caractères éditables et de caractères «mouvement de curseur»);

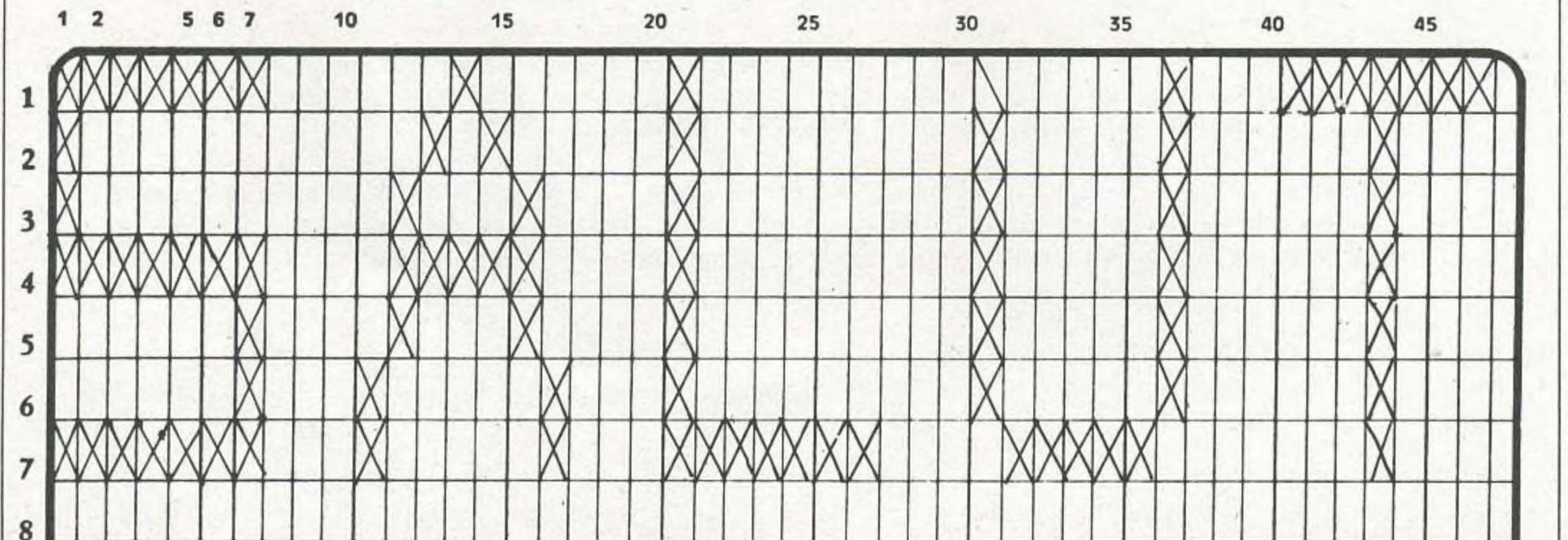
- d'autre part de l'exercice A70 (réalisation de dessins en mode texte);

rédigez un programme constitué de petits modules fournissant :

- dans la chaîne A\$, le dessin de la lettre A composée de X dans une grille de 7 colonnes et 9 lignes;
- dans la chaîne B\$, le dessin de la lettre B dans la même grille;
- .....
- dans la chaîne SP\$, le dessin du caractère «espace»;
- dans la chaîne CF\$, l'équivalent d'un CR + LF (retour au début de la ligne suivante).

L'exécution de ce programme devra permettre à l'utilisateur d'afficher sur son écran 2 lignes de 10 très gros caractères du texte qu'il tapera. Si votre ordinateur comporte moins de 24 lignes et 80 colonnes, réduisez la taille de la grille pour obtenir le même résultat.

Si votre ordinateur comporte des «pavés» comme caractères graphiques (voir LED MICRO n° 15 page 21), utilisez-les de préférence aux «X».



### Exercice de récapitulation R32

Toutes les nouvelles fonctions chaîne de caractères : SPACE\$, STRING\$, INSTR, SWAP sont bien commodes mais n'existent pas sur tous les ordinateurs. Ecrivez des petits modules qui permettent de s'en passer.

### Exercice de récapitulation R33 : la pagaille de plus en plus vite

Considérons la chaîne de caractères A\$ contenant toutes les lettres de l'alphabet :

A\$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

A partir de cette chaîne A\$,

On tire au hasard un nombre compris entre 1 et 26, soit par exemple 7, qui correspond à la lettre G. On tire ensuite une autre lettre au hasard, par exemple la lettre T, etc., jusqu'à obtenir une chaîne de caractères telle que :

B\$ = "GTCNABQJZPFRIYMKXWDUELSOH"

B\$ contenant les 26 lettres de l'alphabet sans omission ni répétition.

Nous vous demandons d'écrire **deux** programmes fournissant une chaîne B\$.

- Pour le premier programme, ne vous cassez pas la tête, vous tirez au hasard une lettre parmi 26, puis une autre lettre parmi 26, en vérifiant qu'elle n'a pas déjà été choisie, puis... etc.

- Mais cette solution n'est pas très élégante : lorsqu'on a déjà écrit une vingtaine de lettres, l'ordinateur met un temps fou à éliminer les lettres déjà tirées. Dans le deuxième programme que vous écrivez, l'ordinateur **ne** devra **pas** avoir besoin de vérifier que chaque nouvelle lettre tirée n'a pas déjà été utilisée : il ne tirera que des lettres restantes.

Ceux d'entre vous qui ont un ordinateur pourront comparer les temps d'exécution de ces deux programmes. Communiquez-moi vos résultats. Merci.

### Exercice de récapitulation R34 : recherche des codes inconnus

Supposons que vous ayez égaré la liste des codes ASCII des caractères non éditables. Ecrivez un programme qui vous permet de retrouver cette correspondance.

C'est facile : il suffit d'explorer les codes ASCII de 1 à 31... mais il faut remédier aux mauvaises surprises (par exemple : un CLS au milieu qui efface tout ce que vous avez affiché).

Continuez votre exploration. Certains BASICs ne connaissent pas le CLS. Mais on peut généralement se fabriquer un CLS en utilisant le code ASCII du caractère non éditable convenable. Par exemple, avec le SANCO 8001, on pourra remplacer le CLS manquant par un :

```
PRINT CHR$(26)
```

Avec d'autres systèmes, il faudra envoyer successivement deux caractères de commande de l'écran :

```
PRINT "CHR$(A) + CHR$(B)"
```

avec : A = code ASCII de la commande «effacement de l'écran»

B = code ASCII du retour du curseur en position HOME.

Notez les résultats : vous en aurez besoin dans la première leçon de PASCAL, car le PASCAL ne connaît pas le CLS. (Rassurez-vous : le PASCAL, ce n'est pas pour tout de suite).

## Le Samideano va encore plus loin

Fidèle à son habitude, M. Sipra nous envoie une solution qui fait plus que ce que je demande : la traduction alphabétique → Morse de son inverse. Je manque de place pour publier ses explications. En vertu du principe de Rockefeller, cela sera un avantage : essayez de le comprendre tel quel. Ce sera votre première application du cas (fréquent !) où on vous demandera de reprendre le programme d'un collègue qui a quitté la Société il y a deux ans sans laisser aucune documentation.

```
10 code$="0012211121210211000111210221111100111222021212110022002102221221221201
2101110002011211120122211221222211"
20 t$="-":p$="."
30 CLS:PRINT "Voulez-vous la TRADUCTION d'un message en Morse ,"
40 PRINT "ou le DECODAGE d'un message redige en Morse ?"
50 INPUT "TRADUCTION = 1          DECODAGE = 2          "r
60 IF r<>1 AND r<>2 THEN 30
70 CLS:ON r GOTO 100,300
100 PRINT "Entrer votre message":PRINT
110 INPUT message$:message$=UPPER$(message$)          'tout en MAJUSCULES
120 PRINT #8,message$                                  'impression message
130 PRINT #8:PRINT #8,"Est code en MORSE ==>"
140 PRINT #8:PRINT #8
150 lo=LEN(message$)                                  'longueur du texte
160 i=1:CLS                                           'Possibilite FOR...NEXT
170 a$=MID$(message$,i,1)                              'lettre Par lettre
180 IF a$=" " THEN PRINT #8:PRINT #8:GOTO 260         'couPure des mots
190 n=ASC(a$)-65:m=4*n+1                               'n est le numero de la lettre
200 code=VAL(MID$(code$,m,4))                          'val Pour eliminer les 0
210 trad$=STR$(code):l=LEN(trad$)                     'Place dans une var. chaine
220 FOR j=2 TO l                                       '2 evite l'esPace-signe
230     k=VAL(MID$(trad$,j,1))                          'k=1 ou k=2
240     IF k=1 THEN PRINT #8,p$: ELSE PRINT #8,t$:    'Point ou trait
250 NEXT j:PRINT #8," ";                               'on recommence...
260 i=i+1:IF i<=lo THEN 170                            'ici aussi...
270 PRINT #8:GOTO 30
300 PRINT "Utilisez - et . Pour le texte":PRINT
310 PRINT "[esPace] Pour seParer les LETTRES":PRINT
320 PRINT "[esPace] suivi de / Pour seParer les MOTS et Pour TERMINER le texte"
330 PRINT:INPUT texte$
340 lo=LEN(texte$):message$=""                          'longueur du texte
350 CLS:i=1
360 b$=""
370 a$=MID$(texte$,i,1)
380 IF a$="/" THEN 490
390 IF a$=" " THEN 420
400 IF a$="." THEN b$=b$+"1" ELSE b$=b$+"2"
410 i=i+1:GOTO 370
420 IF b$="" THEN 510
430 FOR j=1 TO 26
440     c$=MID$(code$,4*j-3,4)
450     IF VAL(b$)=VAL(c$) THEN 470                    'identification lettre
460 NEXT j
470 message$=message$+CHR$(j+64)                       'Par code ASCII
480 i=i+1:GOTO 360
490 IF i=lo THEN 520
500 message$=message$+" "
510 i=i+1:GOTO 360
520 PRINT #8,"Message code ==> ";texte$
530 PRINT #8:PRINT #8,"Message decode ==> ";message$
540 GOTO 30
```

## ERRATUM

### Une erreur... qui tombe bien

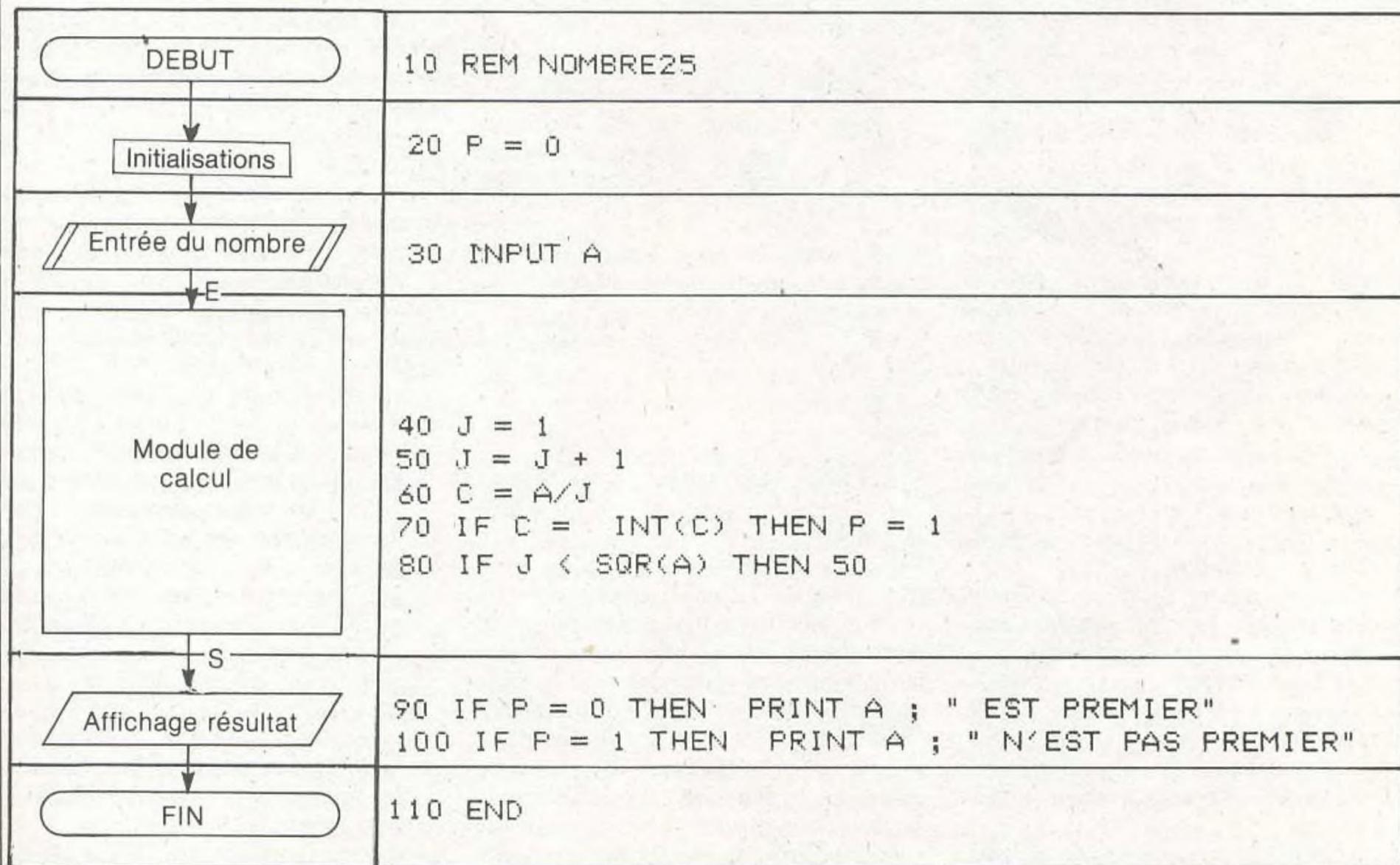
Dans le numéro 19 de LED MICRO, une erreur de transmission entre notre service de montage et moi-même n'a pas permis de mettre parfaitement en évidence l'idée de base de l'exercice d'application A51 (pages 8 et 9).

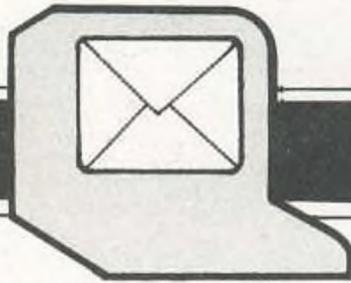
Les figures ci-dessous reproduisent ce qui aurait dû être les figures de la page 9 de LED MICRO n° 19. La différence est minime : il ne s'agit que de préciser le contenu des 6 modules du programme NOMBRE25. Mais elle porte sur un point fondamental :

- + le (vilain) programme NOMBRE3 comportait 2 «pseudo-modules»
  - un module de calcul (lignes 30 à 70) comportant deux points de sortie (ligne 60 et ligne 70)
  - Un module d'affichage (lignes 90 et 100) comportant deux points d'entrée (ligne 90 et ligne 100).
- + Le (joli) programme NOMBRE25 est constitué de «vrais» modules ne comportant chacun qu'un seul point d'entrée et un seul point de sortie.

La décomposition d'un programme en «vrais» modules est la base de toute la structuration des programmes. Remercions le bon Dieu qui, en nous envoyant cette (petite) erreur nous a donné l'occasion d'insister sur ce point fondamental qui sera à la base de la «structuration des programmes» (prochain chapitre 3.19).

Dans «notre» structuration des programmes, vous aurez le droit d'utiliser des GOTO à l'intérieur d'un module (voir exercice A71 programme SECRET21) mais jamais d'un module à l'autre. Les adeptes du PASCAL sont plus intransigeants !





□ *Fidèle lecteur de votre revue, j'aurais aimé pouvoir apprendre à construire un programme de confection d'emploi du temps scolaire :*

— d'abord celui d'une classe à un cours, par exemple cours moyen 1<sup>re</sup> année, les paramètres me paraissant être seulement :

• les 6 tranches horaires (de 9 à 12 h et de 14 h à 17 h)

• les 4 jours et 1/2 de classe (du lundi matin au samedi midi)

• les différentes disciplines réglementairement pratiquées, avec : leurs horaires hebdomadaires imposés ; la répartition de ces horaires dans la semaine ; les critères pédagogiques de choix (du genre «il vaut mieux Calcul le matin et Ed. Physique l'après-midi») ;

— puis celui d'une classe à 2 cours (par ex. CMI et CM2)

— puis celui d'une classe unique (où travaillent des enfants de 5 à 11 ans).

Mais le summum pourrait être un programme organisant l'activité pédagogique dans un collège ou un lycée et qui devrait tenir compte :

— des différentes classes (6<sup>e</sup>A, 6<sup>e</sup>B... 5<sup>e</sup>A, 5<sup>e</sup>B..., etc.)

— des salles disponibles (spécialisées ou pas : trav. prat., gymnase...)

— des horaires réglementaires de chaque discipline

— de la mixité (garçons, filles) si elle est refusée pour certaines activités

— des horaires imposés aux différentes catégories de profs (P.E.G. certifiés...)

— des horaires de fonctionnement du collège (liés aux transports scolaires)

— des regroupements de classes, des contraintes liées à la rénovation

*pédagogique (réforme Legrand) avec nécessité d'heures de concertation...*

— des souhaits des différents professeurs quant à l'aménagement de leurs propres horaires (éloignement du domicile et horaires des trains, limitation maximale des «trous» dans l'horaire du professeur... comme dans celui de l'élève sinon horaires des permanences avec problème de locaux et de surveillance)

— de l'intérêt et de l'agrément des élèves (internes ou externes, temps normal de repos après le repas

— pas de cours à 13 h — foyer socio-éducatif, pas 3 h de cours d'affilée dans une même discipline. — etc.

*Ce programme serait donc complexe. Mais le professionnel qui en serait l'auteur répondrait à un besoin que ressentent avidement des dizaines de milliers d'enseignants.*

*En existe-t-il sur le marché ? Votre revue pourrait-elle aborder ce sujet.*

*R.C. 47800 Miramont-de-Guyenne*

Merci pour votre lettre qui me fournit une excellente illustration de l'idée (classique) suivante :

Programmer en Basic, cela s'apprend facilement — et on s'en lasse vite. La vraie difficulté (et le vrai plaisir) réside dans l'analyse des problèmes. Votre lettre constitue (presque) une analyse d'un problème réel de programmation. Bien sûr, notre but est d'arriver à analyser des problèmes du genre de celui que vous posez... mais la simple lecture de votre lettre montre, d'une part que nous ne sommes pas encore prêts à rédiger un tel programme,

d'autre part qu'avant de le rédiger il faudra expliciter les diverses conditions que vous citez : sacré travail !

J'ajouterai deux choses :

— Une fois qu'on a épuisé les joies de la programmation élémentaire en Basic, il y a toute une collection de problèmes difficiles (donc passionnants) qui ne sont pas à proprement parler de l'analyse (recherche d'algorithmes performants, optimisation...). Ceci déborde du cadre de LED MICRO.

— Un de mes amis est justement en train de réaliser le programme que vous souhaitez (avec un beau graphisme couleur, en interactif et avec quelques astuces simplificatrices que je ne suis pas autorisé à vous communiquer pour le moment). Je vous tiendrai personnellement au courant quand il sera commercialisé.

□ *A la suite des récents reproches de certains de vos lecteurs : lenteur de votre cours ou importance excessive donnée aux corrigés des exercices, je voulais à mon tour vous donner mon humble avis.*

*J'ai découvert LED MICRO par hasard à la vitrine d'un marchand de journaux, c'était le n° 13. Votre revue m'a plu et j'ai aussitôt commandé les numéros précédents que vous m'avez fait parvenir avec une promptitude remarquable. Il m'a fallu mettre les bouchées doubles pour combler mon retard, mais ce ne fut pas une corvée, au contraire, car j'ai pris un grand plaisir à étudier votre cours qui me paraît excellent, clair, précis, parfaitement gradué, ce qui est important lorsque l'on travaille seul, sans le secours d'un professeur. Quant à vos exercices, toujours bien adaptés au cours, ils ne seront jamais trop nombreux. J'applaudis à*

*voire intention d'intensifier la partie «Exercices de récapitulation» et leur développement. Les exercices obligent à faire l'effort nécessaire pour bien assimiler le cours, pour contrôler des connaissances nouvellement acquises et les graver durablement dans son esprit. J'ajouterai que je prends plaisir à traiter les questions mathématiques que vous proposez ; elles m'obligent à rafraîchir de vieux souvenirs quelque peu estompés.*

*R.C. 41000 Blois*

Merci pour votre lettre. Les lettres de compliments que je reçois me font toujours plaisir (envoyez m'en beaucoup !) mais j'évite de les publier. J'utilise cependant votre texte aujourd'hui comme «bouclier» ou comme «contre-feu» pour m'aider à justifier l'exceptionnelle surabondance d'exercices d'application dont le présent cours est rempli.

En ce qui concerne les exercices de récapitulation je pense que vous êtes trop indulgent avec moi :

— Leur but était, certes, de faire faire des révisions d'anciennes notions, mais surtout de commencer à apprendre (par la pratique) à analyser un problème. Je les ai lancés trop tôt. J'aurais dû attendre que les lecteurs sachent structurer leurs programmes.

— Certains sont trop longs.

j'envisage de les continuer «avec intensité», dès que nos lecteurs sauront manipuler le GOSUB (futur n° 22).

□ *Le jour où j'ai eu connaissance de votre revue, je me suis dépêché d'acheter le n° 11 et de commander les 10 premiers qui étaient déjà parus. Je pensais que votre revue était ce qu'il me fallait. Depuis, nous voilà au n° 18 et je fais le bilan : je suis complètement «largué». Je connais des tas d'instructions, des tas de commandes mais*

*je suis dans un trou noir. Je ne sais pas comment les joindre les unes aux autres de façon à réaliser un programme un peu compliqué. Alors, lorsque je lis le «Courrier des lecteurs», et surtout les solutions (certainement brillantes pour les débutants), j'ai le sentiment que je ne comprends rien à rien et que je suis un abruti.*

*Cependant, à y regarder de plus près, force m'est de constater que ces solutions sont la plupart du temps signées des mêmes personnes et ceci d'un numéro à l'autre.*

*J'en conclus que ces «débutants» n'en sont peut-être pas et qu'ils pratiquent l'informatique depuis plus longtemps qu'ils ne le laissent entendre et, surtout, qu'ils la pratiquent d'une autre façon qu'en amateurs.*

*Alors Mr Polgar, je me permets de vous faire une suggestion : consultez vos lecteurs par questionnaire pour savoir combien y-a-t-il d'idiots comme moi qui n'arrivent plus à vous suivre. Si nous ne sommes que quelques-uns dans mon cas, tant pis.*

*Mais s'il s'avérait que nous sommes une majorité, par pitié, dites à ces pseudo-amateurs de laisser LED MICRO aux vrais amateurs et adaptez dans vos cours une progression plus adaptée aux gens qui, comme moi, n'ont jamais touché un clavier que des yeux avant d'en acheter un pour suivre les cours de LED MICRO.*

*A.P. 84100 Orange*

Commençons par examiner votre premier thème : «Je connais des tas d'instructions mais je ne sais pas comment les joindre les unes aux autres de façon à réaliser un programme un peu compliqué».

Voici ma réponse :

Ne cherchez pas pour le moment à réaliser des programmes compliqués. Contentez-vous de travailler les «exercices d'application» qui sont de tout petits modules. Bientôt vous verrez comment assembler ces petits modules (grâce à la «structuration» des programmes).

Rédiger des programmes compliqués, c'est l'idéal Schadock (pourquoi faire simple alors qu'il est plus facile de faire compliqué). Un programme bien structuré est peut-être long, mais il est toujours simple et la rédaction d'un tel programme est toujours très facile.

Dans votre deuxième thème : «Dites aux faux-amateurs de laisser LED MICRO aux vrais «débutants», vous reprenez sous une autre forme ce que je constatais dans LED MICRO n° 17 page 5 («quand les élèves sont trop forts»).

Je n'ai pas l'intention de refouler les élèves «trop forts». Ils apportent des idées nouvelles, dont nous profitons tous (moi le premier !). Mais je pense que vous allez être satisfaits par la solution que prépare notre Directeur : créer, en parallèle avec LED MICRO, une revue plus spécialement destinée aux fortiches qui veulent optimiser leurs programmes.

En conclusion (et ceci est valable pour tous les lecteurs débutants qui éprouvent les mêmes difficultés que vous) :

- 1) N'étudiez que les exercices d'application pour le moment.
- 2) D'ici quelque temps, vous serez vous aussi devenu un fortiche, et vous pourrez vous attaquer aux exercices de récapitulation. C'est le principe de la «deuxième vague» de (l'excellente) méthode ASSIMIL.
- 3) Signalez-moi de façon concrète les points que vous ne comprenez pas (dans le cours, pas dans les exercices de récapitulation). Je m'efforcerais de vous répondre.

# GESTION DE FICHIERS

Nous allons étudier la suite du logiciel intitulé «Gestion de fiches». Nous allons apprendre à dupliquer, à trier et à détruire des fichiers. Ainsi, nous aurons un ensemble de fonctions suffisamment complet pour une utilisation courante. Les listings publiés ne correspondent qu'aux lignes de programmes qui ont été publiées au mois de mars et qui doivent être modifiées, ou les lignes supplémentaires (nouvelles fonctions).

## 1. LES MODIFICATIONS

### Sur le menu

Auparavant, le menu principal donnait directement accès à une fonction en agissant sur le numéro correspondant. Les nouvelles fonctions font apparaître un autre menu. Ainsi, en agissant sur 7 (Arrangement d'ensemble), nous voyons apparaître un menu nous laissant le choix entre un tri ou une duplication. Pour diminuer les manipulations, il est possible de choisir directement entre la duplication (D) et le tri (T) à partir du menu principal.

Par exemple, une personne connaissant le logiciel et qui désire trier des fiches pourra directement agir sur la touche T pour avoir accès à cette fonction, alors qu'une personne ne le connaissant pas, agira d'abord sur la touche 7 puis sur la touche T.

### Sur consultation d'une fiche (accès direct)

Pour accéder directement à une fiche, nous n'avons que la possibilité d'indiquer le numéro de la fiche. Cela n'est pas pratique si nous ne connaissons pas l'emplacement de la fiche désirée. Une recherche par ordre alphabétique est beaucoup plus pratique.

Il est maintenant possible de faire cette recherche. En 509, le programme demande à l'opérateur si celui-ci veut accéder à une fiche par son numéro (n) ou en imposant des conditions sur le contenu d'une zone (z). Dans ce dernier cas, nous exécutons le sous-programme de recherche d'une fiche que nous examinerons plus loin.

### Sur le changement d'un type de fiche

Dans les nouveaux sous-programmes, nous sommes souvent amenés à connaître la forme d'une fiche sans pour autant avoir besoin de l'afficher à l'écran. Un sous-programme plus rapide, ne chargeant que la structure d'un type de fiche est devenu nécessaire. Les lignes 619 à 625 ont ce rôle. Nous pouvons constater qu'elles ne diffèrent de la ligne 604 à la ligne 611 que par la ligne 610.

## 2. LES NOUVELLES FONCTIONS

### Destruction de fichier (700 → 734)

De la ligne 701 à la ligne 709, nous faisons apparaître un menu qui demande la nature du fichier à détruire. Seule la destruction d'un ensemble de fiches ou d'un type de fiche est permise.

La destruction de *cfiche* et *censemble* est interdite car elle désorganiserait la disquette. Le programme ne serait plus capable de savoir quels sont les fichiers restants.

### Destruction d'un type de fiche (710 → 719)

Après avoir demandé le nom du type de fiche (710), nous demandons la confirmation de cette destruction (711 → 714). En 715, nous plaçons le pointeur du fichier *cfiche* sur l'enregistrement contenant le nom du type de fiche puis nous décalons la suite du fichier de 1 enregistrement (car  $i = 1$ ) grâce au sous-programme destruction partiel.

Après avoir modifié le fichier *cfiche*, nous effaçons le fichier portant le nom du type de fiche grâce à la commande SCRATCH (ligne n° 718).

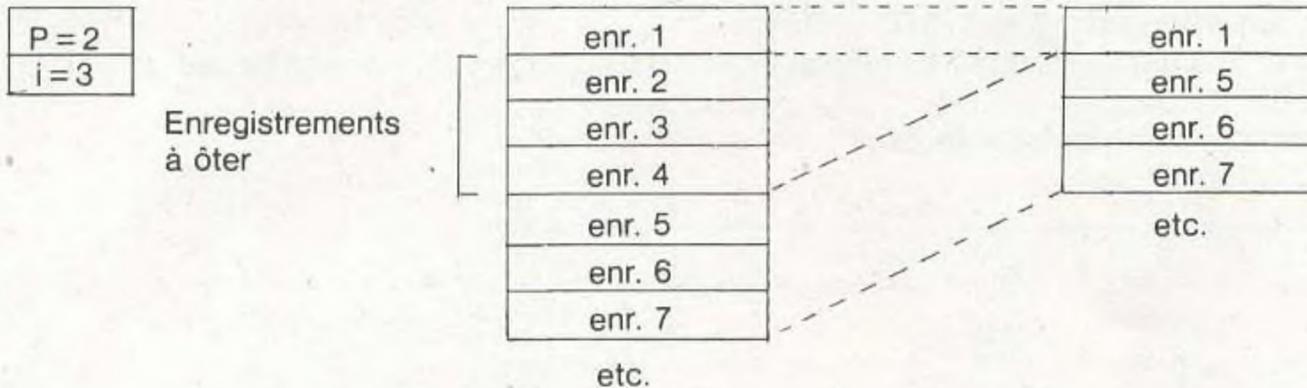
### Destruction d'un ensemble de fiches (725 → 734)

La seule différence avec la destruction d'un type de fiche est l'effacement des deux enregistrements (nom de l'ensemble de fiche, nom du type de fiche correspondant dans le fichier *censemble* ( $i = 2$  à la ligne 732).

### Destruction de $i$ enregistrements à partir du $P^{ème}$ (751 → 763)

Le but de ce sous-programme est d'éliminer des enregistrements au milieu d'un fichier.

Par exemple : l'élimination de trois enregistrements à partir du deuxième.



De 754 à 757, nous chargeons les 23 premiers enregistrements suivant ceux à détruire.

De 758 à 761, nous remettons les 23 enregistrements au bon endroit, compte tenu des enregistrements à détruire.

Si nous ne sommes pas arrivés à la fin de l'enregistrement, nous recommençons à partir du  $23 + P^{\text{ème}}$  enregistrement (763). Il est à noter que le fichier doit se terminer par «DIN» pour que ce sous-programme puisse correctement fonctionner.

### Arrangement d'ensemble

Cette fonction permet de trier les fiches par ordre croissant ou décroissant ou de dupliquer des fichiers.

### Duplication de fichiers

Trois possibilités nous sont alors offertes ; une recopie de toute la disquette sur une autre disquette ou la copie d'un ensemble, d'un type de fiche sur la même disquette ou sur une autre.

Si nous avons choisi l'une des deux dernières options (ligne 1113), nous devons indiquer au programme si on duplique sur la même disquette ou sur une autre (1114 → 1115), on retrouve alors dans P\$(1) le numéro du lecteur de disquette qui recevra les copies (ligne 1116).

En 1120, le programme est orienté en fonction du choix précédent.

### Recopie de toute la disquette (1150 → 1169)

En 1155, nous utilisons une dénomination ambiguë pour détruire tous les fichiers de la disquette de destination de façon à disposer de la plus grande place possible pour les fichiers de ce logiciel.

En 1156 et 1157, les deux fichiers de type catalogue sont recopiés.

De 1158 à 1169, nous recopions tous les fichiers de type ensemble de fiches puis type de fiche.

### Recopie d'ensemble de fiches (1200 → 1216)

De 1200 à 1205, nous demandons le nom de l'ensemble et nous vérifions que cet ensemble est présent sur la disquette, on charge dans *nf\$* le type de fiche correspondant à cet ensemble.

En 1206, nous recopions l'ensemble de nom *p\$* de la disquette 0 sur la disquette *p\$(1)* et sous le nom «temporaire».

De 1207 à 1216, nous renommons le fichier «temporaire» avec le nom final (nous vérifions que ce nom n'est pas déjà utilisé) et nous remettons le fichier *censemble* à jour (avec le nouvel ensemble de fiches).

La recopie d'un type de fiche (1250 → 1263) a la même structure et suit le même déroulement.

## TRI D'UN ENSEMBLE DE FICHES

### Entrée des paramètres servant au tri (1000 → 1027)

Dans cette partie, nous entrons toutes les informations nécessaires.

#### ● 1000 → 1014

Nous demandons le nom de l'ensemble de fiches à trier et nous vérifions que le type de fiche correspondant soit présent sur la disquette (il est nécessaire pour calculer la position des enregistrements).

#### ● 1015 → 1024

Nous vérifions d'abord qu'il n'y ait pas qu'une seule fiche. Nous demandons alors s'il faut faire un tri par ordre alphabétique ou l'inverse (1018 → 1021). Nous avons alors respectivement «i» ou «s» dans *P\$* pour indiquer le sens du tri.

Puis nous entrons le numéro de la zone servant à déterminer l'ordre des fiches.

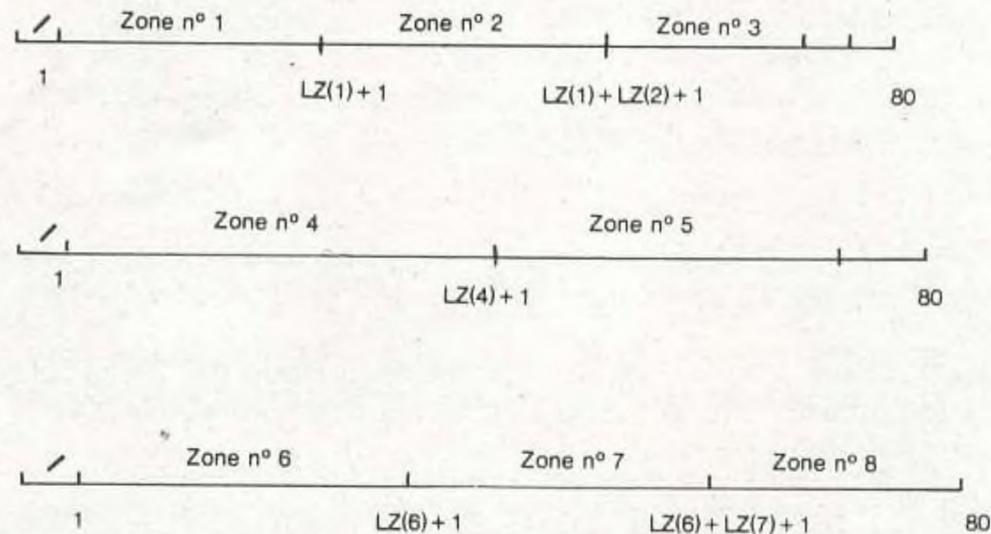
A ce stade, nous avons :

- l'ensemble de fiches est ouvert sous le numéro 2 ;
- P\$ = «j» ou «s» pour un ordre alphabétique ou l'inverse ;
- V = numéro de la zone servant au tri ;
- chargement partiel de la fiche.

### Calcul de la position d'un enregistrement (970 - 974)

Le but de ce sous-programme est de connaître l'emplacement d'une zone sur la disquette par rapport à la première zone d'une fiche.

Nous rappelons la structure des enregistrements :



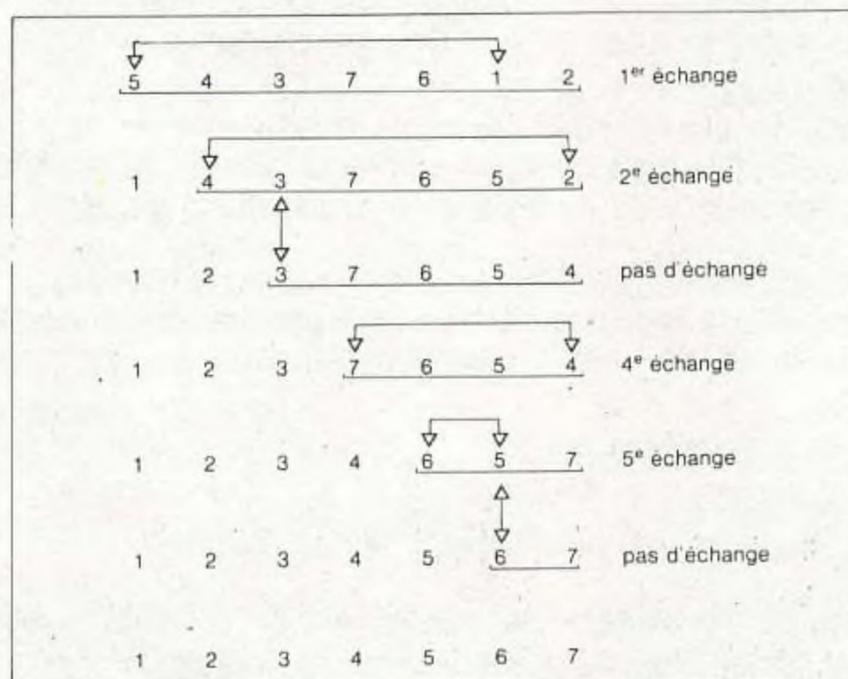
Nous devons avoir, à la sortie de ce sous-programme dans NE, le numéro de l'enregistrement et dans L1, la position de la zone dans l'enregistrement. Si nous faisons un tri avec la zone n° 5, nous aurions NE = 2 et L1 = LZ(4) + 1.

### Méthode de tri

Il existe un grand nombre de manière de trier des fichiers, nous utiliserons une méthode simple à comprendre :

1. On recherche le plus petit (ou grand) parmi un ensemble ;
2. Une fois trouvé, on l'échange avec le premier de l'ensemble ;
3. On recommence avec le même ensemble moins le premier.

Ainsi, pour trier : 5, 4, 3, 7, 6, 1, 2, nous aurions fait la démarche suivante :



A l'aide de l'organigramme et de la liste des variables, nous pouvons suivre le déroulement du programme.

### Sélection d'une fiche (950 - 968)

Ce sous-programme permet de sélectionner la première fiche qui est supérieure, inférieure ou égale (en fonction de P\$, respectivement S, i, e).

De 953 à 956, nous chargeons dans Z\$ le contenu de la zone servant à la sélection.

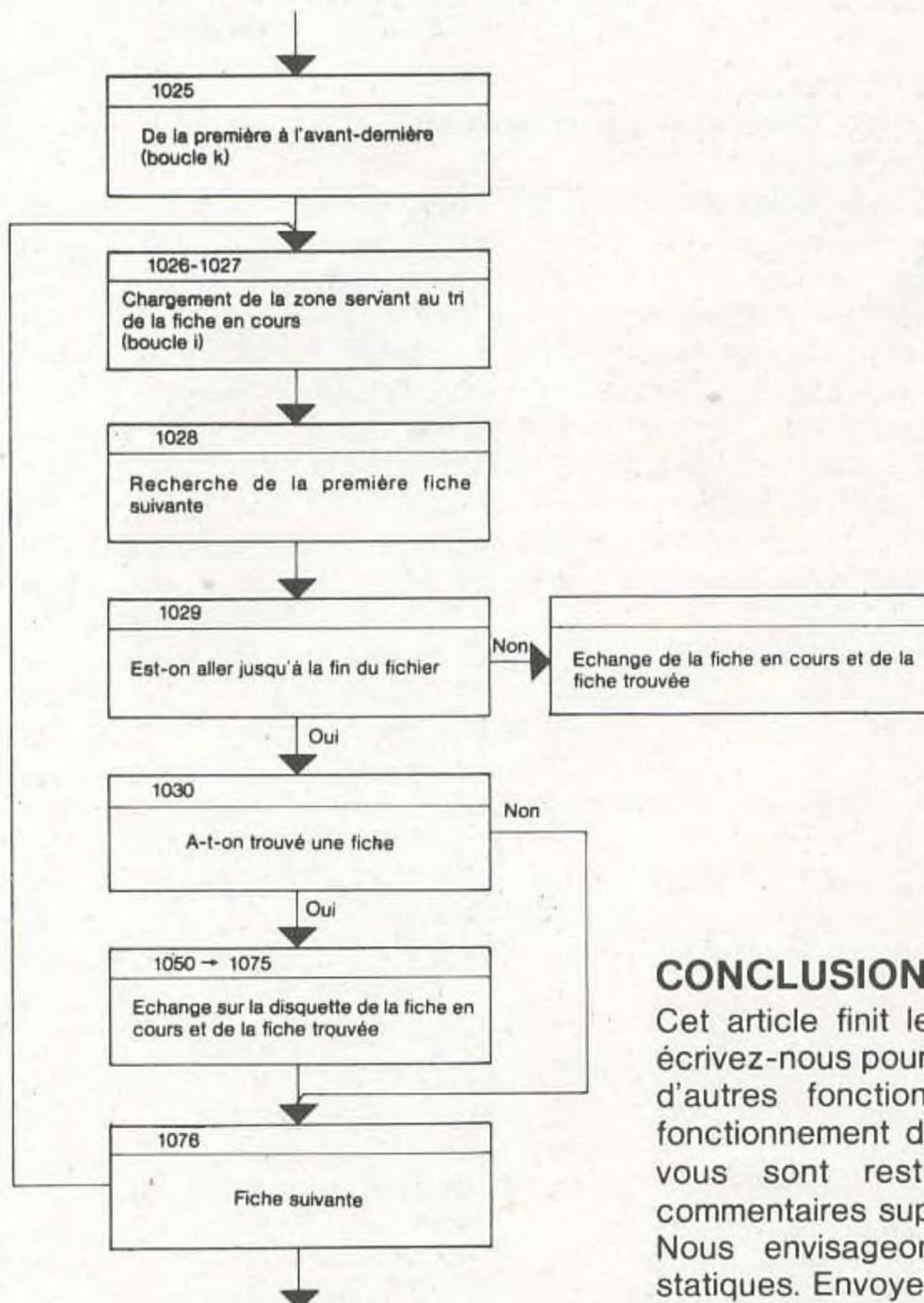
# FORTICHES LE COIN DES FORTICHES

De 957 à 959, nous effectuons un test (en fonction de P\$) pour savoir si la fiche doit être sélectionnée. Dans le cas contraire, nous passons à la suivante.  
De 966 à 968, nous pointons la fiche suivante si elle existe (J#Nf), sinon nous mettons le label V\$ à «DIN» pour indiquer que nous avons examiné les fiches jusqu'à la dernière.

## Liste des variables

Nom	Utilisation	Initialisation
LZ	Nombre d'enregistrements par fiche	Chargement partiel
NE	N° de l'enregistrement contenant la zone servant au tri	Calcul de la position d'un enregistrement
L1	Position de la zone dans l'enregistrement	Idem
nf	Nombre total de fiches de l'ensemble	1016
k	N° de la première fiche à partir de laquelle on recherche une fiche plus grande ou plus petite	1025
pf	N° de la fiche trouvée	1028
J	N° de la fiche en cours de comparaison	1027 et Sélection d'une fiche
nf\$	Contenu de la zone servant à la comparaison	1027
Z\$	Contenu de la zone de la nouvelle fiche	Sélection d'une fiche
V\$	Label de sélection d'une fiche indiquant si on est allé jusqu'à la fin d'un ensemble	Sélection d'une fiche
P\$(1-11)	Les enregistrements de la n <sup>ème</sup> fiche	1050 - 1056
P\$(1-23)	Les enregistrements de la P <sup>fème</sup> fiche	1056 - 1065

## Organigramme de tri



## CONCLUSION

Cet article finit le logiciel de gestion de fiches, néanmoins, écrivez-nous pour nous dire si vous désirez que nous écrivions d'autres fonctions qui faciliteraient ou complèteraient le fonctionnement de ce progiciel. De même, si certains points vous sont restés obscurs, nous pourrions faire des commentaires supplémentaires.

Nous envisageons un logiciel d'aide aux interprétations statiques. Envoyez-nous vos suggestions.

B. Lilamand

```

1 rem*****
2 rem*****consultation et creation de fiche*****
3 rem*****ecrit Par b. lilamand Pour led micro*****
4 rem*****
5 rem-----initialisation-----
6 dim P$(23),Pz(23,2),lz(23)
7 open 15,8,15:rem ouverture du canal de commande
8 poke550,128:poke53272,23:rem toute touche repetitive et 2eme jeu de caractere
9 ef$="XXXXXXXXXXXXXXXXXXXXXXXXXXXX":rem Pour dePlacer le curseur verticalement
10 m$="PB"+chr$(0)+chr$(0)+chr$(0):rem Pour remettre le Pointeur de 2 a zero
11 m1$="*****"
50 rem
51 rem-----menu-----
52 rem
53 Print chr$(147)
54 Print m1$;"*****  GESTION DE FICHE  *****";m1$
57 Print:Print "          *****MENU*****"
58 Print:Print "Tapez le numero de l'action desiree"
59 Print
60 Print "Initialisation d'une disquette-----:1"
61 Print:Print "Creation d'un type de fiche-----:2"
62 Print "Creation d'un ensemble de fiche-----:3"
63 Print:Print "Consultation des types de fiche-----:4"
64 Print "Consultation des ensembles de fiche--:5"
65 Print "Consultation des fiches d'un ensemble:6"
66 Print:Print "Arrangement d'ensemble(T,D)-----:7"
67 Print "Destruction de fichier(F,E)-----:8"
68 Print:Print "Sortie du Programme-----:9"
70 get P$:if P$="" then 70
71 if P$="e"then P=1 :goto 82
72 if P$="f"then P=2 :goto 82
73 if P$="t"then P=3 :goto82
74 if P$="d" then P=4 :goto82
77 P=asc(P$)-asc("0")
78 if P<1 or P>9 then 70
79 Print chr$(147);:on P gosub 112,200,351,151,401,451,801,700,90
80 goto 53
82 Print chr$(147);:on P gosub 725,710,1000,1100
83 goto 53

ready.

508 rem acces direct a une fiche*****
509 Print"Par le Numero ou Par une zone ";left$(ef$,24);
510 get P$:if P$<>"n" and P$<>"z" then 510

```

```

511 Print "                                     ":if P$="z" then gosub 920:goto 472
512 Print left$(ef$,24);:input "numero de la fiche":P$
513 P=val(P$):if P=<0 or P>nf then 484
514 j=P:goto 472

```

ready.

```

619 rem***** chargement Partiel d'une fiche*****
620 open 2,8,2,P$
621 input#2,nz :lz=1:l=1
622 for P=1 to nz:input#2,Pz(P,1),Pz(P,2),lz(P)
623 if l+lz(P)> 80 then lz=lz+1:l=0
624 l=l+lz(P):next P:Print chr$(147);
625 close 2:return

```

ready.

```

701 Print m1$:"***** DESTRUCTION DE FICHIER *****":m1$
702 Print " Mettre la disquette de donnee et faire:" :Print
703 Print " RETURN Pour revenir au menu"
704 Print " F Pour detruire un type de fiche"
705 Print " E Pour detruire un ensemble de fiche"
706 get P$:if P$="" then 706
707 if P$=chr$(13) then return
708 Print#15,"i":if P$="e" then 725
709 if P$<>"f" then 706
710 input " Nom du type de fiche":P$
711 nf$="f"+P$:Print " Etes vous sur de vouloir detruire le type de fiche ";
712 Print P$;" [o/☐n☐]?"
713 get v$:if v$="" then 713
714 if v$="n" or v$=chr$(13) then return
715 open 2,8,2,"cfiche":P$=nf$:gosub 251 :rem*****recherche*
716 if P$<>nf$ then close 2:return
717 i=1:gosub 753 :rem *****destruction Partielle*
718 close 2:Print#15,"s":P$
719 return
725 input " Nom de l'ensemble de fiche":P$
726 nf$="e"+P$:Print " Etes vous sur de vouloir detruire l'ensemble de fiche ";
727 Print P$;" [o/☐n☐]?"
728 get v$:if v$="" then 728
729 if v$="n" or v$=chr$(13) then return
730 open 2,8,2,"censemble":P$=nf$:gosub 251:rem *****recherche*
731 if nf$<>P$ then close 2:return
732 i=2:gosub 753:rem *****destruction Partielle*
733 close 2:Print#15,"s":P$
734 return
750 rem*****

```

```

751 rem** destruction de i enregistrements*
752 rem** a. Partir du P eme *****
753 i=p+i
754 Print#15,"PB"chr$(i-256*int(i/256))chr$(int(i/256))chr$(1)
755 for j=1 to 23:input#2,p$(j)
756 if p$(j)="din" then j=23
757 next j
758 Print#15,"PB"chr$(p-256*int(p/256))chr$(int(p/256))chr$(1)
759 for j=1 to 23:Print#2,p$(j)
760 if p$(j)="din" then j=25
761 next j
762 if j=26 then return
763 p=p+23:i=i+23:goto 754

```

ready.

```

800 rem
801 Print#1;"***** RECOPIE D'ENSEMBLE *****";m1$
802 Print"  T  :Trier les fiches"
803 Print"  D  :Dupliquer un ensemble"
804 Print"RETURN:Retour au menu"
810 getp$:if p$="" then 810
811 if p$="t" then 1000
812 if p$="d" then 1100
818 if p$=chr$(13) then return
ready.

```

```

919 rem***entre des Parametres Pour la selection de fiche*****
920 Print left$(ef$,24);"
921 Print"
";left$(ef$,24);
922 p$="":input"no de la zone Pour la selection";p$:if p$="" then return
923 v=val(p$):if v<1 or v>nz then 920
924 gosub 970:rem*****calcul de la Position d'un enregistrement*
928 Print left$(ef$,24);"La zone doit-elle etre superieur(S), "
929 Print "inferieur(I),ou egalle(E)?";
930 get p$:if p$<>"e" and p$<>"i" and p$<>"s" then 930
931 Print left$(ef$,24);"
";left$(ef$,24);
932 Print"
";left$(ef$,24);
933 Print"contenu de la zone de reference. ";lz(v);"car":Print"☐";
934 nf$="":for i=1 to lz(v)
935 get v$:if v$="" then 935
936 p=asc(v$):if (p<31 or p=34 or p>128) and p<159 then 935
937 Print v$;nf$=nf$+v$
938 next i:Print"☐";

```

```

939 Print left$(ef$,24);"
940 Print "          PATIENTEZ          ";
942 gosub 966:rem*****selection d'une fiche*
943 get v$:if v$<>" " then 943
944 return
950 *****
951 ***** selection d'une fiche *****
952 *****
953 P=(j-1)*lz+1+ne
954 Print#15,"PB"chr$(P-256*int(P/256))chr$(int(P/256))chr$(11)
955 z$="":for i=1 to lz(v)
956 get#2,v$:z$=z$+v$:next i:Print left$(ef$,24);z$
957 if P$="e" then if z$=nf$ then return:la fiche est ok
958 if P$="s" then if z$>nf$ then return
959 if P$="i" then if z$<nf$ then return
960 rem    fiche suivante
966 if j<>nf then j=j+1:goto 953
967 Printleft$(ef$,24);"Plus de fiche          "
968 v$="din":return
969 rem***** calcul de la Position d'un enregistrement *****
970 l1=2:ne=1:if v=1 then return
971 for i=1 to v-1
972 if l1+lz(i)>80 then ne=ne+1:l1=2
973 l1=l1+lz(i):next i
974 return
999 *****
1000 Print m1$;"***** tri d'un ensemble de fiche *****";m1$
1001 Print "    Faire RETURN Pour revenir au menu ou"
1002 P$="":input"indiquez le nom de l'ensemble de fiche a trier";P$
1003 if P$="" then return
1004 Print#15,"i":open2,8,2,"censemble"
1005 P$="e"+P$:nf$=P$
1006 gosub 251:rem *****recherche*
1007 if P$<>nf$ then Print" Cet ensemble n'existe Pas":close 2:goto 1002
1008 input#2,P$:input#2,nf$(1)
1009 close 2:open2,8,2,"cfiche"
1010 P$=nf$(1):gosub 251:rem*****recherche*
1011 close 2:if P$=nf$(1) then 1015
1012 Print chr$(147)
1013 Print"    La fiche demandee Pour cet ensemble n'est Pas decrite sur")
1014 input "cette disquette.    Faire RETURN.":P$:return
1015 gosub 620:rem*****chargement Partiel*
1016 open 2,8,2,nf$:Print chr$(147):input#2,nf
1017 if nf=1 then close 2:Print" Une seule fiche.Faire RETURN":input P$:return
1018 Print "    Voulez-vous faire un tri Par ordre 1)De-croissant ou 2)Croissant"
1019 get P$:if P$<>"c" and P$<>"d" then 1019
1020 if P$="c" then P$="i":Print "Croissant":goto 1022
1021 P$="s":Print "Decroissant"
1022 Print:input"Numero de la zone servant a determiner l'ordre";v$

```

```

1023 v=val(v$):if v<1 or v>nz then Print"*****":goto 1022
1024 gosub 970: rem ***** calcul de la Position d'un enregistrement
1025 for k= 1to nf-1:P=(k-1)*lz+1+ne
1026 Print#15,"PB"chr$(P-256*int(P/256))chr$(int(P/256))chr$(1)
1027 nf$="":for i=1 to lz(v):get#2,z$:nf$=nf$+z$:next i:j=k
1028 Pf=j:gosub 966:rem***** selection d'une fiche*
1029 if v$<>"din" then nf$=z$:goto 1028
1030 if Pf=k then 1076
1050 j=(k-1)*lz+2
1051 Print#15,"PB"chr$(j-256*int(j/256))chr$(int(j/256))chr$(1)
1052 for i=1 to ne:v$=""
1053 for j=1 to 80:get#2,z$
1054 if z$=chr$(13) then j=80:goto1056
1055 v$=v$+z$
1056 next j:P$(i)=v$:next i:j=(Pf-1)*lz+2
1060 Print#15,"PB"chr$(j-256*int(j/256))chr$(int(j/256))chr$(1)
1061 for i=1 to ne:v$=""
1062 for j=1 to 80:get#2,z$
1063 if z$=chr$(13) then j=80:goto 1065
1064 v$=v$+z$
1065 next j:P$(24-i)=v$:next i
1070 j=(Pf-1)*lz+2
1071 Print#15,"PB"chr$(j-256*int(j/256))chr$(int(j/256))chr$(1)
1072 for i=1 to ne:Print#2,P$(i):next i
1073 j=(k-1)*lz+2
1074 Print#15,"PB"chr$(j-256*int(j/256))chr$(int(j/256))chr$(1)
1075 for i=1 to ne:Print#2,P$(24-i):next i:
1076 next k:close 2:return
1100 Print chr$(147);m1$;"***** DUPLICATION *****";m1$
1101 Print " 1: Duplication de toute la disquette"
1102 Print " 2: Duplication d'un ensemble de fiche"
1103 Print " 3: Duplication d'un type de fiche"
1104 Print "      ou RETURN"
1110 get P$:if P$="" then 1110
1111 if P$=chr$(13) then return
1112 P=val(P$):if P<1 or P>3 then 1110
1113 if P=1 then 1120
1114 Print:Print " Voulez copier sur une autre disquette?[O/N]"
1115 get P$:if P$<>"o" and P$<>"n" then 1115
1116 P$(1)="1":if P$="n" then P$(1)="0":goto 1120
1117 Print " Mettre la disquette receveuse dans le lecteur numero 0"
1118 Print " Laisser la disquette source dans le lecteur numero 1"
1119 input " Faire RETURN";P$:Print#15,"i:1"
1120 on P gosub 1150,1200,1250
1121 goto 1100
1149 *****duplication*****
1150 Print " ATTENTION! Vous allez detruire les in- formations Presentent sur";

```

```

1151 Print " la disquette receveuse.Voulez-vous toujours faire la duplication";
1152 v$="":input "[O/☐N☐]";v$
1153 if v$<>"o" then return
1155 Print#15,"s1:*"
1156 Print#15,"c1:cfiche=0:cfiche"
1157 Print#15,"c1:censemble=0:censemble"
1158 open2,8,2,"cfiche"
1159 input#2,p$:if p$="din" then1162
1160 Print#15,"c1: ";p$;"=0: ";p$
1161 goto 1159
1165 close 2:open2,8,2,"censemble"
1166 input#2,p$:if p$="din" then return
1167 input#2,p$
1168 Print#15,"c1: ";p$;"=0: ";p$
1169 goto 1166
1200 p$="":input " Nom de l'ensemble de fiche a copier ou RETURN ";p$
1201 if p$="" then return
1202 open2,8,2,"censemble"
1203 p$="e"+p$:gosub 251:rem *****recherche*
1204 input#2,nf$:input#2,nf$:close 2
1205 if p$="din"then Print " Cet ensemble n'existe Pas.":goto 1200
1206 Print#15,"c";p$(1);":temporaire=0: ";p$
1207 p$="":input " Nom de l'ensemble de fiche recopie (15 car) ";p$
1208 open2,8,2,p$(1)+":censemble"
1209 p$="e"+p$:nf$(1)=p$:gosub 251:rem*****recherche*
1210 if p$="din" then 1215
1211 Print " Choisissez un autre nom."
1213 close 2:goto 1207
1215 Print#2,nf$(1):Print#2,nf$:Print#2,"din"
1216 close 2:Print#15,"r";p$(1);": ";nf$(1);"=temporaire":return
1250 p$="":input " Nom du type de fiche a recopier ou RETURN";p$
1251 if p$="" then return
1252 open2,8,2,"cfiche"
1253 p$="f"+p$:gosub 251:rem*****recherche*
1254 close 2
1255 if p$="din" then Print " Ce type de fiche n'existe Pas.":goto 1250
1256 Print#15,"c";p$(1);":temporaire=0: ";p$
1257 p$="":input " Nom du type de fiche recopie (15 car)";p$
1258 if len(p$)>15 then Print " 15 carateres au maximum.":goto 1257
1259 open 2,8,2,p$(1)+":cfiche"
1260 p$="f"+p$:nf$=p$:gosub 251:rem*****recherche*
1261 if p$<>"din" then Print " Choisissez un autre nom":close 2:goto 1257
1262 Print#2,nf$:Print#2,"din":close 2
1263 Print#15,"r";p$(1);": ";nf$;"=temporaire":return

```

ready.

# Assemblage et systèmes

## – INTERFAÇAGE RS232C –

### 1. APPRENONS A ASSEMBLER DES SYSTEMES

Dans les premiers numéros de LED MICRO, nous avons

- d'une part décrit les principales interfaces standard rencontrées en micro-informatique (RS232C, Centronics, IEE488, prise Péritel) : LED MICRO n° 2 pages 14, 15 et 24 à 29 ;
- d'autre part précisé que l'on ne peut pas connecter n'importe quel périphérique avec n'importe quel ordinateur, même s'ils sont tous deux munis d'une interface portant le même nom. (Ce sera – peut-être – le cas lorsque le standard MSX sera devenu un standard universel en micro-informatique domestique, ne rêvons pas !)
- enfin (et surtout !) insisté sur le fait que la constitution d'un système complet à partir de sous-ensembles provenant de diverses origines était un travail difficile et qu'il fallait impérativement en laisser le soin de l'assemblage à des spécialistes.
- La conclusion était : n'achetez que des systèmes complets chez un même «ensemblier» vous garantira le bon fonctionnement de l'ensemble matériel et logiciel (et encore ! Entre les promesses de la garantie et la réalité, il y a autant de différence qu'entre un programme électoral et ce qui se passe après).

Travail de spécialiste, d'accord. Mais avant de devenir un «spécialiste», chaque technicien a été un débutant. Dans la série d'articles «Assemblage et Systèmes», nous allons vous donner quelques notions pratiques sur les problèmes d'assemblage et de connexion.

Il ne s'agira pas d'un cours dont les chapitres s'enchaînent logiquement (comme les cours de Philippe Duquesne et de Claude Polgar). Il ne s'agira pas non plus d'une suite de «recettes» du genre : «Pour connecter sur l'ordinateur X la machine à dessiner Y, il vous faudra d'abord positionner le switch A3 sur ON puis...». Nous vous fournirons une suite d'exemples simples, progressifs, concrets et d'utilité réelle en vous expliquant le «pourquoi» de chaque connexion ou de chaque ligne de programme. Ceci vous donnera une «culture générale» vous aidant à vous adapter à l'infinie variété des problèmes d'assemblage de systèmes.

### 2. SERIE-PARALLELE ET SYNCHRONE-ASYNCHRONE

#### Série et parallèle

Nous pouvons transmettre les 8 bits d'un octet :

- soit sur 8 fils séparés (transmission parallèle : voir figure 1 page ci-contre) ;
- soit sur un seul fil, les 8 bits étant transmis les uns après les autres (transmission série) : voir figure 2 ;

La première méthode est plus simple et plus rapide.

La seconde est plus économique en fils et moins sensible aux parasites.

De ce fait, on utilise souvent :

- la transmission série pour les transmissions à longue distance (supérieures à 2 m) ;
- la transmission parallèle pour les transmissions à courte distance.

#### Synchrone et asynchrone

L'exemple-type de la transmission asynchrone est la transmission dans l'antique code Morse. On peut faire démarrer l'émission d'un message à n'importe quel instant et chaque caractère y est transmis «isolé». D'où son autre nom «transmission caractère par caractère».

Dans une transmission synchrone au contraire, les bits doivent être envoyés à des instants parfaitement repérés – dans un cycle – ce qui nécessite un dispositif de repérage du temps appelé «horloge».

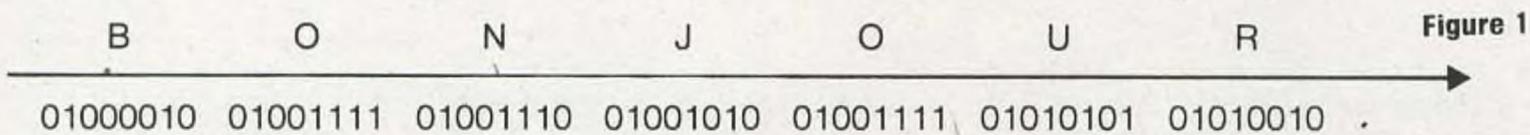
Toutes choses égales par ailleurs, une transmission synchrone est plus rapide qu'une transmission asynchrone, on ne perd pas de temps à séparer les caractères soit par des petits silences (code Morse), soit par des bits de synchronisation (bits de Start-Stop par exemple).

#### Le RS232C et les autres

Le tableau figure 3 de la page ci-contre compare les caractéristiques des trois principales interfaces utilisées en micro-informatique : RS232C, Centronics et IEEE488.

## Transmission série et transmission parallèle

Transmission série



Transmission parallèle



Figure 2

	Centronics	RS232C	IEEE488
Nombre de fils minimum	12	3	16
Vitesse	faible	moyenne	élevée
Complexité de mise en œuvre	simple	moyenne	complexe
Communication dans les deux sens	non	oui	oui
Mode	parallèle	série	parallèle
Fiabilité	moyenne	moyenne	élevée
Nombre maximum d'appareils reliés	2	2	16
Utilisation principale	imprimantes, traceurs et périphériques uniquement «récepteurs».	imprimantes, liaison téléphonique, terminaux... et tous périphériques aussi bien émetteurs et récepteurs de vitesse relativement lente (par exemple pas de disquette, ce serait trop lent).	Tous systèmes industriels et en particulier des appareils de mesure, des disques durs, etc...

## 3. DE LA LIAISON 3 FILS AU HANDSHAKE

### Liaison 3 fils

Le mode de communication le plus simple entre deux appareils est la «liaison 3 fils». Dans le cas du RS232C, les 3 fils utilisés sont :

- TD (= Transmitted Data) : fil pour émettre les données
- RD (= Received Data) : fil pour recevoir les données
- GND (= Ground) : fil de masse électrique.

Lorsque deux appareils communiquent entre eux, le fil de sortie de l'un (son TD) doit être relié à l'entrée de l'autre (le RD) et réciproquement.

Au repos, les deux fils TD et RD sont au niveau logique 1. Le passage au niveau 0 indique le début de la transmission.

### Handshake

Il est facile d'établir un dialogue complet entre deux appareils en utilisant une simple liaison 3 fils. Mais cette liaison 3 fils peut être insuffisante. Par exemple, l'appareil A qui reçoit les informations de l'émetteur B peut avoir d'autre chose à faire et peut avoir besoin de lui dire «Arrête un peu ton flot de paroles, je ne peux pas tout faire à la fois». Puis lui dire «Vas-y, maintenant je t'écoute». Ce dialogue complémentaire s'appelle «handshake» (= poignée de main). Sur le RS232C il s'effectue sur 5 fils supplémentaires qui sont les fils d'état :

- RTS : Request to Send : demande d'émission (S)
- DTR : Data Terminal Ready : terminal prêt à émettre (S)
- DCD : Data Carrier Detector : reçoit des données sur la ligne (E)
- CTS : Clear To Send : prêt à émettre (E)
- DSR : Data Set Ready : poste de données prêt (E).

**N.B.1.** S = Sortie par rapport au terminal. E = Entrée.

**N.B.2.** Ces 5 fils supplémentaires peuvent jouer également des rôles de sécurité (contrôle de fil non coupé, etc.)

Le figures 3 et 4 (page ci-contre) représentent le dialogue qui s'établit entre l'ordinateur et le périphérique. ATTENTION ! Remarquez que ce dialogue **n'est pas** symétrique. L'ordinateur prend la précaution de s'assurer que le périphérique est prêt à recevoir ses données. Le périphérique, lui, suppose que l'ordinateur est toujours prêt à recevoir n'importe quel flot de données.

## 4. LA MISE EN FORME DES DONNÉES

Il s'agit d'agencer les bits d'un octet de façon à les envoyer «convenablement» les uns après les autres sur une ligne.

La norme RS232C impose certaines contraintes :

- au repos, la ligne est au 1 logique ;
- il faut un bit de start (= début) pour signaler le début d'émission d'un octet ;
- il faut au moins un bit de stop pour signaler la fin de l'octet.

### Le bit de start

Il vaut toujours 0 et permet la synchronisation du périphérique et de l'ordinateur grâce au passage du niveau 1 (ligne au repos ou bit de stop) au niveau 0.

### Le bit de stop

Il vaut toujours 1 et permet une vérification de la synchronisation du receveur par rapport à l'émetteur. En effet, si le receveur ne trouve pas de bit de stop, il y a nécessairement eu une erreur de transmission. Par contre, la présence de celui-ci ne permet pas d'assurer à 100 % que la transcription s'est bien passée. La norme RS232C laisse le choix entre 1 ou 2 bits de stop. La présence de 2 bits de stop est préférable pour obtenir une meilleure fiabilité mais elle ralentit la transmission.

Le RS232C n'impose pas de standard sur la nature des informations transmises. C'est ainsi que :

**Le caractère** n'est pas obligatoirement transmis sur 8 bits. On peut utiliser le code Baudot (code télégraphique) qui ne comporte que 5 bits, ou l'ASCII sans bit de parité, qui ne comporte que 7 bits.

**La parité.** Pour augmenter la fiabilité de la transmission, on peut ajouter (facultativement) un bit de parité (ou d'imparité). La «théorie» du bit de parité a été faite dans LED MICRO n° 3 §G2.9.2.D page 14).

La figure 5 (page ci-contre) donne un exemple de caractères émis suivant la convention «1 bit de parité et 2 bits de stop».

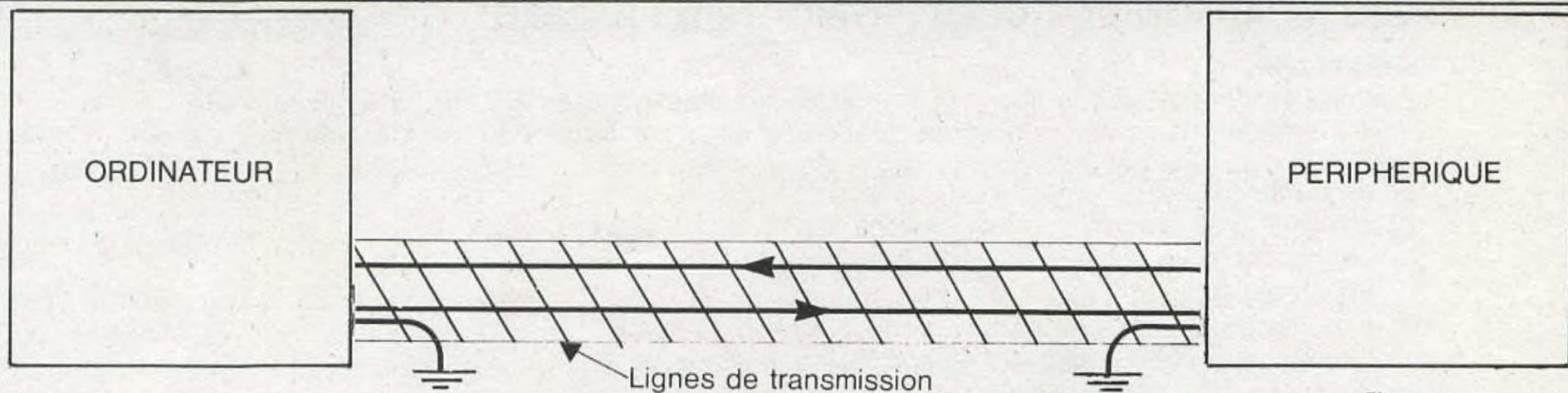


Figure 1

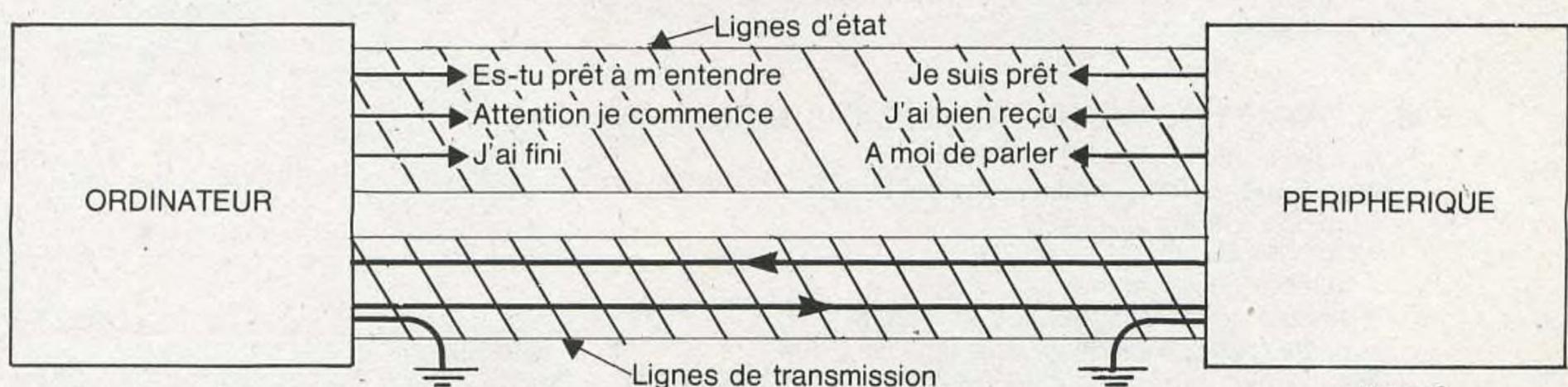


Figure 2

### Procédure d'émission de données de l'ordinateur vers le périphérique en RS232C

1	Ordinateur	RTS = 1	Je veux émettre. Etes-vous prêt ?
2	Périphérique	DSR = 1	Oui, je suis prêt à recevoir.
3	Ordinateur	DTR = 1	Attention je vais émettre.

Figure 3

### Procédure d'émission de données du périphérique vers l'ordinateur en RS232C

1	Périphérique	DCD = 1	Je vais vous envoyer des données. Je suppose que vous êtes prêt.
2	Périphérique	CTS = 1	Je vous envoie ces données.

Figure 4

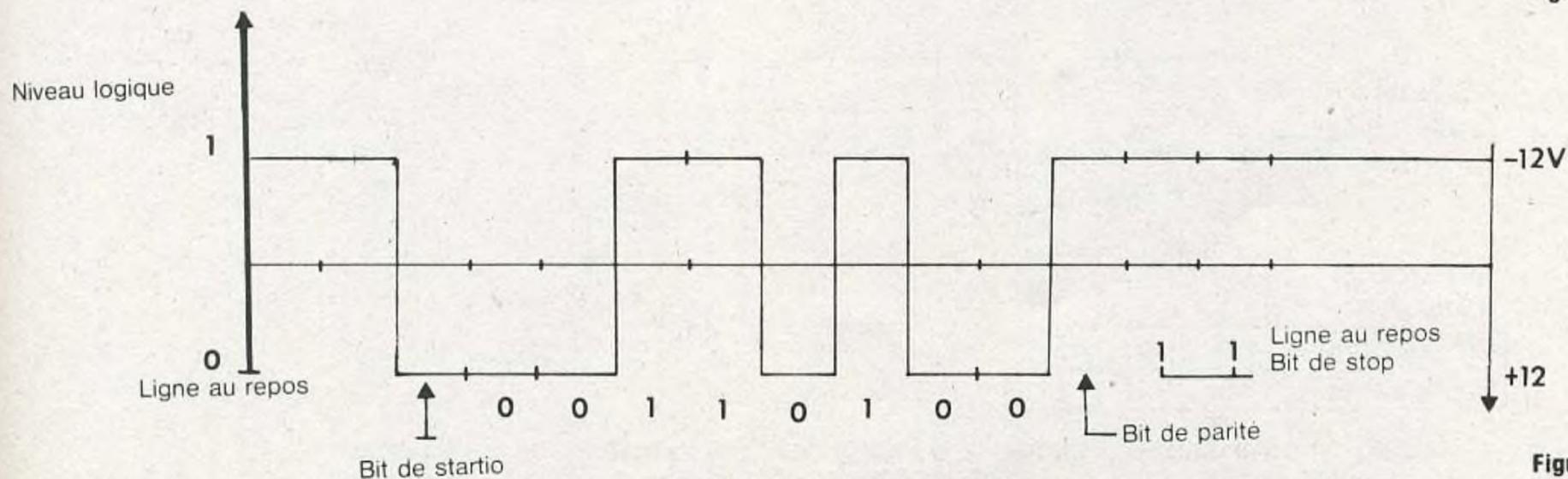


Figure 5

## 5. LES CARACTERISTIQUES PHYSIQUES DU RS232C

### ● Brochage

Pour les brochages, voir la figure 2. Seules les broches intitulées TD, RD, GND, RTS, DCD, CTS et DSR nous intéressent. Les autres broches sont d'une utilisation beaucoup plus délicate et leur emploi n'entre pas de le cadre de cet article.

### ● Les tensions

La norme RS232 précise que les tensions sur les lignes doivent vérifier :

le niveau logique 1 :  $-25 V < U < -3 V$

le niveau logique 0 :  $25 V > U > 3 V$

les valeurs utilisées en général sont  $\pm 12 V$  ou  $\pm 24 V$

les entrées doivent pouvoir supporter des tensions de  $\pm 50 V$ .

### ● La vitesse de transmission

En fonction de la qualité de la ligne, nous pouvons modifier la vitesse de transmission. Il existe des vitesses standardisées réparties de 50 à 19 200 bauds.

## 6. COMMANDE D'UNE RS232C EN BASIC

Pour utiliser une liaison RS232C, il faut :

+ d'une part, définir certaines options :

- nombre de fils utilisés (3 ou 8)
- nombre de bits de stop (1 ou 2)
- nombre de bits transmis (5 à 8)
- vitesse de transmission (50 à 19 200)
- parité (paire, impaire) ou absence de parité.

+ d'autre part, indiquer à l'ordinateur à quelle «adresse», il trouvera le connecteur RS232C avec lequel il travaillera.

Ces deux éléments s'obtiennent en utilisant l'instruction OPEN.

On sait que OPEN sert, à «ouvrir un fichier», c'est-à-dire à envoyer ou recevoir des informations sur le périphérique «lecteur-enregistreur de disquettes». Mais le OPEN peut être utilisé avec d'autres périphériques que le lecteur de disquettes. Il suffit de préciser les caractéristiques du message (les «options») et l'adresse, et le BASIC envoie tout ce qu'on lui demande d'envoyer.

La seule difficulté est que la syntaxe de l'OPEN varie considérablement d'une machine à l'autre. **CONSULTEZ VOTRE NOTICE !**

## 7. EXEMPLE AVEC LE COMMODORE 64

La syntaxe de l'OPEN est la suivante :

OPEN Lfn, 2, 3, "< registre de contrôle > <registre de commande >

**Lfn : numéro du canal** (utilisé avec les PRINT #Lfn)

2 : indique que nous adressons à l'entrée/sortie RS232

3 : est une adresse secondaire

< registre de contrôle > est un caractère qui spécifie la vitesse de transmission, le nombre de bits transmis et le nombre de bits de stop.

Pour connaître le code de ce caractère, nous utilisons le tableau suivant :

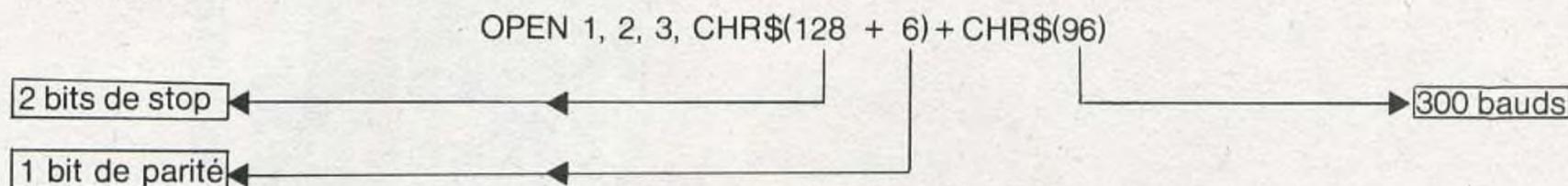
N° du bit	7	6	5	4	3	2	1	0
Poids	128	64	32	16	8	4	2	1
	← Parité →			0	X	X	X	Nbre de fils
Absence de bit de parité	X	X	0					
Bit d'imparité	0	0	1					
Bit de parité	0	1	1					

0 = 3 fils
1 = 8 fils

< registre de commande > permet de définir la parité et le nombre de fils utilisés.

Si nous désirons transmettre 8 bits + 1 bit de parité et 2 bits de stop à une vitesse de 300 bauds sur 3 fils, nous aurons



Pour envoyer un caractère vers le périphérique, nous ferons : PRINT # 1, A\$ ; et pour recevoir un caractère du périphérique : GET # 1, A\$.

Pour transformer un ordinateur en terminal avec une interface RS232, nous pouvons faire le petit programme suivant :

```

5 OPEN 1, 2, 3, CHR$(128 + 6) + CHR$(96)
10 GET # 1, A$ : PRINT A$ ;
15 GET 1, A$ : PRINT # 1, A$ ;
20 GOTO 10
    
```

### Quelques affectations normalisées du connecteur RS232C

N° de broche			V24
7	GND	Terre de signalisation	102
2	TD	Emission de données	103
3	RD	Réception de données	104
4	RTS	Demande d'émission	105
5	DTR	Terminal prêt à émettre	106
6	DCD	Périphérique prêt à émettre	107
8	DS	Détection de porteuse	109



**Attention :** Le RS232C et le V24 ne sont pas entièrement identiques. Consultez (l'excellent) ouvrage de H. Lilen «Interfaces pour microprocesseurs et micro-ordinateur», page 124).

N° du bit	7	6	5	4	3	2	1	0
Poids	128	64	32	16	8	4	2	1

Nombre de bits de stop

0 : 1 bit de stop

1 : 2 bits de stop

Vitesse de transmission

0	0	0	1	50	bauds
0	0	1	0	75	bauds
0	0	1	1	110	bauds
0	1	0	0	134,5	bauds
0	1	0	1	150	bauds
0	1	1	0	300	bauds
0	1	1	1	600	bauds
1	0	0	0	1 200	bauds
1	0	0	1	1 800	bauds
1	0	1	0	2 400	bauds
1	0	1	1	3 600	bauds
1	1	0	0	4 800	bauds
1	1	0	1	7 200	bauds
1	1	1	0	9 600	bauds
1	1	1	1	19 200	bauds

Nombre de bits transmis

00	8 bits
01	7 bits
10	6 bits
11	5 bits

**E**n cette période du SICOB PRINTEMPS, force est de constater que le marché de la micro-informatique est de plus en plus balisé. Il ne reste pour ainsi dire aucune place à l'improvisation, de plus les petits génies sont priés d'aller apprendre la finance et le marketing devenus aujourd'hui les cartes majeures. Il ne fait aucun doute que les deux standards, MSX pour les amateurs et IBM PC pour les professionnels, sont devenus la référence. La tendance en ce printemps concerne, pour la famille des IBM-PC et compatibles, des capacités de stockage en disque dur de plus en plus intéressantes. Cela va de 5 M octets à 70 M octets pour les plus puissantes. Bref, nous entrons dans l'époque des micros ayant des capacités de mini. La mode se tourne de plus en plus sur les transportables et les portables. Les écrans à cristaux liquides deviennent de plus en plus performants. Après avoir fait beaucoup parlé de lui, le standard UNIX reste limité et réservé aux professionnels avertis. Du côté des imprimantes, notons la sortie de nombreux modèles à jet d'encre. Régulièrement, les capacités graphiques gagnent du terrain. Les prix baissent et la qualité augmente, que demander de mieux ! Par contre, à part la série des OLIVETTI, la vitesse de traitement n'évolue guère. Les progiciels par contre font beaucoup de progrès en direction de leur facilité d'emploi, l'interactivité gagne du terrain. La dernière version de dBASE (dBase III) et FRAMEWORK en sont un exemple frappant. Finalement, rien de très nouveau ni à l'ouest ni à l'est. Le professionnalisme arrive enfin ! L'improvisation commence à se calmer.

Notons, à l'aube des vacances, une très nette baisse du taux de natalité des revues spécialisées en informatique. Il y aura sans doute, à la rentrée, des petits changements, et les mensuels qui vivent grâce à la publicité vont sûrement tousser.

LED MICRO, quant à lui, confirme et signe. Votre revue a su trouver son chemin, son succès et sa spécialisation en fait la revue la plus constante du marché.

C.H. Delaleu

# SOMMAIRE

LES LANGAGES ET LOGICIELS DE LA QUATRIÈME GÉNÉRATION : DES INCONNUS. ON PARLE BEAUCOUP DE LA CINQUIÈME GÉNÉRATION ALORS QUE CETTE QUATRIÈME GÉNÉRATION EST LOIN D'AVOIR DONNÉ TOUT CE QU'ELLE POUVAIT **P. 48**. CONTRE-MESURE : LE GUÉPARD, UN MICRO FRANÇAIS A LA LOUPE (SUITE ET FIN) **P. 52**. DBASE III, UNE BASE DE DONNÉES PARMIS LES PLUS CÉLÈBRES ET LES PLUS PERFORMANTES **P. 58**. SHOPPING : NOUVEAUX LOGICIELS ET MATÉRIELS QUE VOUS POURREZ VOIR AU SPÉCIAL SICOB ET DANS LES MAGASINS **P. 61**. A LIRE : LA SÉLECTION DU MOIS DE PHILIPPE FAUGERAS **P. 64**.

## Une méconnue : la 4<sup>ème</sup> génération informatique

**A**ujourd'hui, la 5<sup>e</sup> génération informatique – l'intelligence artificielle – fait la une des journaux. Elle fait naître les rêves les plus fous. Et la 4<sup>e</sup> génération est comme oubliée. Cependant, elle est un trait d'union indispensable entre le «savoir programmer» et la mise en application d'une informatique de puissance.

Aujourd'hui, la IV<sup>e</sup> génération, de par les concepts qu'elle véhicule, par sa facilité et sa convivialité se rapproche de l'intelligence artificielle. Aussi constitue-t-elle une étape charnière. Mais avant d'arriver à la situation actuelle, plusieurs étapes importantes ont été nécessaires..

### DE 1950 A AUJOURD'HUI

Les premiers ordinateurs sont apparus vers 1950. Et dès cette époque les langages de gestion ont été au cœur des préoccupations des chercheurs. De par la masse des informations à traiter il était nécessaire de disposer d'une technique cohérente de classement mais aussi de possibilités efficaces de stockage. En ce sens, les logiciels et langages de la IV<sup>e</sup> génération apparaissent comme la résultante de l'évolution du langage de programmation et des systèmes de gestion de bases de données, les SGBD.

C'est en 1961 que la première base de

données voit le jour. Integrated Data System, tel est son nom, a été conçue par General Electric. Quatre ans plus tard, un chercheur d'IBM décrit un modèle d'organisation des données, qualifié de «relationnel». Les premiers SGBD relationnels sont commercialisés en 1974/75. Ils s'appellent DB2, Oracle, Ingres. Ils constituent la seconde génération des SGBD. En trente-cinq ans, les bases de données ont connu trois générations et les langages quatre. Il faut admettre que les limites ne sont pas toujours très nettes et la classification des trois premières générations de langages donne toujours lieu à des interprétations. Néanmoins les définitions suivantes sont couramment admises.

La première génération est celle du langage machine. Le programmeur agit directement sur le micro-processeur.

La seconde génération est constituée par le langage assembleur. Ce langage en traduisant des mnémotechniques en code machine binaire demeure encore

très proche du langage machine. Avec la troisième génération apparaissent les langages « évolués » comme l'Apl, le Cobol, le Fortran, le PLL, le Pascal, le Basic. La programmation se fait moins abstraite. Elle se rapproche davantage de la conception humaine du langage.

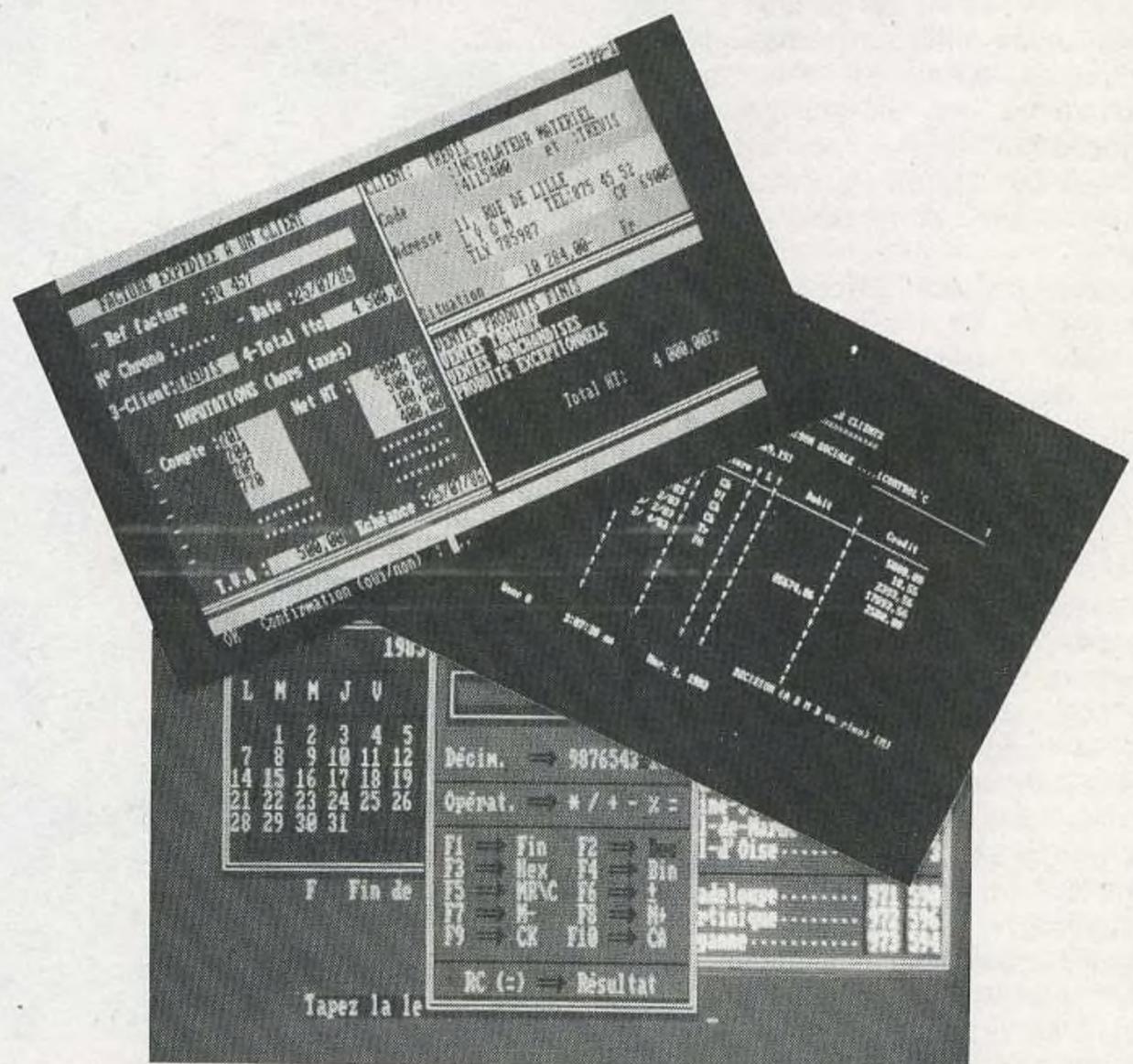
Enfin, la quatrième génération marque une étape importante dans la mesure où elle tente de répondre à une question fondamentale : à quoi sert l'informatique ? On peut également ajouter : à qui est-elle destinée ? A des spécialistes, les informaticiens, ou à tous ceux qui, dans leur vie professionnelle, ont besoin d'un outil de travail performant et simple à utiliser. A ses débuts, l'informatique était une affaire de spécialistes. Le développement d'une application de gestion ou de comptabilité impliquait la connaissance d'un langage comme le Cobol ou le Fortran que seuls les spécialistes maîtrisaient. Pour que le programme réponde parfaitement aux besoins du comptable par exemple, il fallait que celui-ci soit en contact permanent avec l'informaticien afin qu'il le guide dans le déroulement des opérations.

## DES OUTILS CONVIVIAUX

Pouvoir se passer de l'informaticien et réaliser son programme soi-même, voilà qui simplifie la tâche et diminue

les coûts. Ce n'est plus un rêve, mais une réalité depuis près de dix ans, date à laquelle sont apparus les premiers générateurs d'applications ; ces pro-

grammes permettent aux utilisateurs non informaticiens de développer eux-mêmes leurs propres programmes. Parallèlement les premiers systèmes de



# LANGAGE INFORMATIQUE

gestion de bases de données relationnelles en simplifiant les manipulations des données ont rendu l'informatique plus accessible aux non-informaticiens. Tous ces outils étaient destinés aux gros systèmes. Au cours des années suivantes et avec l'avènement de la micro-informatique, des systèmes adaptés aux micro-ordinateurs ont été développés. Et aujourd'hui, il n'est pas de jour ou presque sans qu'un générateur de programmes fasse son entrée sur le marché.

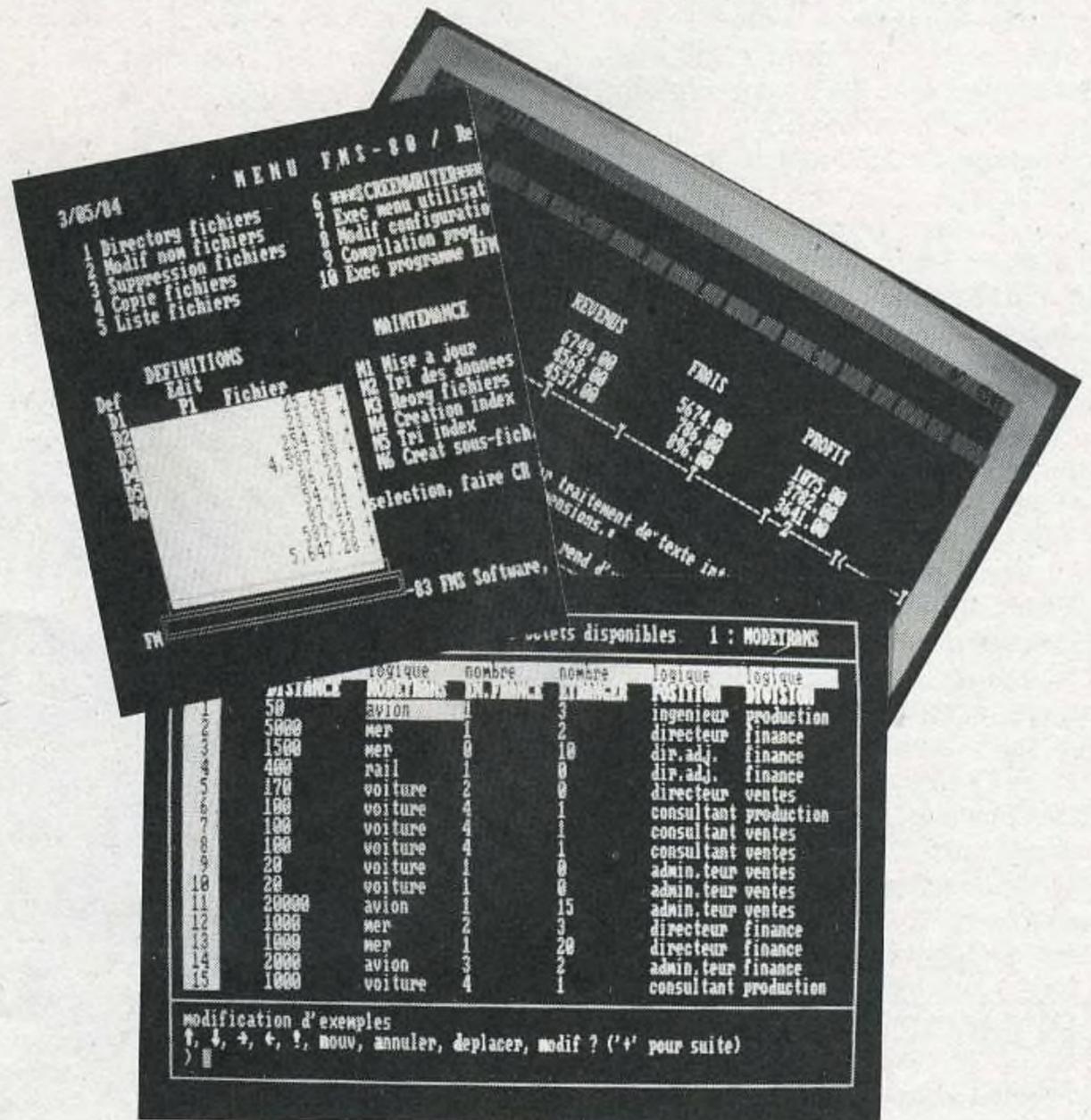
Progressivement l'informatique s'est rapprochée de l'utilisateur : elle est devenue conviviale. C'est d'ailleurs l'une des quatre caractéristiques de base de la IV<sup>e</sup> génération avec la non procéduralité, l'intégration et l'efficacité.

Ce qui distingue la quatrième génération c'est son approche « intelligente ». Le système réfléchit et choisit la meilleure façon d'obtenir le résultat désiré. Les langages de quatrième génération - les langages de manipulation des bases des données et de développement d'applications - ont rompu avec la tradition. Précédemment, on devait décrire pas à pas à l'aide d'instructions détaillées chaque étape de l'exécution d'un programme. Cette méthode est appelée procédurale.

Avec la quatrième génération, il n'y a plus à expliquer. Le logiciel à l'aide d'instructions générales ou de choix

d'options sur des menus décide de la procédure à suivre. C'est le mode non procédural. Il accélère et simplifie

l'interrogation ou la manipulation des bases de données ainsi que la conception et la réalisation des applications.



Mais il y a un inconvénient : il limite le champ d'application en raison de ses instructions générales, spécifiques à l'application visée. On ne peut pas faire, par exemple, de l'enseignement assisté par ordinateur avec un logiciel de la quatrième génération. En revanche, les langages procéduraux de la troisième génération sont universels, ce qui explique pourquoi ils sont toujours autant utilisés.

S'il est un mot à la mode en informatique aujourd'hui, c'est bien convivialité. Il est souvent utilisé à des fins «marketing» mais il recouvre aussi un souci d'améliorer le dialogue homme/machine en permettant aux utilisateurs de concevoir leurs propres applications. Les logiciels de la quatrième génération répondent bien à ce souci de convivialité, à la condition toutefois que le système parle la même langue que l'utilisateur et puisse réagir à ses erreurs par l'affichage de messages clairs indiquant ce qui ne va pas et comment se sortir d'un mauvais pas.

Par ailleurs, l'évolution s'est faite en deux directions : l'intégration et la portabilité. Les logiciels de la quatrième génération intègrent plusieurs fonctions assurées précédemment par divers programmes spécifiques : traitements de texte, tableur, graphique. Par exemple, il est possible d'utiliser des données statistiques d'une base de données pour réaliser avec un logiciel graphique des camemberts ou autres représentations graphiques.

Par portabilité, on entend la facilité d'implantation d'un logiciel d'une machine sur l'autre. Elle est fonction du langage de programmation. Si le logiciel a été écrit dans un langage évolué comme le langage C ou Unix, il sera aisément transportable sur d'autres systèmes. Pour ce faire, il suffit que ces systèmes supportent le même langage de développement. Le plus haut degré de portabilité est obtenu lorsque le logiciel est écrit dans son propre langage. Dans ce cas, une base de données unique gère aussi bien les mécanismes de base du logiciel que les données de l'utilisateur.

### **LANGAGES ET LOGICIELS**

Le terme de quatrième génération peut prêter à confusion car il s'applique aussi bien à des langages qu'à des logiciels. Un logiciel est un système qui réunit divers programmes dont un langage de la quatrième génération. On distingue deux types de logiciels appartenant à cette génération. Le premier regroupe les générations d'application. Le second rassemble les systèmes de gestion de base de données relationnelles dont la principale fonction est la gestion et la manipulation des données. Ils ont en commun de comprendre un langage de la quatrième génération. Un langage de la quatrième génération permet de définir la procédure d'application nécessaire lors du développement d'une application ou les

procédures à suivre dans un système de base de données (interroger la base de données ou la modifier). Donc avec un logiciel de la quatrième génération, il suffit de taper sur le clavier la procédure choisie : par exemple pour que la machine réalise l'ensemble des opérations conduisant à cette opération. Mais par exemple l'édition des résultats est effectuée par des outils de développement comme un générateur d'états. En plus, des logiciels utilitaires complètent le plus souvent ce logiciel : ils permettent de mettre l'information en forme. Ces logiciels sont les traitements de texte, les tableurs, les logiciels graphiques.

### **LES LANGAGES DE 4<sup>e</sup> GÉNÉRATION ET LA MICRO**

La micro-informatique en mettant l'informatique à la portée du plus grand nombre en raison de son faible coût a fait qu'il n'est plus aujourd'hui possible de concevoir de nouveaux systèmes informatiques sans penser micro-informatique. Les grandes entreprises avec la micro ont entrepris une décentralisation de leur informatique et les moyennes et petites entreprises peuvent s'équiper de systèmes n'impliquant pas des investissements trop importants. Planter un logiciel de la quatrième génération sur un micro est aujourd'hui tout à fait possible.

**C.R.**

# LE GUEPARD

(2ème partie)



Deux extensions réalisées par HBN pour le Guépard sont arrivées aux Editions Fréquences, il nous a paru intéressant de livrer nos impressions à nos lecteurs.

**M**

Malgré quelques petits défauts de jeunesse, le Guépard nous a laissé une bonne impression, c'est donc avec un certain plaisir que nous avons pris possession de deux nouvelles cartes. Celles-ci sont destinées à la partie Vidéo de cette machine, puisqu'il s'agit d'une extension moyenne résolution graphique couleur et d'un générateur programmable de caractères.

## **GRAPHIQUE MOYENNE RESOLUTION**

L'emballage protégeant des chocs cette extension contient une carte d'excellente qualité, une documentation et

une disquette renfermant le MR BASIC. Le MR BASIC est un complément au BASIC de base du Guépard qui permet de programmer et de visualiser des tracés graphiques avec une précision de 320 x 250 points. Ce MR BASIC est assez pauvre en nombre d'ordres utilisateur. Toutefois, les fonctions CIRCLE et PAINT seront beaucoup appréciées. L'efficacité des fonctions remplace la multitude des ordres de certaines extensions graphiques.

Nous avons regretté le manque d'exemples abondants. Les utilisateurs de l'ASSEMBLEUR seront comblés car la partie système est bien détaillée. Huit couleurs sont disponibles (noir,

rouge, vert, jaune, bleu, magenta, bleu ciel et blanc), la fonction PLOT est très performante. La mise en œuvre de cette extension ne pose pas de problème majeur. Les personnes hésitant à manipuler des cartes électroniques pourront faire configurer le graphisme dans l'un des 50 magasins HBN. Le processeur graphique utilisé est le TMS

3556 de chez TEXAS INSTRUMENT. Deux liaisons sont effectuées sur cette carte, une sur le bus de fond de panier, et une avec la carte vidéo d'origine. Le schéma de la carte ne pose pas de remarque. Il s'agit là d'un choix classique. Par contre, la qualité de fabrication est très au-dessus de la moyenne. Il s'agit d'un produit sûr et efficace.

## LE GENERATEUR DE CARACTERES

Le générateur de caractères programmables est une extension au système vidéo qui autorise la définition de symboles autres que ceux représentés par le code ASCII. Qui n'a jamais essayé d'afficher à l'écran des caractères grecs qui sont amplement utilisés en mathématique ? Il n'est pas facile d'écrire une intégrale sur un écran cathodique. L'extension générateur de caractères programmables résout ces problèmes. Si sa mise en œuvre ne pose pas de difficulté, la programmation nécessite un peu de méthode. Un logiciel accompagne cette carte. Il permet de réaliser 7 fonctions principales :

- la création d'un jeu de caractères
- le chargement
- la modification
- la duplication
- l'effacement
- la liste de fichiers de caractères
- le retour au DOS.

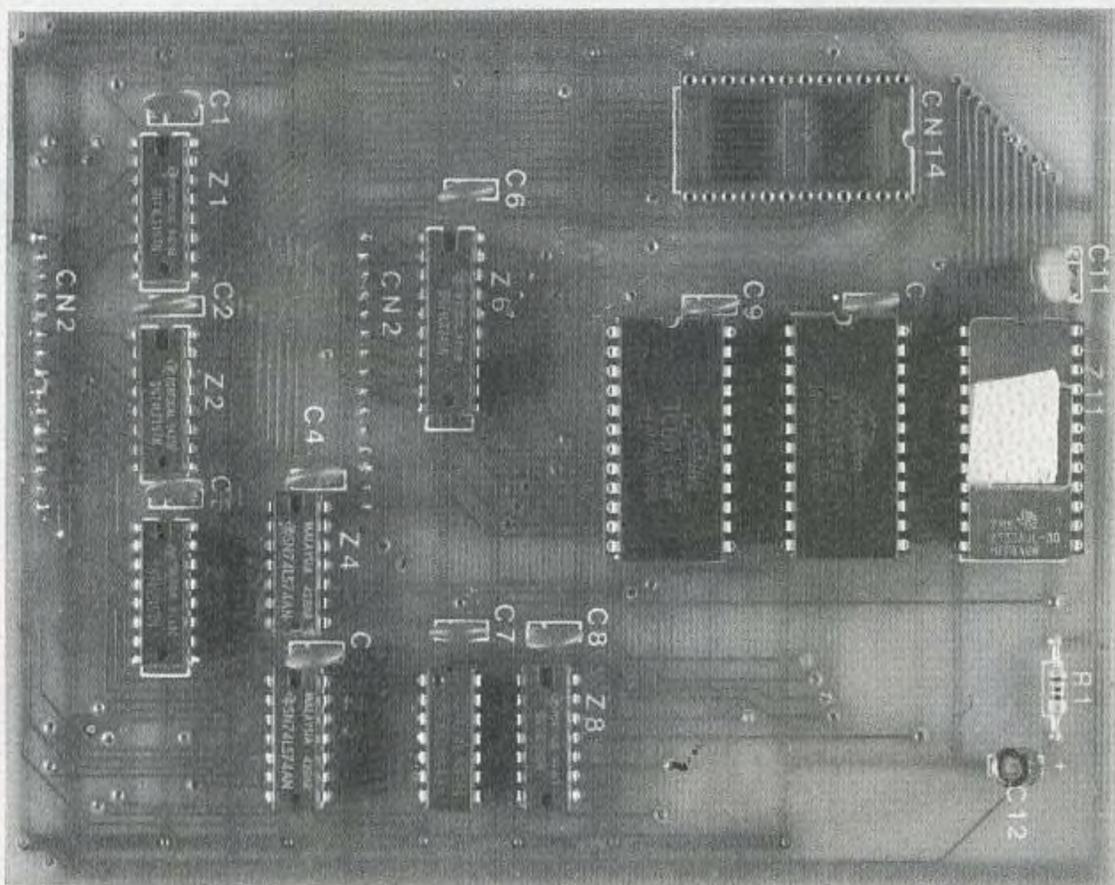
La création permet de réaliser un jeu de 256 caractères.

Si l'affichage écran s'opère sur 24 lignes de 80 caractères, la structure d'un caractère est composée d'une matrice de points rangés en 8x12 pixels (8 en horizontal, 12 en vertical). Il faut donc 12 octets pour réaliser un caractère. La recopie de ces caractères se fait en mode graphique sur une imprimante. Seules les imprimantes équipées d'un mode graphique pourront être utilisées. Un ensemble d'utilitaires contenus dans une disquette accompagnant la carte avec un manuel d'utilisation permet d'effectuer les différentes tâches nécessaires à la gestion de l'ensemble. La partie HARD ne nécessite aucun commentaire et sa mise en œuvre ne pose pas de problème majeur.

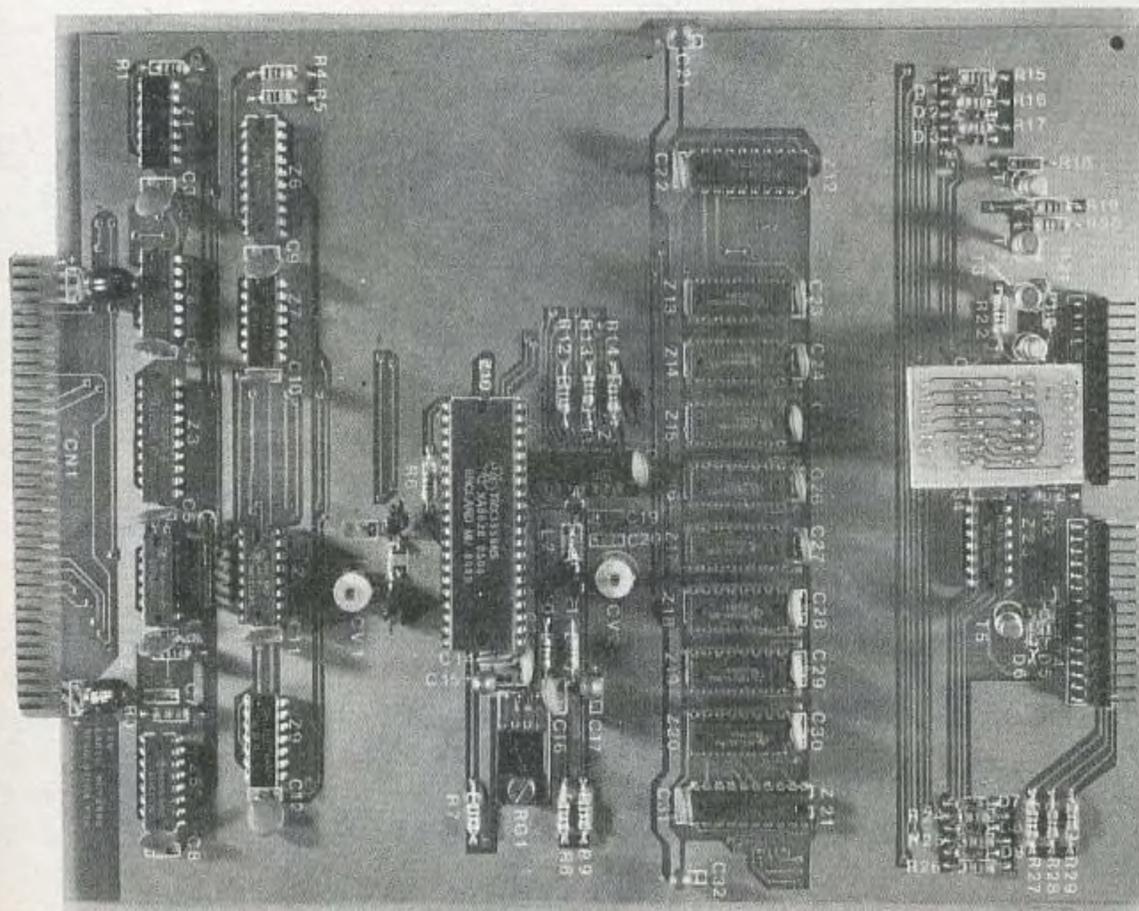
## CONCLUSION

Ces deux extensions sont de bonne qualité, leur construction est même exemplaire, on regrettera seulement une documentation un peu aride en BASIC. Il s'agit de matériel professionnel à un prix amateur. Il est à noter que HBN annonce la sortie prochaine de plusieurs progiciels pour petites et moyennes entreprises très compétitifs.

C.H. Delaleu



La carte graphique moyenne résolution : une belle fabrication, une fiabilité certaine.



La carte générateur de caractères programmable : compacte mais efficace...



## OLIVETTI M21

**L** Après avoir fait parler d'elle, Olivetti est devenue aujourd'hui une société prospère. Il y a peu de temps, le public apprenait un certain rapprochement avec la grande firme américaine ATT. Depuis, une gamme de micro-ordinateurs a fait son entrée sur le marché très prisé des compatibles IBM-PC. C'est le M21 que nous avons choisi pour ces contre-mesures.

Le M 21 est un micro-ordinateur dit «transportable». Il est certain que son ergonomie autorise de nombreux déplacements, toutefois son poids respectable (14 kg) ne lui permet pas l'appellation de «portable». L'appareil se compose de deux blocs : d'une part le clavier, d'autre part l'unité centrale comprenant l'écran ainsi que les unités de disquettes 5" 1/4. Il est possible

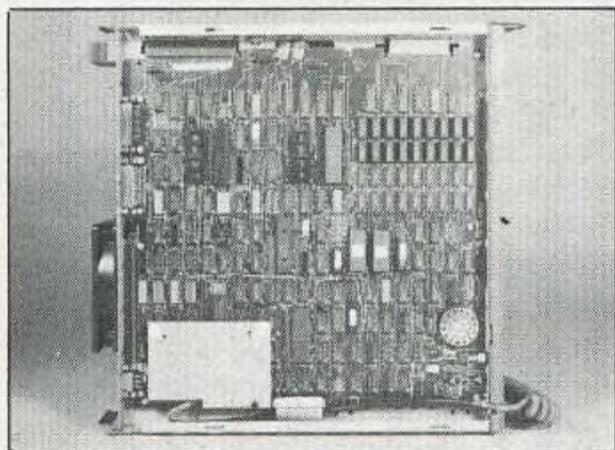
d'opter entre un ou deux lecteurs, de 360 ou 720 koctets, soit un maximum de 1,5 mega-octet en ligne.

L'unité de base comporte également trois emplacements libres pour des cartes compatibles. Il est ainsi possible de connecter toutes sortes de périphériques dont une unité de disques rigides de 10 mega-octets, un écran graphique couleur, etc.

## LA COMPATIBILITE

L'Olivetti M 21 garantit la compatibilité opérationnelle avec le standard IBM-PC. Cela signifie que tous les programmes écrits sous MS-DOS peuvent être utilisés. Il est possible d'ajouter les systèmes d'exploitation CP/M-86 et UCSD.

Le microprocesseur utilisé est le 8086



La carte-mère du M 21. La densité de puces au dm<sup>2</sup> est confortable, c'est une fabrication claire et soignée.

de chez Intel tournant à une fréquence horloge de 8 MHz. L'IBM-PC est équipé, lui, du 8088 (4,7 MHz). Pour être précis, nous dirons que le 8088 est un processeur architecturé en 16 bits, mais il possède un bus d'entrées-sorties (DATA) de 8 bits. Le 8086, lui, est configuré en 16 bits internes et 16 bits en DATA. De ce fait, le M 21 peut être désigné comme un véritable 16 bits, alors que l'IBM-PC est un faux 16 bits. Cette différence associée à la fréquence d'horloge autorise l'Olivetti à être beaucoup plus rapide que l'IBM-PC. En effet, le M 21 fonctionne de deux à deux fois et demie plus vite que l'IBM-PC. Le 8086 existe en trois versions de base : 5,8 et 10 MHz. Il appartient à la famille des HMOS. Il est capable d'adresser 1 Moctet de mémoire vive. Il possède 24 modes d'adressage et est compatible avec le format MULTI-BUS. De ce fait, le 8086 est un microprocesseur complet et assez performant.

Malgré ces différentes hardwares, la compatibilité est parfaite au niveau logiciel, ainsi qu'en ce qui concerne les cartes d'extensions dites compatibles IBM-PC.

## LA MACHINE

En version de base, le M 21 est équipé de 16 koctets de mémoire ROM et de 128 koctets de mémoire RAM. Il est équipé d'une interface série asynchrone (RS 232), d'une interface parallèle (Centronics), d'un contrôleur DMA (accès direct à la mémoire RAM), et de divers emplacements.

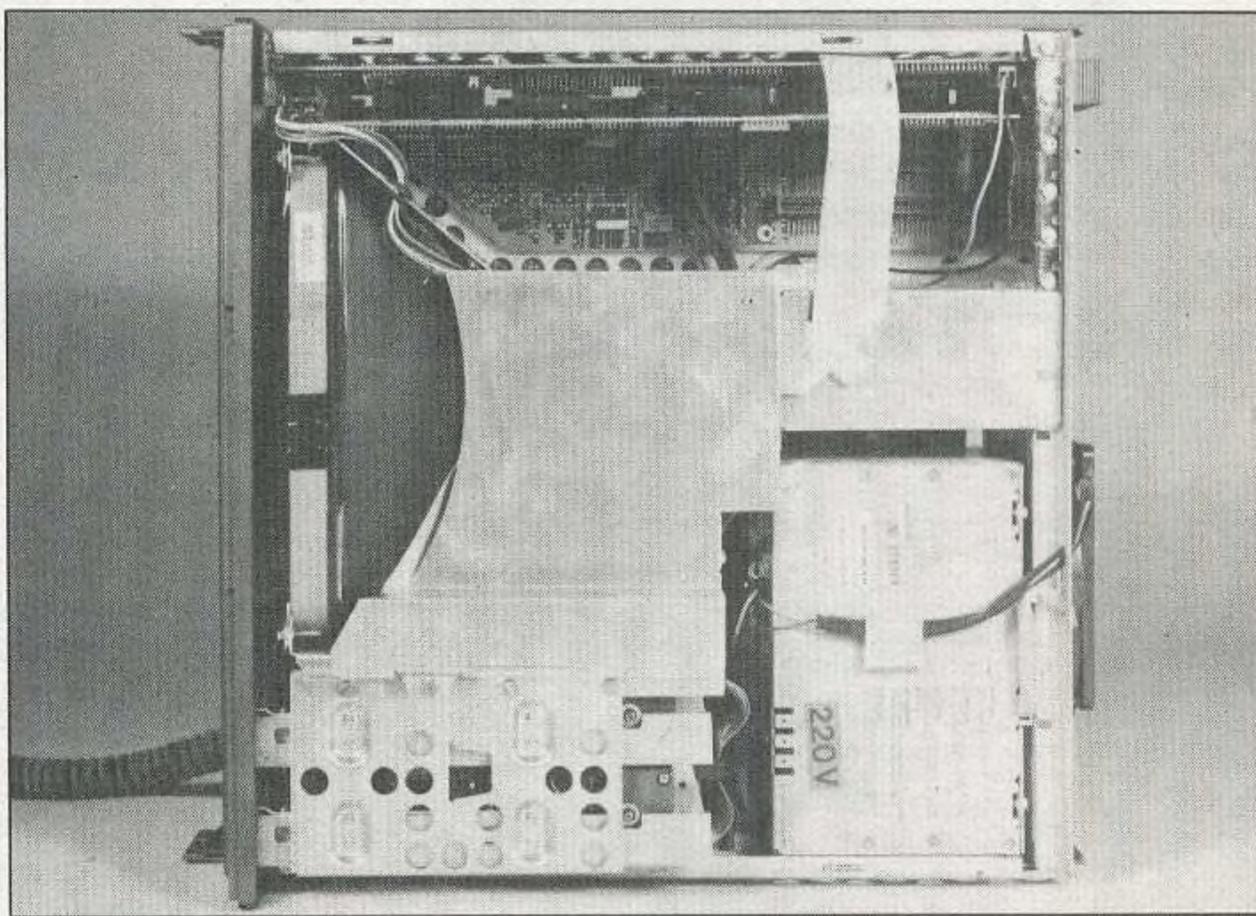
Le contrôleur graphique-alphanumérique dispose d'un mode monochrome ou couleur. Selon l'écran, quatre nuances simultanées (quatre couleurs) sont à choisir sur une palette de seize. Un générateur de sons ainsi qu'une horloge avec calendrier sur batterie accompagnent l'ensemble. Plusieurs options sont disponibles. Il existe un coprocesseur numérique 8087, des bits d'extension mémoire de 128 koctets, une interface de communication asynchrone et synchrone, un convertisseur de bus avec trois emplacements libres destinés à des cartes compatibles. La partie graphique peut être dopée grâce à une carte qui permet

d'afficher simultanément 16 nuances ou 16 couleurs. Enfin, notons la présence d'une carte réseau de type OMMINET/ETHERNET ainsi qu'une carte Z 8000.

Le clavier est bien agencé, son utilisation est agréable et rapide. Il possède un dispositif de signalisation tactile et acoustique très pratique. L'écran est un 9'', monochrome, phosphore ambre. Il dispose de 25 lignes de 80 ou 40 caractères. La définition graphique est soit 640x400, soit 640x200, soit 320x200 pixels, selon le mode de fonctionnement.

## SYSTEME D'EXPLOITATION

Si, en ce qui concerne la «norme» IBM-PC, le système d'exploitation est le MS-DOS, il est possible d'utiliser le C-CP/M-86, le PCOS (nécessitant la carte Z 8000 et un écran 12''), et UCSD P-System. De la sorte, le M 21 pourra sans aucun problème être utilisé dans d'autres environnements que MS-DOS. Ainsi l'accès à UCSD P-



La partie supérieure du M 21, notez la présence de nombreux blindages. Sur la gauche de l'appareil, le fond de panier où deux cartes supplémentaires peuvent être insérées.

# CONTRE-MESURES

System comblera les amoureux du développement logiciel. Il est donc possible d'utiliser l'importante bibliothèque écrite sous Pascal UCSD.

## LA VITESSE D'EXECUTION

Nous avons commencé à utiliser le M 21 avec dBase III (décrit dans ce numéro), les temps de réponse sont très sensiblement différents de ceux rencontrés sur l'ensemble de la famille

des compatibles IBM-PC. Il n'y a aucun doute, cette machine possède ici un atout majeur. Le second essai a consisté à effectuer différentes boucles sous BASIC. Une simple boucle FOR-NEXT de 10 000 passages prend 5 secondes. Il en faut près de 12 pour l'IBM-PC. Sans entrer dans les records de la micro-informatique, le M 21 est sans doute le plus rapide dans cette famille de compatibles. La compatibilité des progiciels écrits pour IBM-PC est ici conforme à cent pour cent.

## LA DOCUMENTATION

Nous n'avons pas pu disposer en temps voulu de la documentation et de la notice d'emploi accompagnant le M 21. De ce fait, nous n'avons pu nous faire une opinion sur la qualité de ces dernières. Il semble toutefois qu'elles soient similaires à celles de nombreux constructeurs (Victor, Guépard HBN, etc.).

Notons que Olivetti dispose d'un centre de formation assez complet. Il est possible, outre la machine et les langages, de suivre des stages sur de nombreux progiciels. Nous avons retenu : Multiplan, Word, dBase III, Framework, Symphony, Open Access, Textor.

La méthode pédagogique est basée sur des séquences d'auto-formation complétées par l'aide régulière d'un animateur.

Chaque groupe se compose d'un maximum de six personnes disposant chacune d'un micro-ordinateur. Les tarifs sont comparables à l'ensemble des autres constructeurs.

## LE PRIX

Si le M 21 surprend par sa vitesse d'exécution, il convient de noter que son prix est tout aussi intéressant. Il s'agit donc dans cette gamme de produits du meilleur rapport qualité/prix actuellement disponible sur le marché.

## CONCLUSION

Ce micro-ordinateur transportable nous a laissé une excellente impression. Ses performances sont nettement au-dessus de la moyenne, son prix est justifié et sa fabrication soignée. Si vous deviez acheter un compatible IBM-PC transportable, c'est sans aucun doute le M 21 qui serait votre meilleur choix.

C.-H. Delaleu

### Unité de base

- Microprocesseur 16 bits: 8086 (8 Mhz).
- Emplacement pour un coprocesseur numérique optionnel 8087.
- Mémoire morte de 16 Koctets pour le chargement du système, le diagnostic à la mise sous tension et le BIOS.
- Mémoire vive de 128 Koctets.
- Emplacement pour 128 Koctets de mémoire supplémentaire.
- Une ou deux unités de minidisquettes de 360 ou 720 Koctets.
- Contrôleur pour écran alphanumérique et graphique monochrome ou couleur.
- Selon l'écran, quatre nuances simultanées (ou quatre couleurs) à choisir sur une palette de 16.
- Interface série asynchrone (RS 232).
- Interface parallèle (Centronics).
- Emplacement pour un circuit optionnel de communication.
- Contrôleur DMA (accès direct en mémoire).
- Générateur de sons.
- Horloge et calendrier sur batterie.

### Clavier

- Configuration compatible de 83 touches.
- Capot de protection détachable.
- Profil bas.
- Touches sculptées.
- Dispositif de signalisation tactile et acoustique.
- Répétition automatique de caractères.

### Ecran

- De 9 pouces intégré dans l'unité de base.
- Monochrome, phosphore ambre.
- 25 lignes de 80 ou 25 lignes de 40 caractères.
- Graphique haute résolution: 640x400, 640x200 ou 320x200 pixels, selon le mode de fonctionnement.
- 16 nuances.
- Réglage de la luminosité.
- Surface anti-reflets.
- En option, écran graphique de 12" monochrome ou couleur.

### Options internes

- Coprocesseur numérique 8087.
- Kit d'extension mémoire de 128 Koctets.
- Interface de communication asynchrone et synchrone (chip 8530).
- Convertisseur de bus avec 3 emplacements libres destinés à des cartes compatibles.
- Carte mémoire de 128 Ko prédisposée à recevoir 2 autres Kits d'extension mémoire de 128 Ko.
- Carte graphique pour l'affichage simultané de 16 nuances ou de 16 couleurs.
- Carte pour une seconde extension de communication.
- Carte pour l'interface série double.
- Cartes OMNINET/ETHERNET.
- Carte Z8000.

### Imprimantes

- Matricielle à impact 80 ou 132 colonnes.
- Copie d'écran avec prestation graphique.
- Qualité courrier.
- Machines à écrire électroniques.

### Systèmes d'exploitation

- MS-DOS.
- C-CP/M-86.
- PCOS (nécessite la carte Z8000 et l'écran 12").
- UCSD P-System.

### Dimensions et poids

Largeur: 396 mm  
Hauteur: 210 mm  
Profondeur: 425 mm  
Poids: 14 kg

MS-DOS est une marque déposée de Microsoft Corporation.  
UCSD P-System est une marque déposée de l'université de Californie.  
Concurrent CP/M-86® est une marque déposée de Digital Research.  
OMNINET est une marque déposée de Corvus Systems, Inc.

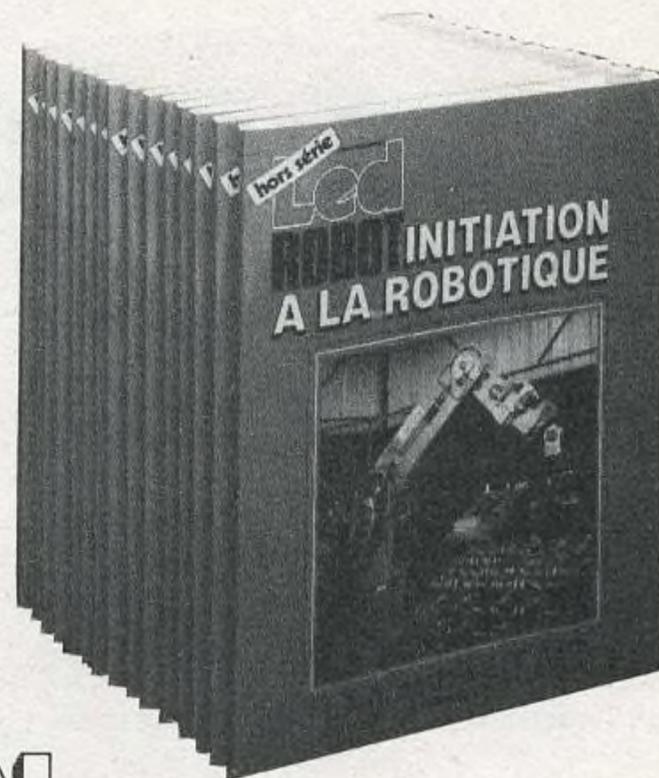
### Option externe

- Disque rigide de 10 Mo.

Caractéristiques générales du M 21. (Doc. Olivetti).

**VOICI ENFIN LA PREMIÈRE PIERRE  
D'UN DOMAINE ENCORE INEXPLORÉ...**

L'ouverture au monde passionnant de la robotique, dans un style simple et direct, travail d'un collectif de spécialistes animé par Claude Polgar.



**hors série**

**PRIX TTC 115 F**

**Led**  
**ROBOT**

# INITIATION A LA ROBOTIQUE

**Format 21 x 27, 100 pages, plus de 130 schémas et illustrations.**

## Le sommaire : une somme !

- **La grande relève des hommes par les robots**
- **L'anatomie de HERO 1** : bras, jambes, ouïe, vue, télémétrie, détection de mouvements.
- **Inventeurs et inventions** : ne confiez pas vos inventions avant de vous être protégé.
- **Cours de conception mécanique** : vocabulaire et notion de base - Ajustement, tolérance, excentricité, etc.
- **Cours de logique générale** : schémas et symboles.
- **Electronique industrielle** : du circuit au démultiplexeur.
- **Vie industrielle** : la CAO, assistante de la création.
- **Conception et construction** : de la tortue au robot.
- **Modules fonctionnels** : construction de la carte de départ pour commander les moteurs pas à pas à partir de votre micro.
- **Maquettes et modélisme** : le modélisme ferroviaire se renouvelle grâce à la micro-informatique.
- **Analyses et méthodes** : les rosaces d'évaluation.

## BON DE COMMANDE

Je désire recevoir Led-Robot «INITIATION A LA ROBOTIQUE» (attention, cet ouvrage n'est pas vendu en kiosque) au prix de **125 F** (port compris).

Nom : ..... Prénom : .....

Adresse : .....

**ATTENTION** : Si je suis abonné soit à LED, soit à LED-MICRO, je bénéficierai d'une réduction de 20 % sur le prix de l'ouvrage et je ne paierai que **100 F** (port compris).

Je vous note, dans le cadre, mon numéro d'abonné :

Ci-joint un chèque bancaire  chèque postal  mandat .

Adressez votre commande et votre règlement aux **EDITIONS FRÉQUENCES 1**, boulevard Ney, 75018 Paris.

## DBASE III

Après avoir testé Multiplan et Word, voici le dernier progiciel appartenant à ce trio réputé dont le but est d'être l'outil de base de la micro-informatique.

**D**

Base III est la dernière version de la famille DBase. En comparaison avec DBase II, DBase III autorise la gestion d'un fichier beaucoup plus important. De même, sa vitesse d'exécution a été largement optimisée. Un gestionnaire de fichiers peut servir à de nombreuses applications. L'utilisation la plus simple concerne la réalisation de fichiers d'adresses. Mais ce serait bien pauvre si le constructeur de DBase n'avait ajouté à ce module de base de nombreux utilitaires tels que les tris qui, dès lors, permettent de larges champs d'applications.

En fait, la majorité des gros ordinateurs utilisés en gestion passe quatre-vingt dix pour cent de leur temps à effectuer des travaux sur fichiers. Ceux-ci d'ailleurs sont essentiellement axés sur des tris. Le logiciel a été réécrit en langage C. Le nombre d'enregistre-

ments possibles est passé à 2 milliards. Egalement dix fichiers peuvent être ouverts en même temps.

Ashton Tate qui fabrique ce progiciel, n'a pas hésité sur la documentation qui accompagne ce programme qui est présenté comme une base de données.

Ce terme nous paraît un peu galvaudé, car une base de données, en informatique, désigne un ensemble infiniment plus important que DBase III. Nous choisirons, pour notre part, l'appellation «gestionnaire de données».

DBase III est un système de gestion de données de type relationnel, il utilise des commandes simples et puissantes. Il est possible d'utiliser le mode interactif ou le mode programmé. Il existe 128 champs et 4 000 caractères maximum par enregistrement. Les tris sont rapides et fonctionnent sur plusieurs champs.

## UTILISATION

La prise en main est assez aisée, en effet lors de l'approche avec DBase III, on ne perd pas le temps habituel qu'occasionne la majorité des autres progiciels. Une technique simple dans ce genre de manipulations consiste à mettre en route le programme principal sans avoir au préalable regardé la documentation. Si le programme est bien construit, tout doit se passer facilement. Sur une configuration classique, la disquette system sera insérée dans le lecteur A et les utilitaires en B. Il suffit dès lors de frapper : DBase puis Return. Un effort important a été

effectué sur la facilité d'utilisation de ce progiciel. Un mode d'assistance permet à l'utilisateur néophyte de manipuler immédiatement les données de ses fichiers. En effet, par le biais de menus, auxquels sont associés des explications, vous pouvez créer votre fichier de données, ajouter, modifier, trier ou indexer des enregistrements et également imprimer des états représentant les enregistrements sélectionnés. La seconde fonction de ce mode assistance est de constituer au fur et à mesure des choix de l'utilisateur, la ligne de commande qu'il aurait été nécessaire de frapper en mode direct. Un système d'aide structuré vient compléter cet aspect ergonomique du logi-

ciel. Il permet d'afficher immédiatement les explications correspondantes à une erreur dans l'utilisation d'une instruction ou bien en fonction des choix et spécifications.

## LA DOCUMENTATION

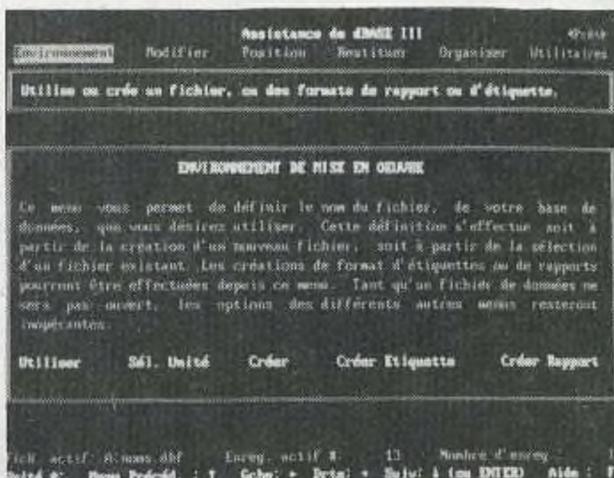
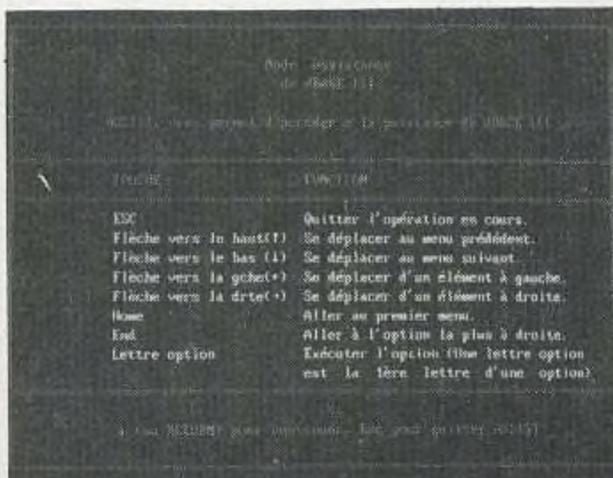
C'est près de 500 pages qui composent le classeur de documentation de DBase III. Cette documentation est divisée en trois parties principales :

- formation
- référence
- annexe.

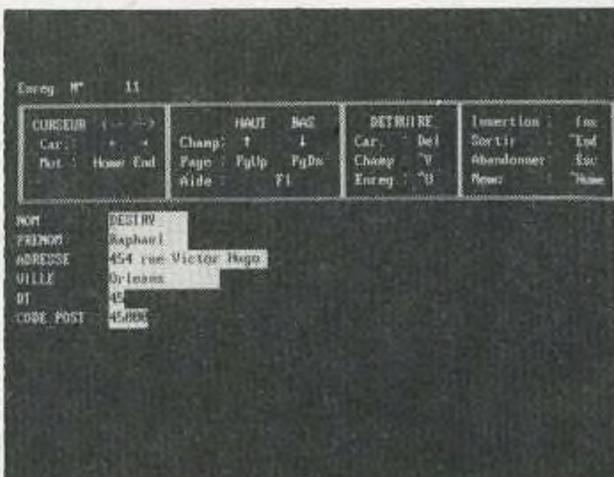
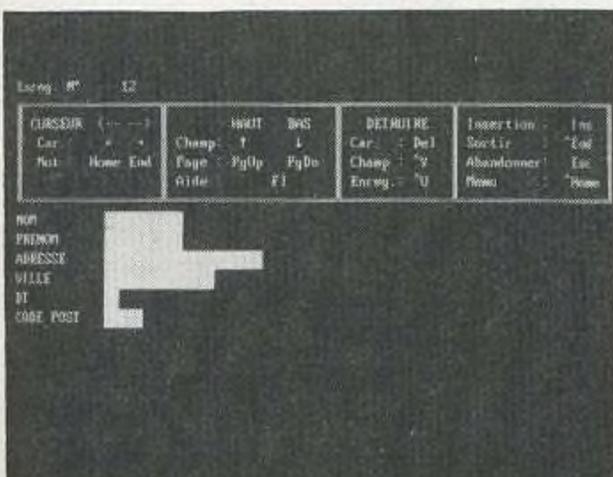
Chaque concept est repris en détail, la partie machine et la partie documentation sont clairement exprimées par l'utilisation d'une couleur verte permettant de visualiser les commandes et l'affichage écran.

## LA GESTION DE FICHIERS

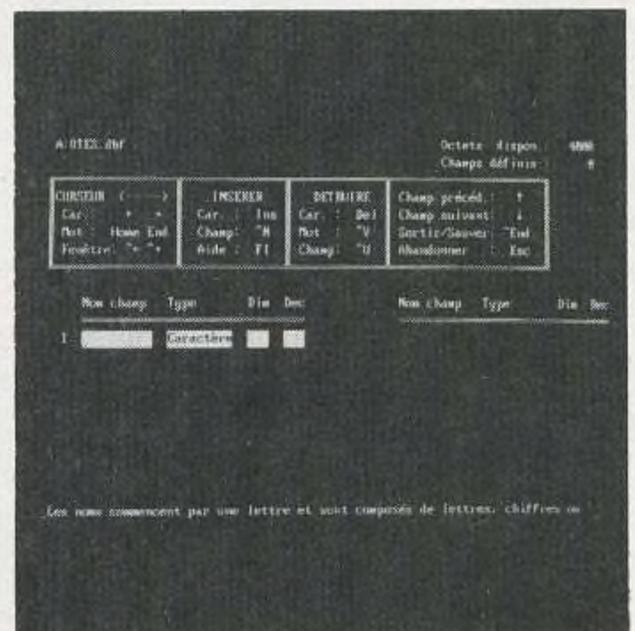
Une base de données est une collection (un ensemble) d'informations concernant un ou plusieurs sujets. Chaque



Grâce à un mode d'assistance, il est assez facile d'accéder à l'ensemble des fonctions de dBase III. Les possibilités de ce concept sont puissantes et fort bien réalisées. a) Le menu maîtrise l'ensemble des possibilités de dBase III. b) Ce menu permet de créer un fichier. Il est possible de partir d'un fichier existant ou de configurer l'ensemble des variables.



Enregistrement composé de six champs : une série de fenêtres autorise une gestion parfaite des chaînes de caractères. Avant saisie en a, après saisie en b.



Programmation d'un champ : a) le nom du champ, b) son type (caractère numérique), c) la taille, d) pour les nombres décimaux, le nombre de décimales.

# PROGICIELS A L'ESSAI

enregistrement correspond en fait à une fiche dans laquelle l'auteur place des informations. Celles-ci sont divisées en champs. Un champ est une rubrique qui désigne soit un nom, soit un taux de T.V.A., etc.

Dans le cas d'un médecin, on trouvera ainsi dans son fichier client un enregistrement par patient où les différents champs comprendront le nom, le prénom, l'adresse, le numéro de téléphone, etc.

Dans une base de données, les différentes informations pourront être manipulées, sélectionnées, triées, imprimées suivant le cahier des charges de l'utilisateur, les données et les traitements s'y rapportant sont complètement indépendants.

DBase III est à la fois un progiciel de traitement de données, mais aussi un langage de programmation. Peu de personnes utilisent cette seconde possibilité. Elle permet de créer ses propres séquences d'instructions.

## CONCLUSION

Le succès de DBase II est indéniable. Il est fort probable que DBase III aura pour les utilisateurs de micro-ordinateurs IBM-PC le même sens. Ce progiciel peut être interfacé sans problème à Framework du même constructeur.

Rappelons que Framework est un progiciel intégré comprenant une table des matières électroniques, un tableur, un gestionnaire de fichiers, un outil graphique, un traitement de texte, un accès DOS, un cadreur, un outil de communication et un outil de programmation.

Ainsi, en maniant ces deux progiciels, l'utilisateur dispose de l'ensemble des outils nécessaires en bureautique.

C.-H. Delaleu

```

DISPLAY STRUCTURE
Structure du fichier : A:\nomm.dbf
Nombre d'enregistrements: 13
Dernière mise à jour : 09/04/85
Champ  Nom champ  Type      Dim  Dec
-----  -
1  NOM      Caractère 10
2  PRENOM   Caractère 10
3  ADRESSE  Caractère 20
4  VILLE   Caractère 14
5  DT      Caractère  2
6  CODE POST Caractère  5
** Total **                62
    
```

Structure d'un enregistrement.

## LES CARACTERISTIQUES TECHNIQUES

Plus d'un milliard d'enregistrements par fichier.

128 champs par enregistrement.

Champs de longueur variable (jusqu'à 5 000 caractères).

4 000 caractères par enregistrement de longueur fixe.

10 fichiers de données ouverts au même instant.

256 variables mémoires.

Précisions des valeurs numériques sur 19 chiffres et 15 décimales.

Enreg. N°	NOM	PRENOM	ADRESSE	VILLE	DT	CODE_POST
1	MARTIN	Jean	86 allée des fleurs	Brest	29	29200
2	DURAND	Lionel	49 rue du port	Calais	62	62100
3	LEMAITRE	Sabine	45 place de l'étang	Rouen	76	76000
4	ROUX	Jacques	98 rue du petit bac	Royan	17	17200
5	DUPONT	Charles	2 allée des orangers	Lyon	69	69002
6	LEGRIS	Arthur	65-rue Bonaparte	Cherbourg	50	50100
7	BLANC	Joseph	57 square Mistral	Cannes	06	06400
8	BEAUFIL	Christian	27 route des écoles	Laon	02	02000
9	PREMIER	François	le château	Blois	41	41000
10	PHILIPPE	Louis	39 rue du Roi	Troyes	10	10000

Listings d'un fichier, comprenant plusieurs enregistrements d'une série d'adresses.

## LES DIFFERENCES ENTRE DBASE II ET DBASE III

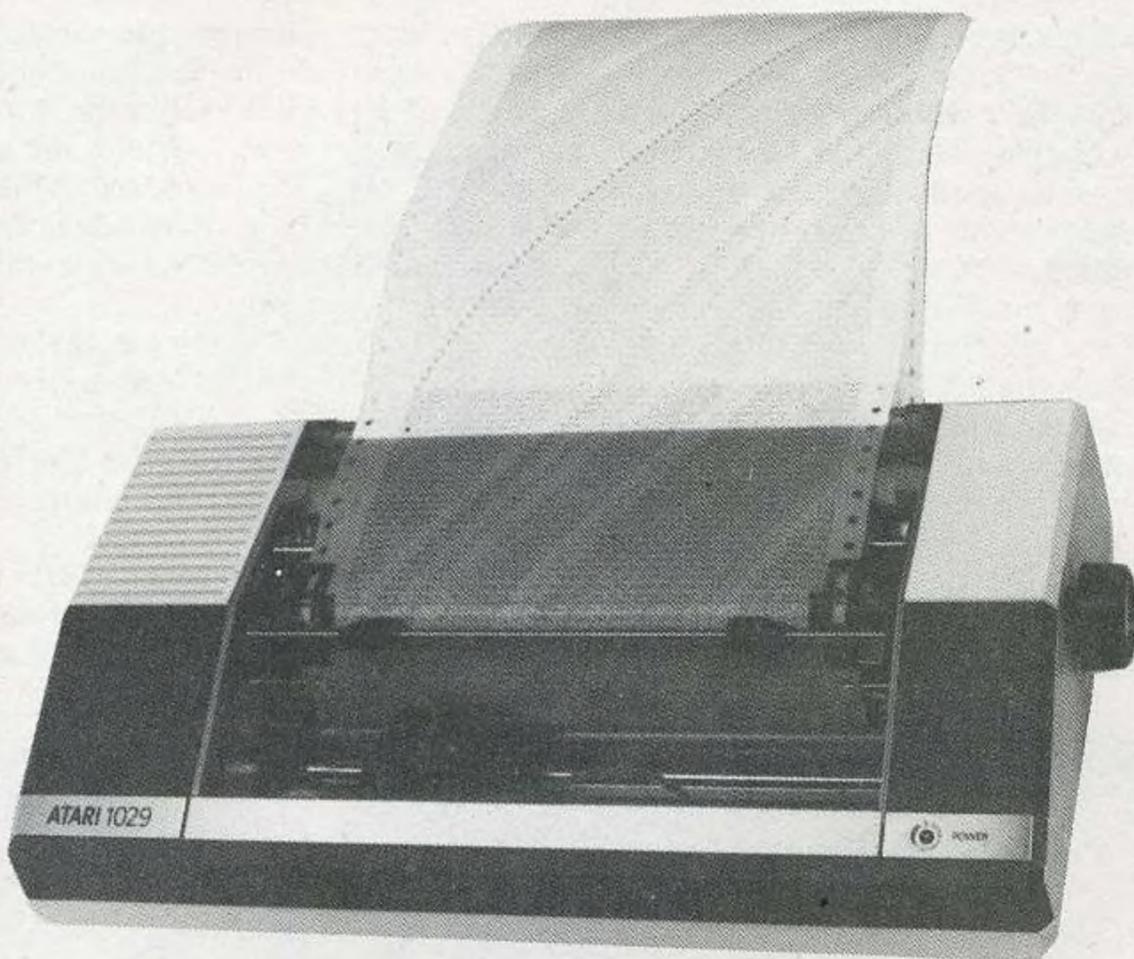
	dBase II	dBase III
Nombre de champs par enregistrement	32	128
Nombre de caractères par enregistrement	1 000	4 000
Nombre d'enregistrements	65 535	1 000 000 000
Nombre de fichiers ouverts	2	10
Précision numérique des champs	10	19

Fabricant : Ashton-Tate

Importateur : La Commande Electronique.

## UNE IMPRIMANTE ABORDABLE

Atari commercialise une nouvelle imprimante matricielle à aiguilles, la 1029, dotée d'une vitesse de 50 cps, qui convient fort bien à un usage personnel domestique et qui est vendue à un prix très abordable.



## CODEWRITER : UNE GAMME DE GENERATEURS DE PROGRAMMES

La société Sofitec distribue Codewriter, une gamme de générateurs de programmes d'origine américaine, composée de sept produits disponibles sur IBM et compatibles, Commodore, Apple, Apricot... Les sept produits sont disponibles actuellement. Trois d'entre eux, Filewriter, Reportwriter et Menuwriter, peuvent être vendus sous forme d'un

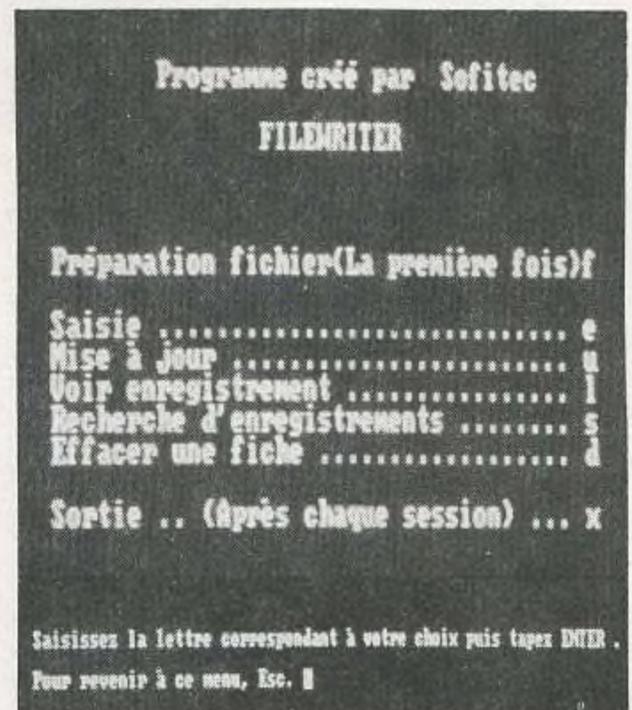
seul produit dénommé Codewriter. Ces produits sont francisés (programme et manuel).

Codewriter gère, crée et manipule des fichiers. Il réclame une capacité mémoire de 64 ko et il est écrit en deux versions : l'une pour SRM, compatible, Victor et Apricot, l'autre pour Apple, Atari et Commodore 8000 et 64.

Filewriter sert à écrire un programme en Basic, Reportwriter réalise des états statistiques à partir des programmes générés par Filewriter. Menuwriter est destiné aux possesseurs des deux autres produits et sert à écrire des menus.

La gamme comporte trois autres produits. Adventurewriter, un programme permettant d'écrire et de faire fonctionner sur un ordinateur son propre jeu d'aventure. Il est écrit en Assembleur pour IBM, Apple, Atari et Commodore 64. Dialog est un système de création de présentation personnalisée de textes pour Apple IIe et IIc et Commodore 64.

Elf fournit un programme de feuilles de calcul électronique pour Commodore 8000 et 64. Tous ces produits demandent une mémoire minimum de 64 ko. Les prix varient selon les produits et la version. Ainsi, Codewriter sur Atari et Commodore est commercialisée environ 1 000 F TTC, sur Apple IIe, IIc et Commodore 8000, 2 300 F, sur IBM et compatibles, Victor et Apricot 2 800 F. Sofitec a été créée en 1981 par Alain Boutoult ; elle a une double activité : distribution d'Apple et IBM essentiellement auprès des grandes entreprises et importation et distribution de Codewriter en France et dans les pays francophones. La distribution de Codewriter devrait permettre de quasiment tripler le chiffre d'affaires qui s'est élevé en 1984, à 5,2 millions de francs ; Codewriter devant représenter 30 à 40 % d'un chiffre d'affaires prévisionnel estimé à 15 millions de francs. En ce qui concerne la distribution, Alain Boutoult a choisi deux créneaux privilégiés : les boutiques micro et les centrales d'achats.



## GRIDCASE : LE PORTATIF DES TEMPS MODERNES

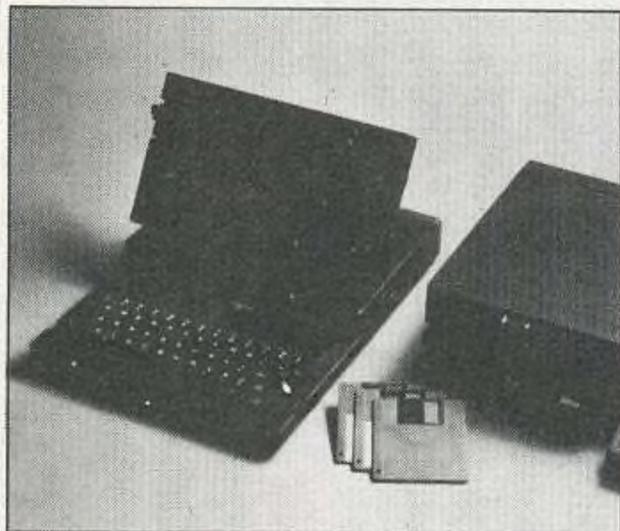
La société américaine lance une nouvelle gamme de portables, particulièrement attrayante par ses possibilités

techniques :

Trois écrans de 25 lignes de 80 caractères, un écran plasma et deux écrans à cristaux liquides (LCD).

Des options multiples : co-processeur arithmétique 8087 pour des applications graphiques ou d'ingénierie, RAM et ROM de 128, 256 ou 512 ko, sortie vidéo couleur, un modem intégré pour le réseau téléphonique, alimentation secteur/batterie amovible pour autonomie.

Tous les Gridcase sont équipés d'un



microprocesseur principal Intel 8086 et d'un lecteur de disquettes 3" 1/2 de 720 ko intégré.

Une gamme de périphériques superposables : lecteurs de disquettes et disques durs ; des logiciels avancés intégrés ; facilement utilisables, les 17 logiciels intégrés Grid sont adaptables à toutes applications, industrielles ou de gestion ; une compatibilité totale IBM. En outre, Grid propose un réseau organisé autour de son Grid Server. Il gère 58 postes qui peuvent, en local ou à distance communiquer entre eux, accéder aux bases de données aux systèmes centraux ou à des réseaux. Selon le type de configuration et d'écran, ces portables valent entre 35 000 et 65 000 F.

---

## TOUT NOUVEAU, LE GOUPIL G4, COMPATIBLE IBM

---

La SMT annonce la commercialisation du micro-ordinateur Goupil G4, pre-

mier modèle d'une nouvelle gamme qui vient compléter et enrichir la gamme actuelle du G3 PC.

Sa conception et son lancement aujourd'hui s'inscrivent dans la stratégie produit développée par la SMT Goupil qui s'articule autour de trois axes principaux : compatibilité avec les grands standards de fait du marché, intégration dans une dynamique systèmes et télécommunications, environnement convivial (logiciels intégrés, fenêtres, souris, etc.).

Les principales caractéristiques sont les suivantes : le Goupil G4 est *entièrement compatible PC («hard» et «soft»*), c'est-à-dire capable d'exploiter immédiatement tous les grands logiciels du marché et de recevoir des cartes électroniques optionnelles au format IBM.

C'est un micro-ordinateur performant grâce à son microprocesseur Intel 80186 tournant à 8 MHz.

Il est doté en standard de l'ensemble des fonctionnalités indispensables à toute utilisation professionnelle, et notamment d'une mémoire vive extensible à 512 K sans ajout de carte, des interfaces série et parallèle, d'une carte d'affichage gérant les écrans monochromes

ou couleur.

Il bénéficie par ailleurs d'un design et d'une ergonomie soigneusement étudiés, afin qu'il s'intègre harmonieusement à divers types d'environnements et offre le plus grand confort d'utilisation. Goupil G4 est disponible à partir du 15 avril 1985.

---

## UN MSX

### A LA MODE COREENNE

---

ASM-Diffusion Electronique vient de signer un accord de distribution avec la société coréenne Lucky Goldstar International Corporation, portant sur un micro-ordinateur au standard MSX. ASN Diffusion a l'exclusivité de la distribution pour les pays d'Afrique francophone et la Suisse.

Lucky Goldstar, troisième groupe coréen, emploie 60 000 personnes et regroupe une trentaine d'entreprises de secteurs très divers. Son chiffre d'affaires s'est élevé en 1983, à 7,5 billions de dollars.

ASM-Diffusion commencera à livrer ce micro MSX, FC 200, à partir de mai, accompagné d'un manuel en français.



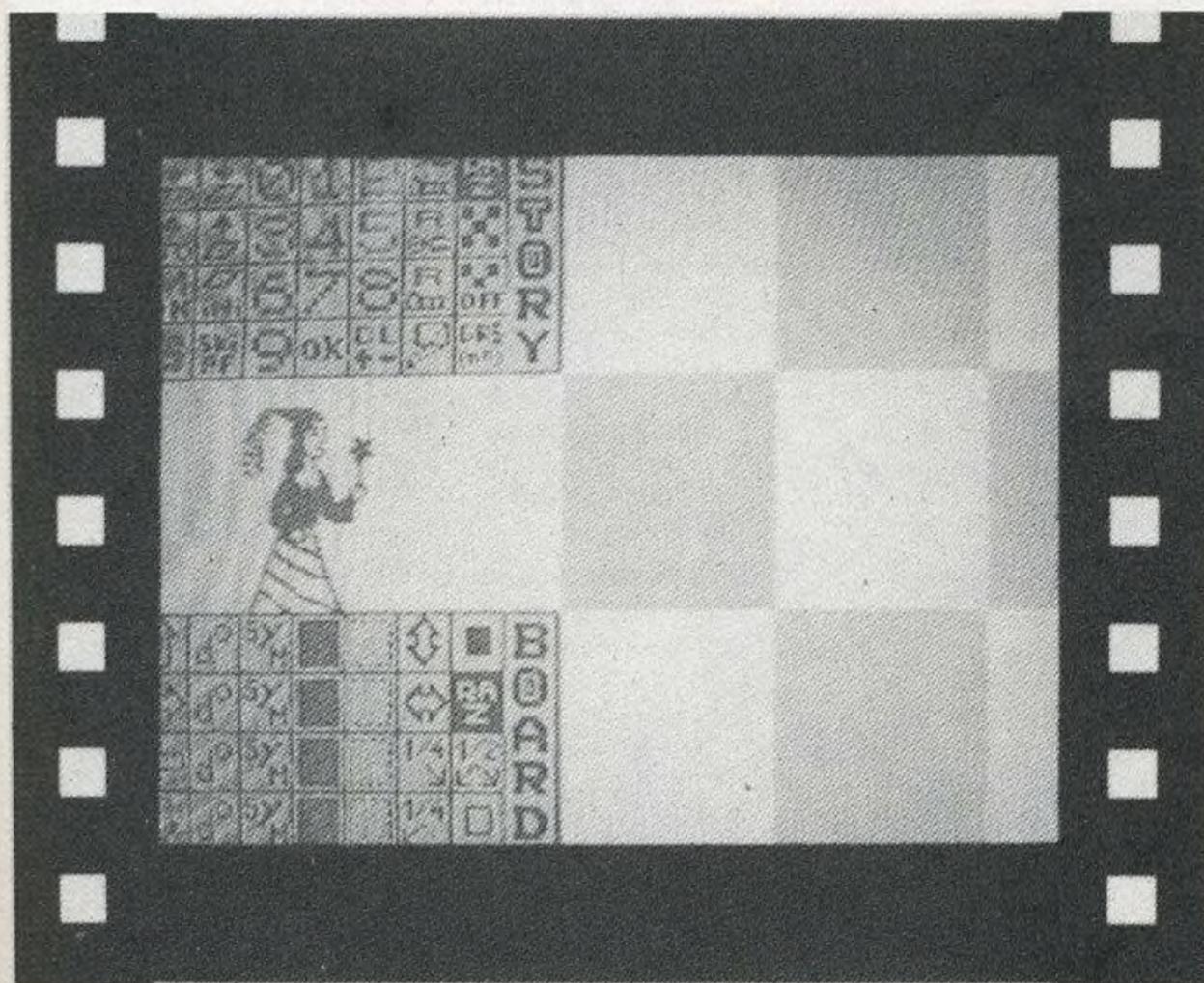
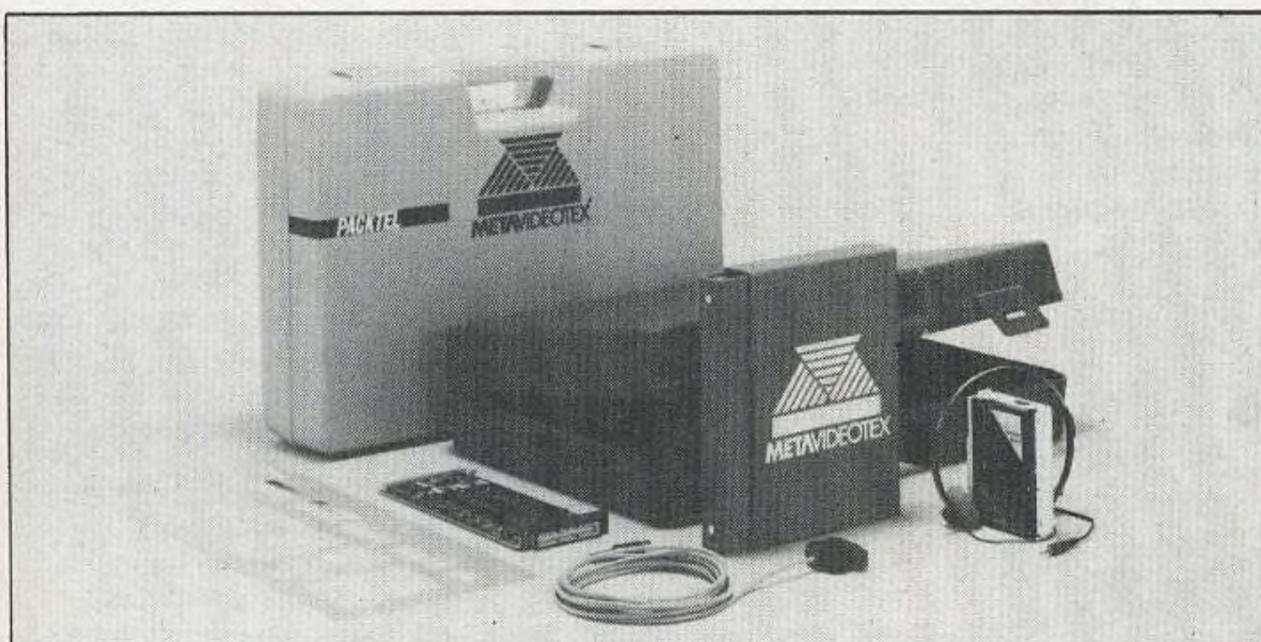
## META VIDEOTEX LANCE COKTEL

Cette société créée en juillet 1983 en collaboration avec SMT-Goupil s'est donnée pour objectif de promouvoir une activité industrielle du progiciel dans le secteur de la télématique. Pour son premier exercice, Meta Videotex a réalisé un chiffre d'affaires de plus 7 millions de francs. L'objectif 1985 est de l'ordre de 13 millions. En 18 mois, Meta Videotex a installé 80 systèmes.

Les deux co-fondateurs de la société, Richard Chêne et Philippe Lenoir, se sont donnés pour but de mettre à la disposition des PME/PMI des systèmes standardisés et packages sur micro à un prix abordable. Leur nouveau produit, Coktel, intègre toutes les fonctions nécessaires à la constitution d'un service d'information par Minitel sur micro-ordinateur : composition des écrans (texte et graphique), organisation et mise à jour du service, gestion du service, jeu de pages graphiques pour

l'illustration. Coktel est commercialisé 14 900 F HT. Les pages graphiques sont constituées d'un jeu de 50 pictogrammes types sur les thèmes suivants : sport, santé, culture, vie quotidienne, service, public. La disquette de 50 pictogrammes est commercialisée 2 900 F HT.

Afin de mettre réellement à la portée des PME/PMI mais aussi de toutes sortes d'entreprises, le Videotex, Meta Videotex a conçu la malette Packtel qui comporte tout ce qui est nécessaire — matériel et logiciel — pour transformer un micro PC en serveur Videotex. Cette malette contient un masque adapté au clavier facilitant la composition par l'affectation des fonctionnalités Videotex aux touches de fonction ; une carte entrée/sortie permettant la connexion du serveur aux modems ou boîtiers Tranapac ; un câble de connexion Minitel, le logiciel de base Videotex, le logiciel Coktel, la documentation et une cassette d'auto-formation. Packtel s'adapte en particulier à l'IBM-PC/XT, au Goupil, à l'Olivetti M24, à l'Ericsson PC et au Philips P3100. Son prix varie de 30 000 à 40 000 F selon les applications.

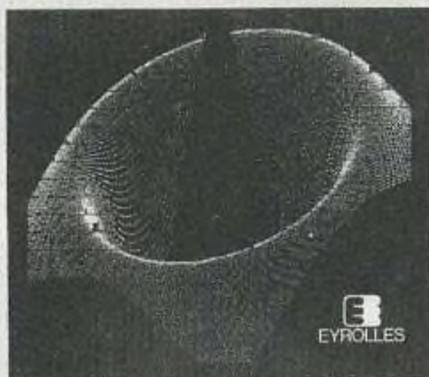


## STORY BOARD

Ce logiciel est un véritable utilitaire graphique qui permet la création de dessins et la réalisation d'animation sur un T07, T07-70 et M05 munis d'un crayon optique. Il est ensuite possible de réutiliser les graphiques et animations créés dans d'autres programmes Basic. Le logiciel est diffusé par Langage Informatique, une société toulousaine qui s'est spécialisée dans le développement de logiciels utilitaires et éducatifs sur micro-ordinateurs. A signaler également Logiformes, un jeu qui passionnera les petits et qui les aidera, à l'aide de leurs animaux favoris, à développer une qualité indispensable tout au long de leurs études : la logique. Ce jeu consiste à retrouver de 3 à six animaux choisis aléatoirement par le micro-ordinateur et à les replacer dans le bon ordre.

## 25 PROGRAMMES GRAPHIQUES EN BASIC MICROSOFT

T.J. O'MALLEY



### 25 Programmes graphiques en Basic Microsoft

T.J. O'Malley - Ed. Eyrolles

La formidable expansion des micro-ordinateurs s'est accompagnée de nombreuses évolutions techniques comme le graphisme. Lorsqu'on compare les possibilités graphiques d'un «vieux» APPLE II ou d'un TRS 80 avec un ORIC ou un TO7, on reste pantois tellement la puissance de ces deux derniers est plus grande.

Avec le graphisme et les micro-ordinateurs de la seconde génération est née une nouvelle «science informatique» le D.A.O. ou Dessin Animé par Ordinateur. C'est à ce thème qu'est consacré cet ouvrage proposé par T.J. O'Malley.

La première partie de ce livre traite des graphismes en deux dimensions. Les scientifiques et les économistes seront très intéressés par cette partie qui donne de nombreux exemples comme le tracé d'une courbe cartésienne ou encore la réalisation d'un histogramme.

Dès la seconde partie, l'auteur aborde le graphisme en trois dimensions et la méthode pour incorporer un troisième axe sur une figure quelconque.

Cette partie plus complexe débute par des explications mathématiques et... quelques équations. Aussitôt ces notions acquises, vous pourrez alors

réaliser des graphismes animés ou encore simuler une navette spatiale.

Un des buts de ce livre est aussi d'initier tous les utilisateurs à la création de ses propres dessins, c'est ce que l'auteur appelle «l'art avec l'ordinateur».

D'un point de vue pratique, chaque listing de cet ouvrage est accompagné de nombreuses explications en ce qui concerne l'algorithme suivi mais aussi sur les instructions basic utilisées. D'un point de vue Basic, l'auteur suit le standard Microsoft implanté sur la plupart des micro-ordinateurs disponibles sur le marché.

A noter chez le même éditeur un second livre traitant du D.A.O.

«Dessins géométriques et artistiques avec votre micro-ordinateur» Jean Paul Delahaye.

### La télématique

Michel Poulet - Ed. Marabout.

La télématique est avant tout un moyen de communication entre plusieurs terminaux informatiques. L'aspect transmission est donc largement abordé dans cet ouvrage à l'aide de

nombreuses descriptions techniques. C'est ainsi que l'on trouve dans ce livre la façon de traiter les informations (codage, modulation) mais

aussi une description des principaux supports de transmission (câble téléphonique, fibre optique...). Une attention toute particulière est donnée au réseau TRANSPAC qui est le principal réseau de transport de données en France.



Dans une première partie, l'auteur analyse quels sont les protocoles que l'on doit mettre en œuvre pour que 2 terminaux puissent communiquer entre eux à moindre coût et... sans conflit.

Dans une seconde partie, l'auteur se place du côté utilisateur et décrit les différents services qu'apporte la télématique. C'est bien sûr le Minitel qui est pris comme exemple dans cette partie qui traite à la fois de la télématique grand public (annuaire électronique, vidéo-texte, monnaie électronique...) mais aussi de la télématique professionnelle et de la bureautique (traitement du courrier, diffusion de l'information, traitement de texte...).

La télématique est déjà présente dans de nombreux foyers et son implantation ne fera que grandir au cours des ans. Pour les personnes qui ne sont pas insensibles à ce nouveau phénomène et qui désirent s'initier, ce livre peut être une excellente introduction.

### Manuel technique du MO5

Michel Oury - Cédic Nathan

Les livres traitant du «hard» sont rares, aussi c'est avec plaisir que je me suis plongé dans ce manuel technique du MO5. J'en profite pour émettre quelques suggestions aux éditeurs et aux réalisateurs de banc d'essais. Pourquoi lorsqu'un micro-ordinateur est «décortiqué» dans une revue spécialisée ne trouve-t-on aucun renseigne-

ment sur la structure de la machine ? Ne serait-il pas possible de mettre un synoptique général du micro-ordinateur ? Dans les micro-ordinateurs modernes, le graphisme est un des points sur lequel les fabricants font beaucoup d'efforts, aussi il est toujours intéressant d'analyser quels sont les principes retenus pour réaliser un contrôleur d'écran. L'auteur débute donc ce livre par une description détaillée de l'interface vidéo et de ses différents sous-ensembles (mémoire écran, base de temps, générateur de caractères...).

L'étude du MO5 continue par l'analyse des organes d'entrées sorties (schéma des circuits d'interfaces, brochage des connecteurs de liaison) : lecteur de K7, clavier, imprimante, générateur de sons...

En parallèle de cette partie matériel, des logiciels d'applications sont donnés afin de mettre en œuvre les différents périphériques que l'on peut relier au MO5. Une description du moniteur et de ses principaux sous-programmes de gestion vient compléter cette partie soft.

En conclusion, l'auteur donne le brochage et la fonction des principaux circuits intégrés

(amplificateurs de bus, mémoire, registre...) constituant le MO5. Un chapitre particulier concerne le microprocesseur 6809 (Motorola) qui équipe tous les micro-ordinateurs bas de gamme de Thomson et le MO5.

Pour exploiter au maximum cet ouvrage, il est conseillé aux lecteurs de posséder de bonnes bases en électronique, ce postulat vérifié, ce livre peut être un bon support pour des extensions ou des adaptations personnelles.

**Basic sans peine MO5, TO7**  
(livre accompagné de 2 cassettes) André Deledicq - Ed. Cedic Nathan.

Ce «kit d'initiation» se compose de deux éléments : un livre et deux cassettes compatibles avec le MO5 ou le TO7 de Thomson.

Le cours proprement dit est donné par les cassettes qui expliquent les principales ins-

tructions du Basic. Le livre vient en support et apporte précisions et compléments nécessaires à une bonne compréhension.

Dans une première partie «introduction», l'auteur explique le fonctionnement de ce double cours et comment utiliser les cassettes. Chaque chapitre est ensuite consacré à un thème particulier du basic.

- Graphique et son
- Variables
- Opérations logique et arithmétique
- Branchements conditionnel et inconditionnel
- Traitement des caractères
- Tableaux...

Des nombreux exercices, contenus dans le livre et les cassettes, viennent illustrer ce cours de Basic. Très classique dans le fond, ce livre est par contre très original dans sa forme. L'idée du kit d'initiation livre cassette est à retenir pour d'autres découvertes.

**Jouer avec son micro-ordinateur**

Olivier Chazole - Ed. Marabout  
D'après une enquête approfondie sur la micro-informatique domestique qui vient d'être effectuée aux Etats-Unis, quatre vingt pour cent des utilisateurs de micro-ordinateurs familiaux s'en servent pour

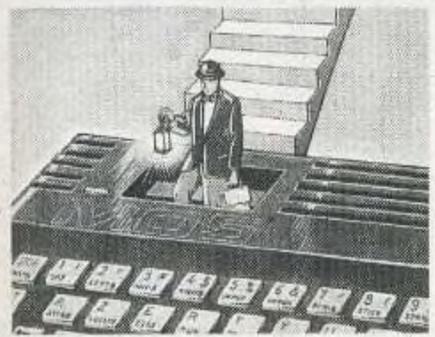
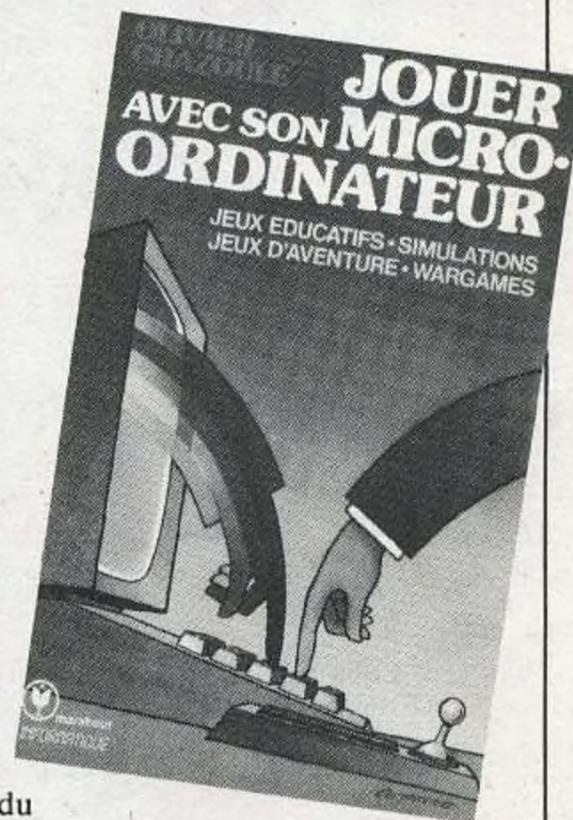
jouer. Il était donc tentant pour un spécialiste comme Olivier Chazole d'analyser ce phénomène.

Amateurs de listing et de longs programmes s'abstenir, ce livre est avant tout un guide dont la fonction principale est de renseigner les futurs utilisateurs ou mordus de micro-informatique et de jeux. Avant de se lancer dans des «jeux infernaux», il faut choisir un bon matériel : les principaux micro-ordinateurs familiaux sont présentés avec leurs accessoires (crayon optique, manettes...). Chaque matériel est analysé avec ses qualités et ses défauts. Il est évident que les micro-ordinateurs de la seconde génération avec leur graphisme, leur générateur de sons... et leur faible prix sont avantagés. La suite de cet ouvrage est consacrée aux nombreux jeux existant sur le marché. Chaque logiciel est testé et commenté par l'auteur.

Suivant leurs caractéristiques, les jeux sont regroupés en chapitre, on trouve ainsi :

- les jeux éducatifs, réservés aux enfants
- les jeux de simulation, simulateur de vol...
- les jeux d'aventure, enquêtes policières...
- les jeux de guerre où chaque utilisateur joue sa vie avec la touche RETURN.

P.F.



TO TEX  
cedic nathan

# PETITES ANNONCES GRATUITES

Vds jeu vidéo Atari + cassettes (Combat, B. Ball, Basic, Super Break Out) + joys., 500 F. Patard 4, sq. Crozatier, 78450 Villepreux. Tél. : (3) 056.16.68. Bur. : (3) 043.81.34.

Vds table traçante HP7475 à format A4, A3, 6 coul. + cordon RS232, 17 000 F. Patard 4, square Crozatier, 78450 Villepreux. Tél. : (3) 056.16.68. Bur. : (3) 043.81.34.

A vendre, cause double emploi : coffret 10 disquettes 3M neuves 5 pouces 1/4, double face, double densité, 96 TPI, 100 francs. M. Béreau, tél. : 331.83.89 (après 18 heures).

Vds TRS80 Mod 3 : 48 Ko + 2 disques (lecteurs SFSD) + imprimante Logabax 120 cps - 132 colonnes + programmes, 10 000 F. Dürr M. 18, rue La Pérouse, 31120 Portet-sur-Garonne. Tél. : (61) 72.23.18.

Vends IC260E, 3 200 F. Divers app. mesures. Liste contre env. sa carte Micro-Mazel-2 6800 + vidéo + alim., 800 F. Micro Hector HRX + cartouche Basic 3X + jeux + manettes, 3 500 F. P. Collin, tél. : (4) 452.04.97 h.b.

Vends Apple II+ avec 64 K de RAM + écran monochrome + imprimante Microline 80 + magnéto K7 (informatique) + livres. Prix : 8 000 F. Contact Laurent, tél. : 895.29.18 le soir.

Vends MPF1+ de Multitech avec interpréteur Basic et manuels d'utilisation au prix de 1 600,00 F. Etudierai autres propositions. Téléphoner au (99) 99.73.42.

Vds C64 PAL + cassette 1530 + diskette 1541 + 2 joysticks Canon VJ-200, le tout sous garantie + boîte 10 diskettes Memorex + cours autoformation Basic + jeux divers, 6 500 F. Robert : 576.84.62.

Club Ronchin recherche des ZX81 bas prix, même en panne. B.P. 47, 59790 ou tél. : (20) 88.06.62.

Prof. de lycée technique, section mécanique générale, serait prêt à fabriquer gratuitement ou à faible coût la partie mécanique (et éventuellement les plans) d'éléments style «robots, bras manipulateur, etc.» avec personnes pleines d'idées prenant en charge la partie électronique et/ou pilotage par micro-ordinateur. Picouveau Michel, Cité Victor Hugo, Bât. H, 93150 Le Blanc-Mesnil. Tél. : 865.61.34.

Vends Alice 4 Ko + livret + prise Péritel + adaptateur N/B. Etat neuf, garanti jusqu'au 17-10-85, embal. origine, 1 000 F exp. comprise (val. 1 500 F). Tél. : (55) 73.30.81.

Vds micro-ordinateur Sinclair ZX 81 + RAM 16 K + clavier ABS + magnéto K7 avec livre et schéma. Le tout 750 F. Riera JB. Tél. : 357.42.33 le soir.

**VOUS DESIREZ  
ECHANGER, VENDRE,  
ACQUERIR UN MATERIEL  
N'HESITEZ PAS A  
UTILISER NOS PETITES  
ANNONCES GRATUITES**



## Bulletin d'Abonnement

Je désire m'abonner à Led Micro (10 numéros). France : 140 F - Etranger : 210 F, à partir du n°....

Nom ..... Prénom .....

N° ..... Rue .....

Ville ..... Code Postal .....

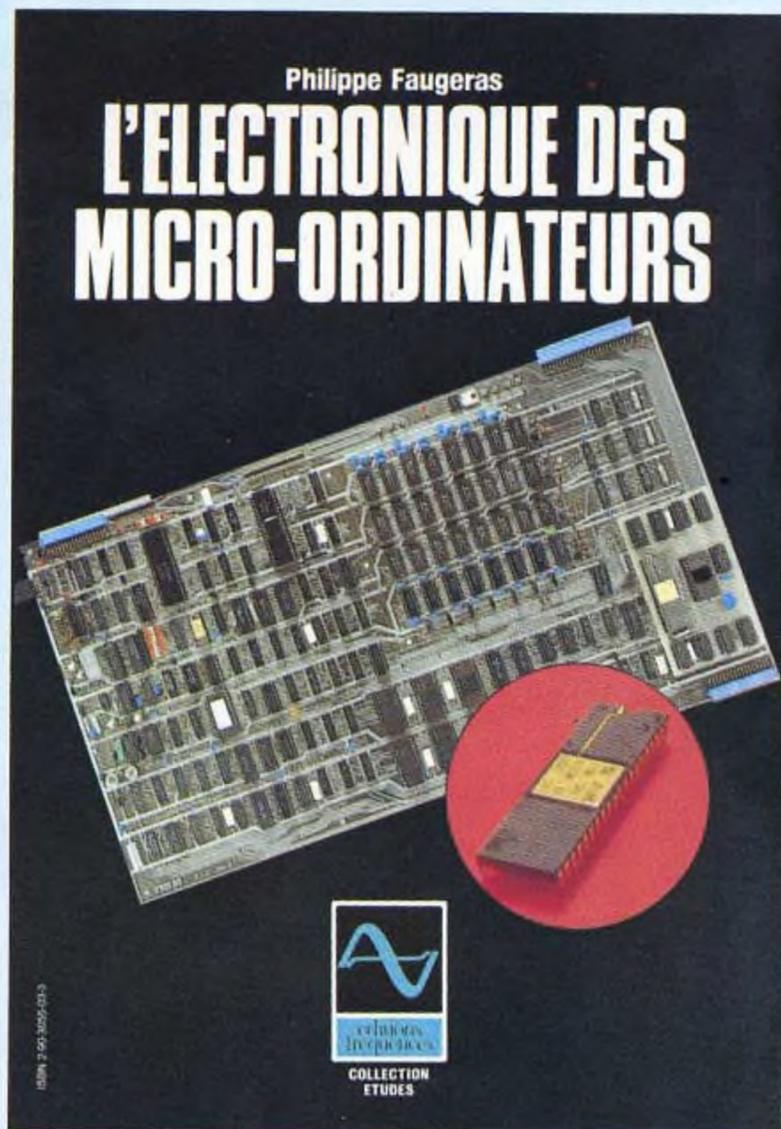
Envoyez ce bon accompagné du règlement à l'ordre des Editions Fréquences à :

EDITIONS FREQUENCES, 1 boulevard Ney, 75018 PARIS

MODE DE PAIEMENT : CCP  - Chèque bancaire  - Mandat

# VOYAGE AU CŒUR DES MICRO-ORDINATEURS

dans la  
COLLECTION  
«ETUDES»  
aux  
éditions  
fréquences



**une véritable  
schémathèque**

- 128 pages
  - 101 schémas
  - 34 tableaux
- Prix : 150 F

Que ce soit pour concevoir des interfaces ou optimiser un programme (utilisation des périphériques, encombrement mémoire...) «un micro-informaticien performant» doit posséder une bonne connaissance de son matériel.

Ce livre s'adresse donc à tous les électroniciens qui désirent découvrir les différents

composants constituant un micro-ordinateur. Articulé autour du microprocesseur Z80, cet ouvrage contient de nombreux schémas (plan mémoire, interfaces série et parallèle, interface clavier, interface vidéo, CAN, CNA...) qui pourraient être le thème... de nouvelles extensions.

En vente chez votre libraire et aux Editions Fréquences

## BON DE COMMANDE

Je désire recevoir l'ouvrage «**l'électronique des micro-ordinateurs**» au prix de **165 F** (150 F + 15 F de port).

Nom .....

Adresse .....

A adresser aux **EDITIONS FREQUENCES 1 boulevard Ney, 75018 Paris**

Règlement ci-joint :

Par chèque bancaire  par chèque postal  par mandat

*Philippe Faugeras, Docteur-ingénieur en électronique a acquis son expérience dans de grandes entreprises françaises où pendant cinq ans, il a travaillé sur des systèmes d'automatismes à base de microprocesseurs. Philippe Faugeras est responsable de la rubrique «Raconte-moi la micro-informatique» dans la revue LED.*

# chaque mois dans...

## MICRO MAGAZINE

La revue de la micro-informatique  
professionnelle et de l'utilisateur Victor



**le dossier**

**l'actualité**

**les applications**

**la technique**

**les fiches cuisine**

**le catalogue  
des progiciels**

**les logiciels**

EDITE PAR SITTELLE CONSEIL, 32, RUE WASHINGTON 75008 PARIS - (1) 359.68.34

**CHEZ VOTRE MARCHAND DE JOURNAUX**