



# **BASIC AMSTRAD CPC 464, 664 ET 6128**

## **2. PROGRAMMES ET FICHIERS**



**JACQUES BOISGONTIER**





# ***BASIC AMSTRAD CPC 464, 664 ET 6128***

## ***2. PROGRAMMES ET FICHIERS***

## CONNAISSEZ-VOUS TOUTE LA COLLECTION AMSTRAD CHEZ P.S.I. ?

---

Pour les Amstrad CPC 464, 664 et 6128 :

---

### Initiation :

- La découverte de l'Amstrad - Daniel-Jean David
- Exercices en Basic pour Amstrad - Maurice Charbit

### Programmation BASIC :

- 102 programmes pour Amstrad - Jacques Deconchat
- Super jeux Amstrad - Jean-François Sehan
- Amstrad en famille - Jean-François Sehan
- Super générateur de caractères sur Amstrad - Jean-François Sehan
- Photographie sur Amstrad et Apple II - Pierrick Moigneau et Xavier de la Tullaye
- Amstrad en musique - Daniel Lemahieu

### A paraître :

- Amstrad à l'école - Daniel Nielsen et Augustin Garcia

### Maîtrise du BASIC :

- BASIC Amstrad - 1. Méthodes pratiques - Jacques Boisgontier et Bruno Césard
- BASIC Plus : 80 routines sur Amstrad - Michel Martin
- Périphériques et fichiers sur Amstrad - Daniel-Jean David

### Assembleur et Pascal :

- Assembleur de l'Amstrad - Marcel Henrot
- Graphisme en assembleur sur Amstrad CPC - Francis Piérot
- Création et animations graphiques sur Amstrad CPC - Gilles Fouchard et Jean-Yves Corre
- Trois étapes vers l'intelligence artificielle sur Amstrad CPC - René Descamps
- Turbo Pascal sur Amstrad - Pierre Brandeis et Frédéric Blanc

### Système :

- CP/M Plus sur Amstrad 6128 et 8256 - Yvon Dargery
- Clefs pour Amstrad - 1. Système de base - Daniel Martin
- Clefs pour Amstrad - 2. Système disque - Daniel Martin et Philippe Jadoul

### A paraître :

- Intelligence artificielle : langage et formes sur Amstrad - Thierry Lévy-Abégnolli et Olivier Magnan
- Clefs pour Amstrad 8256 - Eric Baumarti

Amstrad Amdos et Amstrad CPC 464, 664 et 6128 sont des marques déposées de Amstrad Consumer Electronics.

AMX Mouse est une marque déposée de Advanced Memory Systems Ltd.

Pour tout problème rencontré dans les ouvrages P.S.I.  
vous pouvez nous contacter au numéro ci-dessous :

**Numéro Vert/Appel Gratuit en France**

**05 21 22 01**

(Composer tous les chiffres, même en région parisienne)

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1<sup>er</sup> de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

© Éditions du P.S.I. - B.P. 86 - 77402 Lagny/Marne cedex  
1986

ISBN 2-86595-249-5

**JACQUES BOISGONTIER**

# ***BASIC AMSTRAD CPC 464, 664 ET 6128***

## ***2. PROGRAMMES ET FICHIERS***



**ÉDITIONS DU P.S.I.  
1986**





# SOMMAIRE

|                                                |     |
|------------------------------------------------|-----|
| <b>GRAPHIQUES</b>                              | 7   |
| Rappels sur les graphiques AMSTRAD             | 7   |
| Instructions spécifiques aux CPC 664 et 6128   | 13  |
| Caractères accentués                           | 19  |
| Animation                                      | 21  |
| Dessinateur                                    | 25  |
| Souris                                         | 27  |
| Stockage d'un dessin dans un fichier           | 29  |
| Dessinateur et composition de dessins          | 30  |
| Apprentissage de mots                          | 35  |
| Composition de paysage avec animation          | 37  |
| Mini interpréteur LOGO                         | 40  |
| Musique                                        | 47  |
| <b>FICHIERS</b>                                | 53  |
| Rappels sur les fichiers                       | 53  |
| Fichier d'adresses                             | 54  |
| Etiquettes                                     | 59  |
| Gestion de fichier adaptable                   | 61  |
| Conversion minuscules-majuscules               | 65  |
| Changement d'un mot dans un fichier séquentiel | 65  |
| Composition de textes                          | 67  |
| Bibliothèque                                   | 69  |
| Dictionnaire des synonymes                     | 73  |
| Gestion de repas                               | 75  |
| Gestion de chèques                             | 75  |
| Courrier paramétré                             | 80  |
| Editeur de texte                               | 82  |
| Justification de texte à droite                | 87  |
| Mailing                                        | 90  |
| Gestion de clubs                               | 92  |
| Index de livre                                 | 96  |
| Histogramme circulaire (CPC 664)               | 97  |
| Histogramme double                             | 98  |
| <b>JEUX</b>                                    | 101 |
| Le mot le plus long                            | 101 |
| Le compte est bon                              | 104 |

|                                    |     |
|------------------------------------|-----|
| Jeu de marienbad                   | 107 |
| Jeu de marienbad par apprentissage | 110 |
| Jeu de la vie                      | 113 |
| Obstacle                           | 118 |
| Météorites                         | 120 |
| Casse-briques                      | 121 |
| Gloutons                           | 123 |
| Biorythmes                         | 125 |

|                               |     |
|-------------------------------|-----|
| <b>EDUCATION</b>              | 127 |
| Phrase à compléter            | 127 |
| Dictée                        | 128 |
| Mise en ordre d'un mot        | 129 |
| Mise en ordre d'une phrase    | 130 |
| Liste à compléter             | 131 |
| Questionnaire                 | 132 |
| Quid                          | 133 |
| Addition en base B            | 134 |
| Jeu du pendu                  | 135 |
| Dictionnaire français/anglais | 137 |

# GRAPHIQUES | 1

## RAPPELS SUR LES GRAPHIQUES AMSTRAD

---

La gestion des couleurs sur AMSTRAD est particulière. Pour la plupart des instructions graphiques ce n'est pas le numéro de la couleur qui est spécifié, mais un numéro de case d'une table de correspondance qui contient les numéros des couleurs.

|        |    |   |                   |
|--------|----|---|-------------------|
| Case 0 | 1  | • | numéro de couleur |
| Case 1 | 24 |   |                   |
| Case 2 | 11 |   |                   |
| Case 3 | 6  |   |                   |

Le programmeur peut modifier (par INK) les numéros des couleurs de la table de correspondance qui, à la mise sous tension, contient des numéros des couleurs par défaut.

Suivant la résolution choisie (par MODE) 640\*200, 320\*200 ou 150\*200, 2, 4 ou 16 couleurs parmi 27 peuvent être utilisées simultanément.

En changeant le numéro de couleur d'une case, tout ce qui a été dessiné par l'intermédiaire de cette case change instantanément de couleur.

Les CPC 664 et 6128 possèdent quelques instructions supplémentaires :

|                              |                                                    |
|------------------------------|----------------------------------------------------|
| <b>FILL couleur</b>          | Remplit une figure fermée.                         |
| <b>MASK format</b>           | Permet le dessin de pointillés.                    |
| <b>GRAPHICS PAPER papier</b> | Définit la couleur du papier pour les graphiques.  |
| <b>GRAPHICS PEN stylo</b>    | Définit le stylo par défaut pour les graphiques.   |
| <b>CURSOR 1</b>              | Fait apparaître le curseur pour INKEY\$.           |
| <b>COPYCHR\$ (#fenêtre)</b>  | Fournit un caractère affiché à l'écran.            |
| <b>FRAME</b>                 | Synchronise l'affichage des points d'un caractère. |
| <b>DERR</b>                  | Donne les codes d'erreur pour le disque.           |
| <b>ON BREAK CONT</b>         | Inhibe les interruptions de programme par ESC.     |

### MODE numéro

Définit le mode écran choisi :

MODE 0 : 20 caractères par ligne / 200\*160 points 16 couleurs.

MODE 1 : 40 caractères par ligne / 200\*320 points 4 couleurs.

MODE 2 : 80 caractères par ligne / 200\*640 points 2 couleurs.

### INK case couleur,couleur

Définit les couleurs de la table de correspondance des couleurs.

10 INK 1,2                    couleur 2(bleu vif) dans case couleur 1  
20 INK 2,6                    couleur 6(rouge vif) dans case couleur 2

En MODE 2 seules les cases 0 et 1 sont utilisables.

En MODE 1 seules les cases 0,1,2,3 sont utilisables.

En MODE 0 les 16 cases sont utilisables.

Table de correspondance des couleurs à la mise sous tension :

|    | Mode 2 | Mode 1 | Mode 0 | Couleurs     |                   |
|----|--------|--------|--------|--------------|-------------------|
| 0  | 1      | 1      | 1      | 0 noir       | 14 bleu pastel    |
| 1  | 24     | 24     | 24     | 1 bleu       | 15 orange         |
| 2  | 1      | 20     | 20     | 2 bleu vif   | 16 rose           |
| 3  | 24     | 6      | 6      | 3 rouge      | 17 magenta pastel |
| 4  | 1      | 1      | 26     | 4 magenta    | 18 vert vif       |
| 5  | 24     | 24     | 0      | 5 mauve      | 19 vert marin     |
| 6  | 1      | 20     | 2      | 6 rouge vif  | 20 turquoise vif  |
| 7  | 24     | 6      | 8      | 7 pourpre    | 21 vert citron    |
| 8  | 1      | 1      | 10     | 8 magenta    | 22 vert pastel    |
| 9  | 24     | 24     | 12     | 9 vert       | 23 turquoise      |
| 10 | 1      | 20     | 14     | 10 turquoise | 24 jaune          |
| 11 | 24     | 6      | 16     | 11 bleu ciel | 25 jaune pastel   |
| 12 | 1      | 1      | 18     | 12 jaune     | 26 blanc brillant |
| 13 | 24     | 24     | 22     | 13 blanc     |                   |
| 14 | 1      | 20     | 1,24   |              |                   |
| 15 | 24     | 6      | 16,11  |              |                   |

En MODE 2 par exemple, si vous changez la couleur de la case 0, la couleur de la case 2 est également changée.

**PAPER #fenêtre,case couleur**

**PEN #fenêtre,case couleur**

Permet de choisir les "cases couleur" de la table de correspondance des couleurs pour l'écriture du texte. A la mise sous tension, PAPER pointe vers la case 0 et PEN vers la case 1.

Si vous frappez "PEN 2" (en MODE 1), la couleur d'écriture du texte se fait avec la couleur de la case 2 de la table des couleurs.

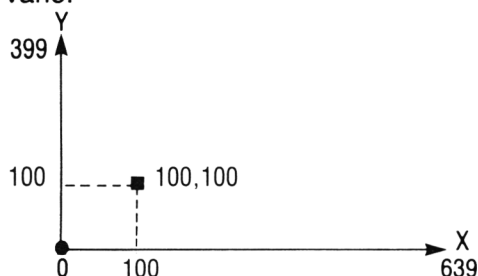
## PLOT X,Y,case couleur

Allume le point X,Y spécifié avec la case couleur (stylo) spécifiée.

```
10 MODE 1
20 INK 0,26 ' blanc dans case 0
30 INK 1,0  ' noir dans case 1
40 INK 2,6  ' rouge dans case 2
50 PLOT 100,100,1 ' point 100,100 avec stylo 1(noir)
60 PLOT 120,120,2 ' point 120,120 avec stylo 2(rouge)
```

L'origine de l'écran est en bas à gauche. X doit être compris entre 0 et 639. Y doit être compris entre 0 et 399.

Les coordonnées spécifiées sont INDÉPENDANTES de la résolution choisie par MODE. Seule la taille du point varie.



Piège : si vous tentez d'utiliser le stylo 4 en MODE 1, vous utilisez en réalité le stylo 0. Naturellement, le point n'apparaît pas.

A la mise sous tension, si la "case couleur" (stylo) n'est pas spécifiée dans l'instruction PLOT, la case par défaut est la case 1.

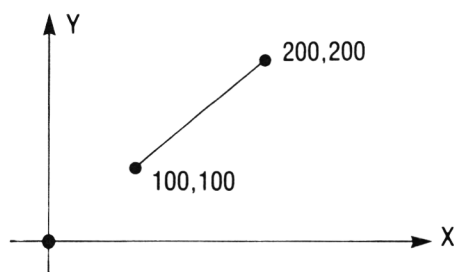
En cours de programme, c'est le dernier stylo spécifié qui est utilisé par défaut.

Il est vivement conseillé de spécifier un stylo. En effet, dans le cas contraire, si un programme précédent a utilisé le stylo 0, la trace n'apparaîtra pas.

## DRAW X,Y,case couleur

Trace une droite entre le point courant et le point spécifié (en absolu).

```
10 MODE 1
20 INK 0,1 ' bleu dans case 0
30 INK 1,24 ' jaune dans case 1
40 PLOT 100,100,1 ' point 100,100
50 DRAW 200,200,1 ' droite entre 100,100 et 200,200
```

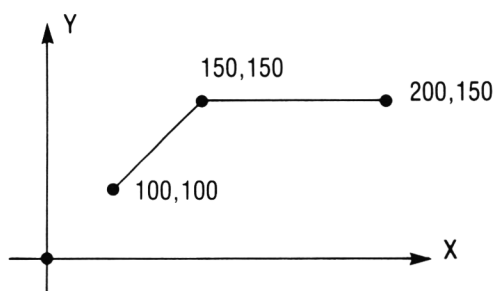


## PLOT DX,DY,case couleur DRAWR DX,DY,case couleur

Ces instructions spécifient des déplacements relatifs au point courant.

```
10 MODE 1
20 INK 0,26      ' blanc dans case 0
30 INK 1,0       ' noir dans case 1
40 INK 2,6       ' rouge dans case 2
50 PLOT 100,100,1 ' point 100,100
60 DRAWR 50,50,1  ' droite entre 100,100 et 150,150
70 DRAWR 50,0,2   ' droite entre 150,150 et 200,150
```

La première droite est dessinée avec le stylo 1(noir), et la seconde avec le stylo 2(rouge).



Si vous aviez utilisé le stylo numéro 4, les droites n'auraient pas été tracées. En MODE 1, le stylo 4 est en réalité le stylo 0.

## MOVE X,Y MOVER DX,DY

Positionne le curseur graphique sans dessiner. Le CPC 664 dispose du paramètre "stylo".

## TAG TAGOFF

TAG spécifie que l'écriture du texte doit se faire avec le curseur graphique.

TAGOFF annule TAG ; l'affichage du texte se fait avec le curseur texte (positionné par LOCATE X,Y).

Pour empêcher l'affichage des caractères CHR\$(10) et CHR\$(13), l'instruction PRINT doit être suivie d'un point-virgule.

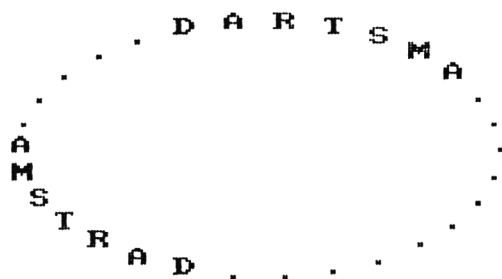
```
10 MODE 1
20 TAG
30 MOVE 100,100:PRINT "AAAAAAA";      affiche en 100,100
40 TAGOFF
50 LOCATE 1,1:PRINT "BBBBBBBBB"      avec curseur texte
```

Le programme ci-dessous fait défiler un message. Sans TAG l'affichage ne serait pas aussi précis.

```

10 ; AFFICHAGE AVEC TAG
20
30 MODE 1
40 INK 0,26:INK 1,0
50 DIM X(100),Y(100)
60 A$="AMSTRAD.....AMSTRAD....."
70 R=100
80 XC=200:YC=200
90 FOR I=1 TO 30
100  A=PI*I/30*2
110  X(I)=R*COS(A)*1.5+XC
120  Y(I)=R*SIN(A)+YC
130 NEXT I
140 '---- ROTATION
150 FOR I=1 TO 30
160  MOVE X(I),Y(I)
170  TAG:PRINT MID$(A$,I,1);
180 NEXT I
190 A$=RIGHT$(A$,1)+LEFT$(A$,LEN(A$)-1)
200 GOTO 150

```



#### ORIGIN X,Y,gauche,droite,haut,bas

Redéfinit l'origine pour le curseur graphique et une fenêtre graphique si les paramètres "gauche", "droite", "haut" et "bas" sont spécifiés.

```

10 MODE 1
20 ORIGIN 100,100 ' origine en 100,100
30 PLOT 1,1,1
40 DRAW 100,100 ' 100,100 a partir de la nouvelle origine

```

```

10 MODE 1:INK 0,26:INK 1,0:INK 2,2
20 ORIGIN 200,100,200,350,100,250 ' fenetre graphique
25 CLG 2
30 PLOT 1,1,1
40 DRAW 50,50,1

```

**CLG papier**

Efface l'écran graphique.

**TEST (X,Y)**  
**TESTR (X,Y)**

Donne le numéro de stylo avec lequel un point a été allumé.

```
10 MODE 1
20 PLOT 100,100,1
30 PLOT 200,200,2
40 PRINT TEST(100,100)
50 PRINT TEST(200,200)
```

RUN

1

2

Le programme ci-dessous recopie un écran sur imprimante EPSON.

```
2000 ----- RECOPIE D'ECRAN SUR IMPRIMANTE EPSON (mode 1)
2004 WIDTH 255
2005 PRINT #8,CHR$(27);"3";CHR$(10)
2010 DEFINIT A-Z
2020 DIM LG(320)
2030 LN=399                               ligne
2040 FOR LG=1 TO 30
2050   FOR X=0 TO 319
2060     LG(X)=0
2070     FOR P=0 TO 6
2080       IF TEST(X*2,LN-P*2)>0 THEN LG(X)=LG(X)+(2^(6-P))
2090     NEXT P
2100   NEXT X
2110 PRINT #8,CHR$(27);"K";CHR$(64);CHR$(1);
2120 FOR Z=0 TO 319:PRINT #8,CHR$(LG(Z));:NEXT Z
2130 PRINT #8
2140 LN=LN-14
2150 NEXT LG
2160 PRINT #8,CHR$(15)
2170 '----- pour MODE 2
2180 ' 2080 IF TEST(x,LN-P*2)>0 THEN .....
2190 ' Prevu pour cable AMSTRAD (adapter 2005 pour autres cables)
```

**WINDOW #numéro fenêtre,gauche,droite,haut,bas**

Définit une fenêtre pour le texte. Dans l'exemple ci-dessous, l'écriture du texte dans la fenêtre 1 se fait en blanc sur fond rouge.

```
10 ----- FENETRE
20 MODE 1
30 INK 0,26
40 INK 1,0
50 INK 2,6
60 INK 3,26
```





```

70 '---- fenetre 1
80 WINDOW #1,20,40,1,12 ' fenetre col 20 et 40-ligne 1 et 12
90 PAPER #1,2:CLS #1
100 PEN #1,3 ' stylo 3 pour fenetre 1
110 PRINT #1,"Fenetre 1"
120 PRINT #1,"Ecriture blanche"
130 PRINT #1,"sur fond rouge"
140 '----- fenetre 2 (dans fenetre 1)
150 WINDOW #2,30,38,6,11
160 PAPER #2,0:CLS #2
170 PRINT #2."fenetre 2"

```

Une fenêtre définie à l'intérieur d'une autre n'est pas protégée contre une écriture dans la fenêtre où elle est incluse.

La fenêtre par défaut "#0" recouvre tout l'écran. Naturellement, les bornes de cette fenêtre peuvent être changées.

## INSTRUCTIONS SPÉCIFIQUES AUX CPC 664 ET 6128 \_\_\_\_\_

Sur CPC 664 et 6128, MOVE possède un paramètre "stylo".

Les instructions PLOT, DRAW et MOVE disposent d'un paramètre "opération" supplémentaire. Ce paramètre égal à 0, 1, 2 ou 3 permet de combiner un tracé existant avec un nouveau tracé en utilisant une opération XOR,AND,OR. Par exemple, si vous tracez une droite sur une droite existante avec l'opération XOR(OU exclusif), l'ancien tracé disparaît (0=nul 1=XOR 2=AND 3=OR).

PEN possède un paramètre "mode transparent" qui annule l'effet de PAPER.

```

10 MODE 1:INK 0,26:INK 1,0:INK 2,6
20 PAPER 0:PEN 1
30 PRINT "AAAAA"
35 '---
40 PAPER 2
50 PEN 1 ' essayer avec PEN 1,1
60 PRINT "BBBBBB"
70 PAPER 0

```

### FILL stylo

Remplit une figure fermée avec le stylo spécifié. Le curseur doit être positionné avec le stylo qui a servi à tracer la figure. En revanche, le stylo spécifié dans FILL peut être différent.

```

10 '----- FILL
20 MODE 1
30 CE=1 ' Couleur ecriture
40 INK 0,26:INK 1,0:INK 2,20
50 PLOT 100,100,CE
60 DRAWR 50,0,CE
70 DRAWR 0,50,CE
80 DRAWR -50,0,CE
90 DRAWR 0,-50,CE

```



```

100 '----- remplissage carre
110 MOVER 10,10,CE      ' Positionnement curseur
120 C=1                  ' Stylo remplissage
130 FILL C
140 '
150 ' La couleur de remplissage peut etre differente de la
    couleur du contour
160 ' ex:C=2

```

### **MASK masque, premier point**

Le tracé des droites est effectué avec le pointillé défini en binaire par "masque". Si "premier point" est égal à 1, celui-ci apparaît.

```

10 MODE 1
20 MASK &X10101010
30 PLOT 100,100
40 DRAW 200,200,1
50 MASK &X1111000
60 DRAW 300,200,1

```

### **COPYCHR\$ (#no fenêtre)**

Donne le caractère affiché sous le curseur.

```

10 MODE 1
20 LOCATE 10,10:PRINT "ABC"
30 LOCATE 10,10:X$= COPYCHR$(#0)
40 LOCATE 1,20:PRINT X$

```

Sur CPC 464, le programme suivant donne le contenu des 8 octets représentant un caractère.

```

10 ' COPYCHR$ SUR CPC464
20 MODE :2
30 X=1
40 Y=1
50 LOCATE X,Y:PRINT "ABCDE"
60 '
70 FOR AM=49152+X-1+(Y-1)*80 TO 65534 STEP 2048
80 M=PEEK(AM)
90 PRINT BIN$(M,8)
100 NEXT AM
    ABCDE
00011000
00111100
01100110
01100110
01111110
01100110
01100110
00000000
Ready

```

## GRAPHICS PAPER stylo

Définit la couleur de fond sur laquelle les droites en pointillé sont dessinées.

```
10 MODE 1
20 INK 0,26:INK 1,0:INK 2,14
30 GRAPHICS PAPER 2
40 MASK &X111110000
50 PLOT 100,100:DRAW 200,200,1
```

## GRAPHICS PEN stylo,mode transparent

Définit le stylo utilisé par défaut pour les instructions graphiques. Si "mode transparent" est égal à 1, l'effet de GRAPHICS PAPER est annulé.

## FRAME

Synchronise l'affichage des points d'un caractère affiché avec TAG.

## CURSOR mode

Lorsque INKEY\$ est utilisé, le curseur est apparent pour CURSOR 1 et invisible pour CURSOR 0.

## EFFACEMENT D'UN DESSIN

Pour effacer un dessin il suffit de tracer le même dessin en utilisant le stylo 0.

```
10 '----- effacement d'un dessin
20 MODE 1:PAPER 0:PEN 1
30 INK 0,26 ' blanc dans case 0
40 INK 1,0 ' noir dans case 1
50 '----- droite noire sur fond blanc
60 PLOT 100,100,1
70 DRAW 200,200,1 ' droite avec case 1
80 '----
90 FOR TP=1 TO 2000:NEXT TP ' temporisation
100 '--- effacement
110 PLOT 100,100,0
120 DRAW 200,200,0 ' droite avec case 0
```

## CHANGEMENT DE LA COULEUR D'UN DESSIN

Pour changer la couleur d'un dessin il suffit de modifier par INK la couleur de la case (stylo) qui a servi à tracer le dessin. On peut également de la même façon faire disparaître momentanément un dessin sans l'effacer.

```
10 '---- comment changer la couleur d'un dessin
20 MODE 1:PAPER 0:PEN 1
30 INK 0,26 ' blanc dans case 0
40 INK 1,0 ' noir dans case 1 →
```

```

50 '---- carre noir sur blanc
60 PLOT 100,100,1
70 DRAWR 100,0,1
80 DRAWR 0,100,1
90 DRAWR -100,0,1
100 DRAWR 0,-100,1
110 '----
120 FOR TP=1 TO 2000:NEXT TP ' temporisation
130 '----- le carre devient bleu
140 INK 1,2 ' bleu dans case 1
150 FOR tp=1 TO 2000:NEXT tp
160 '----- le carre devient invisible
170 INK 1,26 ' blanc dans case 1
180 FOR tp=1 TO 2000:NEXT tp
190 '----- le revoila
200 INK 1,6 ' rouge dans case 1

```

### Pièges :

Lorsque vous utilisez un programme la table de correspondance des couleurs est dans l'état où le programme précédent l'a laissée. Il est donc prudent d'initialiser la table des couleurs en début de programme ou bien d'appuyer sur "CTRL /SHIFT/ESC".

Le programme ci-dessous trace, en MODE 1, trois droites avec les stylos 1, 2 et 3. En MODE 2, la seconde droite n'apparaît pas puisque le stylo 2 est en réalité le stylo 0.

```

10 '----- PIEGE NO 1
20 MODE 1
30 INK 0,26 ' blanc dans case 0
32 INK 1,0 ' noir dans case 1
34 INK 2,2 ' bleu dans case 2
40 '----- droite noire sur fond blanc
50 PLOT 100,100,1
60 DRAW 200,200,1
70 '----- droite bleue sur fond blanc
80 DRAW 300,200,2
90 '----- droite rouge sur fond blanc
100 DRAW 300,300,3
110 '----
120 ' essayez ce programme en MODE 2
130 ' la droite trace avec le stylo 2 n'apparait pas:
140 ' En mode 2, le stylo 2 est le meme que 0.

```

Le programme ci-dessous utilise le stylo 2 pour écrire en MODE 1. En passant en MODE 2, le texte n'apparaît plus puisque PEN 2 est équivalent à PEN 0. Il suffit de frapper PEN 1 pour faire apparaître le texte.

```

10 '----- PIEGE NO 2
20 MODE 1
30 INK 0,26:INK 1,0:INK 2,2
40 PEN 1
50 PRINT "J'ECRIS EN NOIR"
60 PEN 2
70 PRINT "J'ECRIS EN BLEU"

```



```

80 '
90 PRINT "ATTENTION,TOUT VA DISPARAITRE"
100 PRINT "POUR FAIRE APPARAÎTRE,FRAPPEZ :PEN 1"
110 FOR TP=1 TO 4000:NEXT TP ' TEMPORISATION
120 MODE 2
130 '
140 ' EN MODE 2 , 'PEN 2' EST EQUIVALENT A 'PEN 0'

```

Ci-dessous, nous plaçons dans la case 1 la couleur de la case 0 pour rendre invisible une droite ; le texte disparaît en même temps. Pour le faire apparaître à nouveau, frapper INK 1,0.

```

10 '-----PIEGE NO 3
20 MODE 1
30 INK 0,26:INK 1,0
40 '----- droite noire sur fond blanc
50 PLOT 100,100,1
60 DRAWR 100,0,1
70 '-----
80 PRINT "Frapper:INK 1,0"
90 FOR TP=1 TO 2000:NEXT TP ' temporisation
100 '----- droite invisible
110 INK 1,26
120 '
130 ' frapper en mode direct: INK 1,0

```

Nous présentons ici quelques exemples de dessins.

Le programme ci-dessous trace un drapeau français. Nous initialisons d'abord la table des couleurs puis nous traçons trois rectangles.

```

10 '      DRAPEAU FRANCAIS
20 '
30 '
40 MODE 1
50 INK 0,21 ' vert dans case 0
60 INK 1,2 ' bleu dans case 1
70 INK 2,26 ' blanc dans case 2
80 INK 6,3 ' rouge dans case 3
90 '
100 XA=100:YA=100 ' coordonnees depart
110 H=150:L=100 ' hauteur/longueur rectangle
120 CE=1 ' bleu
130 GOSUB 220
140 CE=2 ' blanc
150 XA=XA+L
160 GOSUB 220
170 CE=3 ' rouge
180 XA=XA+L
190 GOSUB 220
200 END
210 '----- Trace d'un rectangle

```



```

220 FOR Y=YA TO YA+H
230 PLOT XA,Y:DRAW L,0,CE
240 NEXT Y
250 RETURN

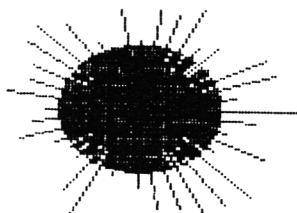
```

Le dessin de soleil ci-dessous est obtenu en traçant un cercle plein puis des rayons aléatoires.

```

10 '----- SOLEIL
20 MODE 1
30 INK 0,26:INK 1,6:PAPER 0:PEN 1
40 R=40 ' rayon
50 XC=200:YC=200 ' centre
60 GOSUB 160
70 '----- RAYONS
80 FOR A=0 TO 2*PI STEP PI/20
90 R1=R+RND(1)*R
100 X=XC+R1*COS(A)
110 Y=YC+R1*SIN(A)
120 PLOT XC,YC,1:DRAW X,Y,1
130 NEXT A
140 END
150 '----- CERCLE
160 FOR A=0 TO 2*PI+0.1 STEP 1/R
170 X=XC+R*COS(A)
180 Y=YC+R*SIN(A)
190 PLOT XC,YC:DRAW X,Y,1
200 NEXT A
210 RETURN

```



Pour tracer des cercles rapidement nous calculons les valeurs des sinus et cosinus dans des tables C() et S().

```

10 ' TRACE DE CERCLE RAPIDE
20 '
30 MODE 1
40 INK 0,26:INK 1,6
50 PRINT "JE CALCULE LES COSINUS ET SINUS"
60 '
70 NP=20 ' NOMBRE DE POINTS
80 DIM C(360),S(360)
90 FOR J=0 TO 360 STEP 360/NP
100 A=2*PI*J/360
110 C(J)=COS(A):S(J)=SIN(A)
120 NEXT J
130 '---
140 CE=1
150 FOR N=1 TO 10
160 XC=100+RND(1)*300
170 YC=100+RND(1)*200
180 R=40+RND(1)*30
190 GOSUB 230
200 NEXT N

```

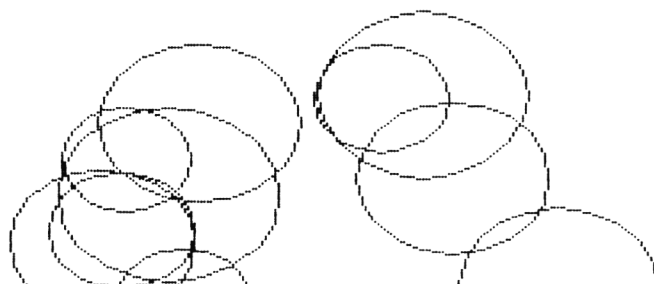


```

210 END
220 '----- CERCLE
230 PLOT XC+R,YC,1
240 FOR A=0 TO 360 STEP 360/NP
250   X=XC+R*C(A)
260   Y=YC+R*S(A)
270   DRAW X,Y,CE
280 NEXT A
290 RETURN

```

**JE CALCULE LES COSINUS ET SINUS**  
**Ready**



## CARACTÈRES ACCENTUÉS

---

Les caractères accentués ne sont pas prévus sur l'AMSTRAD. Ils peuvent être ajoutés en redéfinissant des caractères avec l'instruction SYMBOL.

Si les caractères accentués doivent être édités sur imprimante, il faut utiliser les codes ASCII adaptés à l'imprimante. Par exemple, pour une imprimante DPM1, le code de "à accent" est 64. On redéfinit donc le caractère de code ASCII 64.

En revanche, les touches au clavier peuvent être choisies.

Ci-dessous nous utilisons la touche "1" du clavier numérique pour frapper le "à accent".

```

10 '===== ACCENTS
30 ' redefinition du caractere 'A'
40 ' 1 clavier numerique
50 '
55 SYMBOL AFTER 64
60 SYMBOL 64,96,16,120,12,124,204,118
70 KEY 129,CHR$(64)

```

Ci-dessous nous utilisons ",CTRL/A" pour frapper "à accent".

```

10 '===== ACCENTS
30 ' redefinition du caractere 'A'
40 ' CTRL A --> a accentue
50 '
55 SYMBOL AFTER 64
60 SYMBOL 64,96,16,120,12,124,204,118
70 KEY DEF 69,1,97,65,64

```

Voici quelques exemples de caractères accentués :

```

10 '===== ACCENTS
20 SYMBOL AFTER 64
30 '---7 e accent grave
40 SYMBOL 125,96,16,60,102,126,96,60
50 KEY 135,CHR$(125)
60 '----- 0 accent aigu
70 SYMBOL 123,6,8,60,102,126,96,60
80 KEY 128,CHR$(123)
90 '----- 1 a accent grave
100 SYMBOL 64,96,16,120,12,124,204,118
110 KEY 129,CHR$(64)
120 '--- 4 c cedille
130 SYMBOL 92,0,0,60,102,96,62,8,24
140 KEY 132,CHR$(92)

```

L'exemple ci-dessous inverse les caractères "A" et "Q" au clavier.

```

5 '----- clavier awerty
10 KEY DEF 67,1,97,65
20 KEY DEF 69,1,113,82

```

Le programme ci-dessous permet de dessiner un caractère et donne les valeurs décimales à spécifier dans l'instruction SYMBOL.

```

10 '----- GENERATEUR DE CARACTERES 8X8
20 MODE 1:PAPER 0:PEN 1:INK 0,26:INK 1,0
30 DIM T(9,9)
40 SYMBOL AFTER 134
50 SYMBOL 134,255,129,129,129,129,129,129,255      carre
60 FOR Y=1 TO 8:FOR X=1 TO 8
70   LOCATE X,Y:PRINT CHR$(134)
80 NEXT X:NEXT Y
90 LOCATE 1,12:PRINT "Fleches pour deplacer"
100 LOCATE 1,13:PRINT "A:allumer  E:effacer V:valeurs dec."
110 LOCATE 1,15:PRINT "F:fin"
120 X=5:Y=5
130 CS=1
140 '----- CURSEUR CLIGNOTANT
150 C$=INKEY$:IF LEN(C$)<>0 THEN 210
160 LOCATE X,Y:IF T(X,Y)=1 THEN PRINT CHR$(143) ELSE PRINT CHR$(
231)
170 FOR TP=1 TO 40:NEXT TP
180 LOCATE X,Y:PRINT CHR$(134)
190 GOTO 150
200 '---
210 LOCATE X,Y:IF T(X,Y)=1 THEN PRINT CHR$(143) ELSE PRINT CHR$(
134)
220 '
230 C$=UPPER$(C$)
240 C=ASC(C$)
250 IF C=242 THEN IF X>1 THEN X=X-1

```

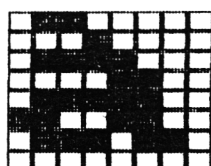




```

260 IF C=243 THEN IF X<8 THEN X=X+1
270 IF C=241 THEN IF Y<8 THEN Y=Y+1
280 IF C=240 THEN IF Y>1 THEN Y=Y-1
290 IF C$="A" THEN LOCATE X,Y:PRINT CHR$(143):T(X,Y)=1
300 IF C$="E" THEN LOCATE X,Y:PRINT CHR$(134):T(X,Y)=0
310 IF C$="V" THEN 350
320 IF C$="F" THEN END
330 GOTO 150
340 '----- calcul valeurs decimales
350 FOR L=1 TO 8      ' 8 lignes
360   ND=0            ' valeur decimale
370   FOR X=1 TO 8    ' 1 caractere
380    A=0:IF T(X,L)=1 THEN A=1
390    ND=ND+A*2^(8-X) ' ^=fleche haut
400   NEXT X
410   LOCATE 10,L:PRINT ND SPC(1)
420   D(L)=ND
430 NEXT L
440 SYMBOL 135,D(1),D(2),D(3),D(4),D(5),D(6),D(7),D(8)
450 LOCATE 15,2:PRINT CHR$(135)
460 X=5:Y=5
470 GOTO 150

```



```

96
16  a
120
12
124
204
118
0

```

**Fleches pour deplacer**  
**A:allumer E:effacer U:valeurs dec.**  
**F:fin**

## ANIMATION

---

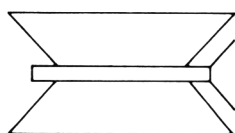
Pour animer des figures on utilise généralement des caractères redéfinis. En affichant un caractère dans différentes positions de l'écran, on obtient un effet d'animation.

Pour simuler un vol de papillon nous le représentons alternativement dans deux positions. Chaque position est représentée par 16×16 points, soit quatre caractères graphiques. CHR\$(8), CHR\$(10) et CHR\$(11) permettent de déplacer le curseur à gauche, en bas et vers le haut. Ainsi le papillon est affiché avec une seule instruction PRINT.

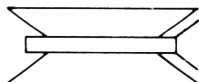
```

10 ; VOL DE PAPILLON
20
30 MODE 1:INK 0,26:INK 1,6
40 C=145 ' 1ER CARACTERE A MODIFIER
50 SYMBOL AFTER C
60 '----- POSITION 1
70 SYMBOL C,0,0,0,127,63,31,15,63
80 SYMBOL C+1,0,0,0,248,240,226,196,248
90 SYMBOL C+2,15,31,63,127,0,0,0,0
100 SYMBOL C+3,196,226,240,248,0,0,0,0
110 P1$=CHR$(C)+CHR$(C+1)+CHR$(10)+CHR$(8)+CHR$(8)+CHR$(C+2)+CHR$(C+3)+CHR$(11)
120 LOCATE 10,2:PRINT L$
130 '----- POSITION 2
140 SYMBOL C+4,0,0,0,0,0,31,15,63
150 SYMBOL C+5,0,0,0,0,0,226,196,248
160 SYMBOL C+6,15,31,0,0,0,0,0,0
170 SYMBOL C+7,196,226,0,0,0,0,0,0
180 P2$=CHR$(C+4)+CHR$(C+5)+CHR$(10)+CHR$(8)+CHR$(8)+CHR$(C+6)+CHR$(C+7)+CHR$(11)
190 '
200 '----- CHAINE EFFACEMENT
210 EF$=CHR$(32)+CHR$(10)+CHR$(8)+CHR$(32)+CHR$(11)
220 '---- AVANCE PAPILLON
230 Y=10
240 FOR X=1 TO 24
250 LOCATE X,Y:PRINT EF$;P1$
260 FOR TP=1 TO 100:NEXT TP
270 LOCATE X,Y:PRINT EF$;P2$
280 FOR TP=1 TO 80:NEXT TP
290 NEXT X
300 LOCATE X,Y:PRINT EF$;EF$
310 INK 1,INT(RND(1)*16)
320 GOTO 240

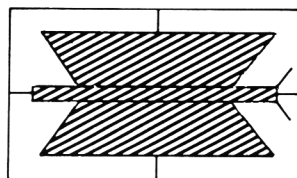
```



Position1



Position2



Voici deux exemples de dessins définis par quatre caractères graphiques (16×16 points).

```

10 ;---- soleil
20
30 MODE 1:INK 0,1:INK 1,24
40 C=145 ' 1ER CARACTERE A MODIFIER
50 SYMBOL AFTER C
60 SYMBOL C,1,65,33,17,11,7,239,31
70 SYMBOL C+1,0,8,16,32,192,192,224,254
80 SYMBOL C+2,15,7,27,34,66,2,2,0
90 SYMBOL C+3,224,192,160,16,8,0,0,0
100 L$=CHR$(C)+CHR$(C+1)+CHR$(10)+CHR$(8)+CHR$(8)+CHR$(C+2)+CHR$(C+3)+CHR$(11)
110 LOCATE 10,2:PRINT L$

```

```

10 '----- bateau
20 '
30 MODE 1:INK 0,1:INK 1,24
40 C=145 ' 1ER CARACTERE A MODIFIER
50 SYMBOL AFTER C
60 SYMBOL C,0,0,0,0,2,2,6,6
70 SYMBOL C+1,0,0,0,0,128,128,192,224
80 SYMBOL C+2,14,30,62,0,255,127,63,0
90 SYMBOL C+3,240,248,252,0,255,254,252,0
100 L$=CHR$(C)+CHR$(C+1)+CHR$(10)+CHR$(8)+CHR$(8)+CHR$(C+2)+CHR$(C+3)+CHR$(11)
110 LOCATE 10,2:PRINT L$

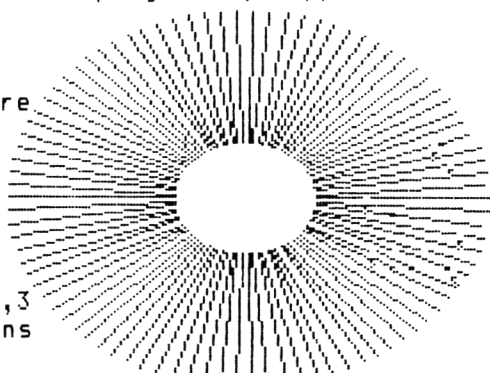
```

Pour réaliser une animation sans caractères graphiques nous dessinons une figure dans différentes positions et n'en "démasquons" qu'une seule à la fois.

```

10 ' ROTATION ROUE
20 '
30 ' On represente successivement une roue
40 ' dans 3 positions pour donner une
50 ' impression d'animation.
60 ' Attention! si vous interrompez le programme,frappez en
70 ' mode direct: INK 1,0
80 '
90 CF=26 ' fond
100 CE=6 ' ecriture
110 MODE 1:PAPER 0:PEN 1
120 INK 0,CF:INK 1,CF
130 INK 2,CF:INK 3,CF
140 XA=200:YA=200 ' centre
150 R=150 ' rayon
160 GOTO 180
170 '----- dessin positions 1,2,3
180 NR=30 ' Nombre de rayons
190 V=200 ' vitesse
200 DC=0:S=2:GOSUB 330
210 DC=(2*PI)/(3*NR):S=3:GOSUB 330
220 DC=DC*2:S=1:GOSUB 330
230 '----- rotation roue
240 INK 1,CF:INK 3,CE ' position 1
250 FOR TP=1 TO V:NEXT TP
260 INK 2,CE:INK 3,CF ' position 2
270 FOR TP=1 TO V:NEXT TP
280 INK 2,CF:INK 1,CE ' position 3
290 FOR TP=1 TO V:NEXT TP
300 V=V-2:IF V=0 THEN END
310 GOTO 240
320 '----- RAYONS
330 FOR P=1 TO NR
340 A=2*PI*P/NR+DC
350 DX=R*COS(A):DY=R*SIN(A)
360 PLOT XA+DX*0.3,YA+DY*0.3,S:DRAW XA+DX,YA+DY,S
370 NEXT P
380 RETURN

```



Pour simuler l'agrandissement d'une figure (carré sur l'exemple) nous la dessinons en mode "masque" puis nous la démasquons. Ainsi, la figure apparaît instantanément. Nous procédons de la même façon pour l'effacement.

```

10 '----- AGRANDISSEMENT D'UNE FIGURE
20 MODE 1
30 CF=0      ' fond
40 CE=26     ' ecriture
50 INK 0,CF
60 '
70 XC=200:YC=200      ' ORIGINE
80 R=10
90 '-----
100 STYLO=1:INK 1,CF:GOSUB 260      ' 1ER FIGURE
110 '
120 INK 1,CE:INK 2,CF      ' DEMASQ 1ERE/MASQUAGE 2EME
130 '
140 STYLO=0:R=R-4:XC=XC+2:YC=YC+2:GOSUB 260      ' EFFACEMENT 2ME
150 '
160 STYLO=2:INK 2,CF:R=R+8:XC=XC-4:YC=YC-4: GOSUB 260      ' 2EME FIG
URE
170 '
180 INK 1,CF:INK 2,CE      ' MASQUAGE 1ER,DEMASQUAGE 2EME
190 '
200 STYLO=0:R=R-4:XC=XC+2:YC=YC+2:GOSUB 260      ' EFFACEMENT 1ER
210 '
220 R=R+8:XC=XC-4:YC=YC-4
230 IF R>150 THEN INK 1,CE:END
240 GOTO 100
250 '----- FIGURE
260 PLOT XC,YC,STYLO
270 DRAWR R,0,STYLO
280 DRAWR 0,R,STYLO
290 DRAWR -R,0,STYLO
300 DRAWR 0,-R,STYLO
310 RETURN

```

Un éclair est dessiné en mode masque. Nous le démasquons momentanément et en même temps nous changeons la couleur de fond.

```

10 ' ECLAIR
20 '
30 MODE 1
40 PAPER 0:PEN 1:BORDER 1
50 INK 0,1:INK 1,26:INK 2,0:INK 3,0
60 GOSUB 250      ' OPTIONNEL
70 '--- DESSIN ECLAIR
80 PLOT 200,300,2
90 DRAW 234,340,2
100 DRAW 232,314,2
110 DRAW 280,362,2
120 '---
130 INK 2,6      ' ECLAIR APPARENT
140 FOR TP=1 TO 100:NEXT TP

```



```

150 INK 0,2:BORDER 2
160 '
170 FOR TP=1 TO 100:NEXT TP
180 '--
190 INK 2,1 ' ECLAIR MASQUE
200 INK 0,1:BORDER 1
210 '-
220 FOR TP=1 TO 3000:NEXT TP
230 GOTO 130
240 '----- IMMEUBLE
250 C=125 ' caractere a modifier
260 C$=CHR$(C)
270 SYMBOL AFTER C
280 SYMBOL C,255,129,129,129,129,129,129,255
290 '
300 XB=1:YB=23
310 H=15:L=7
320 PEN 3
330 FOR Y=YB TO YB-H STEP-1
340   FOR X=XB TO XB+L
350     LOCATE X,Y:PRINT C$
360   NEXT X
370 NEXT Y
380 PEN 1
390 RETURN
400 '----- sur ecran vert:
410 ' 50 ink 0,1:ink 1,26:ink 3,20
415 ' 130 ink 2,24
420 ' 150 ink 0,12:border 12

```

## DESSINATEUR

Le dessinateur présenté permet de choisir un caractère parmi ceux affichés en haut de l'écran à l'aide d'un curseur. Lorsque le caractère est choisi vous réalisez un dessin en déplaçant le curseur avec les quatre flèches. "L" permet de se déplacer sans écriture, "E" d'effacer une partie des dessins. Pour dessiner à nouveau, utiliser "B".

□ ★ ▢ ▣ ■ . ' - L , | r t - J - + ↵ ↶ ↷



★

**Flèches pour déplacer**  
**P:prend**  
**B:baissér L:lever E:effacer**  
**Couleurs:1,2,3**  
**FIN**



```

10 '----- DESSIN (CPC464 ET CPC664)
20 CE=1:CF=0 ' ecriture/fond
30 MODE 1
40 INK 0,26:INK 1,0:INK 2,2:PAPER 0:PEN 1
50 CC=139 ' debut caracteres graphiques
60 SYMBOL AFTER CC
70 SYMBOL CC,255,129,129,129,129,129,129,255
80 SYMBOL CC+1,8,8,28,127,28,34,65,0
90 SYMBOL CC+2,255,129,129,255,0,0,0,0
100 SYMBOL CC+3,255,129,129,255,129,129,129,255
110 X=10:Y=10 ' ccordonnees curseur
120 '
130 XM=1:YM=1 ' affichage caracteres de base
140 LOCATE XM,YM:FOR I=CC TO CC+18:PRINT CHR$(I) SPC(1):NEXT I
150 CP$=CHR$(CC):L=1:LOCATE X,Y:PRINT CP$
160 '
170 LOCATE 1,20:PRINT "Fleches pour deplacer"
180 LOCATE 1,21:PRINT "P:prend"
190 LOCATE 1,22:PRINT "B:baisser L:lever E:effacer F:fin"
200 LOCATE 1,23:PRINT "Couleurs:1,2,3"
210 '----- CURSEUR
220 LOCATE X,Y
230 XG=(X-1)*16+8:YG=399-(Y-1)*16-8
240 T=TEST(XG,YG):TB=TEST(XG+2,YG)
250 C$=INKEY$:IF LEN(C$)<>0 THEN 290
260 PLOT XG,YG,CE:PLOT XG+2,YG,CE:PLOT XG,YG,CF:PLOT XG+2,YG,CF
270 GOTO 250
280 '-----
290 PLOT XG,YG,T:PLOT XG+2,YG,TB
300 C$=UPPER$(C$)
310 C=ASC(C$)
320 IF C=242 THEN IF X>1 THEN X=X-1
330 IF C=243 THEN IF X<38 THEN X=X+1
340 IF C=240 THEN IF Y>1 THEN Y=Y-1
350 IF C=241 THEN IF Y<24 THEN Y=Y+1
360 IF C$>="1" AND C$<="3" THEN CE=VAL(C$):PEN CE
370 IF C$="P" THEN GOSUB 470
380 IF C$="B" THEN L=1
390 IF C$="L" THEN L=0
400 IF C$="E" THEN L=2
410 IF C$="F" THEN END
420 IF L=1 THEN IF Y>YM+1 THEN LOCATE X,Y:PRINT CP$
430 IF L=2 THEN IF Y>YM+1 THEN LOCATE X,Y:PRINT CHR$(32)
440 LOCATE 1,16:PRINT c$
450 GOTO 220
460 '----- ON PREND
470 IF Y=YM THEN CP$=CHR$(CC+INT((X-XM)/2))
480 LOCATE 1,18:PRINT CP$:PRINT CHR$(7);
490 RETURN
500 '-----
510 ' Pour CPC664:
520 ' Ajouter : 115 CURSOR 1
530 ' Supprimer: 230 240 260

```

## SOURIS

---

Ce programme permet de déplacer des caractères affichés à l'écran à l'aide d'un curseur que vous positionnez avec les quatre flèches.

Sur CPC 664, grâce à la fonction COPYCHR\$(#0) qui donne un caractère affiché à l'écran, et à la fonction CURSOR, le programme est assez simple.

En revanche, pour le CPC 464, il faut représenter un curseur et gérer directement la mémoire écran pour simuler COPYCHR\$(#0).

```

10 '----- SOURIS (CPC664)
20 CE=1:CF=0 ' ecriture/fond
30 MODE 2
40 INK 0,26:INK 1,0:PAPER 0:PEN 1
50 X=10:Y=10 ' coordonnees curseur
60 CURSOR 1
70 LOCATE 1,5:PRINT "DEPLACEZ LE CURSEUR AVEC LES FLECHES"
80 LOCATE 1,6:PRINT "'PRENEZ' UNE LETTRE EN APPUYANT SUR 'P'"
90 LOCATE 1,7:PRINT "DEPLACEZ LE CURSEUR PUIS 'POSEZ' LA LETTRE
EN APPUYANT SUR 'D'"
100 LOCATE 1,24:PRINT "F: fin"
110 '----- CURSEUR
120 LOCATE X,Y
130 C$=INKEY$:IF LEN(C$)<>0 THEN 160
140 GOTO 130
150 '--
160 C=ASC(C$)
170 C$=UPPER$(C$)
180 IF C=242 THEN IF X>1 THEN X=X-1
190 IF C=243 THEN IF X<79 THEN X=X+1
200 IF C=240 THEN IF Y>1 THEN Y=Y-1
210 IF C=241 THEN IF Y<25 THEN Y=Y+1
220 IF C$="P" THEN GOSUB 270
230 IF C$="D" THEN GOSUB 320
240 IF C$="F" THEN END
250 GOTO 120
260 '----- ON PREND
270 CP$=COPYCHR$(#0)
280 PRINT CHR$(32)
290 LOCATE 1,20:PRINT CP$
300 RETURN
310 '----- ON POSE
320 IF COPYCHR$(#0)<>CHR$(32) THEN RETURN
330 PRINT CP$
340 RETURN

```

```

DEPLACEZ LE      EUR AVEC LES FLECHES
'PRENEZ' UNE LETTRE EN APPUYANT SUR 'P'
DEPLACEZ LE CURSEUR PUIS 'POSEZ' LA LETTRE EN APPUYANT SUR 'D'

```

C  
U  
R  
S

La version ci-dessous fonctionne sur CPC 464, 664 et 6128.

```

10 '----- SOURIS (CP464+CPC664)
20 CE=1:CF=0 ' ecriture/fond
30 MODE 2
40 INK 0,26:INK 1,0:INK 2,2:INK 3,6:PAPER 0:PEN 1
50 X=10:Y=10 ' coordonnees curseur
60 '
70 LOCATE 1,5:PRINT "DEPLACEZ LE CURSEUR AVEC LES FLECHES"
80 LOCATE 1,6:PRINT "'PRENEZ' UNE LETTRE EN APPUYANT SUR 'P'"
90 LOCATE 1,7:PRINT "DEPLACEZ LE CURSEUR PUIS 'POSEZ' LA LETTRE
EN APPUYANT SUR 'D'"
100 LOCATE 1,24:PRINT "F:fin"
110 '----- curseur clignotant
120 LOCATE X,Y
130 GOSUB 470 ' Sauvegarde caractere sous le curseur
140 '
150 LOCATE X,Y
160 C$=INKEY$:IF LEN(C$)<>0 THEN 210
170 PRINT CHR$(143)
180 GOSUB 530
190 GOTO 150
200 '
210 GOSUB 530 ' Restitution caractere sous curseur
220 C$=UPPER$(C$)
230 C=ASC(C$)
240 IF C=242 THEN IF X>1 THEN X=X-1
250 IF C=243 THEN IF X<78 THEN X=X+1
260 IF C=240 THEN IF Y>1 THEN Y=Y-1
270 IF C=241 THEN IF Y<25 THEN Y=Y+1
280 IF C$="P" THEN GOSUB 330
290 IF C$="D" THEN GOSUB 410
300 IF C$="F" THEN END
310 GOTO 120
320 '----- ON PREND
330 N=1
340 FOR A=49152+X-1+(Y-1)*80 TO 65534 STEP 2048
350 XP(N)=PEEK(A):N=N+1
360 NEXT A
370 PRINT CHR$(32)
380 PRINT CHR$(7);
390 RETURN
400 '----- ON POSE
410 N=1
420 FOR A=49152+X-1+(Y-1)*80 TO 65534 STEP 2048
430 POKE A,XP(N):N=N+1
440 NEXT A
450 RETURN
460 '----- sauvegarde caractere sous curseur
470 N=1
480 FOR A=49152+X-1+(Y-1)*80 TO 65534 STEP 2048
490 X(N)=PEEK(A):N=N+1
500 NEXT A
510 RETURN
520 '----- restitution caractere

```





```

530 N=1
540 FOR A=49152+X-1+(Y-1)*80 TO 65534 STEP 2048
550 POKE A,X(N):N=N+1
560 NEXT A
570 RETURN

```

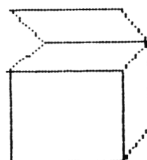
## STOCKAGE D'UN DESSIN DANS UN FICHIER

Nous réalisons un dessin à l'aide de segments de droites. Les coordonnées des points sont stockées dans un fichier séquentiel sur cassette ou disquette.

```

10 '----- STOCKAGE D'UN DESSIN DANS UN FICHIER
20 CE=1:CF=0 ' ecriture/fond
30 MODE 1
40 INK 0,0:INK 1,26:INK 2,2:INK 3,6:PAPER 0:PEN 1
50 LOCATE 1,20:PRINT "PREMIER POINT: fleches puis 'P'"
60 LOCATE 1,21:PRINT "AUTRES POINTS: fleches puis 'D'"
70 LOCATE 1,22:PRINT "1,2,3:couleurs"
80 LOCATE 1,23:PRINT "F:FIN"
90 X=200:Y=200
100 INPUT "NOM FICHIER ";NF$
110 OPENOUT NF$
120 '----- CURSEUR CLIGNOTANT
130 T=TEST(X,Y)
140 '
150 C$=INKEY$:IF LEN(C$)<>0 THEN 200
160 PLOT X,Y,CE
170 PLOT X,Y,CF
180 GOTO 150
190 '
200 PLOT X,Y,T
210 C=ASC(C$)
220 C$=UPPER$(C$)
230 IF C=242 THEN X=X-2
240 IF C=243 THEN X=X+2
250 IF C=240 THEN Y=Y+2
260 IF C=241 THEN Y=Y-2
270 LOCATE 1,17:PRINT C$
280 IF C$="P" THEN PLOT X,Y,CE:GOSUB 340:XA=X:YA=Y
290 IF C$="D" THEN PLOT XA,YA,CE:GOSUB 340:DRAW X,Y,CE:YA=Y:XA=X
300 IF C$>="0" AND C$<="3" THEN CE=VAL(C$):GOSUB 340
310 IF C$="F" THEN CLOSEOUT:END
320 GOTO 130
330 '----- STOCKAGE FICHIER
340 PRINT #9,C$
350 PRINT #9,X
360 PRINT #9,Y
370 RETURN

```



```

10 '----- LECTURE FICHIER DESSIN
20 MODE 1
30 CF=0:CE=1          ' fond/ecriture
40 INK 0,0:INK 1,26:INK 2,2:INK 3,6
50 '
60 INPUT "NOM FICHIER ";NF$
70 OPENIN NF$
80 '
90 '
100 IF EOF THEN CLOSEIN:END
110 INPUT #9,C$,X,Y
120 IF C$="P" THEN PLOT X,Y,CE
130 IF C$="D" THEN DRAW X,Y,CE
140 IF C$>="0" AND C$<="3" THEN CE=VAL(C$)
150 GOTO 100

```

## DESSINATEUR ET COMPOSITION DE DESSINS

Nous reprenons le dessinateur présenté dans le tome 1, page 127, mais cette fois nous enregistrons toutes les commandes et les coordonnées des points dans des tables CM\$(),X(),Y() et CE().

| CM\$() | X() | Y() | CE() |
|--------|-----|-----|------|
| P      | 81  | 371 | 1    |
| D      | 81  | 315 | 1    |
| D      | 47  | 315 | 1    |
| D      | 81  | 371 | 1    |
| F      | 77  | 343 | 1    |

Cette méthode présente plusieurs avantages :

- ☐ Elle permet d'exécuter à nouveau un dessin en supprimant une ou plusieurs commandes.
- ☐ La sauvegarde du dessin est moins encombrante et plus rapide qu'une sauvegarde binaire.
- ☐ Le dessin réalisé peut être agrandi.
- ☐ Nous pouvons ajouter plusieurs dessins.

Le dessin se fait dans une fenêtre en haut à gauche de l'écran.

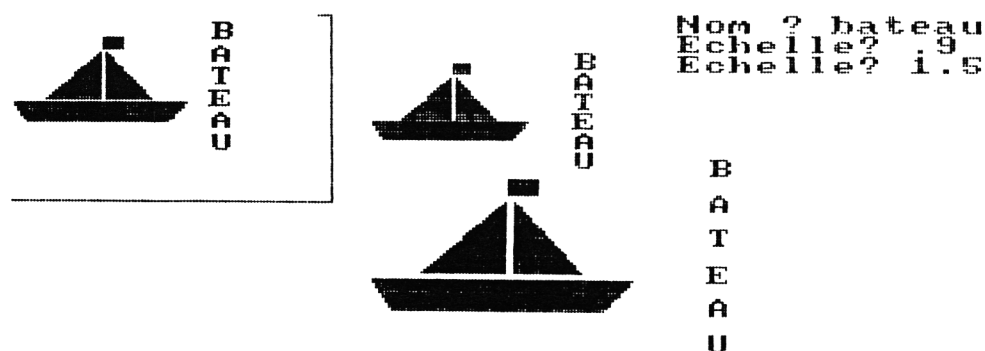
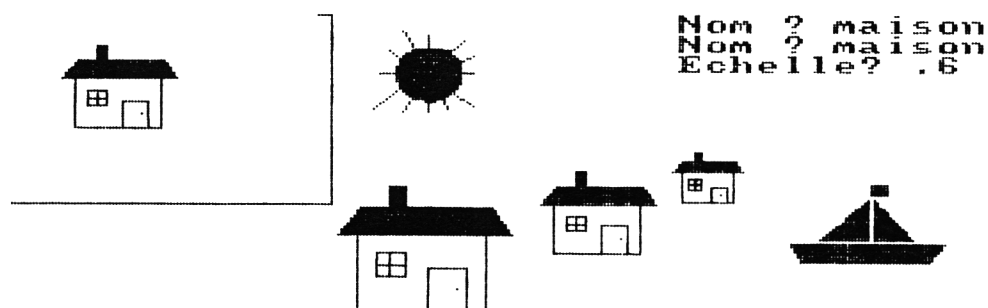
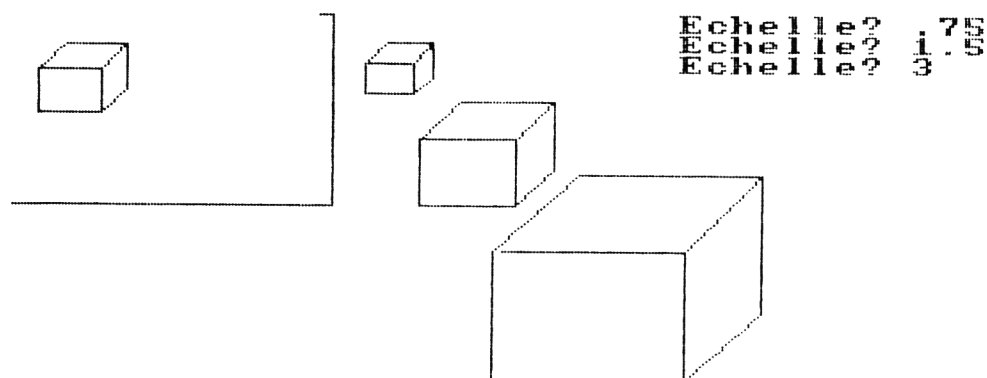
Par exemple, pour tracer une droite, positionnez le curseur avec les flèches puis frappez "P". Ensuite déplacez le curseur et frappez "D".

Pour tracer un rectangle, frappez "P" puis déplacez le curseur vers le sommet opposé et frappez "R".

Lorsque le dessin est réalisé, vous le sauvegardez en frappant "#".

Vous pouvez ensuite l'appeler avec "\$". Il est affiché dans la fenêtre. Pour l'afficher dans un autre endroit de l'écran, utiliser "%". Auparavant, positionnez le curseur à l'endroit où doit être affiché le dessin.

"+" et "-" permettent d'augmenter et de diminuer la vitesse de déplacement du curseur.



```

10 ' COMPOSITION DE DESSINS
20 '
30 GOSUB 1400 ' Affichage menu
40 DIM X(200),Y(200),CM$(200),CE(200)
50 DIM XX(200),YY(200)
60 CE=1:CF=0 ' couleur ecriture et fond
70 INK 0,26:INK 1,0:PAPER CF:PEN CE ' fond blanc/encre noire
80 INK 2,2:INK 3,6
90 RPT=3
100 SPEED KEY 20,3 ' repetition plus rapide
110 MODE 1
120 PLOT 1,399,1:DRAWR 200,0:DRAWR 0,-150:DRAWR -200,0
130 WINDOW #1,27,40,1,3
140 WINDOW #2,1,12,1,9
150 PRINT #2,"Des de base"
160 XA=100:YA=350 ' point precedent
170 XB=XA:YB=YA:X=XA+20:Y=YA
180 LOCATE 1,24:PRINT "P:point D:droite R:rect C:cercle Y:tgls"
190 LOCATE 1,25:PRINT "coul:0,1,2,3 F:peindre #:sauv $:lect"
200 '----- curseur clignotant
210 T=TEST(X,Y)
220 '
230 C$=INKEY$:IF LEN(C$)<>0 THEN 270
240 PLOT X,Y,CE:PLOT X,Y,CF
250 GOTO 230
260 '
270 PLOT X,Y,T
280 C=ASC(C$)
290 IF C=242 THEN IF X>2 THEN X=X-2:GOTO 210
300 IF C=243 THEN IF X<600 THEN X=X+2:GOTO 210
310 IF C=240 THEN IF Y<397 THEN Y=Y+2:GOTO 210
320 IF C=241 THEN IF Y>2 THEN Y=Y-2:GOTO 210
330 C$=UPPER$(C$):LOCATE 1,21:PRINT C$
340 IF C$="+" THEN IF RPT>1 THEN RPT=RPT-1:SPEED KEY 20,RPT
350 IF C$="-" THEN IF RPT<6 THEN RPT=RPT+1:SPEED KEY 20,RPT
360 IF C$="!" THEN TT=ABS(TT-1):GOSUB 770:GOTO 210
370 IF TT=1 THEN TAG:PLOT X,Y,1:PRINT C$;:X=X+16:TAGOFF:GOSUB 770:GOTO 210
380 IF C$="P" THEN PLOT X,Y,CE:XB=XA:YB=YA:XA=X:YA=Y:GOSUB 770
390 IF C$="D" THEN PLOT XA,YA,CE:DRAW X,Y,CE:XB=XA:YB=YA:YA=Y:XA=X:GOSUB 770
400 IF C$="C" THEN GOSUB 600:GOSUB 770
410 IF C$="R" THEN GOSUB 550:XA=X:YA=Y:GOSUB 770
420 IF C$="Y" THEN GOSUB 680:XB=XA:YB=YA:XA=X:YA=Y:GOSUB 770
430 IF C$="G" THEN AC=CE:CE=0:GOSUB 550:CE=AC:GOSUB 770 ' gomme
440 IF C$>="0" AND C$<="9" THEN CE=VAL(C$)
450 IF C$="F" THEN MOVE X,Y,CE:FILL CE:GOSUB 770 ' CPC664
460 IF C$="S" THEN SAVE "DES",B,&C000,&4000
470 IF C$="L" THEN LOAD "DES"
480 IF C$="#" THEN GOSUB 800 ' sauvegarde dessin de base
490 IF C$="$" THEN GOSUB 920 ' lecture dessin de base
500 IF C$="%" THEN GOSUB 1040 ' affichage dessin base avec ech
510 IF C$="&" THEN IF P>1 THEN P=P-1:CLS #2:GOSUB 1220
520 IF C$="A" THEN END
530 GOTO 210

```

```

540 '----- rectangle plein
550 FOR Y1=YA TO Y STEP SGN(Y-YA)
560 PLOT XA,Y1,CE: DRAW X,Y1,CE
570 NEXT Y1
580 RETURN
590 '----- Cercle plein
600 R=SQR((XA-X)^2+(YA-Y)^2):R2=R^2 ' ^=fleche haut
610 FOR DX=-R TO R
620 DY=SQR(R2-(DX^2)) ' ^=fleche haut
630 PLOT XA+DX,YA+DY,CE: DRAW XA+DX,YA-DY,CE
640 NEXT DX
650 RETURN
660 '----- triangle plein
670 ' valider 2 points avec 'P' puis un troisieme avec 'Y'
680 D=SQR((YB-YA)^2+(XB-XA)^2)
690 IF D=0 THEN RETURN
700 CX=(XA-XB)/D:CY=(YA-YB)/D
710 FOR DD=0 TO D STEP 0.5
720 X3=XB+DD*CX:Y3=YB+DD*CY
730 PLOT X,Y,CE: DRAW X3,Y3,CE
740 NEXT DD
750 RETURN
760 '----- MAJ TABLES
770 P=P+1:X(P)=X:Y(P)=Y:CM$(P)=C$:CE(P)=CE
780 XX(P)=X:YY(P)=Y:RETURN
790 '----- SAUVEGARDE COMMANDES
800 INPUT #1,"Nom";NF$
810 OPENOUT NF$
820 PRINT #9,P
830 FOR J=1 TO P
840 PRINT #9,XX(J)
850 PRINT #9,YY(J)
860 PRINT #9,CM$(J)
870 PRINT #9,CE(J)
880 NEXT J
890 CLOSEOUT
900 RETURN
910 '----- AFFICHAGE DESSIN BASE
920 INPUT #1,"Nom ";NF$
930 CLS #2
940 GOSUB 1140
950 SX=X:SY=Y
960 X=1:Y=399:ECH=1
970 FOR J=1 TO P
980 X(J)=XX(J)*ECH+X:Y(J)=(YY(J)-400)*ECH+Y
990 NEXT J
1000 GOSUB 1220
1010 X=SX:Y=SY
1020 RETURN
1030 '----- DESSIN AVEC ECHELLE
1040 INPUT #1,"Echelle";ECH
1050 IF ECH<=0 THEN ECH=1
1060 FOR J=1 TO P
1070 X(J)=XX(J)*ECH+X:Y(J)=(YY(J)-400)*ECH+Y
1080 NEXT J
1090 SX=X:SY=Y
1100 GOSUB 1220
1110 X=SX:Y=SY:FOR J=1 TO P:X(J)=XX(J):Y(J)=YY(J):NEXT J
1120 RETURN

```

```

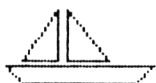
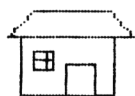
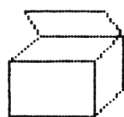
1130 '----- LECTURE DESSIN BASE
1140 OPENIN NF$
1150 INPUT #9,P
1160 FOR J=1 TO P
1170 INPUT #9,XX(J),YY(J),CM$(J),CE(J)
1180 NEXT J
1190 CLOSEIN
1200 RETURN
1210 '----- REDESSINE
1220 FOR J=1 TO P
1230 X=X(J):Y=Y(J):CE=CE(J)
1240 IF CM$(J)="!" THEN TT=ABS(TT-1):GOTO 1340
1250 IF TT=1 THEN TAG:PLOT X,Y,CE:PRINT CM$(J);:X=X+16:TAGOFF:G
OTO 1340
1260 IF CM$(J)="P" THEN PLOT X,Y,CE:XB=XA:YB=YA:XA=X:YA=Y
1270 IF CM$(J)="D"
THEN PLOT XA,YA,CE:DRAW X,Y,CE:XB=XA:YB=YA:XA=X:YA=Y
1280 IF CM$(J)="C" THEN GOSUB 600
1290 IF CM$(J)="R" THEN GOSUB 550:XB=XA:YB=YA:XA=X:YA=Y
1300 IF CM$(J)="Y" THEN GOSUB 680:XB=XA:YB=YA:XA=X:YA=Y
1310 IF CM$(J)="F" THEN MOVE X,Y,CE:FILL CE ' CPC664
1320 IF CM$(J)="!" THEN TT=ABS(TT-1)
1330 IF CM$(J)="G" THEN AC=CE:CE=0:GOSUB 550:CE=AC
1340 NEXT J
1350 RETURN
1360 '
1370 XB=XA:YB=YA:YA=Y:XA=X
1380 RETURN
1390 '----- MENU (OPTIONNEL)
1400 MODE 2:PAPER 0:PEN 1:INK 0,26:INK 1,0
1410 PRINT "Permet de dessiner des figures de base dans une fene
tre"
1420 PRINT "en haut a gauche"
1430 PRINT "Fleches,P=premier point,D=droite avec point preceden
t,C=cercle"
1440 PRINT "R=rectangle,Y=triangle,F=peindre(CPC664)"
1450 PRINT
1460 PRINT "EX: -DROITE: Fleches puis 'P'.Ensuite fleches puis '
D'."
1470 PRINT " -CERCLE: 'P' pour centre.Ensuite fleches et 'C'
"
1480 PRINT "Couleurs: 0,1,2,3 (avec 0,le curseur apparait seulem
ent sur surface coloriee)"
1490 PRINT "G gomme un rectangle"
1500 PRINT "+ et - changent la vitesse du curseur"
1510 PRINT "& annule la derniere commande"
1520 PRINT "! fait passer du mode graphique au texte et inverse
ment"
1530 PRINT:PRINT "SAUVEGARDE DESSIN DE BASE: '#' "
1540 PRINT "LECTURE DESSIN DE BASE: '$' "
1550 PRINT "% permet de les afficher avec une echelle"
1560 PRINT "dans differents endroits de l'ecran grace au curseur
"
1570 PRINT "L'ensemble du dessin compose peut etre sauvegarde pa
r 'S' et lu par 'L'"
1580 PRINT:PRINT "A: arret programme"
1590 PRINT:INPUT "APPUYER SUR ENTER";X$
1600 RETURN
1610 '-----
1620 ' Pour disque ajouter: 25 ON ERROR GOTO 1800
1630 ' 1800 IF DERR=146 THEN RESUME 920

```

## APPRENTISSAGE DE MOTS

---

Nous affichons une silhouette d'objet et nous demandons le nom de cet objet. Si une seule lettre de la réponse est oubliée (ou fausse), nous affichons la bonne orthographe du mot.



Les dessins sont tracés par segments de droite. Une discontinuité dans le tracé est spécifiée par "0,0". "999,999" spécifie la fin d'un dessin.

Naturellement, les dessins auraient pu être stockés dans un fichier à l'aide d'un programme dessinateur évitant ainsi la recopie des coordonnées en DATA.

```

10 ' APMOT      APPRENTISSAGE DE MOTS
20 '
30 MODE 1:INK 0,0:INK 1,26
40 NMOT=5      ' nombre de mots
50 '
60 '----- BOITE
70 DATA 16,354, 70,354, 70,310
80 DATA 16,310, 16,354, 36,374
90 DATA 86,374, 70,354, 70,310
100 DATA 88,332, 88,374, 76,394
110 DATA 26,394, 36,374, 999,999
120 DATA BOITE
130 '
140 '----- MAISON
150 DATA 12,364, 90,364, 76,384
160 DATA 28,384, 12,364, 20,364
170 DATA 20,318, 78,318, 78,364
180 DATA 0,0
190 DATA 66,342, 48,342, 48,318
200 DATA 66,318, 66,342
210 DATA 0,0
220 DATA 28,352, 40,352, 40,338
230 DATA 28,338, 28,352, 28,346
240 DATA 40,346, 40,338, 36,338
250 DATA 36,352
260 DATA 999,999
270 DATA MAISON
280 '----- BOUGIE
290 DATA 42,358, 42,286, 66,286
300 DATA 66,358, 42,358, 52,358
310 DATA 48,366, 46,374, 46,384
320 DATA 52,390, 56,380, 56,374
330 DATA 54,366, 52,360
340 DATA 999,999
350 DATA BOUGIE

```

```

360 '----- BATEAU
370 DATA 38,382, 16,342, 38,342
380 DATA 38,382
390 DATA 0,0
400 DATA 44,382, 44,342, 70,342
410 DATA 44,382
420 DATA 0,0
430 DATA 6,338, 98,338, 86,324
440 DATA 16,324, 16,324, 6,338
450 DATA 999,999
460 DATA BATEAU
470 '----- ETOILE
480 DATA 12,366, 38,366, 48,392
490 DATA 64,366, 92,366, 74,344
500 DATA 88,320, 52,330, 28,310
510 DATA 32,342, 12,364
520 DATA 999,999
530 DATA ETOILE
540 '===== LECTURE DESSIN AU HASARD
550 P=INT(RND(1)*NMOT):IF P=AP THEN 550
560 AP=P:RESTORE
570 IF P=0 THEN 630
580 FOR J=1 TO P
590 READ X,Y:IF X=999 THEN 600 ELSE 590
600 READ MOT$
610 NEXT J
620 '----- AFFICHAGE
630 MODE 1
640 READ XA,YA
650 PLOT XA,YA,1
660 '
670 READ X,Y:IF X=999 THEN 730
680 IF X=0 THEN READ X,Y:PLOT X,Y,1:GOTO 700
690 DRAW X,Y,1
700 XA=X:YA=Y
710 GOTO 670
720 '----- REPONSE
730 READ MOT$
740 LOCATE 1,20:INPUT "Reponse (ou FIN) ";R$
750 R$=UPPER$(R$):IF R$="FIN" THEN END
760 IF R$=UPPER$(MOT$) THEN PRINT "OK":GOTO 830
770 '
780 GOSUB 860
790 '
800 IF R=LEN(MOT$)-1
    THEN PRINT "LA BONNE ORTHOGRAPHE EST: ";MOT$:GOTO 830
810 PRINT "LA BONNE REPONSE EST: ";MOT$
820 '
830 FOR TP=1 TO 1000:NEXT TP
840 GOTO 550
850 '----- VOISINAGE DE LA REPONSE
860 R=0 ' nombre de bonnes lettres
870 FOR K=1 TO LEN(R$)

```





```

880 C$=MID$(R$,K,1)
890 FOR J=1 TO LEN(MOT$)
900 IF C$=MID$(MOT$,J,1) THEN R=R+1:GOTO 920
910 NEXT J
920 NEXT K
930 RETURN

```

## COMPOSITION DE PAYSAGE AVEC ANIMATION \_\_\_\_\_

A l'aide de trois figures de base (arbre, maison, locomotive), vous composez un paysage. Le choix des figures se fait à l'aide d'un curseur que vous déplacez avec les quatre flèches.

En frappant "P", vous "prenez" une des trois figures de base (BEEP signale que la figure a été choisie). "D" permet de "déposer" la figure choisie. La couleur se choisit en frappant 1,2,3.

Pour animer une figure, placer le curseur devant elle puis frapper "+" ou "-" (une ou plusieurs fois). Une figure animée peut également être stoppée.

S\$() contient les dessins.

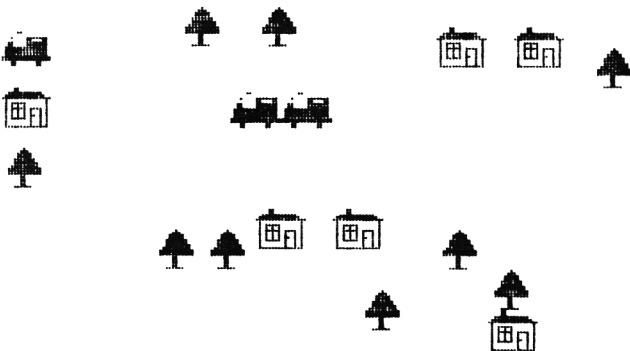
X() et Y() contiennent les coordonnées des dessins.

V() contient les vitesses.

### Exemple :

Placez le curseur sur le centre de la locomotive et appuyez sur "P" puis déplacez le curseur et appuyez sur "D". Pour animer la figure, placez le curseur devant la locomotive et appuyez sur "+" ou "-".

Pour "accrocher" une figure derrière une autre, placez le curseur derrière la première figure et appuyez sur "A".



Fleches puis P:prendre D:deposer A:accro  
+:avancer -:reguler F:fin  
Couleurs: 1,2,3

```

10 '----- COMPOSITION DE PAYSAGE AVEC ANIMATION
20 MODE 1
30 CE=1:CF=0          ' ecriture/fond
40 PAPER 0:PEN 1
50 INK 0,0:INK 1,26:INK 2,2:INK 3,6
60 DIM X(32),Y(32),V(32) ' coordonnees/vitesse
70 DIM S$(32)
80 NS=3          ' nombre de figures de base
90 NN=NS          ' nombre total de figures
100 '
110 LOCATE 1,23:PRINT "Fleches puis P:prendre D:deposer A:accroc
her"
120 LOCATE 1,24:PRINT "+:avancer -:reculer F:fin"
130 LOCATE 1,25:PRINT "Couleurs: 1,2,3"
140 EF$=CHR$(32)+CHR$(10)+CHR$(8)+CHR$(32)+CHR$(11) ' effacement
150 '----- locomotive
160 C=145          ' 1er caractere a modifier
170 SYMBOL AFTER C
180 SYMBOL C,0,0,6,0,16,0,48,48
190 SYMBOL C+1,0,0,0,0,254,198,254,254
200 SYMBOL C+2,127,127,127,127,255,255,24,24
210 SYMBOL C+3,254,254,254,254,255,255,24,24
220 S$(1)=CHR$(C)+CHR$(C+1)+CHR$(10)+CHR$(8)+CHR$(8)+CHR$(C+2)+C
HR$(C+3)+CHR$(11)
230 '----- maison
240 SYMBOL C+4,12,12,63,127,255,64,95,85
250 SYMBOL C+5,0,0,248,252,255,2,2,2
260 SYMBOL C+6,95,85,85,85,95,64,64,127
270 SYMBOL C+7,122,74,74,106,74,74,74,254
280 S$(2)=CHR$(C+4)+CHR$(C+5)+CHR$(10)+CHR$(8)+CHR$(8)+CHR$(C+6)
+CHR$(C+7)+CHR$(11)
290 '----- arbre
300 SYMBOL C+8,1,3,7,7,15,15,15,31
310 SYMBOL C+9,0,128,192,192,224,224,240,240
320 SYMBOL C+10,63,63,63,3,3,3,3,15
330 SYMBOL C+11,248,248,248,0,0,0,0,192
340 S$(3)=CHR$(C+8)+CHR$(C+9)+CHR$(10)+CHR$(8)+CHR$(8)+CHR$(C+10)
)+CHR$(C+11)+CHR$(11)
350 '----- AFFICHAGE FIGURES DE BASE
360 FOR S=1 TO NS
370   X(S)=1:Y(S)=S*3
380   LOCATE X(S),Y(S):PRINT S$(S)
390 NEXT S
400 '
410 X=64:Y=384
420 '----- Curseur clignotant
430 T=TEST(X,Y):TB=TEST(X+2,Y)
440 '
450 C$=INKEY$:IF LEN(C$)<>0 THEN 510
460 PLOT X+2,Y,CE
470 PLOT X,Y,CE:FOR TP=1 TO 10:NEXT TP:PLOT X,Y,CF
480 GOSUB 870
490 GOTO 450
500 '---
510 PLOT X,Y,T:PLOT X+2,Y,TB

```

```

520 C$=UPPER$(C$)
530 C=ASC(C$)
540 IF C=242 THEN IF X>15 THEN X=X-16:GOTO 430
550 IF C=243 THEN IF X<580 THEN X=X+16:GOTO 430
560 IF C=240 THEN IF Y<384 THEN Y=Y+16:GOTO 430
570 IF C=241 THEN IF Y>15 THEN Y=Y-16:GOTO 430
580 IF C$="F" THEN END
590 IF C$>"0" AND C$<"4" THEN CE=VAL(C$):PEN CE
600 IF C$="P" THEN GOSUB 670
610 IF C$="D" THEN GOSUB 780
620 IF C$="A" THEN GOSUB 1080
630 IF C$="+" THEN V=1:GOSUB 980
640 IF C$="-" THEN V=2:GOSUB 980
650 GOTO 430
660 '----- ON PREND
670 SS=0
680 XC=INT(X/16)+1:YC=INT((400-Y)/16)+1
690 FOR S=1 TO NS
700 IF XC>=X(S) AND XC<=X(S)+2 AND YC>=Y(S) AND YC<=Y(S)+2
    THEN 730
710 NEXT S
720 RETURN
730 SS=S
740 PRINT CHR$(7);
750 LOCATE 1,21:PRINT S$(SS)
760 RETURN
770 '----- ON POSE
780 IF SS=0 THEN RETURN
790 IF NN>30 THEN RETURN
800 XC=INT(X/16)+1:YC=INT((400-Y)/16)+1
810 IF XC<4 THEN RETURN
820 NN=NN+1:S$(NN)=S$(SS)
830 LOCATE XC,YC:PRINT S$(NN)
840 Y(NN)=YC:X(NN)=XC
850 RETURN
860 '----- AVANCE
870 IF NN=NS THEN RETURN
880 FOR S=NS+1 TO NN
890 IF V(S)=0 THEN 950
900 X(S)=X(S)-V(S)
910 IF X(S)<6
    THEN LOCATE X(S),Y(S):PRINT EF$;EF$;EF$;EF$;EF$:X(S)=34
920 IF X(S)>34
    THEN LOCATE X(S),Y(S):PRINT EF$;EF$;EF$;EF$:X(S)=6
930 IF V(S)>0 THEN LOCATE X(S),Y(S):PRINT S$(S);EF$;EF$
940 IF V(S)<0 THEN LOCATE X(S)-1,Y(S):PRINT EF$;EF$;S$(S);EF$
950 NEXT S
960 RETURN
970 '----- REGLAGE VITESSE
980 XC=INT(X/16)+1:YC=INT((400-Y)/16)+1
990 FOR S=NS+1 TO NN
1000 IF YC>=Y(S) AND YC<=Y(S)+1 THEN 1040
1010 NEXT S

```



```

1020 RETURN
1030 '
1040 IF V=1 THEN IF XC>3 THEN V(S)=V(S)+0.5:V=0
1050 IF V=2 THEN IF XC>3 THEN V(S)=V(S)-0.5:V=0
1060 RETURN
1070 '----- ACCROCHAGE
1080 XC=INT(X/16)+1:YC=INT((400-Y)/16)+1
1090 '
1100 FOR S=NS+1 TO NN
1110   IF YC<>Y(S) THEN 1130
1120   IF XC-2=X(S)
      THEN S$(S)=S$(S)+S$(SS):LOCATE XC,YC:PRINT S$(SS):RETURN
1130 NEXT S
1140 RETURN

```

## MINI INTERPRÉTEUR LOGO

La plupart des langages proposent des ordres graphiques où sont spécifiées les coordonnées X et Y des droites à tracer.

Le langage LOGO dispose d'ordres graphiques originaux :

**AVANCE distance** Trace une droite d'une longueur égale à la distance spécifiée.

**ROTATION angle** Change la direction du tracé en lui ajoutant l'angle spécifié.

Un curseur indique à chaque instant la direction courante. En frappant "ROTATION 90" vous voyez le curseur tourner de 90 degrés.

```

EX:  AVANCE 40
      ROTAT 90

FINI pour stopper

CMDE ? avance 70
CMDE ? rotat 90
CMDE ? avance 70
CMDE ? rotat 90
CMDE ? avance 70
CMDE ? rotat 90
CMDE ?

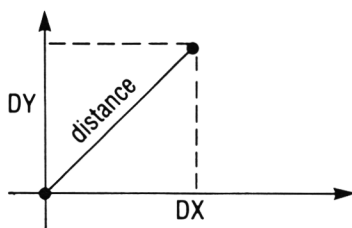
```



Le tracé de la droite en fonction de l'angle courant se fait ainsi :

$DX = \text{distance} \times \cos(\text{angle})$

$DY = \text{distance} \times \sin(\text{angle})$



Les instructions de base sont :

|                 |                                     |
|-----------------|-------------------------------------|
| LEVE            | Le tracé n'apparaît pas.            |
| BAISSE          | Le tracé apparaît.                  |
| EFFACE          | Efface l'écran.                     |
| COULEUR numéro  | Change la couleur (1,2,3).          |
| COLONNE numéro  | Positionne le curseur (de 1 à 600). |
| LIGNE numéro    | Positionne le curseur (de 1,399).   |
| DISTANCE valeur | Change la distance par défaut.      |
| DIS+ valeur     | Augmente la distance courante.      |
| CENTRE          | Positionne le curseur au centre.    |

## BOUCLES

Les instructions entre REPETE X et ENCORE sont exécutées X fois.

```
CMDE ? REPETE 4
CMDE ? AVANCE 50
CMDE ? ROTAT 90
CMDE ? ENCORE
```



Cette séquence dessine un carré.

## FONCTIONS

Des "fonctions" se déclarent ainsi :

```
CMDE ? DEFN TRIANGLE
CD FCT ? REPETE 3
CD FCT ? AVANCE 70
CD FCT ? ROTAT 120
CD FCT ? ENCORE
CD FCT ? FINF
CMDE ?
CMDE ? TRIANGLE
```



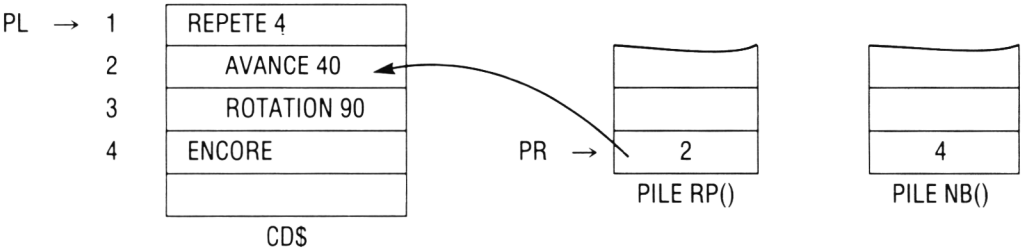
Pour dessiner un triangle, il suffit ensuite d'écrire :

TRIANGLE

Au moment de la lecture des instructions, la "fonction" est remplacée par la suite des instructions qui y sont écrites. Jusqu'à cinq fonctions peuvent être définies.

Remarques sur le programme :

- Pour les boucles “REPETE”, les commandes sont stockées dans une table des commandes CD\$(.). Un système de pile permet de gérer des boucles emboîtées.



- Les fonctions sont stockées dans une table à deux dimensions F\$(.). Chaque ligne de la table peut contenir jusqu'à 20 commandes.

Table F\$(.)

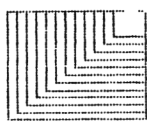
|   |          |          |        |             |        |
|---|----------|----------|--------|-------------|--------|
| 1 | CARRE    | REPETE 4 | AVANCE | ROTATION 90 | ENCORE |
| 2 | TRIANGLE |          |        |             |        |
|   |          |          |        |             |        |

Nous donnons ici plusieurs exemples de dessins obtenus avec le simulateur LOGO.

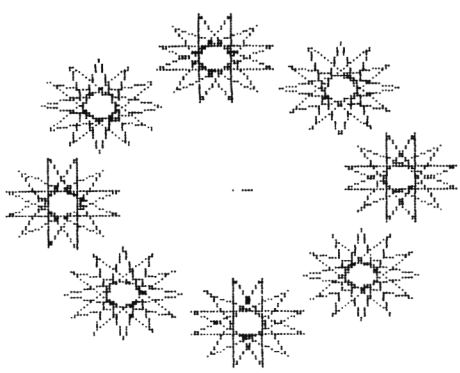
```
CMDE ? REPETE 36
CMDE ? AVANCE 10
CMDE ? ROTAT 10
CMDE ? ENCORE
CMDE ?
```



```
CMDE ? DEFI CARRE
CD FCT ? REPETE 4
CD FCT ? AVANCE
CD FCT ? ROTAT 90
CD FCT ? ENCORE
CD FCT ? FINF
CMDE ?
CMDE ? REPETE 12
CMDE ? CARRE
CMDE ? DIS+ 6
CMDE ? ENCORE
```



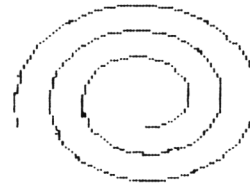
```
CMDE ? DEFI ETOILE
CD FCT ? REPETE 12
CD FCT ? AVANCE 70
CD FCT ? ROTAT 150
CD FCT ? ENCORE
CD FCT ? FINF
CMDE ?
CMDE ? REPETE 8
CMDE ? LEVE
CMDE ? AVANCE 70
CMDE ? BAISSSE
CMDE ? ETOILE
CMDE ? LEVE
CMDE ? AVANCE -70
CMDE ? ROTAT 45
CMDE ? ENCORE
```



```

CMDE  ?  DISTANCE 4
CMDE  ?  ?
CMDE  ?  REPETE 100
CMDE  ?  AVANCE
CMDE  ?  DIS+ 0.1
CMDE  ?  ROTAT 10
CMDE  ?  ENCORE
CMDE  ?

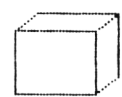
```



```

CMDE  ?  REPETE 4
CMDE  ?  AVANCE 50
CMDE  ?  ROTAT 90
CMDE  ?  ENCORE
CMDE  ?  AVANCE 50
CMDE  ?  ROTAT 45
CMDE  ?  AVANCE 20
CMDE  ?  ROTAT 45
CMDE  ?  AVANCE 50
CMDE  ?  ROTAT 90
CMDE  ?  AVANCE 50
CMDE  ?  ROTAT 45
CMDE  ?  AVANCE 20
CMDE  ?  ROTAT 135
CMDE  ?  AVANCE 50
CMDE  ?  ROTAT 45
CMDE  ?  AVANCE 20

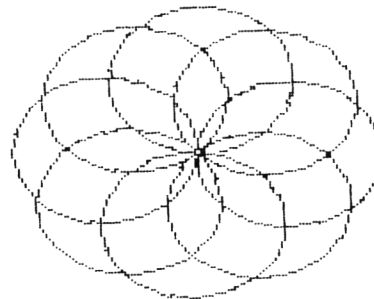
```



```

CMDE  ?  DEFI CERCLE
CD  FCT  ?  REPETE 45
CD  FCT  ?  AVANCE 8
CD  FCT  ?  ROTAT 8
CD  FCT  ?  ENCORE
CD  FCT  ?  FIN
CMDE  ?  REPETE 8
CMDE  ?  CERCLE
CMDE  ?  ROTAT 45
CMDE  ?  ENCORE

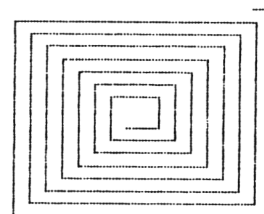
```



```

CMDE  ?  REPETE 30
CMDE  ?  AVANCE
CMDE  ?  ROTAT 90
CMDE  ?  DIS+ 5
CMDE  ?  ENCORE

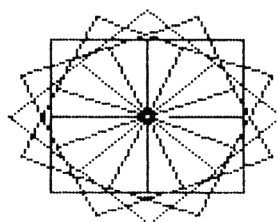
```



```

CMDE  ?  DEFI CARRE
CD FCT ?  REPETE 4
CD FCT ?  AVANCE 60
CD FCT ?  ROTAT 90
CD FCT ?  ENCORE
CD FCT ?  FINF
CMDE  ?  ?
CMDE  ?  REPETE 16
CMDE  ?  CARRE
CMDE  ?  ROTAT 22.5
CMDE  ?  ENCORE

```

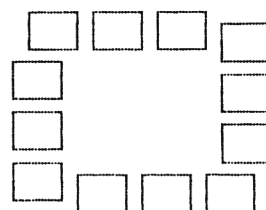


```

CMDE  ?  DEFI CARRE
CD FCT ?  REPETE 4
CD FCT ?  AVANCE 30
CD FCT ?  ROTAT 90
CD FCT ?  ENCORE
CD FCT ?  FINF
CMDE  ?  ?
CMDE  ?  REPETE 4
CMDE  ?  REPETE 3
CMDE  ?  CARRE
CMDE  ?  LEVE
CMDE  ?  AVANCE 40
CMDE  ?  BAISSSE
CMDE  ?  ENCORE
CMDE  ?  LEVE
CMDE  ?  ROTATION 90
CMDE  ?  AVANCE 40
CMDE  ?  BAISSSE
CMDE  ?  ENCORE

```

0

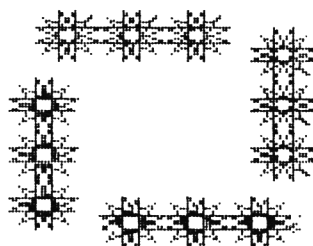


```

CMDE  ?  DEFI ETOILE
CD FCT ?  REPETE 12
CD FCT ?  AVANCE 40
CD FCT ?  ROTAT 150
CD FCT ?  ENCORE
CD FCT ?  FINF
CMDE  ?  ?
CMDE  ?  REPETE 4
CMDE  ?  REPETE 3
CMDE  ?  ETOILE
CMDE  ?  LEVE
CMDE  ?  AVANCE 40
CMDE  ?  BAISSSE
CMDE  ?  ENCORE
CMDE  ?  LEVE
CMDE  ?  ROTAT 90
CMDE  ?  AVANCE 40
CMDE  ?  BAISSSE
CMDE  ?  ENCORE

```

0



```

10 ' SIMULATEUR LOGO
20 '
30 MODE 1
40 INK 0,26:INK 1,0:INK 2,2:INK 3,6
50 DIM CD$(100):XC=450:YC=150:CX=1:CY=1:CY=0:DIST=20
60 DIM F$(5,20) ' table fonctions
70 XX=XC:YY=YC ' memoire centre
80 CE=1 ' ecriture

```





```

90 CF=0 ' fond
100 '
110 WINDOW #0,1,21,1,23
120 WINDOW #1,35,39,1,3
130 PRINT "EX: AVANCE 40
140 PRINT "      ROTAT 90"
150 PRINT
160 PRINT "FINI pour stopper":PRINT
170 '----
180 L=1 ' pointeur ligne de commande
190 '
200 DS=1:GOSUB 1080 ' appel curseur apparent
210 INPUT "CMDE ";X$
220 X$=UPPER$(X$)
230 DS=0:GOSUB 1080 ' effacement curseur
240 IF X$="" THEN 200
250 '
260 IF LEFT$(X$,4)="DEFI" THEN GOSUB 930:GOTO 200
270 FOR K=1 TO 5
280 IF X$=F$(K,0) THEN GOSUB 1020:GOTO 380
290 NEXT K
300 '
310 IF LEFT$(X$,4)="REPE" THEN B=B+1
320 '
330 IF LEFT$(X$,4)="ENCO" THEN B=B-1
340 '
350 CD$(L)=X$ ' stockage commande dans CD$()
360 L=L+1
370 '
380 IF B>0 THEN 200
390 '
400 NC=L-1 ' nombre de commandes
410 PL=1
420 '
430 LG$=CD$(PL)
440 IF PL>NC THEN 180
450 C$=LEFT$(LG$,4)
460 IF C$="REPE" THEN GOSUB 670:GOTO 430
470 IF C$="ENCO" THEN GOSUB 710:GOTO 430
480 IF C$="AVAN" THEN GOSUB 810
490 IF C$="ROTA" THEN GOSUB 740
500 IF C$="DIST" THEN GOSUB 630:IF V<>0 THEN DIST=V
510 IF C$="DIS+" THEN GOSUB 630:IF V<>0 THEN DIST=DIST+V
520 IF C$="LEVE" THEN LV=1:LOCATE #1,1,1:PRINT #1,C$
530 IF C$="BAIS" THEN LV=0:LOCATE #1,1,1:PRINT #1,C$
540 IF C$="COUL" THEN GOSUB 630:IF V>0 THEN CE=V
550 IF C$="EFA" THEN CLG
560 IF C$="COLO" THEN GOSUB 630:IF V<>0 THEN XC=V
570 IF C$="LIGN" THEN GOSUB 630:IF V<>0 THEN YC=V
580 IF C$="CENT" THEN XC=XX:YC=YY
590 IF C$="FINI" THEN END
600 PL=PL+1:IF PL>NC THEN 180
610 GOTO 430

```

```

620 '----- recherche espace
630 P=INSTR(LG$,CHR$(32))
640 IF P=0 THEN V=0:RETURN
650 V=VAL(RIGHT$(LG$,LEN(LG$)-P)):RETURN
660 '----- repete
670 GOSUB 630
680 PR=PR+1:RP(PR)=PL+1:NB(PR)=V
690 PL=PL+1:RETURN
700 '----- ENCORE
710 NB(PR)=NB(PR)-1:IF NB(PR)>0 THEN PL=RP(PR):RETURN
720 PR=PR-1:PL=PL+1:RETURN
730 '----- ROTATION ANGLE
740 GOSUB 630
750 ANG=ANG+V
760 IF ANG>=360 THEN ANG=ANG-360
770 AR=ANG/360*PI*2
780 CX=COS(AR):CY=SIN(AR)
790 RETURN
800 '----- TRACE DROITE
810 GOSUB 630
820 IF P=0 THEN 840
830 DIST=V
840 DX=DIST*CX:DY=DIST*CY
850 IF XC+DX<1 OR XC+DX>640 THEN RETURN
860 IF YC+DY<1 OR YC+DY>399 THEN RETURN
870 IF LV=1 THEN 900
880 '
890 PLOT XC,YC,CE:DRAWR DX,DY,CE
900 XC=XC+DX:YC=YC+DY
910 RETURN
920 '----- STOCKAGE FONCTION DANS F$(,)
930 NF=NF+1
940 F$(NF,0)=RIGHT$(X$,LEN(X$)-5)
950 FOR P=1 TO 20
960   INPUT "CD FCT ";F$
970   F$(NF,P)=UPPER$(F$)
980   IF F$(NF,P)="FINF" THEN RETURN
990 NEXT P
1000 STOP
1010 '----- INSERTION FONCTION
1020 P=1
1030 '
1040 IF F$(K,P)="FINF" THEN RETURN
1050 CD$(L)=F$(K,P)
1060 P=P+1:L=L+1:GOTO 1040
1070 '----- CURSEUR
1080 DX=CX*6:DY=CY*6
1090 IF DS=1 THEN CL=CE:X=XC+DX:Y=YC+DY:IF TEST(X,Y)<>0 THEN CL=0
1100 IF DS=0 THEN CL=CF:X=XC+DX:Y=YC+DY:IF TEST(X,Y)=0 THEN CL=CE
1110 PLOT XC,YC,CL:PLOT XC+DX,YC+DY,CL:DRAWR DX,DY,CL
1120 RETURN
1130 '-----
1140   ex: AVANCE 50
1150       ROTAT 90
1160       AVANCE 60

```

```

1170 '
1180 '      DEFI CARRE
1190 '      REPETE 4
1200 '      AVANCE 60
1210 '      ROTAT 90
1220 '      ENCORE
1230 '      FINF
1240 '
1250 '      CARRE

```

## MUSIQUE

---

Sur AMSTRAD la définition des notes se fait en donnant la période associée, ce qui est moins pratique que de définir la note et le niveau d'octave.

Ci-dessous, nous indiquons en DATA la correspondance note-période pour le niveau d'octave -1. Pour les autres niveaux d'octave, nous calculons la période nécessaire à l'instruction "SOUND canal,période".

```

10 '----- MUSIQUE
20 DIM NT$(12),PER(12)
30 '--- VALEURS PERIODES POUR OCTAVE -1
40 DATA 956,DO
50 DATA 902,DO#
60 DATA 851,RE
70 DATA 804,Mib
80 DATA 758,MI
90 DATA 716,FA
100 DATA 676,FA#
110 DATA 638,SOL
120 DATA 602,SOL#
130 DATA 568,LA
140 DATA 536,Sib
150 DATA 506,SI
160 '
170 FOR J=1 TO 12:READ PER(J),NT$(J):NEXT J
180 '----- MORCEAU A JOUER
190 DATA LA,2,20,SOL,2,20,LA,2,50
200 DATA SOL,2,20,FA,2,20,MI,2,20
210 DATA RE,2,20,DO#,2,50,RE,2,90
220 DATA LA,1,20,SOL,1,20,LA,1,50
230 DATA MI,1,35,FA,1,35,DO#,1,35
240 DATA RE,1,80,LA,0,20,SOL,0,20
250 DATA LA,0,50,SOL,0,20,FA,0,20
260 DATA MI,0,20,RE,0,20,DO#,0,20
270 DATA RE,0,90,RE,2,20,DO,2,20
280 DATA RE,2,20,Sib,1,20,RE,2,20
290 DATA LA,1,20,RE,2,20,SOL,1,20
300 DATA RE,2,20,FA#,1,20,RE,2,20
310 DATA SOL,1,20,RE,2,20,LA,1,20
320 DATA RE,2,20,Sib,1,20,RE,2,20
330 DATA RE,0,20,RE,2,20,MI,0,20
340 DATA RE,2,20,FA#,0,20,RE,2,20
350 DATA FIN,0,0

```



```

360 '-----
370 READ NT$,OCT,DUR
380 IF NT$="FIN" THEN END
390 PRINT NT$,OCT,DUR
400 FOR N=1 TO 12
410 IF NT$=NT$(N) THEN SOUND 1,PER(N)/(2^(OCT+1)),DUR
420 NEXT N
430 GOTO 370
440 '----- ^=fleche haut

```

Le programme ci-dessous décode un morceau défini sous forme de périodes pour donner la note et l'octave.

```

10 '----- DECODAGE PERIODE --> NOTE ET OCTAVE
20 DIM NT$(12),PER(12)
30 '----- VALEURS POUR OCTAVE -1
40 DATA 956,DO
50 DATA 902,DO#
60 DATA 851,RE
70 DATA 804,Mib
80 DATA 758,MI
90 DATA 716,FA
100 DATA 676,FA#
110 DATA 638,SOL
120 DATA 602,SOL#
130 DATA 568,LA
140 DATA 536,Sib
150 DATA 506,SI
160 '
170 FOR N=1 TO 12:READ PER(N),NT$(N):NEXT N
180 '-MORCEAU A DECODER(PERIODE+DUREE) --> NOTE+OCTAVE+DUREE
190 DATA 71,20,80,20,71,50,80,20,89,20,95,20,106,20,113,55,106,9
0,142,20,159,20,142,50,190,35,179,35,213,60,213,80,284,20,319,20
,284,50,319,20,358,20,379,20,426,20,426,50,426,90
200 DATA 999,999
210 '
220 READ PER,DUR:IF PER=999 THEN END
230 SOUND 1,PER,DUR
240 FOR N=1 TO 12
250 FOR OCT=-1 TO 4
260 R=PER(N)/(2^(OCT+1)) ' ^=fleche haut
270 IF PER>R-1 AND PER<R+1 THEN PRINT PER,NT$(N),OCT,DUR
280 NEXT OCT
290 NEXT N
300 GOTO 220

```

|     |     |   |    |
|-----|-----|---|----|
| RUN |     |   |    |
| 71  | LA  | 2 | 20 |
| 80  | SOL | 2 | 20 |
| 71  | LA  | 2 | 50 |
| 80  | SOL | 2 | 20 |
| 89  | FA  | 2 | 20 |
| 95  | MI  | 2 | 20 |

Nous proposons un programme permettant d'enregistrer dans un fichier un morceau à partir du clavier.

L'opérateur frappe la note et l'octave qui sont enregistrées dans un fichier.

La durée de base prévue pour une note est de 20 unités. En appuyant trois fois de suite sur la même touche, l'opérateur définit une note de  $20 \times 3 = 60$  unités.

Naturellement, pendant l'enregistrement au clavier les notes ne sont pas audibles. C'est un programme de lecture du fichier qui permet d'écouter le morceau enregistré.

```

10 '----- ENREGISTREMENT D'UN MORCEAU DANS UN FICHIER
20 MODE 2:PAPER 0:PEN 1:INK 0,26:INK 1,0
30 '----- octave -1
40 DIM NT$(12),PER(12),CL$(12)
50 DATA DO,956,D,DO#,900,E,RE,851,R,Mib,804,N
60 DATA MI,758,M,FA,716,F,FA#,676,G,SOL,638,S
70 DATA SOL#,602,D,LA,568,L,Sib,536,A,SI,506,Z
80 '
90 FOR N=1 TO 12
100 READ NT$(N),PER(N),CL$(N)
110 NEXT N
120 CLS
130 '
140 FOR N=1 TO 12
150 LOCATE N*5,10:PRINT NT$(N)
160 LOCATE N*5,11:PRINT CL$(N)
170 NEXT N
180 LOCATE 1,20:PRINT "OCTAVE: 0,1,2 "
190 LOCATE 1,21:PRINT "Q: FIN"
200 OCT=2 ' OCTAVE
210 LOCATE 1,4:INPUT "NOM FICHIER ";NF$
220 OPENOUT NF$
230 '-----
240 C$=INKEY$:IF LEN(C$)=0 THEN 240
250 C$=UPPER$(C$)
260 IF C$>="0" AND C$<="2"
    THEN OCT=VAL(C$):LOCATE 1,1:PRINT "OCTAVE:";OCT
270 IF C$="Q" THEN CLOSEOUT:END
280 FOR N=1 TO 12
290 IF C$=CL$(N) THEN 330
300 NEXT N
310 GOTO 240
320 '
330 PRINT #9,NT$(N)
340 PRINT #9,OCT
350 PRINT #9,20
360 LOCATE 20,1:PRINT NT$(N);SPC(2)
370 SOUND 1,PER(N)/(2^(OCT+1)),20 ' ^=fleche haut
380 GOTO 240
    →

```

OCTAVE: 2                      SI

OCTAVE: 0,1,2  
Q: FIN

NOM FICHER ? M2

|    |     |    |     |    |    |     |     |      |    |     |    |
|----|-----|----|-----|----|----|-----|-----|------|----|-----|----|
| DO | DO# | RE | Mib | MI | FA | FA# | SOL | SOL# | LA | Sib | SI |
| D  | E   | R  | N   | M  | F  | G   | S   | D    | L  | Ü   | Z  |

```

10 '- JOUE UN MORCEAU ENREGISTRE DANS UN FICHER(PAR 'EMUS')
20 MODE 2
30 DIM NT$(12),PER(12)
40 '--- VALEURS PERIODES POUR OCTAVE -1
50 DATA 956,DO
60 DATA 902,DO#
70 DATA 851,RE
80 DATA 804,Mib
90 DATA 758,MI
100 DATA 716,FA
110 DATA 676,FA#
120 DATA 638,SOL
130 DATA 602,SOL#
140 DATA 568,LA
150 DATA 536,Sib
160 DATA 506,SI
170 '
180 FOR N=1 TO 12:READ PER(N),NT$(N):NEXT N
190 '-----
200 INPUT "NOM FICHER ";NF$
210 OPENIN NF$
220 '
230 IF EOF THEN CLOSEIN:END
240 INPUT #9, NT$,OCT,DUR
250 PRINT NT$,OCT,DUR
260 FOR N=1 TO 12
270 IF NT$=NT$(N) THEN SOUND 1,PER(N)/(2^(OCT+1)),DUR
280 NEXT N
290 GOTO 230

```

Ce programme vous permet d'apprendre l'alphabet MORSE. Par exemple, en appuyant sur la lettre S, vous entendez trois sons brefs. En appuyant sur O, vous entendrez trois sons longs.

```

10 '----- MORSE
20 '
30 MODE 2:PAPER 0:PEN 1
40 DIM MORSE$(26)
50 DATA .-,-...,-.-.
60 DATA -...,,...-
70 DATA --.,...,,...-,-.-,-.-,-.-,-
80 DATA -.,-.-,-.-,-.-,-.-,-.-,-
90 DATA ...-,-.-,-.-,-.-,-.-.
100 '
110 FOR I=1 TO 26:READ MORSE$(I):NEXT I
120 '
130 C$=INKEY$:IF LEN(C$)=0 THEN 130
140 C$=UPPER$(C$)
150 P=ASC(C$)-64:IF P<1 OR P>26 THEN 130
160 '
170 X$=MORSE$(P)
180 PRINT X$;SPC(2);
190 FOR P=1 TO LEN(X$)
200   IF MID$(X$,P,1)="."
      THEN SOUND 1,350,8:FOR TP=1 TO 150:NEXT TP
210   IF MID$(X$,P,1)="-"
      THEN SOUND 1,350,40:FOR TP=1 TO 400:NEXT TP
220 NEXT P
230 FOR TP=1 TO 300:NEXT TP
240 GOTO 130

```

|   | MORSE\$( ) |
|---|------------|
| A | • —        |
| B | — • • •    |
| C | — • — •    |
|   |            |
| Z | — — • •    |





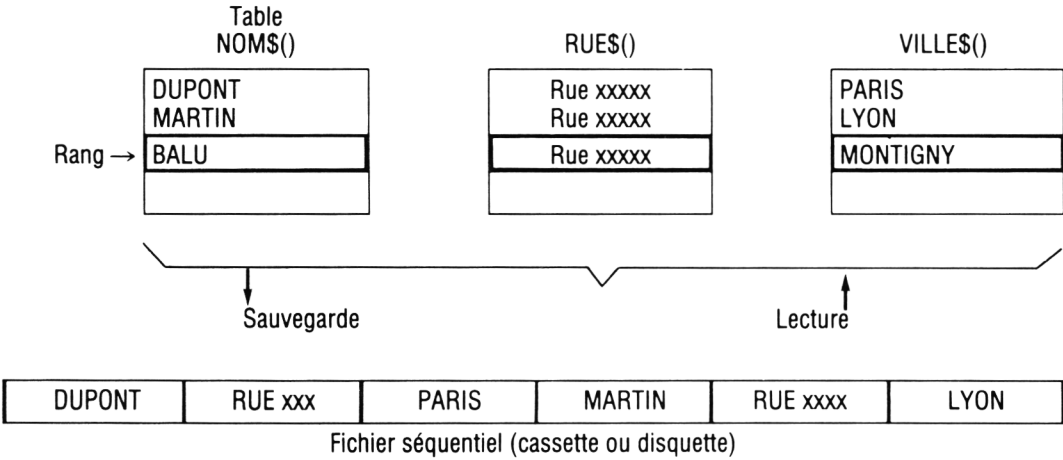
# FICHIERS | 2

## RAPPELS SUR LES FICHIERS

Pour l'instant les fichiers sur AMSTRAD sont seulement du type séquentiel.

Une exploitation d'un fichier séquentiel, enregistrement par enregistrement, est très inconfortable, la gestion des ajouts et suppressions nécessite l'utilisation de fichiers intermédiaires. Aussi est-on amené à gérer les informations dans des tables en mémoire centrale et à les sauvegarder périodiquement.

Naturellement, la taille des fichiers séquentiels se trouve ainsi limitée à celle de la mémoire centrale. Par exemple, si chaque fiche occupe 100 caractères le fichier ne peut comporter que 300 fiches.



## FICHER D'ADRESSES

---

Le programme de fichier d'adresses est organisé avec des sous-programmes :  
CREATION,LISTE,SAUVEGARDE,LECTURE.

```
MENU: INPUT "Mode ";M$
```

```
IF M$="C" THEN GOSUB CREATION
```

```
IF M$="L" THEN GOSUB LISTE
```

```
GOTO MENU
```

CREATION

RETURN

LISTE

RETURN

Un menu de type numérique se présenterait ainsi :

```
MENU: INPUT "Mode 1,2,3,..";M
```

```
ON M GOSUB GOSUB no ligne 1,no ligne 2,...
```

```
GOTO MENU
```

Nous n'utilisons pas ce type de menu ; les codes numériques sont peu mnémoniques.  
En outre, la lisibilité d'une instruction ON GOSUB comportant plus de 10 numéros de ligne nous semble douteuse.

Pour convertir des codes alphabétiques en codes numériques, on pourra faire :

```
MENU: INPUT "Mode (C,L,M....) ";M$
```

```
M=INSTR("XCLM",M$
```

```
IF M<2 THEN GOTO MENU
```

```
ON M-1 GOSUB ligne1,ligne 2,ligne 3,..
```

```
GOTO MENU
```

```
10 ' FICHER D'ADRESSES SIMPLE
```

```
20 '
```

```
30 MODE 2:INK 0,0:INK 1,26:PAPER 0:PEN 1
```

```
40 DIM NOM$(300),RUE$(300),VILLE$(300),CPST$(300)
```

```
50 DIM CLE$(200),INDEX(200)
```

```
60 '
```

```
70 NFICH=0 ' nombre de fiches
```

```
80 INPUT "Nouveau fichier (O/N) ";R$
```

```
90 IF R$<>"O" AND R$<>"o" THEN GOSUB 550
```

```
100 '===== MENU
```

```
110 CLS:PRINT "Modes:":PRINT
```

```
120 PRINT TAB(3) "C: creation"
```

```
130 PRINT TAB(3) "LF: liste du fichier"
```

```
140 PRINT TAB(3) "S: suppression"
```

```
150 PRINT TAB(3) "M: modification"
```

```
160 PRINT TAB(3) "LT: liste triee"
```

```
170 PRINT TAB(3) "ETIQ: etiquettes"
```

```
180 PRINT TAB(3) "FIN: fin de session(sauvegarde)"
```

```
190 '
```

→

```

200 PRINT:INPUT "Mode ";M$
210 M$=UPPER$(M$)
220 IF M$="C" THEN GOSUB 310
230 IF M$="FIN" THEN GOSUB 440:END
240 IF M$="LF" THEN GOSUB 670
250 IF M$="M" THEN GOSUB 760
260 IF M$="S" THEN GOSUB 980
270 IF M$="LT" THEN GOSUB 1170
280 IF M$="ETIQ" THEN GOSUB 1460      ' programme suivant
290 GOTO 110
300 '===== CREATION
310 PRINT
320 INPUT "Nom (ENTER pour fin) ";NOM$
330 IF LEN(NOM$)=0 THEN RETURN      ' fin de mode?
340 NFICH=NFICH+1
350 RANG=NFICH                      ' adresse de rangement
360 NOM$(RANG)=NOM$
370 '----- entree des zones
380 PRINT
390 LINE INPUT "Rue? ",RUE$: RUE$(RANG)=RUE$
400 LINE INPUT "Ville? ",VILLE$:VILLE$(RANG)=VILLE$
410 LINE INPUT "Code postal? ",CPST$:CPST$(RANG)=CPST$
420 GOTO 310
430 '===== SAUVEGARDE TABLES
440 OPENOUT "ADR"
450 PRINT #9,NFICH
460 FOR F=1 TO NFICH
470   PRINT #9,NOM$(F)
480   PRINT #9,RUE$(F)
490   PRINT #9,VILLE$(F)
500   PRINT #9,CPST$(F)
510 NEXT F
520 CLOSEOUT
530 RETURN
540 '===== LECTURE FICHIER DANS TABLES
550 OPENIN "ADR"
560 INPUT #9,NFICH
570 FOR F=1 TO NFICH
580   LINE INPUT #9,NOM$(F)
590   LINE INPUT #9,RUE$(F)
600   LINE INPUT #9,VILLE$(F)
610   LINE INPUT #9,CPST$(F)
620 NEXT F
630 CLOSEIN
640 PRINT:PRINT NFICH;"Fiches"
650 FOR TP=1 TO 2000:NEXT TP
660 RETURN
670 '===== liste du fichier
680 PRINT:PRINT "Liste du fichier":PRINT
690 FOR F=1 TO NFICH
700   IF F MOD 20=0 THEN INPUT "APPUYER SUR ENTER";X$
710   PRINT NOM$(F) TAB(13) RUE$(F) TAB(40) VILLE$(F)
720 NEXT F
730 PRINT:INPUT "Appuyer sur ENTER ";X$
740 RETURN

```

Lors d'une création de fiche nous ne vérifions pas si la fiche existe déjà comme il conviendrait de le faire.

```
332 FOR RANG=1 TO NFICH
333 IF NOM$=NOM$(RANG) THEN PRINT "Existe deja":GOTO 310
334 NEXT RANG
```

Pour procéder à des modifications de fiches, il faut ajouter le sous-programme suivant et prévoir son appel au menu.

Nous affichons les anciennes valeurs des zones et si l'opérateur ne désire pas les modifier, il appuie sur ENTER. On s'aperçoit que les modes CREATION et MODIFICATION pourraient être fusionnés. C'est ce que nous ferons dans les programmes suivants.

Si un nom a été écrit en majuscules lors de la création il doit être écrit aussi en majuscules lors de la recherche. En changeant l'instruction 800, la recherche peut s'effectuer en majuscules ou minuscules :

```
800 IF UPPER$(NOM$)=UPPER$(NOM$(RANG)) THEN GOTO 840
```

On pourra également "normaliser" les noms en majuscules avant de les placer dans le fichier :

```
360 NOM$(RANG)=UPPER$(NOM$)
```

Mode ? M

Quel nom(ENTER pour fin) ? DUPONT

ENTER pour zones inchangées

|              |                             |
|--------------|-----------------------------|
| 11,RUE NOBEL | Nouvelle rue ? 13,RUE NOBEL |
| MONTIGNY     | Nouvelle ville ?            |
| 78180        | Nouveau code postal         |

---

```
750 '===== MODIFICATION
760 PRINT:INPUT "Quel nom(ENTER pour fin) ";NOM$
770 IF LEN(NOM$)=0 THEN RETURN ' fin de mode?
780 '
790 FOR RANG=1 TO NFICH ' recherche du nom
800 IF NOM$=NOM$(RANG) THEN GOTO 840
810 NEXT RANG
820 PRINT:PRINT "N'existe pas":GOTO 760
830 '
840 PRINT:PRINT "ENTER pour zones inchangées":PRINT
850 PRINT RUE$(RANG) TAB(20) ' affichage ancienne valeur
860 LINE INPUT "Nouvelle rue? ",RUE$
870 IF LEN(RUE$)<>0 THEN RUE$(RANG)=RUE$
880 '
→
```

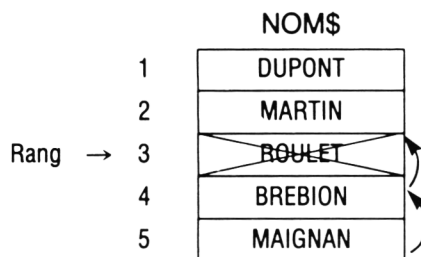
```

890 PRINT VILLE$(RANG) TAB(20) ' affichage ancienne valeur
900 LINE INPUT "Nouvelle ville? ",VILLE$
910 IF LEN(VILLE$)<>0 THEN VILLE$(RANG)=VILLE$
920 '
930 PRINT CPST$(RANG) TAB(20)
940 LINE INPUT "Nouveau code postal? ",CPST$
950 IF LEN(CPST$)<>0 THEN CPST$(RANG)=CPST$
960 GOTO 760

```

## SUPPRESSION D'UNE FICHE

Pour supprimer une fiche nous décalons toutes les fiches en aval de la fiche supprimée.



```

970 '===== SUPPRESSION
980 PRINT:INPUT "NOM ";NOM$:IF LEN(NOM$)=0 THEN RETURN
990 '
1000 FOR RANG=1 TO NFICH
1010 IF NOM$(RANG)=NOM$ THEN 1050
1020 NEXT RANG
1030 PRINT:PRINT "N'EXISTE PAS":PRINT:GOTO 980
1040 '
1050 PRINT
1060 INPUT "SUPPRESSION OK (O/N) ";R$:IF R$<>"O" THEN 980
1070 FOR J=RANG TO NFICH-1
1080 NOM$(J)=NOM$(J+1)
1090 RUE$(J)=RUE$(J+1)
1100 VILLE$(J)=VILLE$(J+1)
1110 CPST$(J)=CPST$(J+1)
1120 NEXT J
1130 NOM$(NFICH)="":RUE$(NFICH)="":VILLE$(NFICH)="":CPST$(NFICH)
    =" "
1140 NFICH=NFICH-1
1150 GOTO 980

```

## TRI

Pour éditer une liste triée des fiches dans l'ordre des noms ou des villes nous ne trions pas les tables contenant les informations ; nous utilisons des tables intermédiaires CLE\$() et INDEX().

Dans la table CLE\$(), nous plaçons les clés à trier, et dans la table INDEX() les adresses de rangement des informations dans les tables NOM\$(), RUE\$() et VILLE\$().

Après avoir trié les tables CLE\$() et INDEX(), nous éditons la liste triée. Avant le tri, les tables se présentent ainsi :

|   | CLE\$()  | INDEX() |   | VILLE\$()  |  | RUE\$() |  | NOM\$() |
|---|----------|---------|---|------------|--|---------|--|---------|
| 1 | MONTIGNY | 1       | → | 1 MONTIGNY |  | RUE xxx |  | DUPONT  |
| 2 | LYON     | 2       | → | 2 LYON     |  | RUE xxx |  | BALU    |
| 3 | PARIS    | 3       | → | 3 PARIS    |  | RUE xxx |  | MARTIN  |

Après le tri, les tables se présentent ainsi :

|   | CLE\$()  | INDEX() |   | VILLE\$()  |  | RUE\$() |  | NOM\$() |
|---|----------|---------|---|------------|--|---------|--|---------|
| 1 | LYON     | 2       | → | 1 MONTIGNY |  |         |  | DUPONT  |
| 2 | MONTIGNY | 1       | → | 2 LYON     |  |         |  | BALU    |
| 3 | PARIS    | 3       | → | 3 PARIS    |  |         |  | MARTIN  |

En lisant séquentiellement la table INDEX() et en se servant des numéros qu'elle contient, on obtient la liste triée. Pour obtenir une liste triée par noms, placer les noms dans la table CLE\$() (au lieu des villes).

```

1160 '===== SELECTION/TRI PAR VILLE
1170 INPUT "Quelle ville (ENTER pour toutes) ";V$
1180 NC=0 ' nombre de cles
1190 FOR F=1 TO NFICH
1200 IF LEN(V$)<>0 THEN IF V$<>VILLE$(F) THEN 1220
1210 NC=NC+1:CLE$(NC)=VILLE$(F):INDEX(NC)=F
1220 NEXT F
1230 '
1240 GOSUB 1330 ' appel tri
1250 '----- edition
1260 FOR F=1 TO NC
1270 X=INDEX(F)
1280 PRINT VILLE$(X) TAB(15) NOM$(X)
1290 NEXT F
1300 PRINT:INPUT "APPUYER SUR ENTER";X$
1310 RETURN
1320 '----- tri shell
1330 ECART=NC
1340 ECART=INT(ECART/2):IF ECART<1 THEN RETURN
1350 IV=0
1360 FOR K=1 TO NC-ECART
1370 J=K+ECART
1380 IF CLE$(J)>=CLE$(K) THEN 1410
1390 X$=CLE$(K):CLE$(K)=CLE$(J):CLE$(J)=X$:IV=1
1400 X=INDEX(K):INDEX(K)=INDEX(J):INDEX(J)=X
1410 NEXT K
1420 IF IV=1 THEN 1350
1430 GOTO 1340

```

## ÉTIQUETTES

---

Nous utilisons le fichier créé par le programme "fichier d'adresses".

Le nombre d'étiquettes par ligne est défini par la variable "NE" ; l'espacement vertical entre deux lignes d'étiquettes par la variable "IVERT". Les étiquettes sont éditées dans l'ordre des villes avec éventuellement une sélection par ville. Pour une édition dans l'ordre alphabétique, il suffit de placer dans la table CLE\$( ) les noms :

CLE\$(NC)=NOM\$(F) au lieu de CLE\$(NC)=VILLE\$(F)

Quelle ville (ENTER pour toutes) ?

LEBOSSE

DUPONT

RUE XXX  
14110 CONDE

11,RUE NOBEL  
78180 MONTIGNY

BREBION

MARTIN

RUE XXX  
78180 MONTIGNY

14,RUE DE MILAN  
75010 PARIS

ROULET

12,RUE XXXX  
75009 PARIS

```

1440 '===== ETIQUETTES
1445 ' A ajouter au programme 'fichier d'adresses'
1450 '
1460 INPUT "Quelle ville (ENTER pour toutes) ";V$
1470 INPUT "Ecran ou imprimante (E/I) ";R$
1480 C=0:IF R$="I" THEN C=8      ' canal
1490 NC=0                        ' nombre de cles
1500 FOR F=1 TO NFICH
1510 IF LEN(V$)<>0 THEN IF V$<>VILLE$(F) THEN 1530
1520 NC=NC+1:CLE$(NC)=VILLE$(F):INDEX(NC)=F
1530 NEXT F
1540 '
1550 GOSUB 1600                  ' appel tri
1560 '
1570 GOSUB 1720
1580 RETURN
1590 '----- tri shell
1600 ECART=NC
1610 ECART=INT(ECART/2):IF ECART<1 THEN RETURN
1620 IV=0
1630 FOR K=1 TO NC-ECART
1640 J=K+ECART
1650 IF CLE$(J)>=CLE$(K) THEN 1680
1660 X$=CLE$(K):CLE$(K)=CLE$(J):CLE$(J)=X$:IV=1

```



```

1670 X=INDEX(K):INDEX(K)=INDEX(J):INDEX(J)=X
1680 NEXT K
1690 IF IV=1 THEN 1620
1700 GOTO 1610
1710 '-----
1720 NE=2 ' nombre etiquettes par ligne
1730 IHRIZ=25 ' intervalle horizontal
1740 IVERT=6 ' intervalle vertical
1750 NLP=71 ' nombre de lignes par page
1760 MARGE=2
1770 '
1780 NRANG=INT(NLP)/IVERT
1790 SP=NLP-NRANG*IVERT
1800 '
1810 TRANG=0:PRINT
1820 M=1
1830 '
1840 IF M>NC THEN 1910
1850 GOSUB 1940
1860 M=M+NE
1870 TRANG=TRANG+1
1880 IF TRANG=NRANG THEN FOR K=1 TO SP:PRINT #C:NEXT K:TRANG=0
1890 GOTO 1840
1900 '
1910 PRINT:INPUT "APPUYER SUR ENTER";X$
1920 RETURN
1930 '-----
1940 FOR K=0 TO NE-1
1950 PRINT #C, TAB(MARGE+K*IHRIZ) NOM$(INDEX(M+K));
1960 NEXT K
1970 PRINT #C:PRINT #C
1980 FOR K=0 TO NE-1
1990 PRINT #C, TAB(MARGE+3+K*IHRIZ) RUE$(INDEX(M+K));
2000 NEXT K
2010 PRINT #C
2020 FOR K=0 TO NE-1
2030 PRINT #C, TAB(MARGE+3+K*IHRIZ) CPST$(INDEX(M+K)) SPC(1) VI
LLE$(INDEX(M+K));
2040 NEXT K
2050 PRINT #C
2060 FOR K=1 TO IVERT-4:PRINT #C:NEXT K
2070 RETURN

```



## GESTION DE FICHIER ADAPTABLE

Le programme de gestion de fichier présenté est adaptable. Les noms des zones à saisir sont définis en début de programme.

Prévu pour un fichier de stocks, ce fichier peut être adapté à un fichier d'adresses, par exemple. Les informations sont placées dans une table à deux dimensions FICH\$(,). Chaque ligne contient une fiche. Les modes création et modification sont communs.

|          | CODE | LIBELLÉ | PRIX   | STOCK | TABLE  |
|----------|------|---------|--------|-------|--------|
| 1        | R5   | XXXXX   | 50000  | 20    | FICH\$ |
| Rang → 2 | R18  | XXXX    | 60000  | 12    |        |
| 3        | R4   | XXXXX   | 400000 | 30    |        |
| 4        | R35  |         | 700000 | 4     |        |
| 5        |      |         |        |       |        |

Mode ? C

CODE (ENTER pour fin) ? PAPIER

Nouvelle cle OK (O/N) ? O

R:zone arriere

Appuyer sur ENTER pour zone inchangée en mode MODIF

```
LIBELLE?PAPIER XXXXXX
PRIX   ?2
STOCK  ?300
```

Liste du fichier

| CODE   | LIBELLE                 | PRIX | STOCK | VALEUR |
|--------|-------------------------|------|-------|--------|
| CRAYON | CRAYON XXXXXXXX         | 2    | 1000  | 2000   |
| GOMME  | GOMME XXXXXX            | 3    | 300   | 900    |
| DISQUE | DISQUE XXXXXXXXXXXXXXXX | 30   | 1000  | 30000  |
| PAPIER | PAPIER XXXXX            | 3    | 300   | 900    |

### Remarques sur le programme :

Pour éviter que l'opérateur entre des codes en minuscules on pourra "normaliser" les codes par :

```
420 CLE$=UPPER$(CLE$)
```

```

10 'FICHER STOCK (PROGRAMME ADAPTABLE A D'AUTRES FICHERS)
20 '
30 '      FICH$(,): table a 2 dimensions contenant les fiches
40 '      NFICH:    nombre de fiches
50 '      NRUB$():  noms des rubriques
60 '
70 MODE 2:PAPER 0:PEN 1:INK 0,1:INK 1,24
80 '----- A ADAPTER
90 NF$="STOCK"          ' nom du fichier
100 NRUB=4              ' nombre de rubriques
110 NRUB$(1)="CODE"     ' nom rubrique 1
120 NRUB$(2)="LIBELLE"  ' nom rubrique 2
130 NRUB$(3)="PRIX"
140 NRUB$(4)="STOCK"
150 '-----
160 DIM FICH$(200,NRUB) ' 200 fiches maxi
170 DIM CLE$(200),INDEX(200)
180 NFICH=0              ' nombre de fiches
190 '
200 INPUT "Nouveau fichier (O/N) ";R$
210 IF R$="O" OR R$="o" THEN 230
220 GOSUB 770
230 CLS:PRINT "Modes:":PRINT
240 PRINT TAB(3) "C:      creation/modification"
250 PRINT TAB(3) "LF:     liste du fichier"
260 PRINT TAB(3) "S:      suppression"
270 PRINT TAB(3) "LTC:    liste triee par code"
280 PRINT TAB(3) "FIN:    fin de session(sauvegarde)"
290 PRINT:INPUT "Mode ";M$
300 M$=UPPER$(M$)
310 IF M$="C" THEN GOSUB 380
320 IF M$="FIN" THEN GOSUB 670:END
330 IF M$="LF" THEN GOSUB 900
340 IF M$="S" THEN GOSUB 1080
350 IF M$="LTC" THEN GOSUB 1260
360 GOTO 230
370 '===== CREATION/MODIFICATION
380 PRINT
390 PRINT NRUB$(1);
400 LINE INPUT " (ENTER pour fin ) ? ",CLE$
410 IF LEN(CLE$)=0 THEN RETURN ' fin de mode?
420 '
430 LG=LEN(CLE$)
440 IF NFICH=0 THEN 490
450 FOR RANG=1 TO NFICH
460 IF CLE$=LEFT$(FICH$(RANG,1),LG) THEN 550 ' nom existe?
470 NEXT RANG
480 '----- nouvelle cle
490 PRINT:INPUT "Nouvelle cle OK (O/N) ";R$
500 IF R$<>"O" AND R$<>"o" THEN 380
510 NFICH=NFICH+1
520 RANG=NFICH
530 FICH$(RANG,1)=CLE$

```

```

540 '----- entree/modif zones
550 PRINT
560 PRINT "R:zone arriere"
570 PRINT "Appuyer sur ENTER pour zone inchangee en mode MODIF":P
RINT
580 FOR R=2 TO NRUB
590 PRINT FICH$(RANG,R) TAB(25) ' ancienne valeur
600 PRINT NRUB$(R) TAB(32) ' nom de zone
610 LINE INPUT "?",X$
620 IF X$="R" THEN IF R>1 THEN R=R-1:GOTO 590
630 IF LEN(X$)<>0 THEN FICH$(RANG,R)=X$
640 NEXT R
650 GOTO 380
660 '===== sauvegarde
670 OPENOUT NF$
680 PRINT #9,NFICH
690 FOR F=1 TO NFICH
700 FOR R=1 TO NRUB
710 PRINT #9,FICH$(F,R)
720 NEXT R
730 NEXT F
740 CLOSEOUT
750 RETURN
760 '===== lecture fichier
770 OPENIN NF$
780 '
790 INPUT #9,NFICH
800 FOR F=1 TO NFICH
810 FOR R=1 TO NRUB
820 LINE INPUT #9,FICH$(F,R)
830 NEXT R
840 NEXT F
850 CLOSEIN
860 PRINT:PRINT nfich;"FICHES"
870 FOR TP=1 TO 2000:NEXT TP
880 RETURN
890 '===== liste du fichier
900 CLS
910 INPUT "Ecran ou imprimante (E/I) ";R$
920 CN=0:IF R$="I" THEN CN=8
930 PRINT #CN,"Liste du fichier":PRINT #CN
940 '
950 PRINT #CN,NRUB$(1) TAB(10) NRUB$(2) TAB(35) NRUB$(3) TAB(45)
960 PRINT #CN,NRUB$(4) TAB(52) "VALEUR":PRINT #CN
970 FOR F=1 TO NFICH
980 IF CN=0 THEN IF F MOD 20=0 THEN INPUT "Appuyer sur ENTER";X$
990 PRINT #CN,FICH$(F,1) TAB(10) ' zone 1 (code)
1000 PRINT #CN,FICH$(F,2) TAB(35) ' zone 2 (libelle)
1010 PRINT #CN,FICH$(F,3) TAB(45) ' zone 3 (prix)
1020 PRINT #CN,FICH$(F,4) TAB(52)
1030 PRINT #CN,VAL(FICH$(F,3))*VAL(FICH$(F,4))
1040 NEXT F
1050 PRINT:IF CN=0 THEN INPUT "Appuyer sur <ENTER> ";X$
1060 RETURN

```

Pour supprimer des fiches, ajoutez les instructions suivantes :

```

1070 '===== SUPPRESSION FICHE
1080 PRINT :PRINT NRUB$(1);:INPUT CLE$
1090 IF LEN(CLE$)=0 THEN RETURN
1100 FOR RANG=1 TO NFICH
1110 IF CLE$=FICH$(RANG,1) THEN 1150
1120 NEXT RANG
1130 PRINT:PRINT "N'existe pas":GOTO 1080
1140 '
1150 PRINT:INPUT "ANNULE OK(O/N) ";R$:IF R$<>"O" THEN 1080
1160 FOR J=RANG TO NFICH-1
1170 FOR R=1 TO NRUB
1180 FICH$(J,R)=FICH$(J+1,R)
1190 NEXT R
1200 NEXT J
1210 FOR R=1 TO NRUB:FICH$(NFICH,R)="":NEXT R
1220 NFICH=NFICH-1
1230 GOTO 1080
1240 '---Attention! Il n'y a pas d'espace entre ""

```

Une liste triée s'obtient suivant le principe présenté pour le programme "fichier d'adresses".

```

1250 '===== SELECTION/TRI PAR CODE
1260 PTRI=1 ' position de tri (CODE)
1270 INPUT "Quelle cle (ENTER pour toutes) ";CLE$
1280 NC=0 ' nc: nombre de cles
1290 LG=LEN(CLE$)
1300 FOR F=1 TO NFICH
1310 IF LEN(CLE$)<>0
1320 THEN IF LEFT$(FICH$(F,PTRI),LG)<>CLE$ THEN 1330
1330 NC=NC+1:CLE$(NC)=FICH$(F,PTRI):INDEX(NC)=F
1340 GOSUB 1440 ' appel tri
1350 PRINT:PRINT "LISTE TRIEE DES FICHES ":PRINT
1360 '----- EDITION
1370 FOR F=1 TO NC
1380 X=INDEX(F)
1390 PRINT FICH$(X,1) TAB(10) FICH$(X,2) TAB(40) FICH$(X,3)
1400 NEXT F
1410 PRINT:INPUT "APPUYER SUR ENTER ";X$
1420 RETURN
1430 '----- tri SHELL-METZNER
1440 ECART=NC
1450 PRINT:PRINT "JE TRIE POUR VOUS ":PRINT
1460 ECART=INT(ECART/2):IF ECART<1 THEN RETURN
1470 J=1 :K=NC-ECART
1480 L=J
1490 M=L+ECART
1500 IF CLE$(L)<=CLE$(M) THEN 1560
1510 X$=CLE$(L):CLE$(L)=CLE$(M):CLE$(M)=X$
1520 X=INDEX(L):INDEX(L)=INDEX(M):INDEX(M)=X
1530 L=L-ECART:IF L<1 THEN 1560
1540 GOTO 1490
1550 '
1560 J=J+1:IF J>K THEN 1460
1570 GOTO 1480

```

## CONVERSION MINUSCULES – MAJUSCULES

---

Le programme ci-dessous convertit tous les caractères d'un texte en majuscules. Le fichier texte est d'abord lu dans une table LIG\$(). Si le fichier texte est un programme, celui-ci doit avoir été sauvegardé en ASCII par "SAVE "XX",A".

```

10 '  CONVERSION MINUSCULE ---> MAJUSCULE
20 '
30 MODE 2:PAPER 0:PEN 1
40 DIM LIG$(1000)
50 INPUT "Nom fichier ";NF$
60 OPENIN NF$
70 '-----LECTURE FICHIER DANS TABLE LIG$()
80 FOR L=1 TO 1000
90 IF EOF=-1 THEN CLOSEIN:NL=L-1:GOTO 150
100 LINE INPUT #9,LIG$(L)
110 PRINT LIG$(L)
120 NEXT L
130 STOP
140 '----- CONVERSION
150 FOR L=1 TO NL
160 LIG$(L)=UPPER$(LIG$(L))
170 PRINT LIG$(L)
180 NEXT L
190 '----- ECRITURE FICHIER
200 OPENOUT NF$
210 FOR L=1 TO NL
220 PRINT #9,LIG$(L)
230 NEXT L
240 CLOSEOUT
250 '-----
260 ' SUR CPC664,ON PEUT OUVRIR UN FICHIER EN LECTURE
270 ' ET UN AUTRE EN ECRITURE SIMULTANEMENT(AVEC DES NOMS
280 ' DIFFERENTS).

```

## CHANGEMENT D'UN MOT DANS UN FICHIER SÉQUENTIEL

---

Pour changer un mot dans un fichier séquentiel nous utilisons l'instruction INSTR qui nous donne la position du mot dans chaque ligne.

Ce programme peut servir à changer des noms de variables d'un programme sauvegardé par "SAVE "XX",A".

```

10 '  CHANGEMENT D'UN MOT DANS UN FICHIER SEQUENTIEL
20 '
30 MODE 2:PAPER 0:PEN 1
40 DIM TXT$(1000)
50 '
60 INPUT "NOM DU FICHIER ";NF$
70 OPENIN NF$
80 INPUT "ANCIEN MOT ";AM$

```



```

90 INPUT "NOUVEAU MOT ";NM$
100 '----- LECTURE FICHIER DANS TABLE TXT$( )
110 FOR I=1 TO 1000
120 IF EOF THEN NL=I-1:CLOSEIN:GOTO 170
130 LINE INPUT #9,TXT$(I)
140 PRINT TXT$(I)
150 NEXT I
160 '-----
170 FOR I=1 TO NL
180 DR=1 ' debut recherche dans la ligne
190 '
200 P=INSTR(DR,TXT$(I),AM$) ' recherche position ancien mot
210 IF P=0 THEN 240
220 TXT$(I)=LEFT$(TXT$(I),P-1)+NM$+RIGHT$(TXT$(I),LEN(TXT$(I))-
P-LEN(AM$)+1)
230 DR=P+LEN(NM$)+1:GOTO 200
240 NEXT I
250 '----- ECRITURE FICHIER
260 OPENOUT NF$+"B"
270 FOR I=1 TO NL
280 PRINT #9,TXT$(I)
290 PRINT TXT$(I)
300 NEXT I
310 PRINT:PRINT "NOUVEAU FICHIER: ";NF$+"B"
320 CLOSEOUT
330 '
340 'Attention! pour un programme,faire 'SAVE "XX",A'

```

```

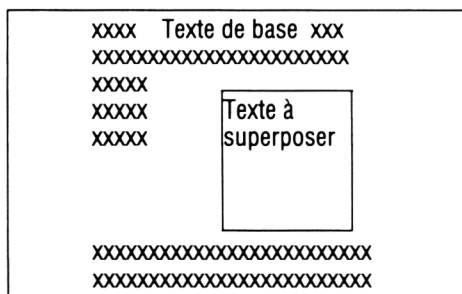
NOM DU FICHIER ? X
ANCIEN MOT ? LIGNE
NOUVEAU MOT ? LG
10 '----- COPIE ECRAN(TEXTE)
20 FOR LIGNE=1 TO 24
30 FOR COL=1 TO 80
40 LOCATE COL,LIGNE:C$=COPYCHR$(#0)
50 PRINT #8,C$;
60 NEXT COL
70 PRINT #8
80 NEXT LIGNE
10 '----- COPIE ECRAN(TEXTE)
20 FOR LG=1 TO 24
30 FOR COL=1 TO 80
40 LOCATE COL,LG:C$=COPYCHR$(#0)
50 PRINT #8,C$;
60 NEXT COL
70 PRINT #8
80 NEXT LG

```

NOUVEAU FICHIER:XB

## COMPOSITION DE TEXTES

Ce programme "superpose" un texte sur un autre. Dans l'exemple ci-dessous nous avons écrit un texte standard en laissant un emplacement pour un autre texte issu, soit d'un traitement de texte, soit de l'exécution d'un programme.



NOM DU FICHIER DE BASE? tbase

Veillez trouver ci joint les statistiques mensuelles  
de ventes des ouvrages du PSI.

====>

J.L VEROHYE

NOM DU FICHIER A SUPERPOSER ? stv

LIGNE ? 4

COLONNE ? 15

VENTES DE JANVIER 1985

BSF : 999  
 AMSTRDAD : 1001  
 APPLE : 997

Veillez trouver ci joint les statistiques mensuelles  
de ventes des ouvrages du PSI.

====>

VENTES DE JANVIER 1985

BSF : 999  
 AMSTRDAD : 1001  
 APPLE : 997

J.L VEROHYE

```

10 ' COMPOSITION DE TEXTES
20 '
30 ' Le texte de base et le texte a ajouter doivent exister.
40 '
50 MODE 2:PAPER 0:PEN 1
60 INPUT "NOM DU FICHIER DE BASE";L$
70 '
80 '----- Lecture texte de base dans table LIG$()
90 OPENIN L$
100 DIM LIG$(100)
110 FOR L=1 TO 100
120 IF EOF THEN NL=L-1:GOTO 180
130 LINE INPUT #9,LIG$(L)
140 PRINT LIG$(L)
150 LIG$(L)=LEFT$(LIG$(L)+STRING$(78,CHR$(32)),78)
160 NEXT L
170 '
180 CLOSEIN
190 '-----
200 INPUT "NOM DU FICHIER A SUPERPOSER ";TS$
210 '
220 INPUT "LIGNE ";LIGNE
230 INPUT "COLONNE ";COL
240 OPENIN TS$
250 '
260 LINE INPUT #9,LIG$
270 PRINT LIG$
280 IF EOF THEN CLOSEIN:GOTO 330
290 MID$(LIG$(LIGNE),COL,LEN(LIG$))=LIG$
300 LIGNE=LIGNE+1
310 GOTO 260
320 '----- Edition
330 FOR L=1 TO NL
340 PRINT LIG$(L)
350 NEXT L

```



## BIBLIOTHÈQUE

---

Les titres, auteurs et emplacement des ouvrages sont stockés dans des tables : TIT\$(),AUT\$(),ETAG\$(). La recherche par titre, auteur ou emplacement se fait à l'aide de la fonction INSTR.

Ce programme est adaptable à plusieurs types d'applications :

- ☐ Recherche de synonymes.
- ☐ Dictionnaire français/anglais.

|   | TIT\$()                  | AUT\$()   | ETAG\$() |
|---|--------------------------|-----------|----------|
| 1 | PROGRAMMER EN ASSEMBLEUR | PINAUD    | ETAG=A   |
| 2 | LA PRATIQUE DE L'APPLE   | POULIQUEN | ETAG=B   |
| 3 |                          |           |          |

### LISTE DU FICHIER

|                            |           |        |
|----------------------------|-----------|--------|
| 1 BASIC ET SES FICHIERS    | DUPONT    | ETAG=A |
| 2 PROGRAMMER EN ASSEMBLEUR | PINAUD    | ETAG=B |
| 3 LA PRATIQUE DE L'APPLE   | POULIQUEN | ETAG=A |
| 4 CLEFS POUR APPLE         | POULIQUEN | ETAG=B |
| 5 LA PRATIQUE DU TRS80     | PINAUD    | ETAG=B |
| 6 BASIC DE A A Z           | MARTIN    | ETAG=A |
| 7 AMSTRAD POUR TOUS        | DUPONT    | ETAG=A |

MOT CLE ? APPLE

|                        |           |        |
|------------------------|-----------|--------|
| LA PRATIQUE DE L'APPLE | POULIQUEN | ETAG=A |
| CLEFS POUR APPLE       | POULIQUEN | ETAG=B |

MOT CLE ? ETAG=A

|                        |           |        |
|------------------------|-----------|--------|
| BASIC ET SES FICHIERS  | DUPONT    | ETAG=A |
| LA PRATIQUE DE L'APPLE | POULIQUEN | ETAG=A |
| BASIC DE A A Z         | MARTIN    | ETAG=A |
| AMSTRAD POUR TOUS      | DUPONT    | ETAG=A |

MOT CLE ? POUL

|                        |           |        |
|------------------------|-----------|--------|
| LA PRATIQUE DE L'APPLE | POULIQUEN | ETAG=A |
| CLEFS POUR APPLE       | POULIQUEN | ETAG=B |

### Remarques sur le programme :

Même si l'opérateur entre le titre en minuscules celui-ci est stocké dans le fichier en majuscules (par UPPER\$). Pour les recherches, l'opérateur peut frapper les mots-clés en majuscules ou minuscules.

```

10 '          BIBLIOTHEQUE
20 '
30 MODE 2
40 INK 0,26:INK 1,0:PAPER 0:PEN 1
50 NF$="BIBE" ' nom fichier
60 DIM TIT$(200),AUT$(200),ETAG$(200)
70 DIM CLE$(200),INDEX(200)
80 NF=0 ' nombre de fiches
90 INPUT "NOUVEAU FICHIER (O/N) ";R$
100 IF R$<>"O" AND R$<>"o" THEN GOSUB 670
110 '
120 CLS
130 PRINT "MODES:":PRINT
140 PRINT "C  :CREATION"
150 PRINT "R  :RECHERCHE"
160 PRINT "L  :LISTE DU FICHIER"
170 PRINT "S  :SUPPRESSION"
180 PRINT "M  :MODIFICATION"
190 PRINT "LTA :LISTE TRIEE PAR AUTEUR"
200 PRINT "FIN :SAUVEGARDE FICHIER"
210 '
220 PRINT:INPUT "MODE ";M$
230 M$=UPPER$(M$)
240 IF M$="C" THEN GOSUB 330
250 IF M$="R" THEN GOSUB 490
260 IF M$="L" THEN GOSUB 590
270 IF M$="S" THEN GOSUB 900
280 IF M$="LTA" THEN GOSUB 1250
290 IF M$="M" THEN GOSUB 1090
300 IF M$="FIN" THEN GOSUB 800:END
310 GOTO 120
320 '===== CREATION
330 PRINT:INPUT "TITRE (ou ENTER) ";TIT$
340 IF LEN(TIT$)=0 THEN RETURN ' fin de mode?
350 TIT$=UPPER$(TIT$):L=LEN(TIT$) ' titre majuscules
360 FOR F=1 TO NF ' titre existe t-il?
370 IF TIT$=LEFT$(TIT$(F),L) THEN PRINT TIT$(F):GOTO 330
380 NEXT F
390 NF=NF+1
400 '-----
410 PRINT
420 INPUT "AUTEUR ";AUT$
430 INPUT "ETAGERE ";ETAG$
440 TIT$(F)=TIT$
450 AUT$(F)=AUT$
460 ETAG$(F)="ETAG="+ETAG$
470 GOTO 330
480 '===== RECHERCHE
490 PRINT:INPUT "Mot cle (ou ENTER) ";MC$
500 IF LEN(MC$)=0 THEN RETURN
510 PRINT
520 FOR F=1 TO NF
530 LIG$=UPPER$(TIT$(F)+AUT$(F)+ETAG$(F))
540 IF INSTR(LIG$,UPPER$(MC$))=0 THEN 560
550 PRINT TIT$(F) TAB(30) AUT$(F) TAB(45) ETAG$(F)
560 NEXT F
570 GOTO 490

```

```

580 '===== LISTE DU FICHIER
590 PRINT:PRINT "LISTE DU FICHIER":PRINT
600 FOR F=1 TO NF
610 PRINT F;TIT$(F) TAB(30) AUT$(F) TAB(45) ETAG$(F)
620 IF F MOD 20=0 THEN INPUT "APPUYER SUR 'ENTER'";X$
630 NEXT F
640 PRINT:INPUT "APPUYER SUR ENTER ";X$
650 RETURN
660 '===== LECTURE FICHIER DANS TABLES
670 OPENIN NF$
680 INPUT #9,NF
690 FOR F=1 TO NF
700 INPUT #9,TIT$(F)
710 INPUT #9,AUT$(F)
720 INPUT #9,ETAG$(F)
730 PRINT TIT$(F)
740 NEXT F
750 CLOSEIN
760 PRINT NF;"FICHES":PRINT
770 FOR TP=1 TO 1000:NEXT TP
780 RETURN
790 '===== SAUVEGARDE TABLES DANS FICHIER
800 OPENOUT NF$
810 PRINT #9,NF
820 FOR F=1 TO NF
830 PRINT #9,TIT$(F)
840 PRINT #9,AUT$(F)
850 PRINT #9,ETAG$(F)
860 NEXT F
870 CLOSEOUT
880 RETURN
890 '===== SUPPRESSION
900 PRINT:INPUT "TITRE (premieres lettres ou ENTER) ";TIT$
910 IF LEN(TIT$)=0 THEN RETURN
920 TIT$=UPPER$(TIT$):L=LEN(TIT$)
930 FOR F=1 TO NF
940 IF TIT$=LEFT$(UPPER$(TIT$(F)),L) THEN GOTO 980
950 NEXT F
960 PRINT:PRINT "N'EXISTE PAS":PRINT:GOTO 900
970 '
980 PRINT:PRINT TIT$(F);:INPUT " OK (O/N) ";R$
990 IF UPPER$(R$)<>"O" THEN 900
1000 FOR J=F TO NF-1
1010 TIT$(J)=TIT$(J+1)
1020 AUT$(J)=AUT$(J+1)
1030 ETAG$(J)=ETAG$(J+1)
1040 NEXT J
1050 TIT$(NF)="" : AUT$(NF)="" : ETAG$(NF)=""
1060 NF=NF-1
1070 GOTO 900

```

Pour modifier une fiche ajouter :

```

1080 '===== MODIFICATION
1090 PRINT:INPUT "TITRE (premieres lettres ou ENTER) ";TIT$
1100 IF LEN(TIT$)=0 THEN RETURN ' fin de mode?
1110 TIT$=UPPER$(TIT$):L=LEN(TIT$)
1120 FOR F=1 TO NF ' titre existe t-il?
1130 IF TIT$=LEFT$(UPPER$(TIT$(F)),L) THEN 1170
1140 NEXT F
1150 PRINT:PRINT "N'EXISTE PAS":PRINT:GOTO 1090
1160 '-----
1170 PRINT:PRINT TIT$(F):PRINT
1180 PRINT AUT$(F) TAB(20) ' ancienne valeur
1190 INPUT "AUTEUR ";X$:IF LEN(X$)<>0 THEN AUT$(F)=X$
1200 PRINT ETAG$(F) TAB(20)
1210 INPUT "ETAGERE ";X$:IF LEN(X$)<>0 THEN ETAG$(F)="ETAG="+X$
1220 '
1230 GOTO 1090

```

Le tri par auteur s'obtient par :

```

1240 '===== SELECTION/TRI PAR AUTEUR
1250 INPUT "Quelle AUTEUR (ENTER pour tous) ";A$
1260 NC=0 ' nombre de cles
1270 L=LEN(A$)
1280 FOR F=1 TO NF
1290 IF UPPER$(A$)<>UPPER$(LEFT$(AUT$(F),L)) THEN 1310
1300 NC=NC+1:CLE$(NC)=AUT$(F):INDEX(NC)=F
1310 NEXT F
1320 '
1330 GOSUB 1430 ' appel tri
1340 '----- edition
1350 PRINT:PRINT "LISTE TRIEE PAR AUTEUR":PRINT
1360 FOR F=1 TO NC
1370 X=INDEX(F)
1380 PRINT AUT$(X) TAB(15) TIT$(X)
1390 NEXT F
1400 PRINT:INPUT "APPUYER SUR ENTER";X$
1410 RETURN
1420 '----- tri shell
1430 ECART=NC
1440 ECART=INT(ECART/2):IF ECART<1 THEN RETURN
1450 IV=0
1460 FOR K=1 TO NC-ECART
1470 J=K+ECART
1480 IF CLE$(J)>=CLE$(K) THEN 1510
1490 X$=CLE$(K):CLE$(K)=CLE$(J):CLE$(J)=X$:IV=1
1500 X=INDEX(K):INDEX(K)=INDEX(J):INDEX(J)=X
1510 NEXT K
1520 IF IV=1 THEN 1450
1530 GOTO 1440

```

## DICTIONNAIRE DES SYNONYMES

---

Nous reprenons le principe du programme "bibliothèque" pour gérer un dictionnaire des synonymes. Il suffit de placer dans chaque enregistrement la liste des synonymes d'un mot.

MODE ? L

LISTE DU FICHIER

```
1 AVOUER RECONNAITRE APPROUVER RATIFIER
2 DELAI REPIT SURSIS PROLONGATION
3 FAIRE CONSTRUIRE FABRIQUER CONFECTIONNER
4 FAIBLESSE ABATTEMENT AFFAIBLISSEMENT DEFAILLANCE
5 MORDRE CROQUER DECHIQUETTER
6 PERSISTER S'OBSTINER PERSEVERER
7 PERPETUEL ETERNEL INCESSANT IFINI
```

MODE ? R

MOT CLE ? SURSIS

DELAJ REPIT SURSIS PROLONGATION

MOT CLE ? APPROUVER

AVOUER RECONNAITRE APPROUVER RATIFIER

```
10 ' DICTIONNAIRE DE SYNONYMES
20 '
30 MODE 2:INK 0,26:INK 1,0:PAPER 0:PEN 1
40 NF$="SYNO" ' nom fichier
50 DIM LIG$(200)
60 NF=0 ' nombre de fiches
70 INPUT "NOUVEAU FICHIER (O/N) ";R$
80 IF R$<>"O" AND R$<>"o" THEN GOSUB 470
90 '
100 CLS:PRINT "MODES:":PRINT
110 PRINT "C :CREATION"
120 PRINT "R :RECHERCHE"
130 PRINT "L :LISTE DU FICHIER"
140 PRINT "FIN :SAUVEGARDE FICHIER"
150 '
160 PRINT:INPUT "MODE ";M$
170 M$=UPPER$(M$)
180 IF M$="C" THEN GOSUB 240
190 IF M$="R" THEN GOSUB 300
200 IF M$="L" THEN GOSUB 390
210 IF M$="FIN" THEN GOSUB 570:END
220 GOTO 100
230 '===== CREATION
```



```

240 PRINT:INPUT "LISTE DES SYNONYMES (ou ENTER) ";LIG$
250 IF LEN(LIG$)=0 THEN RETURN      ' fin de mode?
260 NF=NF+1
270 LIG$(NF)=LIG$      ' ou LIG$(NF)=UPPER$(LIG$)
280 GOTO 240
290 '===== RECHERCHE
300 PRINT:INPUT "MOT CLE (ou ENTER) ";MC$
310 IF LEN(MC$)=0 THEN RETURN
320 PRINT
330 FOR F=1 TO NF
340   IF INSTR(UPPER$(LIG$(F)),UPPER$(MC$))=0 THEN 360
350   PRINT LIG$(F)
360 NEXT F
370 GOTO 300
380 '===== LISTE DU FICHIER
390 PRINT:PRINT "LISTE DU FICHIER":PRINT
400 FOR F=1 TO NF
410   PRINT F;LIG$(F)
420   IF F MOD 20=0 THEN INPUT "APPUYER SUR ENTER";X$
430 NEXT F
440 PRINT:INPUT "APPUYER SUR ENTER ";X$
450 RETURN
460 '===== LECTURE FICHIER DANS TABLE LIG$()
470 OPENIN NF$
480 INPUT #9,NF
490 FOR F=1 TO NF
500   LINE INPUT #9,LIG$(F):PRINT LIG$(F)
510 NEXT F
520 CLOSEIN
530 PRINT NF;"FICHES":PRINT
540 FOR TP=1 TO 1000:NEXT TP
550 RETURN
560 '===== SAUVEGARDE FICHIER
570 OPENOUT NF$
580 PRINT #9,NF
590 FOR F=1 TO NF:PRINT #9,LIG$(F):NEXT F
600 CLOSEOUT
610 RETURN

```

## GESTION DE REPAS

---

Avec le programme précédent vous pouvez stocker les menus des repas servis à vos invités. Ainsi, pour une prochaine invitation, vous pourrez connaître tous les menus des repas déjà servis à vos invités.

MODE ? L

LISTE DU FICHIER

```
1 10 JUILLET 1985 /MENU=CAROTTES STEAK SALADE /INVITE=DUPONT
2 11 JUILLET 1985 /MENU=QUICHE STEAK YADURT /INVITE=MARTIN
3 12 JUILLET 1985 /MENU=THON COTE DE PORC RIZ AU LAIT/INVITE=DUPONT
4 14 JUILLET 1985 /MENU=CREPES CREPES CREPES /INVITE=VEROHYE
5 16 JUILLET 1985 /MENU=CAROTTES GRATIN DAUPHINOIS RIZ AU LAIT
/INVITE=DUPONT
```

MODE ? R

MOT CLE ? INVITE=DUPONT

```
10 JUILLET 1985 /MENU=CAROTTES STEAK SALADE /INVITE=DUPONT
12 JUILLET 1985 /MENU=THON COTE DE PORC RIZ AU LAIT/INVITE=DUPONT
16 JUILLET 1985 /MENU=CAROTTES GRATIN DAUPHINOIS RIZ AU LAIT
/INVITE=DUPONT
MOT CLE ? RIZ AU LAIT
```

```
12 JUILLET 1985 /MENU=THON COTE DE PORC RIZ AU LAIT/INVITE=DUPONT
16 JUILLET 1985 /MENU=CAROTTES GRATIN DAUPHINOIS RIZ AU LAIT
/INVITE=DUPONT
```

## GESTION DE CHÈQUES

---

Des chèques sont stockés dans des tables. Nous éditons la liste des chèques pour la période choisie ainsi que le solde. Les dates doivent être entrées sous la forme "année/mois/jour" afin de permettre la sélection entre deux dates. Nous avons également prévu une ventilation par catégorie de dépenses.

|        | NCHEQS()             | LIB\$()                       | CT\$()               | DT\$()                           | MT()                     |
|--------|----------------------|-------------------------------|----------------------|----------------------------------|--------------------------|
| Rang → | 1111<br>2222<br>3333 | PSI xxx<br>xxxxxxx<br>xxxxxxx | PSI<br>LOYER<br>ALIM | 85/07/01<br>85/07/02<br>85/07/04 | 100000<br>-2500<br>-1500 |

Mode ? C

Numero cheque (ENTER pour fin) ? 9999

```
0
    Date (AA/MM/JJ) ? 85/07/21
    Libelle ? REMISE DE CHEQUES
    Montant ? 1300
    Categorie ? PSI
```

Date debut (AA/MM/JJ) (ou ENTER) ?

Date fin (AA/MM/JJ) (ou ENTER) ?

| No   | cheque   | Libelle |                | Credit    | Debit    |
|------|----------|---------|----------------|-----------|----------|
| 1111 |          | PSI     | PSI 85/07/01   | 100000.00 |          |
| 2222 | XXXXXXX  |         | LOYER 85/07/02 |           | -1000.00 |
| 3333 | XXXXXX   |         | ALIM 85/07/08  |           | -2000.00 |
| 4444 | XXXXX    |         | LOIS 85/07/10  |           | -1500.00 |
| 5555 | XXXXXX   |         | PSI 85/07/12   | 50000.00  |          |
| 7777 | XXXXXXXX |         | DIV 85/07/16   |           | -350.00  |
| 8888 | XXXXXXX  |         | LOIS 85/07/20  |           | -1100.00 |
| 9999 | XXXXXX   |         | ALIM 85/07/22  |           | -500.00  |
| CCCC | XXXXXXX  |         | PSI 85/08/15   | 100000.00 |          |
|      |          |         |                | 250000.00 | -6450.00 |

ANCIEN SOLDE: 0

SOLDE: 243550

Mode ? LC

|       |        |
|-------|--------|
| PSI   | 150000 |
| LOYER | -1000  |
| ALIM  | -3200  |
| LOIS  | -2600  |
| DIV   | -350   |



```

10 '      GESTION DE CHEQUES
20 '
30 MODE 2:INK 0,0:INK 1,26:PAPER 0:PEN 1
40 NCH=0 ' nombre de cheques
50 DIM NCHEQ$(200),DT$(200),LIB$(200),MT(200),CT$(200)
60 DIM C$(20),MC(20)
70 '
80 INPUT "Nouveau fichier (O/N) ";R$
90 IF R$<>"O" AND R$<>"o" THEN GOSUB 900
100 '===== MENU
110 CLS:PRINT "MODES:":PRINT
120 PRINT TAB(3) "C :CREATION/MODIFICATION CHEQUE"
130 PRINT TAB(3) "L :LISTE DES CHEQUES"
140 PRINT TAB(3) "S :SUPPRESSION CHEQUE"
150 PRINT TAB(3) "LC :MONTANT PAR CATEGORIE"
160 PRINT TAB(3) "FIN:SAUVEGARDE FICHIER"
170 '
180 PRINT:INPUT "Mode ";M$
190 M$=UPPER$(M$)
200 IF M$="C" THEN GOSUB 270
210 IF M$="L" THEN GOSUB 510
220 IF M$="S" THEN GOSUB 1000
230 IF M$="LC" THEN GOSUB 1180
240 IF M$="FIN" THEN GOSUB 780:END
250 GOTO 110
260 '===== CREATION/MODIFICATION
270 PRINT:INPUT "Numero cheque (ENTER pour fin) ";X$
280 IF LEN(X$)=0 THEN RETURN ' fin de mode?
290 FOR RANG=1 TO NCH ' cheque existe t-il?
300 IF UPPER$(X$)=UPPER$(NCHEQ$(RANG)) THEN 360
310 NEXT RANG
320 NCH=NCH+1
330 RANG=NCH
340 NCHEQ$(NCH)=X$
350 '---
360 PRINT
370 PRINT DT$(RANG) TAB(20) ' ancienne valeur
380 INPUT "Date (AA/MM/JJ) ";X$
390 IF LEN(X$)<>0 THEN DT$(RANG)=X$
400 PRINT LIB$(RANG) TAB(20)
410 INPUT "Libelle ";X$
420 IF LEN(X$)<>0 THEN LIB$(RANG)=X$
430 PRINT MT(RANG) TAB(20)
440 INPUT "Montant ";X
450 IF X<>0 THEN MT(RANG)=X
460 PRINT CT$(RANG) TAB(20)
470 INPUT "Categorie ";X$
480 IF LEN(X$)<>0 THEN CT$(RANG)=X$
490 GOTO 270

```

```

500 '===== LISTE DES CHEQUES
510 PRINT
520 INPUT "Date debut (AA/MM/JJ)(ou ENTER) ";DB$
530 INPUT "Date fin (AA/MM/JJ)(ou ENTER) ";DF$
540 ASOLDE=0:DEBIT=0:CREDIT=0
550 PRINT
560 PRINT "No cheque" TAB(11) "Libelle" TAB(52) "Credit" TAB(64)
    "Debit"
570 PRINT
580 FOR F=1 TO NCH
590 IF LEN(DB$)<>0 AND LEN(DF$)<>0 AND DT$(F)<DB$
    THEN ASOLDE=ASOLDE+MT(F)
600 IF LEN(DB$)<>0 AND LEN(DF$)<>0 AND
    (DT$(F)<DB$ OR DT$(F)>DF$) THEN 690
610 PRINT NCHEQ$(F) TAB(11) LIB$(F) TAB(34) CT$(F) TAB(40)
620 PRINT DT$(F) TAB(50)
630 IF MT(F)>0 THEN PRINT TAB(50) USING "#####.##";MT(F)
640 IF MT(F)<0 THEN PRINT TAB(62) USING "#####.##";MT(F)
650 IF F MOD 15=0 THEN INPUT "Appuyer sur ENTER";X$
660 '
670 IF MT(F)>0 THEN CREDIT=CREDIT+MT(F)
680 IF MT(F)<0 THEN DEBIT=DEBIT+MT(F)
690 NEXT F
700 PRINT
710 PRINT TAB(50) USING "#####.## ";CREDIT;
720 PRINT TAB(62) USING "#####.## ";DEBIT
730 PRINT:PRINT "ANCIEN SOLDE:";ASOLDE
740 PRINT:PRINT "SOLDE:";CREDIT+DEBIT+ASOLDE
750 PRINT:INPUT "Appuyer sur ENTER" ;X$
760 RETURN
770 '===== SAUVEGARDE
780 OPENOUT "CHEQ"
790 PRINT #9,NCH
800 FOR F=1 TO NCH
810 PRINT #9,NCHEQ$(F)
820 PRINT #9,DT$(F)
830 PRINT #9,LIB$(F)
840 PRINT #9,MT(F)
850 PRINT #9,CT$(F)
860 NEXT F
870 CLOSEOUT
880 RETURN
890 '===== LECTURE FICHIER
900 OPENIN "CHEQ"
910 INPUT #9,NCH
920 FOR F=1 TO NCH
930 INPUT #9,NCHEQ$(F),DT$(F),LIB$(F),MT(F),CT$(F)
940 NEXT F
950 CLOSEIN
960 PRINT:PRINT NCH;"CHEQUES":PRINT
970 FOR TP=1 TO 2000:NEXT TP
980 RETURN

```

```

990 '===== SUPPRESSION
1000 INPUT "Numero de cheque(ENTER pour fin) ";X$
1010 IF LEN(X$)=0 THEN RETURN
1020 FOR F=1 TO NCH ' recherche cheque
1030 IF UPPER$(X$)=UPPER$(NCHEQ$(F)) THEN 1070
1040 NEXT F
1050 PRINT:PRINT "N'existe pas ":PRINT:GOTO 1000
1060 '
1070 FOR J=F TO NCH-1
1080 NCHEQ$(J)=NCHEQ$(J+1)
1090 LIB$(J)=LIB$(J+1)
1100 DT$(J)=DT$(J+1)
1110 MT(J)=MT(J+1)
1120 CT$(J)=CT$(J+1)
1130 NEXT J
1140 NCHEQ$(NCH)="":LIB$(NCH)="":DT$(NCH)="":MT(NCH)=0
1150 NCH=NCH-1
1160 GOTO 1000
1170 '===== VENTILATION PAR CATEGORIES
1180 PRINT
1190 NC=0 ' nombre categories
1200 FOR I=1 TO 20:C$(J)="":MC(J)=0:NEXT I
1210 FOR J=1 TO NCH
1220 FOR K=1 TO NC
1230 IF CT$(J)=C$(K) THEN 1290 ' categorie existe?
1240 NEXT K
1250 NC=NC+1
1260 C$(NC)=CT$(J)
1270 K=NC
1280 '
1290 MC(K)=MC(K)+MT(J) ' cumul
1300 NEXT J
1310 '
1320 FOR J=1 TO NC
1330 PRINT C$(J) TAB(10) MC(J)
1340 NEXT J
1350 PRINT:INPUT "APPUYER SUR ENTER ";X$
1360 RETURN

```



```

10 ' EDLB                      EDITION LETTRE
20 '
30 OPENIN "LETTRE"
40 '----- Lecture lettre standard
50 DIM TXT$(100)
60 FOR L=1 TO 100
70 IF EOF THEN CLOSEIN:NL=L:GOTO 120
80 LINE INPUT #9,TXT$(L)
90 PRINT TXT$(L)
100 NEXT L
110 '----- Introduction parametres reels
120 PRINT
130 FOR P=1 TO 10
140 PRINT P;
150 LINE INPUT "Parametre (ex:NOM=BALU /* POUR FIN ) ";X$
160 IF X$="*" THEN 210
170 EG=INSTR(X$,"="):IF EG=0 THEN 140
180 PRM$(P)=LEFT$(X$,EG-1):T$(P)=RIGHT$(X$,LEN(X$)-EG)
190 NEXT P
200 '----- Edition lettre
210 PRINT
220 FOR LG=1 TO NL
230 LIG$=TXT$(LG)
240 P1=INSTR(LIG$,"<")
250 IF P1=0 THEN 340
260 P2=INSTR(LIG$,">")
270 M$=MID$(LIG$,P1+1,P2-P1-1)
280 FOR NP=1 TO 10
290 IF UPPER$(M$)=UPPER$(PRM$(NP)) THEN 320
300 NEXT NP
310 PRINT "PARAMETRE NON TROUVE":STOP
320 LIG$=LEFT$(LIG$,P1-1)+T$(NP)+RIGHT$(LIG$,LEN(LIG$)-P2)
330
340 PRINT LIG$
350 NEXT LG

1 Parametre (ex:NOM=BALU /* POUR FIN ) ? NOM=BALU
2 Parametre (ex:NOM=BALU /* POUR FIN ) ? RUE=13,RUE DE MILAN
3 Parametre (ex:NOM=BALU /* POUR FIN ) ? VILLE=PARIS
4 Parametre (ex:NOM=BALU /* POUR FIN ) ? *

```

LIBERT PIERRE  
RUE DE PARIS  
53000 SOUCE

LE 21.7.1985

BALU

13,RUE DE MILAN  
PARIS

Cher BALU,veuillez trouver ci joint .....  
.....

P.LIBERT

## ÉDITEUR DE TEXTE

---

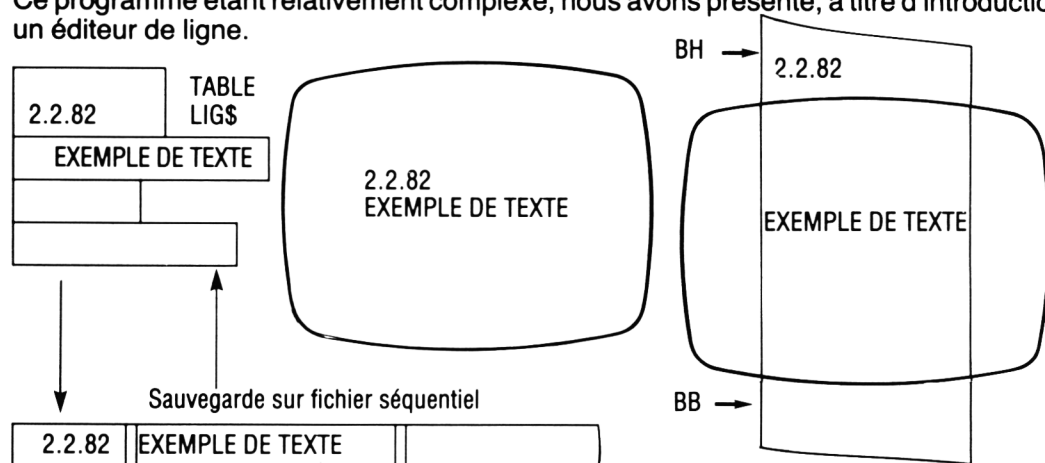
Cet éditeur de texte saisit en texte en mode "plein écran". L'image de l'écran représente à chaque instant l'image du texte.

- ☐ Le curseur se déplace avec les quatre flèches.
- ☐ L'insertion des caractères est automatique. Il n'y a pas de mode insertion.
- ☐ La touche "DEL" supprime le caractère à gauche du curseur.
- ☐ CTRL/B permet d'insérer une ligne.
- ☐ CTRL/C supprime une ligne.
- ☐ CTRL/A sauvegarde.
- ☐ CTRL/D liste sur imprimante.
- ☐ CTRL/E et CTRL/F déplacent le curseur de dix lignes.

Le défilement vertical du texte est automatique.

Pour justifier le texte à droite voir le programme "justification à droite".

Ce programme étant relativement complexe, nous avons présenté, à titre d'introduction, un éditeur de ligne.



Voici quelques exemples d'utilisation d'un éditeur de texte pleine-page :

- ☐ Courrier.
- ☐ Emploi du temps.
- ☐ Dessin.

J. DUPONT  
11, RUE NOBEL  
78180 MONTIGNY LE BX

LE 1.7.1985

Cher Monsieur

Veuillez trouver ci joint .....

## EMPLOI DU TEMPS

|       | LUNDI    | MARDI    | MERCREDI | JEUDI | VENREDI |
|-------|----------|----------|----------|-------|---------|
| 8-9   | MATHS    |          |          |       |         |
| 9-10  | FRANCAIS | MATHS    |          |       |         |
| 10-11 |          |          |          |       |         |
| 11-12 | PHYSIQUE | HISTOIRE |          |       |         |
|       |          |          |          |       |         |
| 14-15 | GEO      | PHYSIQUE |          |       |         |
| 15-16 | GYM      |          |          |       |         |

[illegible]

Ce premier programme saisit une ligne. L'insertion de caractères est automatique. La suppression se fait avec la touche "DEL".

```

10 '   EDITEUR DE LIGNE
20 '
30 MODE 2:PAPER 0:PEN 1:INK 0,26:INK 1,0
40 COL=5:LIG=10          ' ligne et colonne
50 GOSUB 80
60 END
70 '----- SAISIE D'UNE LIGNE
80 PC=1                  ' position curseur colonne
90 '
100 LOCATE COL+PC-1,LIG:PRINT CHR$(143) ' curseur
110 '
120 C$=INKEY$
130 IF LEN(C$)=0
    THEN LOCATE COL+PC-1,LIG:PRINT MID$(LIG$,PC,1):GOTO 100
140 '
150 C=ASC(C$)
160 L=LEN(LIG$)
170 IF C=13 OR C=241
    THEN LOCATE COL+PC-1,LIG:PRINT MID$(LIG$,PC,1):RETURN
175 '----- suppression
180 IF C<>127 THEN 220
190 IF PC<=1 THEN 120
200 LIG$=LEFT$(LIG$,PC-2)+RIGHT$(LIG$,L-PC+1):PC=PC-1
205 LOCATE COL,LIG:PRINT LIG$ SPC(2):GOTO 100
210 '----- curseur gauche
220 IF C=242 THEN IF PC>L THEN LOCATE COL+PC-1,LIG:PRINT SPC(1)
230 IF C=242 THEN IF PC>1 THEN LOCATE COL+PC-1,
    LIG:PRINT MID$(LIG$,PC,1):PC=PC-1:GOTO 100 ELSE 100
240 '----- curseur droite
250 IF C=243 THEN IF PC<=L THEN LOCATE COL+PC-1,
    LIG:PRINT MID$(LIG$,PC,1):PC=PC+1:GOTO 100 ELSE 100
260 '
270 IF C=240 THEN RETURN
280 '----- caractere normal ajoute
290 IF L>78 THEN 120
300 LIG$=LEFT$(LIG$,PC-1)+C$+RIGHT$(LIG$,L-PC+1)
310 LOCATE COL,LIG:PRINT LIG$
320 '
330 PC=PC+1
340 GOTO 100

```



```

10 '   EDITEUR  DE TEXTE PLEIN ECRAN
20 '
30 MODE 2:PAPER 0:PEN 1:INK 0,1:INK 1,24
40 WINDOW #0,1,80,1,21          ' fenetre texte
50 WINDOW #1,1,80,22,25         ' fenetre commandes
60 PRINT #1,"Insertion caracteres automatique / Fleches pour depl
acer DEL pour sup"
70 PRINT #1,"CTRL A:sauvegarde  CTRL B:ajout ligne  CTRL C:suppre
ssion ligne"
80 PRINT #1,"CTRL D:edition imprimante  CTRL E et F: saut 10 lign
es CTRL Z:fin"
90 C$="":C=0:LIG$="":L=0:PC=0
100 DIM LIG$(200)
110 '
120 PRINT
130 INPUT "NOM TEXTE(ENTER pour nouveau) ";NF$
140 IF LEN(NF$)=0 THEN CLS#0:GOTO 210
150 OPENIN NF$
160 FOR L=1 TO 200
170   IF EOF=-1 THEN MX=L-1:GOTO 210
180   LINE INPUT #9,LIG$(L)
190 NEXT L
200 '
210 BH=1:HP=19:BB=BH+HP          ' bornes haut et bas/hauteur page
220 PC=1                          ' pointeur curseur colonne
230 LIG=BH
240 GOSUB 440
250 '-----
260 IF LIG>MX THEN MX=LIG
270 LIG$=LIG$(LIG):GOSUB 530
280 LOCATE 1,LIG-BH+1:PRINT LIG$:SPC(1)
290 LIG$(LIG)=LIG$
300 IF R=2 THEN  IF LIG>1 THEN LIG=LIG-1:GOTO 390 ELSE 390
310 IF R=3 THEN LIG=LIG+1:GOTO 390
320 IF R=4 THEN GOSUB 470
330 IF R=5 THEN GOSUB 500
340 IF R=6 THEN GOSUB 900
350 IF R=7 THEN GOSUB 980
360 IF R=8 THEN IF LIG>10
      THEN LIG=LIG-10:GOTO 390 ELSE LIG=1:GOTO 390
370 IF R=9 THEN LIG=LIG+10:GOTO 390 ELSE 390
380 '---
390 IF LIG<=BB AND LIG>=BH THEN 260
400 '
410 IF LIG>BB THEN BB=LIG:BH=BB-HP:GOSUB 440:GOTO 260
420 IF LIG<BH THEN IF BH>1
      THEN BH=LIG:BB=BH+HP:GOSUB 440:GOTO 260 ELSE 260
430 '---- Affichage page ecran
440 CLS #0
450 FOR L=BH TO BB:LOCATE 1,L-BH+1:PRINT LIG$(L):NEXT L:RETURN
460 '--- Ajout ligne
470 FOR L=MX TO LIG STEP-1:LIG$(L+1)=LIG$(L):NEXT L
480 LIG$(LIG)="" :GOSUB 440:MX=MX+1:RETURN
490 '---- Suppression ligne
500 FOR L=LIG TO MX:LIG$(L)=LIG$(L+1):NEXT L:LIG$(MX)=""

```



```

510 MX=MX-1:GOSUB 440:RETURN
520 '----- SAISIE D'UNE LIGNE
530 IF PC>LEN(LIG$) THEN PC=LEN(LIG$)+1
540 IF LEN(LIG$)=0 THEN PC=1
550 '
560 LOCATE PC,LIG-BH+1:PRINT CHR$(143) ' curseur
570 '
580 C$=INKEY$
590 IF LEN(C$)=0
    THEN LOCATE PC,LIG-BH+1:PRINT MID$(LIG$,PC,1):GOTO 560
600 '
610 C=ASC(C$)
620 L=LEN(LIG$)
630 IF C=240 THEN R=2:RETURN ' fleche haut
640 IF C=13 OR C=241 THEN R=3:RETURN ' fleche bas
650 IF C<>127 THEN 710
660 '---- suppression caractere
670 IF PC<=1 THEN 580
680 LIG$=LEFT$(LIG$,PC-2)+RIGHT$(LIG$,L-PC+1):PC=PC-1
690 LOCATE 1,LIG-BH+1:PRINT LIG$ SPC(2):GOTO 560
700 '----- curseur gauche
710 IF C=242 THEN IF PC>L THEN LOCATE PC,LIG-BH+1:PRINT SPC(1)
720 IF C=242 THEN IF PC>1 THEN LOCATE PC,
    LIG-BH+1:PRINT MID$(LIG$,PC,1):PC=PC-1:GOTO 560 ELSE 560
730 '----- curseur droite
740 IF C=243 THEN IF PC<=L THEN LOCATE PC,
    LIG-BH+1:PRINT MID$(LIG$,PC,1):PC=PC+1:GOTO 560 ELSE 560
750 '
760 IF C=1 THEN R=6:RETURN ' sauvegarde
770 IF C=2 THEN R=4:RETURN ' insertion ligne
780 IF C=3 THEN R=5:RETURN ' suppression ligne
790 IF C=4 THEN R=7:RETURN ' affichage imprimante
800 IF C=5 THEN R=8:RETURN
810 IF C=6 THEN R=9:RETURN
820 IF C=26 THEN END
825 IF C<32 THEN 560
830 '----- caractere normal ajoute
840 IF L>78 THEN 580
850 LIG$=LEFT$(LIG$,PC-1)+C$+RIGHT$(LIG$,L-PC+1)
860 LOCATE 1,LIG-BH+1:PRINT LIG$
870 PC=PC+1
880 GOTO 560
890 '===== SAUVEGARDE
900 LOCATE #1,1,4:INPUT #1,"NOM ";NF$
910 OPENOUT NF$
920 FOR L=1 TO MX
930 PRINT #9,LIG$(L)
940 NEXT L
950 CLOSEOUT
960 RETURN
970 '===== EDITION IMPRIMANTE
980 FOR L=1 TO MX
990 PRINT #8,LIG$(L)
1000 NEXT L
1010 RETURN
1020 '--- Les instructions: LIG$="" ne comportent pas d'espace en
    tre les ""

```

## JUSTIFICATION DE TEXTE

Le premier programme ci-dessous justifie une ligne à droite en ajoutant des espaces entre les mots. Le second programme justifie un paragraphe. Le paragraphe est d'abord mis sous forme de lignes de 65 caractères au maximum. Le découpage se fait sur des espaces ou des virgules. Le programme insère ensuite des espaces pour justifier à droite.

```

10 ' JUSTIFICATION A DROITE D'UNE LIGNE
20 '
30 MODE 2
40 LL=30 ' longueur maxi
50 LIG$="LE PETIT CHAT RONRONNE"
60 '
70 S=1
80 '
90 IF LEN(LIG$)>=LL THEN END
100 FOR K=S TO LEN(LIG$) ' recherche debut espace
110 X$=MID$(LIG$,K,1)
120 IF X$=CHR$(32) THEN GOTO 170
130 NEXT K
140 IF LEN(LIG$)<LL THEN 70
150 END
160 '
170 FOR V=K+1 TO LEN(LIG$) ' recherche fin espace
180 X$=MID$(LIG$,V,1)
190 IF X$<>CHR$(32) THEN 220
200 NEXT V
210 '
220 LIG$=LEFT$(LIG$,V-1)+SPACE$(1)+RIGHT$(LIG$,LEN(LIG$)-V+1)
230 PRINT LIG$
240 S=V+1
250 GOTO 90

```

[illegible]

```

10 '          JUSTIFICATION A DROITE D'UN PARAGRAPHE
20 '
30 MODE 2:PAPER 0:PEN 1:INK 0,0:INK 1,26
40 DIM TXT$(1000)
50 LIG=60 ' Longueur decoupage ligne
60 '
70 INPUT "NOM DU FICHIER ";NF$
80 OPENIN NF$
90 '===== NORMALISATION DES LIGNES
100 ' Les lignes sont lues dans une memoire tampon
110 ' puis decoupees en lignes de 65 caracteres.
120 '
130 NL=0 ' nombre de lignes dans TXT$()
140 '
150 IF EOF THEN CLOSEIN:GOSUB 230:NL=NL+1:TXT$(NL)=TAMP$:GOTO 400
160 '
170 IF LEN(TAMP$)>150 THEN GOSUB 230
180 LINE INPUT #9,LIG$
190 PRINT LIG$
200 TAMP$=TAMP$+LIG$+CHR$(32)
210 GOTO 150
220 '----- Decoupage memoire tampon dans TXT$()
230 IF LEN(TAMP$)<LLIG THEN RETURN
240 D=1:AP=0 ' debut -ancienne position
250 '
260 P1=INSTR(D,TAMP$,CHR$(32)) ' recherche espace
270 P2=INSTR(D,TAMP$,"") ' recherche virgule
280 IF P1=0 AND P2=0 THEN 340
290 P=MIN(P1,P2)
300 IF P1=0 THEN P=P2
310 IF P2=0 THEN P=P1
320 IF P<LLIG THEN AP=P:D=P+1:GOTO 260
330 '
340 X$=LEFT$(TAMP$,AP)
350 IF ASC(RIGHT$(X$,1))=32 THEN X$=LEFT$(X$,LEN(X$)-1)
360 NL=NL+1:TXT$(NL)=X$ ' transfert TXT$()
370 TAMP$=RIGHT$(TAMP$,LEN(TAMP$)-AP) ' reste
380 GOTO 230
390 '----- EDITION TEXTE NORMALISE
400 PRINT "TEXTE NORMALISE":PRINT
410 FOR I=1 TO NL
420 PRINT TXT$(I)
430 NEXT I
440 '===== JUSTIFICATION A DROITE
450 PRINT:PRINT "TEXTE JUSTIFIE A DROITE":PRINT
460 '
470 FOR L=1 TO NL
480 LIG$=TXT$(L)
490 IF LEN(LIG$)<LLIG*0.75 THEN 700
500 R=0 ' temoin espace
510 S=1 ' debut recherche
520 IF LEN(LIG$)>=LLIG THEN 700
530 FOR K=S TO LEN(LIG$) ' recherche espace
540 X$=MID$(LIG$,K,1)
550 IF X$=CHR$(32) THEN R=1:GOTO 610

```

```

560 NEXT K
570 IF R=0 THEN 700
580 IF LEN(LIG$)<LLIG THEN GOTO 500
590 GOTO 700
600 '
610 FOR V=K+1 TO LEN(LIG$)           ' recherche fin espace
620   X$=MID$(LIG$,V,1)
630   IF X$<>CHR$(32) THEN 660
640 NEXT V
650 '
660 LIG$=LEFT$(LIG$,V-1)+SPACE$(1)+RIGHT$(LIG$,LEN(LIG$)-V+1)
670 S=V+1
680 GOTO 520
690
700 PRINT LIG$
710 NEXT L

```

### Exemple de texte à justifier :

L'auteur du "MIGRATEUR" n'a évidemment jamais habité quelque part. Il a pourtant fini par poser ses valises dans l'île d'Houat, en Bretagne. Dans la solitude, il écrit l'histoire de sa jeunesse, "sa chute hors du lycée, dans le hasard". Lire pages 18 à 30.

#### □ Texte normalisé

L'auteur du "MIGRATEUR" n'a évidemment jamais habité quelque part. Il a pourtant fini par poser ses valises dans l'île d'Houat, en Bretagne. Dans la solitude, il écrit l'histoire de sa jeunesse, "sa chute hors du lycée, dans le hasard". Lire pages 18 à 30.

#### □ Texte justifié à droite

L'auteur du "MIGRATEUR" n'a évidemment jamais habité quelque part. Il a pourtant fini par poser ses valises dans l'île d'Houat, en Bretagne. Dans la solitude, il écrit l'histoire de sa jeunesse, "sa chute hors du lycée, dans le hasard". Lire pages 18 à 30.

## MAILING

---

Ce programme permet d'envoyer des lettres personnalisées. Il utilise un fichier d'adresses déjà constitué par le programme "fichier d'adresses". La lettre, stockée dans un fichier séquentiel, doit déjà exister (cf. éditeur de texte).

NOM DU FICHIER LETTRE ? LMAIL  
NOM DU FICHIER ADRESSE ? ADR

Nous vous prions de bien vouloir trouver ci joint  
un exemplaire de notre nouveau catalogue 1986.

Nous sommes a votre disposition pour tout renseignement  
complementaire.

J.MARTIN

QUELLE VILLE (ENTER POUR TOUTES)?  
6 FICHES

DUPONT

11,RUE NOBEL  
78180 MONTIGNY

Nous vous prions de bien vouloir trouver ci joint  
un exemplaire de notre nouveau catalogue 1986.

Nous sommes a votre disposition pour tout renseignement  
complementaire.

J.MARTIN

```

10 ' MAILING
20 '
30 ' Les fichiers lettre et adresses doivent deja exister
40 ' Cf programme 'fichier d'adresses' et editeur de texte.
50 '
60 MODE 2:PAPER 0:PEN 1
70 INPUT "NOM DU FICHIER LETTRE ";L$
80 INPUT "NOM DU FICHIER ADRESSE ";NF$
90 '----- Lecture lettre dans table LIG$()
100 OPENIN L$
110 DIM LIG$(100)
120 FOR L=1 TO 100
130 IF EOF THEN NL=L-1:GOTO 180
140 LINE INPUT #9,LIG$(L)
150 PRINT LIG$(L)
160 NEXT L
170 '
180 CLOSEIN
190 '-----
200 INPUT "QUELLE VILLE (ENTER POUR TOUTES)";V$
210 '
220 OPENIN NF$
230 INPUT #9,NFICH:PRINT NFICH;"FICHES":PRINT
240 '
250 IF EOF THEN CLOSEIN:END
260 LINE INPUT #9,NOM$
270 LINE INPUT #9,RUE$
280 LINE INPUT #9,VILLE$
290 LINE INPUT #9,CPST$
300 IF LEN(V$)<>0 THEN IF VILLE$<>V$ THEN 250
310 '----- Edition lettre
320 PRINT TAB(30) NOM$
330 PRINT
340 PRINT TAB(35) RUE$
350 PRINT TAB(35) CPST$ SPC(1) VILLE$
360 PRINT
370 PRINT
380 FOR L=1 TO NL
390 PRINT LIG$(L)
400 NEXT L
410 PRINT:INPUT "APPUYER SUR ENTER ";X$
420 GOTO 250

```

## GESTION DE CLUBS

Des tables contiennent les noms des membres de clubs sportifs ainsi que le nombre de points obtenus par chacun des membres. On désire obtenir les listes suivantes :

- ☐ Classement individuel.
- ☐ Total par club.
- ☐ Liste triée par noms.
- ☐ Liste triée par clubs.

|        | NOM\$()                                                  | CLUB\$()                                                       | POINT()                          |
|--------|----------------------------------------------------------|----------------------------------------------------------------|----------------------------------|
| Rang → | DUPONT<br>LEBOSSE<br>BALU<br>BREBION<br>MARTIN<br>ROULET | BOULOGNE<br>BOULOGNE<br>MONTIGNY<br>MONTIGNY<br>PARIS<br>PARIS | 12<br>16<br>14<br>13<br>14<br>15 |

### LISTE TRIÉE PAR NOMS

La liste triée par noms est obtenue à l'aide d'une table CLE\$() et une table INDEX() (cf. "tri d'un fichier d'adresses").

### TOTAUX PAR CLUBS

Pour chaque membre on effectue la totalisation dans une table TP(). Les noms des clubs sont placés dans une table CLUB\$(). Les tables pourraient être triées pour afficher le résultat dans l'ordre des clubs.

| CLUB\$() | TP() |
|----------|------|
| BOULOGNE | 28   |
| PARIS    | 29   |
| MONTIGNY | 27   |

### CLASSEMENT INDIVIDUEL PAR POINTS

Afin d'utiliser le programme de tri du classement par noms nous convertissons le nombre de points sous forme d'une chaîne de caractères :

```
CLE$(NCL)=RIGHT$(" " +STR$(POINT(F)),4)
```

#### LISTE TRIÉE PAR CLUBS

|          |         |    |                |
|----------|---------|----|----------------|
|          |         |    | Mode ? TC      |
| BOULOGNE | DUPONT  | 12 |                |
| BOULOGNE | LEBOSSE | 16 | TOTAL PAR CLUB |
| MONTIGNY | BALU    | 14 |                |
| MONTIGNY | BREBION | 13 | BOULOGNE 28    |
| PARIS    | MARTIN  | 14 | PARIS 29       |
| PARIS    | ROULET  | 15 | MONTIGNY 27    |



```

10 '          GESTION DE CLUBS SPORTIFS
20 '
30 MODE 2:PAPER 0:PEN 1:INK 0,1:INK 1,24
40 DIM NOM$(100),CLUB$(100),POINT(100)
50 DIM CLE$(100),INDEX(100)
60 NFICH=0 ' nombre de fiches
70 INPUT "Nouveau fichier (O/N) ";R$
80 IF R$="O" OR R$="o" THEN 110
90 GOSUB 660 ' appel lecture du fichier
100 '
110 CLS:PRINT "Modes:";PRINT
120 PRINT TAB(3) "C:      creation/modification"
130 PRINT TAB(3) "LF:     liste du fichier"
140 PRINT TAB(3) "LFN:    liste triee par nom"
150 PRINT TAB(3) "LFC:    liste triee par club"
160 PRINT TAB(3) "LFP:    liste du fichier par points"
170 PRINT TAB(3) "TC:     total par clubs"
180 PRINT TAB(3) "FIN:    fin de session(sauvegarde)"
190 PRINT:INPUT "Mode ";M$
200 M$=UPPER$(M$)
210 IF M$="C" THEN GOSUB 300
220 IF M$="LFN" THEN GOSUB 1000
230 IF M$="LFC" THEN GOSUB 1140
240 IF M$="LFP" THEN GOSUB 1280
250 IF M$="TC" THEN GOSUB 1430
260 IF M$="FIN" THEN GOSUB 550:END
270 IF M$="LF" THEN GOSUB 780
280 GOTO 110
290 '===== CREATION/MODIFICATION
300 PRINT
310 PRINT "ENTER pour zone inchangee(en modif)";PRINT
320 INPUT "Nom (ENTER pour fin) ";NOM$:NOM$=UPPER$(NOM$)
330 IF LEN(NOM$)=0 THEN RETURN ' fin de mode?
340 '
350 IF NFICH=0 THEN 400
360 FOR RANG=1 TO NFICH
370 IF NOM$=NOM$(RANG) THEN 460 ' nom existe t-il?
380 NEXT RANG
390 '----- nouveau nom
400 PRINT:INPUT "Nouveau nom OK (O/N) ";R$
410 IF R$<>"O" AND R$<>"o" THEN 300
420 NFICH=NFICH+1
430 RANG=NFICH
440 NOM$(RANG)=NOM$
450 '----- entree/modification zones
460 PRINT
470 PRINT CLUB$(RANG) TAB(15) ' ancienne valeur
480 INPUT "Club ";CLUB$:CLUB$=UPPER$(CLUB$)
490 IF LEN(CLUB$)<>0 THEN CLUB$(RANG)=CLUB$
500 PRINT POINT(RANG) TAB(15)
510 INPUT "Points ";POINT
520 IF POINT<>0 THEN POINT(RANG)=POINT
530 GOTO 300

```

```

540 '===== SAUVEGARDE
550 OPENOUT "CLUBS"
560 '
570 PRINT #9,NFICH
580 FOR F=1 TO NFICH
590   PRINT #9,NOM$(F)
600   PRINT #9,CLUB$(F)
610   PRINT #9,POINT(F)
620 NEXT F
630 CLOSEOUT
640 RETURN
650 '===== LECTURE DU FICHIER
660 OPENIN "CLUBS"
670 '
680 INPUT #9,NFICH
690 FOR F=1 TO NFICH
700   INPUT #9,NOM$(F)
710   INPUT #9,CLUB$(F)
720   INPUT #9,POINT(F)
730 NEXT F
740 CLOSEIN
750 PRINT:PRINT NFICH;"Fiches"
760 FOR TP=1 TO 2000:NEXT TP
770 RETURN
780 '===== liste du fichier
790 PRINT "liste du fichier":PRINT
800 '
810 FOR F=1 TO NFICH
820   IF F MOD 20=0 THEN INPUT "APPUYER SUR ENTER";X$
830   PRINT NOM$(F) TAB(15) CLUB$(F) TAB(30) POINT(F)
840 NEXT F
850 PRINT:INPUT "Appuyer sur <ENTER> ";X$
860 RETURN
870 '----- tri shell
880 ECART=NCL
890 ECART=INT(ECART/2):IF ECART<1 THEN RETURN
900 IV=0
910 FOR K=1 TO NCL-ECART
920   J=K+ECART
930   IF CLE$(J)>CLE$(K) THEN 960
940   X$=CLE$(K):CLE$(K)=CLE$(J):CLE$(J)=X$:IV=1
950   X=INDEX(K):INDEX(K)=INDEX(J):INDEX(J)=X
960 NEXT K
970 IF IV=1 THEN 900
980 GOTO 890
990 '===== LISTE TRIEE PAR NOM
1000 NCL=0 'nombre de cles
1010 FOR F=1 TO NFICH
1020   NCL=NCL+1:CLE$(NCL)=NOM$(F):INDEX(NCL)=F
1030 NEXT F
1040 GOSUB 880
1050 '----- EDITION
1060 PRINT:PRINT "LISTE TRIEE PAR NOMS":PRINT
1070 FOR F=1 TO NCL
1080   X=INDEX(F)
1090   PRINT NOM$(X) TAB(15) CLUB$(X) TAB(40) POINT(X)
1100 NEXT F
1110 PRINT:INPUT "APPUYER SUR ENTER ";X$
1120 RETURN

```

```

1130 '===== TRI PAR CLUB
1140 NCL=0 ' nombre de cles
1150 FOR F=1 TO NFICH
1160 NCL=NCL+1:CLE$(NCL)=CLUB$(F)+NOM$(F):INDEX(NCL)=F
1170 NEXT F
1180 GOSUB 880
1190 '-----EDITION
1200 PRINT:PRINT "LISTE TRIEE PAR CLUBS":PRINT
1210 FOR F=1 TO NCL
1220 X=INDEX(F)
1230 PRINT CLUB$(X) TAB(15) NOM$(X) TAB(30) POINT(X)
1240 NEXT F
1250 PRINT:INPUT "Appuyer sur ENTER";X$
1260 RETURN
1270 '===== TRI PAR POINTS
1280 NCL=0
1290 FOR F=1 TO NFICH
1300 NCL=NCL+1
1310 CLE$(NCL)=RIGHT$(" "+STR$(POINT(F)),4)
1320 INDEX(NCL)=F
1330 NEXT F
1340 GOSUB 880
1350 '----- EDITION
1360 FOR F=1 TO NCL
1370 X=INDEX(F)
1380 PRINT POINT(X) TAB(10) NOM$(X)
1390 NEXT F
1400 PRINT:INPUT "APPUYER SUR ENTER ";X$
1410 RETURN
1420 '===== TOTAL PAR CLUB
1430 NCLUB=0
1440 FOR F=1 TO 10:TP(F)=0:NEXT F
1450 FOR F=1 TO NFICH
1460 FOR C=1 TO NCLUB
1470 IF CLUB$(F)=CLUB$(C) THEN 1520
1480 NEXT C
1490 NCLUB=NCLUB+1 ' nouveau club
1500 C=NCLUB:CLUB$(NCLUB)=CLUB$(F)
1510 '
1520 TP(C)=TP(C)+POINT(F)
1530 NEXT F
1540 '-----EDITION
1550 PRINT:PRINT "TOTAL PAR CLUB":PRINT
1560 FOR C=1 TO NCLUB
1570 PRINT CLUB$(C),TP(C)
1580 NEXT C
1590 PRINT:INPUT "APPUYER SUR ENTER";X$
1600 RETURN

```

## INDEX DE LIVRE

---

Les mots-clés et les numéros de page sont définis en DATA. Ils sont lus dans deux tables CLE\$( ) et PG\$( ) puis triés et édités.

```

10 'idlv          INDEX DE LIVRE
20 '
25 MODE 2:PAPER 0:PEN 1
30 DIM CLE$(200),PG$(200)
40 DATA 81,ABS
50 DATA 22,AND
60 DATA 59,ASC
70 DATA 94,CLEAR
80 DATA 15,AUTO
90 DATA 60,CHR$
100 DATA 63,BIN$
110 DATA 28,BORDER
120 DATA *
130 '----- LECTURE DATAS DANS TABLES
140 FOR J=1 TO 200
150 READ PG$(J):IF PG$(J)="*" THEN NM=J-1:GOTO 200
160 READ CLE$(J)
170 PRINT PG$(J),CLE$(J)
180 NEXT J
190 '----- TRI
200 IV=0
210 FOR J=1 TO NM-1
220 IF CLE$(J+1)>=CLE$(J) THEN 250
230 X$=CLE$(J):CLE$(J)=CLE$(J+1):CLE$(J+1)=X$
240 X$=PG$(J):PG$(J)=PG$(J+1):PG$(J+1)=X$:IV=1
250 NEXT J
260 IF IV=1 THEN 200
270 '----- EDITION
280 CN=0
290 FOR J=1 TO NM
300 IF LEFT$(CLE$(J-1),1)=LEFT$(CLE$(J),1) THEN 330
310 PRINT #CN:PRINT #CN, LEFT$(CLE$(J),1):PRINT #CN
320 '
330 PRINT #CN,TAB(3) CLE$(J) STRING$(18-LEN(CLE$(J)),".") ;
340 PRINT #CN,TAB(22) PG$(J)
350 NEXT J

```

A

```

ABS..... 81
AND..... 22
ASC..... 59
AUTO..... 15

```

B

```

BIN$..... 63
BORDER..... 28

```

C

```

CHR$..... 60
CLEAR..... 94

```

## HISTOGRAMME CIRCULAIRE (CPC 664)

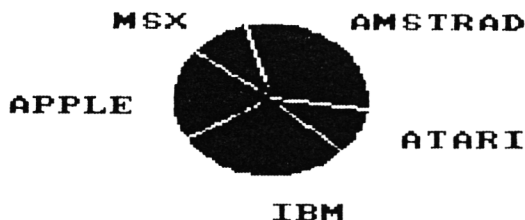
L'histogramme circulaire présenté utilise la fonction "FILL couleur". Chaque secteur de l'histogramme est colorié avec une des trois couleurs disponibles en MODE1.

Sur CPC 464, on pourra remplacer 400 par 400 PLOT XA, YA: DRAW, X, Y, CE.

```

10 ' HISTOGRAMME CIRCULAIRE (CPC664)
20 '
30 MODE 1
40 INK 0,26:INK 1,0:PAPER 0:PEN 1
50 INK 2,2:INK 3,6
60 '
70 DATA AMSTRAD,0.3, MSX,0.1
80 DATA APPLE,0.2, IBM,0.3
90 DATA ATARI,0.1
100 '
110 FOR I=1 TO 5:READ H$(I),H(I):NEXT I
120 '
130 XA=200:YA=200 ' centre
140 R=60 ' rayon
150 '-----
160 AD=0 ' angle debut
170 FOR P=1 TO 5
180 CE=(P MOD 3)+1 ' couleur
190 XD=XA+R*COS(AD):YD=YA+R*SIN(AD)
200 PLOT XA,YA: DRAW XD,YD,CE
210 AF=AD+PI*2*H(P) ' angle fin
220 GOSUB 370
230 X=XA+R*COS(AF):Y=YA+R*SIN(AF)
240 PLOT XA,YA: DRAW X,Y,CE
250 '
260 AT=AD+PI*H(P) ' affichage texte
270 X=XA+R*1.4*COS(AT):Y=YA+R*1.4*SIN(AT)
280 MOVE XA+R/2*COS(AT),YA+R/2*SIN(AT)
290 FILL CE
300 IF AT>PI/2 AND AT<3*PI/2 THEN X=X-16*LEN(H$(P))
310 PLOT X,Y:TAG:PRINT H$(P);
320 AD=AF
330 NEXT P
340 PEN 1
350 END
360 '----- PORTION DE CERCLE
370 FOR A=AD TO AF+0.05 STEP 0.05
380 X=XA+R*COS(A)
390 Y=YA+R*SIN(A)
400 DRAW X,Y,CE
410 NEXT A
420 RETURN

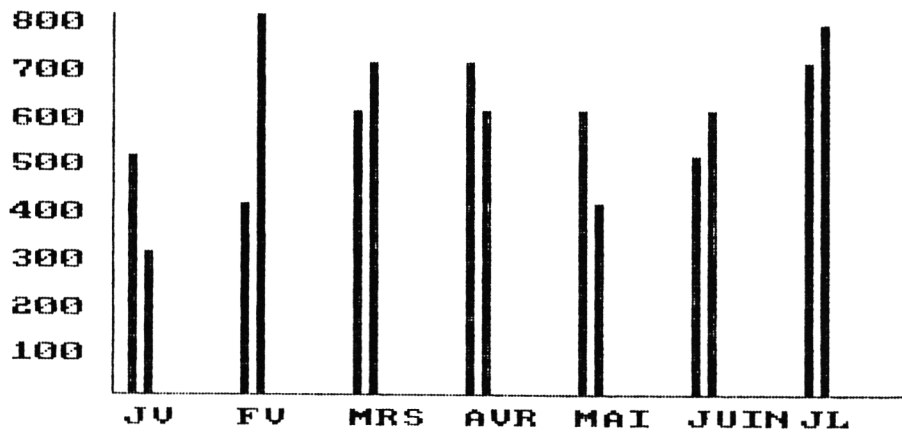
```



# HISTOGRAMME DOUBLE

---

Ce type d'histogramme permet de comparer deux séries de valeurs, par exemple les ventes mensuelles pour deux années différentes.



```

10 '----- HISTOGRAMME DOUBLE
20 MODE 1
30 INK 0,26:INK 1,6:INK 2,0:PAPER 0:PEN 1
35 DIM MOIS$(12),VNTE1(12),VNTE2(12)
40 NM=7 ' nombre de mois
50 DATA JV,500,300, FV,400,800
60 DATA MRS,600,700, AVR,700,600
70 DATA MAI,600,400, JUIN,500,600
80 DATA JL,700,780
90 '--
100 FOR M=1 TO NM:READ MOIS$(M),VNTE1(M),VNTE2(M):NEXT M
110 '
120 XA=80:YA=30 ' depart axes
130 IX=70 ' intervalle X
140 HECR=300 ' hauteur ecran
150 '----- recherche maxi
160 MX=VNTE(1)
170 FOR M=2 TO NM
180 IF VNTE1(M)>MX THEN MX=VNTE1(M)
190 IF VNTE2(M)>MX THEN MX=VNTE2(M)
200 NEXT M
210 ECH=HECR/MX ' echelle
220 '----- axes
230 PLOT XA,YA:DRAW XA+NM*IX,YA,1
240 PLOT XA,YA:DRAW XA,HECR+YA,1
250 '----- affichage mois
260 FOR M=1 TO NM
270 X=XA+8+IX*(M-1):Y=YA-12
280 PLOT X,Y:TAG:PRINT MOIS$(M);
290 NEXT M
300 '----- courbe 1
310 FOR M=1 TO NM
320 X1=XA+10+IX*(M-1):Y1=YA+VNTE1(M)*ECH
330 FOR DX=0 TO 5
340 PLOT X1+DX,YA:DRAW X1+DX,Y1,1
350 NEXT DX
360 NEXT M
370 '----- courbe 2
380 FOR M=1 TO NM
390 X1=XA+20+IX*(M-1):Y1=YA+VNTE2(M)*ECH
400 FOR DX=0 TO 5
410 PLOT X1+DX,YA:DRAW X1+DX,Y1,2
420 NEXT DX
430 NEXT M
440 '----- graduations
450 NG=10 ' nombre de graduations
460 PAS=INT(MX/NG)
470 IF PAS>50 AND PAS<100 THEN PAS=100
480 IF PAS>10 AND PAS<50 THEN PAS=50
490 NG=INT(MX/PAS)
500 IG=HECR/NG ' intervalle graduations
510 FOR G=1 TO NG
520 MOVE 1,YA+G*IG
530 TAG:PRINT G*PAS;
540 NEXT G

```





# JEUX | 3

## LE MOT LE PLUS LONG \_\_\_\_\_

Dans chaque ligne de DATA nous indiquons :

- Les 9 lettres qui sont affichées à l'écran.
- La liste des réponses possibles (\* représente la fin de la liste).

La première version ne comporte pas de chronomètre (lignes 10 à 490). Pour une version avec chronomètre ajoutez les lignes 510 à 850 et faire : 340 GOSUB 520.

A E E P T D S O L

PEDALES  
PETALES  
PEDALOS

VOTRE REPONSE? ■

<sup>11</sup>  
TROP TARD

```

10 'MOT      MOT LE PLUS LONG
20 '
30 MODE 1:INK 0,0:INK 1,26:PAPER 0:PEN 1
40 NM=6      ' Nombre de lignes
50 DATA FICHIERSW,FICHIERS,*
60 DATA PEDALESTO,PEDALES,PETALES,PEDALOS,*
70 DATA MOQUETTEA,MOQUETTE,MAQUETTE,*
80 DATA ARMOIRESI,ARMOIRIES,*
90 DATA FAUTEUILY,FAUTEUIL,*
100 DATA CHEVELUAR,CHEVELU,CHALEUR,CHAULER,*
110 '-----
120 RESTORE      ' debut DATA
130 CLS
140 FOR J=1 TO 9:F(J)=0:NEXT J
150 X=INT(RND(1)*NM)      ' mot au hasard
160 FOR J=1 TO X      ' lecture des precedents
170   READ X$:IF X$<>"*" THEN 170
180 NEXT J
190 '---- stockage reponses
200 READ MP$
210 FOR J=1 TO 5
220   READ MOT$(J)
230   IF MOT$(J)="*" THEN 260
240 NEXT J
250 '----- Affichage des lettres au hasard
260 LOCATE 10,10
270 FOR J=1 TO 9
280   P=1+INT(RND(1)*9)
290   IF F(P)=0 THEN PRINT MID$(MP$,P,1);SPC(1);:F(P)=1 ELSE 280
300 NEXT J
310 PRINT
320 HD=TIME
330 '---- POUR VERSION CHRONO: 340 GOSUB 520
340 LOCATE 6,20:INPUT "Votre reponse(ou FIN) ";LIG$
350 LOCATE 1,22:PRINT INT((TIME-HD)/400);"secondes"
360 '
370 LIG$=UPPER$(LIG$):IF LIG$="FIN" THEN END
380 FOR J=1 TO 5
390   IF MOT$(J)="*" THEN 440
400   IF LIG$=MOT$(J) THEN LOCATE 1,15:PRINT "OK":GOTO 480
410 NEXT J
420 '
430 '--- affichage bonnes reponses
440 FOR J=1 TO 5
450   IF MOT$(J)="*" THEN 480
460   LOCATE 1,14+J:PRINT MOT$(J)
470 NEXT J
480 FOR TP=1 TO 3000:NEXT TP
490 GOTO 120

```

Pour une version avec chronomètre ajoutez les instructions suivantes :

```

500 '----- ENTREE AVEC CHRONO--
510 ' Ajouter les lignes 10 a 490
520 HD=TIME
530 AFTER 800,0 GOSUB 600
540 XS=6:YS=20
550 LOCATE XS,YS:PRINT "VOTRE REPONSE?(ou FIN) ";SPC(12);
560 XS=XS+23:GOSUB 640
570 X=REMAIN(0)
580 RETURN
590 '----- SOUS PROGRAMME CHRONO
600 LOCATE 1,23:PRINT "TROP TARD"
610 TT=1
620 RETURN
630 '----- saisie d'une ligne avec INKEY$
640 LIG$=""          ' pas d'espace entre les ""
650 TT=0            ' temps
660 '
670 LG=LEN(LIG$):LOCATE XS+LG,YS:PRINT CHR$(143);CHR$(32) ' 143
:curseur
680 LOCATE XS+LG,YS
690 '
700 C$=INKEY$
710 IF TT=1 THEN RETURN
720 IF LEN(C$)=0
    THEN LOCATE 1,22:PRINT INT((TIME-HD)/400):GOTO 670
730 C=ASC(C$)
740 '
750 IF C<>127 THEN 780          ' code suppression
760 IF LG>0 THEN LIG$=
    LEFT$(LIG$,LG-1):PRINT CHR$(8);CHR$(32):GOTO 670 ELSE 670
770 '
780 IF C=13 THEN 840          ' code ENTER?
790 IF C<32 OR C>127 THEN PRINT CHR$(7);:GOTO 700
800 LIG$=LIG$+C$          ' ajout caractere
810 PRINT C$              ' affichage caract.
820 GOTO 670
830 '
840 LOCATE XS+LG,YS:PRINT CHR$(32)
850 RETURN

```

## LE COMPTE EST BON

Nous avons programmé "le compte est bon" d'après la méthode suivante (JC BUISSON l'Ordinateur Individuel numéro 54bis) :

- Nous essayons de diviser le nombre cherché par un des six nombres proposés de façon à obtenir un quotient entier.
- Si nous ne trouvons pas de quotient entier, nous procédons de la même façon, mais en ajoutant ou en retranchant un des six nombres proposés au nombre cherché.
- En cas de succès, nous procédons comme ci-dessus avec le quotient obtenu.

Si nous parvenons à un quotient égal à 1, nous sommes en mesure de proposer un bon compte.

### Exemple :

Soit à trouver 840 avec 5,5,8,25,4,3.

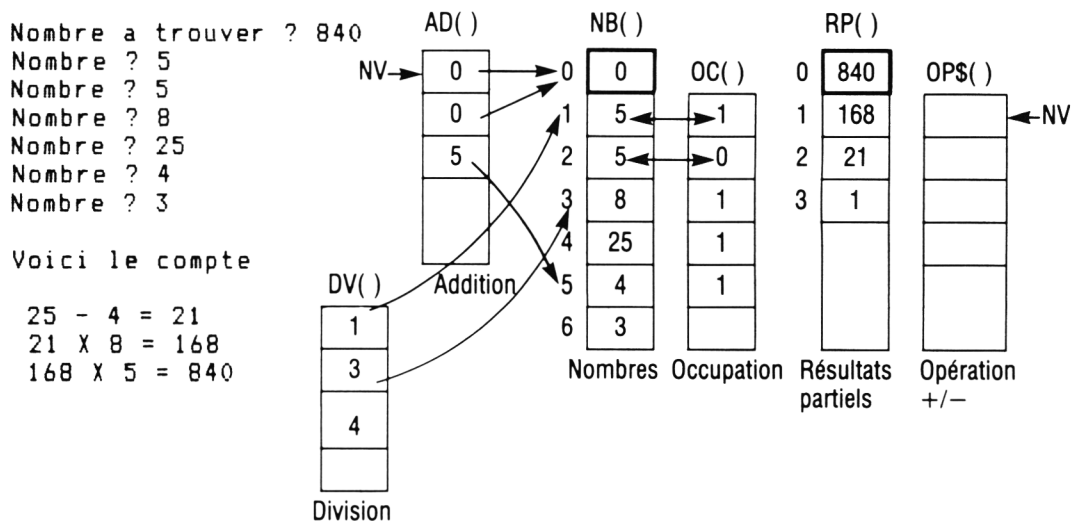
On fait : 1/  $\frac{840-0}{5} = 168$

2/  $\frac{168-0}{8} = 21$

3/  $\frac{21+4}{25} = 1$

Il suffit alors de faire :  $25-4=21$   
 $21 \times 8 = 168$   
 $168 \times 5 = 840$

Cette méthode n'envisage qu'un certain nombre de solutions. Le "bon compte" est trouvé deux fois sur trois.



```

10 '      LE COMPTE EST BON
20 '
30 '      D'apres J.BUISSON (ORDINATEUR INDIVIDUEL 54BI
40 '
50 '      NB() : nombres
60 '      RP() : resultats partiels
70 '      AD() : pointeurs vers additions
80 '      DV() : pointeurs vers divisions
90 '      OC() : temoin occupation de NB()
100 '
110 MODE 2:PAPER 0:PEN 1:INK 0,26:INK 1,0
120 INPUT "Nombre a trouver ";RP(0)
130 '
140 FOR J=1 TO 6:INPUT "Nombre ";NB(J):NEXT J
150 '
160 FOR J=1 TO 6:OC(J)=0:NEXT J
170 NV=1
180 '
190 AD(NV)=0:DV(NV)=1 ' Initialisation pointeur
200 '
210 IF OC(AD(NV))=1 THEN 320 ' Deja utilise
220 IF OC(DV(NV))=1 OR AD(NV)=DV(NV) THEN 290 ' memes nombres
230 '
240 RP(NV)=(RP(NV-1)-NB(AD(NV)))/NB(DV(NV))
250 IF RP(NV)=ABS(INT(RP(NV))) THEN OP$(NV)="+":GOTO 380
260 RP(NV)=(RP(NV-1)+NB(AD(NV)))/NB(DV(NV))
270 IF RP(NV)=ABS(INT(RP(NV))) THEN OP$(NV)="-":GOTO 380
280 '
290 DV(NV)=DV(NV)+1:IF DV(NV)<=6 THEN 220
300 DV(NV)=1
310 '
320 AD(NV)=AD(NV)+1:IF AD(NV)<=6 THEN 210
330 '----- pas de solution de niveau N
340 IF NV=1 THEN PRINT "Pas de solution":STOP
350 '
360 NV=NV-1:OC(AD(NV))=0:OC(DV(NV))=0:GOTO 290
370 '----- solution niveau NV
380 IF RP(NV)=0 OR RP(NV)=1 THEN 440
390 IF AD(NV)<>0 THEN OC(AD(NV))=1
400 OC(DV(NV))=1
410 '---
420 NV=NV+1:GOTO 190
430 '-----Affichage resultats
440 PRINT:PRINT "Voici le compte":PRINT
450 FOR J=NV TO 1 STEP-1
460 IF RP(J)=0 THEN 530
470 IF RP(J)=1 THEN 510
480 IF NB(DV(J))=1 THEN 510
490 PRINT RP(J) "X" NB(DV(J)) "=" RP(J)*NB(DV(J))
500 '
510 IF NB(AD(J))=0 THEN 530
520 PRINT RP(J)*NB(DV(J)) OP$(J) NB(AD(J)) "=" RP(J-1)
530 NEXT J
540 '----- POUR TEST
550 PRINT
560 FOR I=0 TO NV:PRINT RP(I),NB(AD(I)),NB(DV(I)):NEXT I

```

On pourra effectuer des divisions du type suivant en faisant  $NB(7)=1$  :

$$\frac{NCHERCHE + -N}{1} = Q$$

Il faut pour cela ajouter au programme précédent :

```

155 NB(7)=1
290 DV(NV)=DV(NV)+1: IF DV(NV)<=7 THEN 220
380 IF DV(NV)=7 AND AD(NV)= 0 THEN 290
381 IF RP(NV)=1 AND DV(NV)=7 THEN 290
382 IF NV>4 THEN 340
385 IF RP(NV)=0 OR RP(NV)=1 THEN 440
390 IF AD(NV)<>0 THEN OC(AD(NV))=1
400 IF DV(NV)<>7 THEN OC(DV(NV))=1

```

Le nombre de solutions augmente sensiblement mais le temps de recherche augmente aussi très nettement. Essayez de trouver 876 avec 50,8,7,4,3,25.

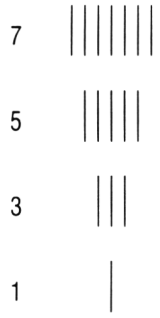
On obtient :

$$\begin{aligned}
 25 - 3 &= 22 \\
 22 * 8 &= 176 \\
 176 - 7 &= 169 \\
 169 + 50 &= 219 \\
 219 * 4 &= 876
 \end{aligned}$$

# JEU DE MARIENBAD

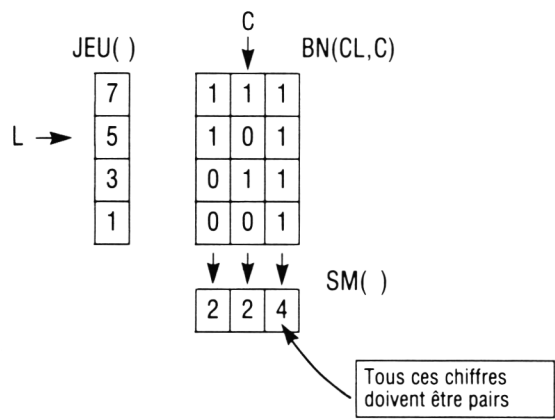
## RÈGLE DU JEU

Quatre rangées contiennent 7, 5, 3, 1 allumettes. Chaque joueur doit prendre à tour de rôle des allumettes dans une seule rangée à la fois. Celui qui prend la dernière perd.



Un algorithme donne le nombre d'allumettes à prendre pour laisser un jeu perdant à l'adversaire :

- On convertit en binaire le nombre d'allumettes de chaque ligne (dans une table BN(,)).
- On effectue la somme de chaque colonne de la table BN(,).



- Pour laisser un jeu perdant à l'adversaire, il suffit que les chiffres de toutes les colonnes de SM() soient pairs.

```

10 '      JEU DE MARIENBAD
20 '
30 '
40 ' SI VOUS COMMENCEZ, VOUS PERDEZ
50 '
60 MODE 2:PAPER 0:PEN 1:INK 0,0:INK 1,26
70 JEU(1)=7:JEU(2)=5
80 JEU(3)=3:JEU(4)=1
90 '
100 GOSUB 910
110 '----
120 PRINT:PRINT "ESC 2 fois pour FIN":PRINT
130 INPUT "Voulez vous commencer (O/N) ";R$
140 JR=0:IF R$="O" OR R$="o" THEN JR=1
150 '
160 FP=0 ' fin de partie
170 '
180 IF JR=1 THEN GOSUB 230:JR=1-JR
190 IF FP=1 THEN 70
200 IF JR=0 THEN GOSUB 350:JR=1-JR
210 IF FP=1 THEN 70
220 GOTO 180
230 '===== JOUEUR
240 PRINT:INPUT "Quelle ligne ";L
250 INPUT "Combien ";N
260 '
270 IF N>JEU(L) OR N<=0
    THEN PRINT:PRINT "Vous trichez":PRINT:PRINT:GOTO 240
280 JEU(L)=JEU(L)-N
290 '
300 GOSUB 910 ' affichage partie en cours
310 GOSUB 980 ' fin de partie?
320 IF S=1 THEN PRINT:PRINT:PRINT "Vous avez gagne":FP=1
330 RETURN
340 '===== MACHINE
350 FOR L=1 TO 4
360 X=JEU(L):GOSUB 700
370 NEXT L
380 '
390 '----- RECHERCHE JEU GAGNANT
400 FOR L=1 TO 4 ' 4 lignes
410 FOR N=1 TO JEU(L) ' on essaie d'en prendre 1 a JEU(L)
420 IF JEU(L)=0 THEN 480
430 X=JEU(L)-N:GOSUB 680
440 GOSUB 760 ' sommation colonnes
450 IF GN=1 THEN 550
460 NEXT N
470 X=JEU(L):GOSUB 680 ' retabliss. ligne binaire de B(,)
480 NEXT L
490 '----- PAS DE SOLUTION GAGNANTE
500 FOR L=1 TO 4
510 IF JEU(L)>0 THEN N=1:GOTO 550
520 NEXT L

```



```

530 STOP
540 '--
550 IF SM(1)<>0 OR SM(2)<>0 THEN 610
560 IF SM(0)=2 AND N=JEU(L) AND JEU(L)>1
    THEN N=JEU(L)-1:GOTO 610
570 IF SM(0)=4 THEN N=JEU(L):GOTO 610
580 IF SM(0)=2 THEN N=JEU(L):GOTO 610
590 IF SM(0)=0 AND N>1 THEN N=JEU(L)-1
600 '
610 PRINT:PRINT "Je joue: ligne:";L;" J'en prends:";N
620 '
630 JEU(L)=JEU(L)-N
640 GOSUB 910:GOSUB 980
650 IF S=1 THEN PRINT:PRINT "J'AI GAGNE ":PRINT:FP=1
660 RETURN
670 '----CV BINAIRE
680 ' ENTREE:X,L SORTIE:REPLIT BN(L,)
690 '
700 FOR P=2 TO 0 STEP-1
710 BN(L,P)=INT(X/(2^P)+0.01) ' ^=fleches haut
720 X=X-BN(L,P)*(2^P)
730 NEXT P
740 RETURN
750 '----- SOMMATION COLONNES
760 FOR C=2 TO 0 STEP-1
770 S=0
780 FOR L1=1 TO 4
790 S=S+BN(L1,C)
800 NEXT L1
810 SM(C)=S
820 NEXT C
830 '
840 GN=0
850 FOR C=2 TO 0 STEP-1
860 IF INT(SM(C)/2)<> SM(C)/2 THEN GN=0:RETURN
870 NEXT C
880 GN=1
890 RETURN
900 '----- DESSIN PARTIE
910 PRINT
920 FOR L=1 TO 4
930 PRINT L;SPC(2);
940 PRINT STRING$(JEU(L),"I")
950 NEXT L
960 PRINT:RETURN
970 '----- FIN DE PARTIE?
980 S=0:FOR L=1 TO 4:S=S+JEU(L):NEXT L:RETURN

```

## JEU DE MARIENBAD PAR APPRENTISSAGE

---

Nous proposons le jeu de Marienbad mais, cette fois-ci, la machine apprend à jouer en "imitant" son adversaire. Pour la première partie, la machine prend une allumette dans une rangée quelconque. Dans deux tables LUI() et J() sont stockés tous les jeux de la partie.

| LUI( ) | J( ) |
|--------|------|
| 1      | 7531 |
| 0      | 7511 |
| 1      | 7411 |
| 0      |      |
|        |      |

Jeu en cours

A la fin de la partie nous cumulons les jeux de la partie achevée dans deux tables.

| TJ() | G() |
|------|-----|
| 7431 | -1  |
| 6431 | 0   |
| 6421 | -1  |
| 5421 | 0   |
| 5411 | -1  |
| 4411 | 1   |
| 4410 | -4  |
| 4400 | 6   |
| 4300 | -6  |
| 3300 | 8   |
| 3200 | -7  |
| 2200 | 9   |
| 2100 | -10 |
| 1000 | 15  |

Récapitulatif  
des jeux

Gains

Dans une table des gains G(), nous faisons +1 si la partie a été gagnée et -1 si elle a été perdue. Ainsi, pour les jeux suivants, la machine consulte ces tables pour chercher des configurations gagnantes.

Au fur et à mesure des parties, la machine dispose de plus en plus de configurations gagnantes. Elle doit jouer de mieux en mieux, surtout si son adversaire joue bien.

Pour tester le programme, laissez-le commencer la partie la première fois et gagnez la partie. Commencez la seconde partie (comme lui) et il jouera comme vous aviez joué lors de la première partie et gagnera lui aussi.

```

10 ' APPR      JEU DE MARIENBAD PAR APPRENTISSAGE
20 '
30 DIM TJ(300),G(300),J(30),LUI(30)
40 MODE 2:PAPER 0:PEN 1:INK 0,0:INK 1,26
50 JEU(1)=7:JEU(2)=5
60 JEU(3)=3:JEU(4)=1
70 '
80 GOSUB 670
90 '--
100 PRINT:PRINT "ESC 2 fois pour stopper":PRINT
110 INPUT "Voulez vous commencer (O/N) ";R$
120 JR=0:IF R$="O" OR R$="o" THEN JR=1
130 '
140 FP=0:COUP=0
150 '
160 IF JR=1 THEN GOSUB 210:JR=1-JR
170 IF FP=1 THEN GOSUB 970:GOTO 50
180 IF JR=0 THEN GOSUB 350:JR=1-JR
190 IF FP=1 THEN GOSUB 970:GOTO 50
200 GOTO 160
210 '===== JOUEUR
220 INPUT "Quelle ligne ";L
230 INPUT "Combien ";N
240 '
250 IF N>JEU(L) OR N<=0 THEN PRINT:PRINT "Vous trichez ":PRINT:G
OTO 220
260 '
270 JEU(L)=JEU(L)-N
280 GOSUB 790:GOSUB 920
290 GOSUB 670 ' Affichage partie en cours
300 GOSUB 730 ' Fin de partie?
310 IF S=1 THEN PRINT:PRINT "Vous avez gagne ":FP=1:G=-1
320 IF S=0 THEN PRINT:PRINT "J'ai gagne ":FP=1:G=1
330 RETURN
340 '===== MACHINE
350 MCOUP=0
360 '----- RECHERCHE DANS TABLE DES JEUX
370 IF TJ(1)=0 THEN N=1:GOTO 620
380 FOR LL=1 TO 4
390   FOR NN=1 TO JEU(LL) ' On essaie d'en prendre NN
400     SJ=JEU(LL) ' Sauvegarde ligne modifiee
410     JEU(LL)=JEU(LL)-NN:GOSUB 790
420     FOR J=1 TO 200 ' Recherche config gagnante
430       IF TJ(J)=0 THEN 470
440       IF JCOD<>TJ(J) THEN 460
450       IF G(J)>MCOUP THEN MCOUP=G(J):NJ=NN:PL=LL
460     NEXT J
470     JEU(LL)=SJ
480   NEXT NN
490 NEXT LL
500 IF MCOUP=0 THEN 620
510 L=PL:N=NJ
520 '
530 PRINT:PRINT "J'en prends: ";N; " Ligne : ";L
540 JEU(L)=JEU(L)-N
550 GOSUB 790:GOSUB 920
560 GOSUB 670
570 GOSUB 730
580 IF S=1 THEN PRINT:PRINT "J'ai gagne ":PRINT :G=1:FP=1
590 IF S=0 THEN PRINT:PRINT "Vous avez gagne":PRINT:G=-1:FP=1
600 RETURN

```

```

610 '----- PAS DE SOLUTION EN TABLE
620 L=INT(RND(1)*4)+1      ' ligne au hasard
630 IF JEU(L)<1 THEN 620
640 N=INT(RND(1)*JEU(L))+1
650 GOTO 530
660 '----- DESSIN PARTIE
670 PRINT
680 FOR L=1 TO 4
690 PRINT L;STRING$(JEU(L),"I")
700 NEXT L
710 PRINT
720 '----- FIN DE PARTIE?
730 S=0
740 FOR L=1 TO 4
750   S=S+JEU(L)
760 NEXT L
770 RETURN
780 '----- CONVERSION JEU() --> JEU CODE
790 FOR J=1 TO 4:X(J)=JEU(J):NEXT J
800 '---- tri
810 IV=0
820 FOR J=1 TO 3
830   IF X(J+1)<X(J) THEN X=X(J):X(J)=X(J+1):X(J+1)=X:IV=1
840 NEXT J
850 IF IV=1 THEN 810
860 '---
870 JCOD=0
880 FOR J=1 TO 4
890   JCOD=JCOD+X(J)*10^(J-1)      ' ^ fleche haut
900 NEXT J
910 RETURN
920 '----- MAJ TABLE JEU ACTUEL
930 COUP=COUP+1:J(COUP)=JCOD
940 LUI(COUP)=JR
950 RETURN
960 '----- MAJ FIN DE PARTIE
970 FOR K=1 TO COUP
980   X=J(K)
990   FOR J=1 TO 300
1000    IF TJ(J)=0 THEN TJ(J)=X
1010    IF X=TJ(J) THEN 1040
1020  NEXT J
1030 '
1040  IF LUI(K)=1 THEN G(J)=G(J)-G
1050  IF LUI(K)=0 THEN G(J)=G(J)+G
1060 NEXT K
1070 RETURN

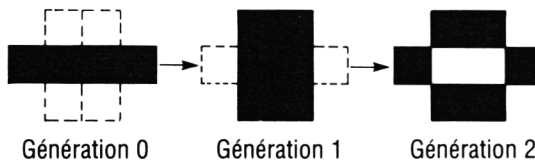
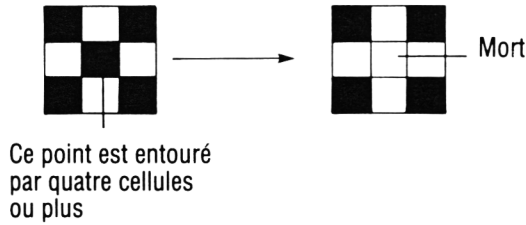
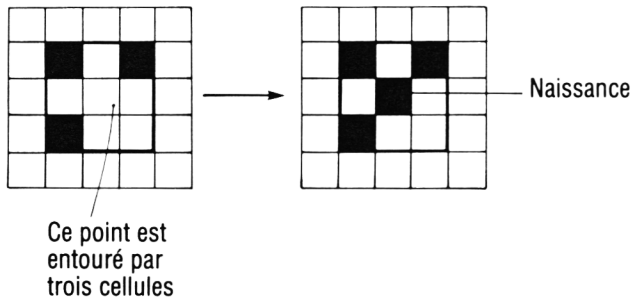
```

# JEU DE LA VIE

CONWAY a imaginé le “jeu de la vie”. Des cellules se reproduisent ou meurent en fonction de leur juxtaposition.

## LES RÈGLES SONT LES SUIVANTES

- Chaque point entouré par trois cellules donne lieu à une naissance.
- Chaque cellule entourée de quatre cellules ou plus meurt (étouffement).
- Chaque cellule entourée par moins de deux cellules meurt (isolement).



## PREMIÈRE MÉTHODE

Nous appliquons directement le principe énoncé. Pour chaque point nous considérons les points voisins. Nous positionnons à 1 les éléments d'une table VN(X,Y) lorsqu'il y a naissance, puis nous affichons le contenu de la table.

|            |  |  |   |  |  |
|------------|--|--|---|--|--|
| TABLE VA() |  |  |   |  |  |
|            |  |  | 1 |  |  |
|            |  |  | 1 |  |  |
|            |  |  | 1 |  |  |
|            |  |  |   |  |  |

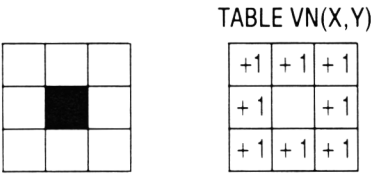
|  |   |   |   |  |  |
|--|---|---|---|--|--|
|  |   |   |   |  |  |
|  |   |   |   |  |  |
|  | 1 | 1 | 1 |  |  |
|  |   |   |   |  |  |
|  |   |   |   |  |  |

 TABLE VN() |

DEUXIÈME MÉTHODE

La première méthode présentée est très lente puisque pour chaque point, nous explorons systématiquement ses huit points voisins. Sachant que seuls les points allumés peuvent provoquer une naissance ou une mort dans leur voisinage, nous procédons ainsi :

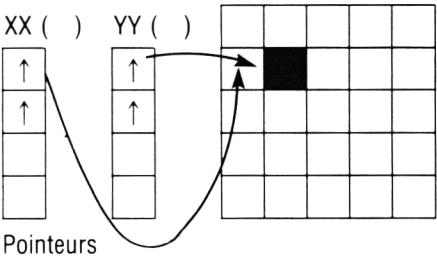
- Pour chaque point allumé, nous augmentons de un les huit points voisins.



- Nous explorons la table VN(X,Y) et allumons tous les points totalisant trois (naissance). Nous éteignons tous les points supérieurs à quatre.

TROISIÈME MÉTHODE

De façon à ne s'intéresser qu'aux points allumés nous ajoutons deux tables XX() et YY() "pointant" vers les cellules en vie. Ainsi, le temps de traitement est notablement réduit ; il est indépendant de la dimension de la table V(X,Y).



**Voici un exemple d'exécution :**

Vous entrez la génération de départ ligne par ligne. Ci-dessous vous appuyez sur ENTER pour les cinq premières lignes. Pour la sixième, vous entrez 11 "\*" puis vous appuyez sur ENTER.

Les générations successives sont affichées toutes les deux minutes pour "VIE1" et toutes les dix secondes pour "VIE3".

```

-----
?
?
?
?
?
?          *****
?
?
?
?          *****
?
?
?
?
?

```

```

-----
Entrer les positions pour chaque ligne
Ex:          *****      (puis ENTER)

```

GENERATION 8

```

    ***
  ** **
** * * **
*  * * *
* * * * *
 *      *

 *      *
* * * * *
*  * * *
** * * **
  ** **
    ***

```

GENERATION 30

```

  * * * *
  * * * *
  * * * *
* ** ** *
*      *
  *      *

  *      *
*      *
* ** ** *
  * * * *
  * * * *
  * * * *
  *      *
  *      *

```

```

10 ' JEU DE LA VIE NO 1
20 '
30 DEFINT A,Z
40 MODE 2:PAPER 0:PEN 1:INK 0,1:INK 1,24
50 X1=2:X2=30 ' bornes territoire
60 Y1=2:Y2=20
70 DIM VA(X2+1,Y2+1) ' ancienne generation
80 DIM VN(X2+1,Y2+1) ' nouvelle generation
90 '
100 LOCATE X1+1,Y1-1:PRINT STRING$(X2-X1,"-")
110 LOCATE X1+1,Y2+1:PRINT STRING$(X2-X1,"-")
120 '----- SAISIE
130 LOCATE 1,24:PRINT "Entrer les positions pour chaque ligne"
140 LOCATE 1,25:PRINT "Ex: ***** (puis ENTER) ou ENTER"
150 FOR Y=Y1 TO Y2
160 LOCATE X1-1,Y
170 LINE INPUT "?",LIG$
180 FOR X=X1 TO X2
190 IF MID$(LIG$,X,1)="*" THEN VA(X,Y)=1
200 NEXT X
210 NEXT Y
220 GOSUB 460
230 '----- Remplissage table VN()
240 FOR Y=Y1 TO Y2
250 FOR X=X1 TO X2
260 VN(X,Y)=0:TV=0
270 FOR DY=-1 TO 1 ' 8 points autour du point X,Y
280 FOR DX=-1 TO 1
290 IF DX=0 THEN IF DY=0 THEN 310
300 IF VA(X+DX,Y+DY)=1 THEN TV=TV+1
310 NEXT DX
320 NEXT DY
330 IF TV>=4 THEN VN(X,Y)=0:GOTO 360
340 IF TV=3 THEN VN(X,Y)=1
350 IF (TV=2 OR TV=3) AND VA(X,Y)=1 THEN VN(X,Y)=1
360 NEXT X
370 NEXT Y
380 '----- transfert VN() -->VA()
390 GN=GN+1
400 FOR Y=Y1 TO Y2
410 FOR X=X1 TO X2:VA(X,Y)=VN(X,Y):NEXT X
420 NEXT Y
430 GOSUB 460
440 GOTO 240
450 '----- AFFICHAGE
460 CLS
470 FOR Y=Y1 TO Y2
480 FOR X=X1 TO X2
490 IF VA(X,Y)=1 THEN LOCATE X,Y:PRINT "*";
500 NEXT X
510 NEXT Y
520 LOCATE 30,1:PRINT "GENERATION ";GN
530 RETURN

```



```

10 '          JEU DE LA VIE NO 3
20 '
30 DEFINT A,Z
40 MODE 2
50 X1=2:X2=50          ' bornes
60 Y1=2:Y2=22
70 DIM VA(X2+1,Y2+1)   ' ancienne generation
80 DIM VN(X2+1,Y2+1)   ' nouvelle generation
90 DIM XX(1000),YY(1000),X1(1000),Y1(1000)
100 '----- Saisie
110 LOCATE X1+1,Y1-1:PRINT STRING$(X2-X1,"-")
120 LOCATE X1+1,Y2+1:PRINT STRING$(X2-X1,"-")
130 LOCATE 1,24:PRINT "Entrer les positions pour chaque ligne"
140 LOCATE 1,25:PRINT "Ex:  ***** (puis ENTER) ou ENTER"
150 N=0
160 FOR Y=Y1 TO Y2
170   LOCATE X1-1,Y
180   LINE INPUT "?",LIG$
190   FOR X=X1 TO X2
200     IF MID$(LIG$,X,1)="*"
        THEN VA(X,Y)=1:VN(X,Y)=3:NC=NC+1:XX(NC)=X:YY(NC)=Y
210   NEXT X
220 NEXT Y
230 CLS
240 GOSUB 400
250 '-----
260 FOR N=1 TO NC
270   X=XX(N):Y=YY(N)
280   FOR DY=-1 TO 1
290     IF Y+DY<Y1 OR Y+DY>Y2 THEN 340
300     FOR DX=-1 TO 1
310       IF X+DX<X1 OR X+DX>X2 THEN 330
320       VN(X+DX,Y+DY)=VN(X+DX,Y+DY)+1
330     NEXT DX
340   NEXT DY
350 NEXT N
360 GN=GN+1
370 GOSUB 390
380 GOTO 260
390 '-----AFFICHAGE
400 N1=0
410 '
420 FOR N=1 TO NC
430   XX=XX(N):YY=YY(N)
440   FOR DY=-1 TO 1
450     FOR DX=-1 TO 1
460       X=XX+DX:Y=YY+DY
470       IF VN(X,Y)=0 THEN 540
480       IF VN(X,Y)<=2
        THEN LOCATE X,Y:PRINT CHR$(32):VA(X,Y)=0:GOTO 540
490       IF VN(X,Y)=3 THEN LOCATE X,Y:PRINT "*":VA(X,Y)=1:GOTO 530
500       IF VN(X,Y)>=5
        THEN LOCATE X,Y:PRINT CHR$(32):VA(X,Y)=0:GOTO 540
510       IF VA(X,Y)=1 AND VN(X,Y)=4 THEN 530
520     GOTO 540

```



```

530   N1=N1+1:X1(N1)=X:Y1(N1)=Y
540   VN(X,Y)=0
550   NEXT DX
560   NEXT DY
570   NEXT N
580   NC=N1
590   FOR N=1 TO NC:YY(N)=Y1(N):XX(N)=X1(N):NEXT N
600   LOCATE 30,1:PRINT "GENERATION ";GN
610   RETURN

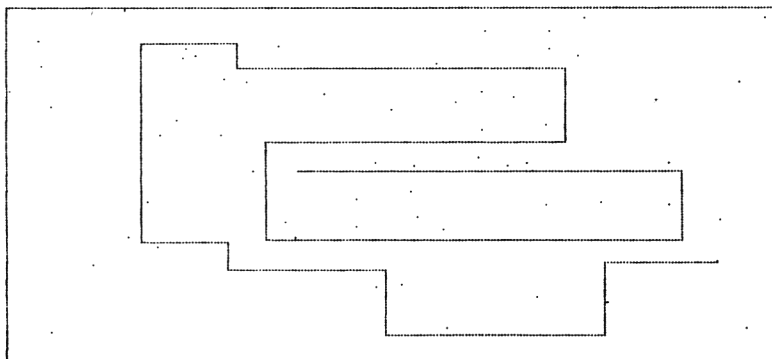
```

## OBSTACLE

---

Un point se déplace à travers des obstacles. Vous devez adapter la direction à l'aide des flèches pour éviter les obstacles.

905 POINTS



Utilisez les fleches

```

10 '   OBSTACLE
20 '
30 MODE 1
40 INK 0,0:INK 1,26
50 INPUT "NIVEAU (1,2,3 (9 pour fin)) ";NV
60 IF NV=9 THEN END
70 CLS
80 LOCATE 1,24:PRINT "Utilisez les fleches"
90 '----- CADRE
100 XD=20:XF=500
110 YD=40:YF=350
120 PLOT XD,YD,1:DRAW XF,YD
130 DRAW XF,YF:DRAW XD,YF:DRAW XD,YD
140 '
150 '----- OBSTACLES
160 FOR P=1 TO (NV+1)*20
170   X=RND*(XF-XD)+XD
180   Y=RND*(YF-YD)+YD
190   PLOT X,Y,1
200 NEXT P
210 '----- COORDONNEES DEPART
220 X=200+RND*10:Y=200+RND*10:IF TEST(X,Y)<>0 THEN 220
230 '----- ATTENTE DEPART
240 C$=INKEY$
250 IF LEN(C$)<>0 THEN GOSUB 300:GOTO 370
260 PLOT X,Y,1
270 PLOT X,Y,0
280 GOTO 240
290 '----- Actualisation X,Y
300 C=ASC(C$)
310 IF C=242 THEN DX=-2:DY=0
320 IF C=243 THEN DX=2:DY=0
330 IF C=241 THEN DX=0:DY=-2
340 IF C=240 THEN DX=0:DY=2
350 RETURN
360 '-----
370 P=0
380 '
390 C$=INKEY$:IF LEN(C$)<>0 THEN GOSUB 300
400 P=P+1
410 X=X+DX:Y=Y+DY
420 IF TEST(X,Y)<>0 THEN 470
430 PLOT X,Y,1
440 FOR TP=1 TO 30-NV*3:NEXT TP
450 GOTO 390
460 '-----
470 PRINT CHR$(7);
480 LOCATE 1,2:PRINT P;"POINTS"
490 FOR TP=1 TO 2000:NEXT TP
500 GOTO 30

```

## MÉTÉORITES

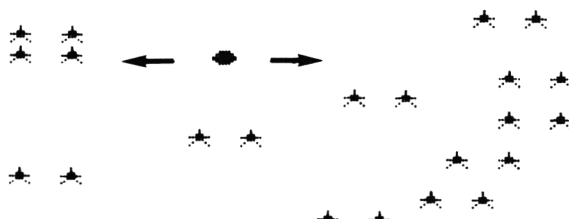
---

Vous déplacez une soucoupe avec les flèches ← → à travers des météorites.

```

10 '----- METEORITES
20 MODE 1:PAPER 0:PEN 1
30 INK 0,26:INK 1,0
40 PRINT "Fleches <- et -> / ESC 2 fois pour fin"
50 FOR TP=1 TO 2000:NEXT TP
60 CLS:F=0
70 XV=15:YV=3                                ' position vehicule
80 C=145:SYMBOL AFTER C
90 SYMBOL C,8,8,28,127,28,34,65,0            ' dessin etoile
100 FOR L=1 TO 10000
110 AX=XV:AY=YV                                ' ancienne position
120 IF INKEY(1)=0 THEN IF XV<33 THEN XV=XV+1 ' droite
130 IF INKEY(8)=0 THEN IF XV>2 THEN XV=XV-1 ' gauche
140 X=RND(1)*32+1:LOCATE 1,25
150 PRINT TAB(X) CHR$(C) TAB(X+2) CHR$(C)      ' 2 etoiles
160 PRINT                                         ' scrolling
170 IF TEST((XV-1)*16+8,399-(YV)*16+8)<>0 THEN F=1:
    PRINT CHR$(7);:FOR TP=1 TO 1000:NEXT TP ' collision?
180 '
190 LOCATE AX,AY-1:PRINT CHR$(32)              ' effacement vehicule
200 LOCATE XV,YV:PRINT CHR$(231)              ' nouvelle position
210 IF F=1 THEN LOCATE 1,23:PRINT "Score:";L:GOTO 240
220 NEXT L
230 '
240 FOR TP=1 TO 3000:NEXT TP:GOTO 60

```

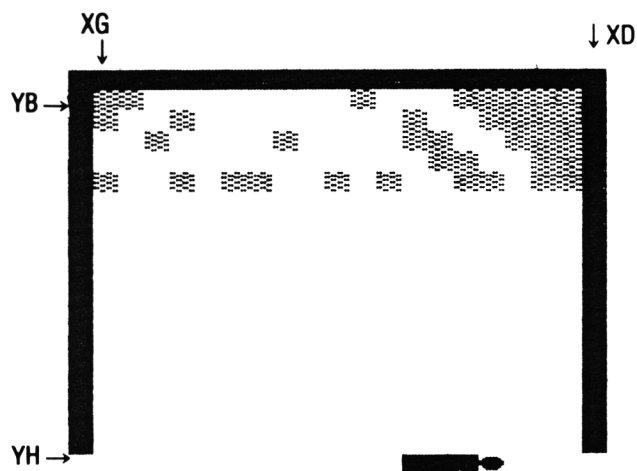


## CASSE-BRIQUES

---

Vous faites rebondir une balle à l'aide d'une raquette que vous déplacez avec les flèches gauche et droite. La balle détruit les briques sur son passage.

La fonction INKEY() permet d'obtenir un déplacement de la raquette sans délai de répétition (comme avec INKEY\$).



Fleches pour raquette

59 POINTS

```

10 '----- CASSE BRIQUES
20 MODE 1:PAPER 0:PEN 1
30 INK 0,26:INK 1,0:INK 2,6
40 INPUT "Niveau (1,2,3) (9 pour fin) ";NV
50 IF NV=9 THEN END
60 '----- dessin terrain
70 MODE 1
80 XD=18+NV*2      ' bornes
90 XG=2:YB=2:YH=21
100 LOCATE 1,23:PRINT "Fleches pour raquette"
110 FOR X=XG TO XD:LOCATE X,YB:PRINT CHR$(143):NEXT X
120 FOR Y=YB TO YH-1
130 LOCATE XG,Y:PRINT CHR$(143):LOCATE XD,Y:PRINT CHR$(143)
140 NEXT Y
150 PEN 2
160 FOR Y=YB+1 TO YB+5
170   FOR X=XG+1 TO XD-1:LOCATE X,Y:PRINT CHR$(143):NEXT X
180 NEXT Y
190 PEN 1
200 '---
210 SC=0              ' score
220 RQ$=CHR$(32)+CHR$(143)+CHR$(143)+CHR$(143)+CHR$(32)
230 DX=1:DY=-1        ' deplacement
240 XL=5+INT(RND(1)*5):YL=10    ' balle
250 XR=10:YR=YH        ' raquette
260 LOCATE XR-1,YR:PRINT RQ$
270 '----- deplacement balle
280 LOCATE XL,YL:PRINT CHR$(32)  ' effacement balle
290 XL=XL+DX:YL=YL+DY            ' nouvelle position
300 IF YL=YB+6 THEN YL=YL+INT(RND*2)*DY
310 '
320 IF TEST((XL-1)*16+1,399-(YL-1)*16)<>0 THEN SC=SC+1
330 LOCATE XL,YL:PRINT CHR$(231)
340 LOCATE XL,YL-1
350 IF YL>YB+1 THEN IF TEST((XL-1)*16+1,399-(YL-2)*16)<>0
      THEN PRINT CHR$(32):SC=SC+1
360 IF XL>XD-1 THEN DX=-DX      ' rebonds
370 IF YL<=YB+1 THEN DY=-DY
380 IF XL<=XG+1 THEN DX=-DX
390 '
400 IF XL>XR-2 AND XL<XR+4 AND YL>YR-2 THEN DY=-DY
410 '
420 IF YL>=YR OR (XD-XG-1)*5=SC THEN 490
430 '----- deplacement raquette
440 IF INKEY(1)=0 THEN IF XR<XD-3 THEN XR=XR+1
450 IF INKEY(8)=0 THEN IF XR>XG+1 THEN XR=XR-1
460 LOCATE XR-1,YR:PRINT RQ$
470 GOTO 280
480 '
490 LOCATE 30,23:PRINT SC;"POINTS"
500 FOR TP=1 TO 4000:NEXT TP
510 GOTO 20

```

## GLOUTONS

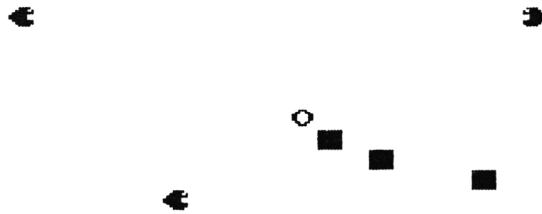
---

Trois gloutons vous poursuivent ; en vous déplaçant avec les flèches vous les attirez dans des pièges.

Les coordonnées des gloutons sont placées dans des tables XM() et YM(). Pour mettre à jour ces coordonnées nous calculons leurs positions par rapport à la vôtre.

Les dessins des gloutons sont définis dans deux instructions SYMBOL.

Grâce à la fonction INKEY() il n'y a pas de délai de répétition pour le déplacement avec les flèches.



### UTILISER LES FLECHES

```

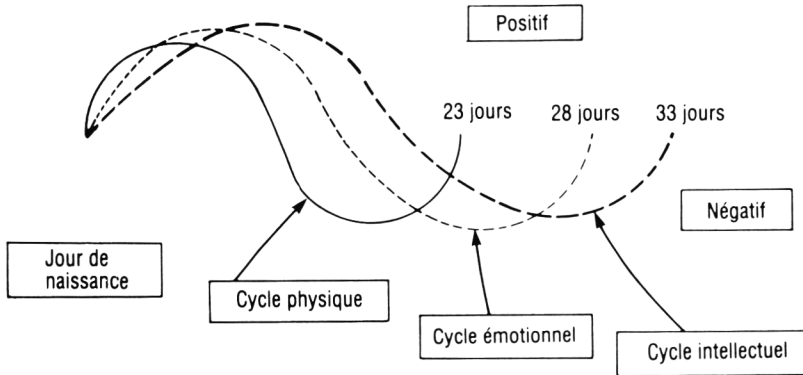
10 '   GLOUTONS
20 '
30 MODE 1:PAPER 0:PEN 1:INK 0,1:INK 1,24
40 LOCATE 1,25:PRINT "UTILISER LES FLECHES / FIN:ESC 2 FOIS"
50 SYMBOL AFTER 145
60 SYMBOL 145,120,188,254,63,63,254,252,120 ' glouton pos 1
70 SYMBOL 146,30,61,127,252,252,127,63,30 ' glouton pos 2
80 M=0 ' gloutons morts
90 XC=18:YC=12 ' vos coordonnees
100 LOCATE XC,YC:PRINT "O"
110 '-----
120 FOR J=1 TO 3
130 XM(J)=INT(RND(1)*38)+1:IF ABS(XM(J)-XC)<8 THEN 130
140 YM(J)=INT(RND(1)*23)+1:IF ABS(YM(J)-YC)<8 THEN 140
150 LOCATE XM(J),YM(J):IF XM(J)>XC THEN PRINT CHR$(145) ELSE PRI
NT CHR$(146)
160 LOCATE RND(1)*7+XC,RND(1)*7+YC:PRINT CHR$(143)
170 NEXT J
180 FOR TP=1 TO 1000:NEXT TP
190 '-----
200 FOR J=1 TO 3
210 LOCATE XC,YC:PRINT CHR$(32) ' effacement
220 IF INKEY(1)=0 THEN IF XC<38 THEN XC=XC+1
230 IF INKEY(8)=0 THEN IF XC>1 THEN XC=XC-1
240 IF INKEY(0)=0 THEN IF YC>1 THEN YC=YC-1
250 IF INKEY(2)=0 THEN IF YC<24 THEN YC=YC+1
260 LOCATE XC,YC:PRINT "O" ' nouvelle position
270 '
280 IF XM(J)=0 THEN 400
290 LOCATE XM(J),YM(J):PRINT CHR$(32) ' effacement glouton
300 XM(J)=XM(J)-(SGN(XM(J)-XC)) ' maj position glouton
310 YM(J)=YM(J)-(SGN(YM(J)-YC))
320 ' --test piege
330 IF TEST((XM(J)-1)*16+1,399-(YM(J)-1)*16)=0 THEN 360
340 XM(J)=0:YM(J)=0:M=M+1:GOTO 390
350 ' -- test devore
360 X1=XM(J):Y1=YM(J)
370 LOCATE X1,Y1:IF X1>XC
THEN PRINT CHR$(145) ELSE PRINT CHR$(146)
380 IF X1=XC AND Y1=YC THEN LOCATE 1,23:PRINT "PERDU":GOTO 430
390 IF M=3 THEN LOCATE 1,23:PRINT "GAGNE":GOTO 430
400 NEXT J
410 GOTO 200
420 '---
430 FOR TP=1 TO 3000:NEXT TP
440 GOTO 30

```



## BIORYTHMES

Il existerait chez l'homme des périodes "positives" et des périodes "négatives" réglées par des "horloges internes" indépendantes de l'environnement extérieur.

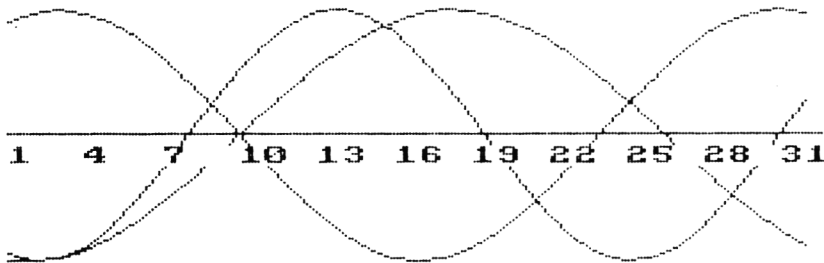


Il y aurait trois types de cycles :

- ☐ Cycle physique de 23 jours.
- ☐ Cycle émotionnel de 28 jours.
- ☐ Cycle intellectuel de 33 jours.

On pourrait ainsi connaître à l'avance les jours favorables.

Noir: physique  
Bleu: émotionnel  
Rouge: intellectuel



Jours vécus: 453  
Naiss: 5 4 1984 Bior: 7 1985

```

10 ' BIORYTHME
20 '
30 MODE 2:PAPER 0:PEN 1:INK 0,26:INK 1,0
40 DIM A(12),JM(12)
50 '
60 DATA 31,28.5,31,30,31,30,31,31,30,31,30,31
70 FOR M=1 TO 12
80 A(M)=A(M-1):READ X:A(M)=A(M)+X:JM(M)=X
90 NEXT M
100 CP=23:CE=28:CI=33 ' Cycles physique/Emot./Intel.
110 '
120 PRINT "NAISSANCE:":PRINT
130 INPUT "Jour,Mois,Annee (ex:5,4,1984) ";JN,MN,AN
140 IF AN<1900 OR AN>1999 THEN 130
150 J=JN:M=MN:A=AN:GOSUB 230:XV=JV
160 '
170 PRINT:PRINT "BIORYTHME":PRINT
180 INPUT "Mois,An (ex:7,1985) ";MB,AB
190 IF AB<1900 OR AB>1999 THEN 180
200 J=1:M=MB:A=AB:GOSUB 230:NJOUR=JV-XV
210 GOTO 270
220 '----- calendrier
230 N=365.25*(A-1901)+A(M)+J
240 JV=INT(N)
250 RETURN
260 '----- Affichage courbes
270 MODE 1:INK 0,26:INK 1,0:INK 2,20:INK 3,6
280 TAGOFF:LOCATE 1,2:PEN 1:PRINT "Noir: physique"
290 LOCATE 1,3:PEN 2:PRINT "Bleu: emotionnel"
300 LOCATE 1,4:PEN 3:PRINT "Rouge:intellectuel":PEN 1
310 PLOT 10,200:DRAW 10+16*31,0,1 ' Axe
320 C=CP:CL=1:GOSUB 420
330 C=CE:CL=2:GOSUB 420
340 C=CI:CL=3:GOSUB 420
350 FOR J=1 TO JM(MB)+1 STEP 3
360 MOVE 10+(J-2)*16,190:TAG:PRINT J;
370 NEXT J
380 TAGOFF:LOCATE 1,22:PRINT "Jours vecus:":NJOUR
390 LOCATE 1,23:PRINT "Naiss:":JN;MN;AN;"Bior:":MB;AB
400 END
410 '----- trace sinusoide
420 PLOT 10,SIN(NJOUR*2*PI/C)*100+200
430 FOR D=1 TO JM(MB)
440 P1=100*SIN((D+NJOUR)*2*PI/C)
450 X=10+D*16:DRAW X,200+P1,CL
460 NEXT D
470 RETURN

```

# ÉDUCATION | 4

## PHRASE A COMPLÉTER

---

Une phrase incomplète est affichée ; l'élève doit la compléter.

Dans chaque ligne de DATA est définie la phrase incomplète suivie du mot manquant. Si les phrases contiennent des virgules ou des caractères spéciaux, elles doivent être placées entre des guillemets.

```

10 ' COMPLETER UNE PHRASE
20 '
30 MODE 2:PAPER 0:PEN 1:INK 0,0:INK 1,26
40 LOCATE 1,20:PRINT "COMPLETER LA PHRASE"
50 DATA IL ETAIT UNE **** DANS L'OUEST,FOIS
60 DATA LUNDI MARDI ***** JEUDI,MERCREDI
70 DATA *
80 XA=1:YA=10 ' Affichage
90 '
100 READ LIG$:IF LIG$="*" THEN END
110 READ MOT$
120 '
130 P=INSTR(LIG$,"*") ' Recherche *
140 '
150 LOCATE XA,YA:PRINT SPC(40)
160 LOCATE XA,YA:PRINT LIG$
170 PL=1
180 '
190 LOCATE XA+P-1,YA:PRINT CHR$(143) ' curseur
200 C$=INKEY$:IF LEN(C$)<>0 THEN 230
210 GOTO 190
220 '--
230 C$=UPPER$(C$)
240 IF C$<>MID$(MOT$,PL,1) THEN PRINT CHR$(7):GOTO 190
250 LOCATE XA+P-1,YA:PRINT C$
260 P=P+1
270 PL=PL+1:IF PL>LEN(MOT$) THEN 100
280 GOTO 190

```

IL ETAIT UNE \*\*\* DANS L'OUEST

COMPLETER LA PHRASE

## DICTÉE

---

Un texte avec des mots incomplets est affiché. L'élève doit compléter les mots.

Dans chaque ligne de DATA est définie la ligne à compléter suivie du caractère du mot incomplet.

```

10 ' DICTEE
20 '
30 MODE 1
40 PAPER 0:PEN 1:INK 1,0:INK 0,26
50 '
60 DATA "QU'EST CE QUE SIG*IFIE APPRIVOISER" ,N
70 DATA "DIT LE P*TIT PRINCE" ,E
80 DATA "C'EST UNE CHOSE TROP OUBLIEE," ,*
90 DATA "D*T LE RENARD" ,I
100 DATA "CA SI*NIFIE CREER DES LIENS." ,G
110 DATA *
120 '----- LECTURE DATAS DANS TABLE TXT$( )
130 DIM TXT$(100),C$(100)
140 FOR L=1 TO 100
150 READ TXT$(L):IF TXT$(L)="*" THEN NL=L-1:GOTO 190
160 READ C$(L)
170 NEXT L
180 '----- AFFICHAGE TEXTE
190 CLS
200 LOCATE 1,20:PRINT "TU DOIS COMPLETER LES MOTS "
210 FOR L=1 TO NL
220 LOCATE 1,L:PRINT TXT$(L)
230 NEXT L
240 '-----
250 FOR L=1 TO NL
260 P=INSTR(TXT$(L),"*") ' position lettre manquante
270 IF P=0 THEN 320
280 LOCATE P,L:PRINT CHR$(143)
290 C$=INKEY$:IF LEN(C$)=0 THEN 290
300 IF UPPER$(C$)<>UPPER$(C$(L)) THEN PRINT CHR$(7);:GOTO 290
310 LOCATE P,L:PRINT C$
320 NEXT L
330 END

```

QU'EST CE QUE SIG\*IFIE APPRIVOISER  
 DIT LE P\*TIT PRINCE  
 C'EST UNE CHOSE TROP OUBLIEE,  
 D\*T LE RENARD  
 CA SI\*NIFIE CREER DES LIENS.

## MISE EN ORDRE D'UN MOT

---

Un mot est affiché dans le désordre ; l'élève doit l'écrire en ordre.

Les mots sont présentés dans l'ordre des DATA. Pour un affichage aléatoire, ajoutez :

```

80 NM=4          ' nombre de mots
81 X=INT(RND(1)*NM)
82 RESTORE:FOR J=1 TO X:READ MOT$:NEXT J
83 READ MOT$

10 '   MISE EN ORDRE D'UN MOT
20 '
30 MODE 1:PAPER 0:PEN 1
40 DIM T(12)     ' témoin
50 DATA AMSTRAD,ECOLE,MAISON,BASIC
60 DATA *
70 '
80 READ MOT$
90 IF MOT$="*" THEN END
100 L=LEN(MOT$)
110 CLS
120 FOR P=1 TO 12:T(P)=0:NEXT P
130 FOR N=1 TO L
140   P=INT(RND(1)*L)+1:IF T(P)=1 THEN 140
150   PRINT MID$(MOT$,P,1);
160   T(P)=1
170 NEXT N
180 '
190 PRINT:PRINT
200 INPUT "MOT";M$
210 IF UPPER$(M$)=UPPER$(MOT$) THEN PRINT "BRAVO"
220 IF UPPER$(M$)<>UPPER$(MOT$) THEN PRINT "NON,C'EST:";MOT$
230 FOR TP=1 TO 2000:NEXT TP
240 GOTO 80

```

DTRSAAM

MOT? AMSTRAD  
BRAVO

## MISE EN ORDRE D'UNE PHRASE

---

Une phrase est affichée dans le désordre ; l'élève doit l'écrire dans l'ordre.

La phrase est d'abord découpée en recherchant les espaces entre les mots. Les mots sont stockés dans une table M\$( ) puis affichés dans le désordre.

```

10 ' REMISE EN ORDRE D'UNE PHRASE
20 '
30 MODE 2
40 CE=1:CF=0 ' ecriture et fond
50 INK 0,26:INK 1,0:PAPER 0:PEN 1
60 '
70 READ L$:IF L$="*" THEN END ' lecture d'une ligne de DATA
80 LIG$=L$
90 GOSUB 230
100 '----- Affichage dans le desordre
110 CLS
120 FOR F=1 TO N
130   P=INT(RND(1)*N)+1 ' position au hasard
140   IF LEN(M$(P))=0 THEN 130
150   LOCATE 1,F:PRINT M$(P):M$(P)=" " ' pas d'espace entre "
160 NEXT F
170 '
180 LOCATE 1,13:INPUT "Frappez la phrase en ordre ";PHR$
190 IF UPPER$(PHR$)=UPPER$(L$) THEN PRINT "OK " ELSE PRINT "Non
c'est: ";L$
200 FOR TP=1 TO 2000:NEXT TP
210 GOTO 70
220 '----- DECOUPAGE PHRASE
230 N=0:Y$="" ' pas d'espace entre "
240 P=INSTR(LIG$,CHR$(32)) ' recherche espace (chr$(32))
250 IF P=0 THEN N=N+1:M$(N)=LIG$:RETURN
260 N=N+1:M$(N)=LEFT$(LIG$,P-1)
270 LIG$=RIGHT$(LIG$,LEN(LIG$)-P)
280 GOTO 240
290 '----
300 DATA LE PETIT CHAT RONRONNE
310 DATA JE M'APPELLE AMSTRAD
320 DATA *
```

```

PETIT
CHAT
LE
RONRONNE
```

```

Frappez la phrase en ordre ? LE PETIT CHAT RONRONNE
OK
```

## LISTE A COMPLÉTER

---

L'élève doit compléter une liste.

```

10 '   LISTE A COMPLETER
20 '
30 MODE 2:INK 0,1:INK 1,24:PAPER 0:PEN 1
40 DIM AN$(100),C$(100)
50 '
60 DATA LE CHIEN,ABOIE
70 DATA LE CHAT,MIAULE
80 DATA LE MOUTON,BELE
90 DATA LE CHEVAL,HENNIT
100 '
110 DATA *,*
120 '-----lecture DATAS dans tables
130 FOR J=1 TO 100
140 READ AN$(J),C$(J)
150 IF AN$(J)="*" THEN NM=J-1:GOTO 190
160 NEXT J
170 STOP
180 '----- Affichage
190 CLS
200 LOCATE 10,22:PRINT "Tu dois completer la liste"
210 LOCATE 10,23:PRINT "ex: LE CHIEN ABOIE"
220 FOR L=1 TO NM
230 LOCATE 2,L:PRINT AN$(L)
240 NEXT L
250 '
260 FOR L=1 TO NM
270 LOCATE 13,L:INPUT LIG$
280 IF UPPER$(C$(L))<>UPPER$(LIG$) THEN PRINT CHR$(7):GOTO 270
290 NEXT L

```

```

LE CHIEN    ? ABOIE
LE CHAT     ? MIAULE
LE MOUTON   ? BELE
LE CHEVAL   ?

```

## QUESTIONNAIRE

---

Chaque ligne de DATA contient une question et l'ensemble des réponses possibles. Grâce à la fonction INSTR on vérifie que la réponse de l'élève appartient à l'ensemble des réponses possibles.

```

10 '      QUESTIONNAIRE
20 '
30 MODE 2:PAPER 0:PEN 1
40 '
50 '      Question                      Reponses possibles
60 '
70 DATA "Donnez un synonyme de DELAI","SURSIS REPIT PROLONGATION
"
80 DATA "Capitale de l'ESPAGNE",      "MADRID"
90 DATA "Capitale de la FRANCE ",    "PARIS"
100 DATA *
110 '
120 '
130 READ Q$:IF Q$="*" THEN END
140 READ RP$
150 PRINT
160 PRINT Q$              ' affichage question
170 INPUT "Votre reponse ";R$
180 P=INSTR(UPPER$(RP$),UPPER$(R$))
190 IF P<>0 THEN PRINT:PRINT "OK" ELSE PRINT RP$
200 GOTO 130

```

Donnez un synonyme de DELAI  
 Votre reponse ? REPIT

OK



## QUID

---

Un ensemble de lignes d'information sont stockées en DATA. Le programme donne toutes les lignes contenant un mot clé cherché.

```

10 '      QUID
20 '
30 MODE 2:PAPER 0:PEN 1:INK 0,0:INK 1,26
40 '
50 DATA TOUR DE FRANCE 1979 HINAULT
60 DATA TOUR D'ITALIE 1979 SARONNI
70 DATA CHAMPIONNAT DU MONDE 1979 RASS
80 DATA TOUR DE FRANCE 1980 HINAULT
90 DATA TOUR D'ITALIE 1980 HINAULT
100 DATA CHAMPIONNAT DU MONDE 1980 HINAULT
110 DATA TOUR DE FRANCE 1981 HINAULT
120 DATA TOUR D'ITALIE 1981 BATAGLIN
130 '
140 DATA *
150 '
160 PRINT:INPUT "Mot cle (ou FIN)";M$
170 M$=UPPER$(M$):IF M$="FIN" THEN END
180 '
190 RESTORE
200 READ LIG$
210 IF LIG$="*" THEN 160
220 P=INSTR(UPPER$(LIG$),UPPER$(M$))
230 IF P<>0 THEN PRINT LIG$
240 GOTO 200

```

```

Mot cle ? 1980
TOUR DE FRANCE 1980 HINAULT
TOUR D'ITALIE 1980 HINAULT
CHAMPIONNAT DU MONDE 1980 HINAULT

```

```

Mot cle ? HINAULT
TOUR DE FRANCE 1979 HINAULT
TOUR DE FRANCE 1980 HINAULT
TOUR D'ITALIE 1980 HINAULT
CHAMPIONNAT DU MONDE 1980 HINAULT
TOUR DE FRANCE 1981 HINAULT

```

```

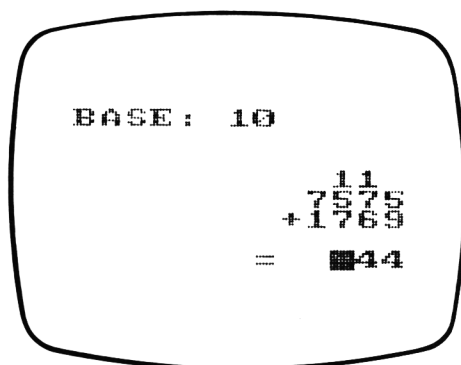
Mot cle ? FRANCE
TOUR DE FRANCE 1979 HINAULT
TOUR DE FRANCE 1980 HINAULT
TOUR DE FRANCE 1981 HINAULT

```

## ADDITION EN BASE B

---

Deux nombres choisis au hasard sont affichés ; l'élève doit donner pour chaque rang le total (sans la retenue) puis la retenue pour le rang suivant. La base et le nombre de chiffres peuvent être choisis.



```

10 ' ADDITION EN BASE B
20 '
30 MODE 1:PAPER 0:PEN 1:INK 0,26:INK 1,0
40 B=10 ' base
50 NCH=4 ' 4 chiffres
60 XL=10:YL=10 ' AFFICHAGE
70 '----- Choix au hasard
80 N1$="":N2$="" ' pas d'espace entre ""
90 FOR J=1 TO NCH
100 X=INT(RND(1)*B)
110 Y=INT(RND(1)*B)
120 N1$=N1$+MID$(STR$(X),2,1)
130 N2$=N2$+MID$(STR$(Y),2,1)
140 NEXT J
150 '
160 CLS
170 LOCATE 1,3:PRINT "BASE: ";B
180 LOCATE XL,YL-3:PRINT N1$
190 LOCATE XL-1,YL-2:PRINT "+ ";N2$
200 '
210 LOCATE XL-2,YL:PRINT "="
220 '----- Saisie resultats
230 P=LEN(N1$)
240 RET=0 ' retenue
250 '
260 IF P=0 THEN TR=RET:GOTO 290
270 TR=VAL(MID$(N1$,P,1))+VAL(MID$(N2$,P,1))+RET
280 '
290 RET=INT(TR/B)
300 RS=TR-RET*B
310 '
320 LOCATE 1,18:PRINT "TOTAL POUR LA COLONNE(SANS RETENUE)"
330 LOCATE XL+P-1,YL:PRINT CHR$(143)
340 LOCATE XL+P-1,YL

```

```

350 R$=INKEY$:IF LEN(R$)=0 THEN 350
360 IF RS<>VAL(R$) THEN PRINT CHR$(7):GOTO 330
370 PRINT R$
380 '-----Retenue de rang N+1
390 IF RET=0 THEN 480
400 '
410 LOCATE 1,18:PRINT "ENTRE LA RETENUE POUR LA COLONNE" SPC(4)
420 LOCATE XL+P-2,YL-4:PRINT CHR$(143)
430 LOCATE XL+P-2,YL-4
440 R$=INKEY$:IF LEN(R$)=0 THEN 440
450 IF VAL(R$)<>RET THEN PRINT CHR$(7):GOTO 420
460 PRINT R$
470 '
480 IF P=0 THEN 530
490 IF P=1 AND RET=0 THEN 530
500 P=P-1
510 GOTO 260
520 '
530 FOR TP=1 TO 3000:NEXT TP
540 GOTO 80

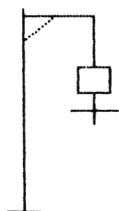
```

## JEU DU PENDU

---

Vous devez retrouver un mot en proposant des lettres. Vous avez droit à sept erreurs.

.BCDEFGH.JKL..OPQR.TUVW.YZ



MAIS.N

Quelle lettre ?

A chaque fois qu'une lettre est utilisée, elle est remplacée dans l'alphabet par le caractère ".". Les mots sont présentés dans l'ordre des DATA ; on pourra les choisir au hasard en ajoutant :

```

124 NM=5 ' nombre de mots
125 X=INT(RND(1)*NM)
126 RESTORE:FOR J=1 TO X:READ MOT$:NEXT J

10 ' JEU DU PENDU
20 '
30 DATA MAISON,SAPIN,VELO,PATIN
40 DATA AMSTRAD
50 DATA *
60 XM=12:YM=10 ' affichage mot
70 XA=2:YA=2 ' affichage alphabet
80 '-
90 MODE 1:PAPER 0:PEN 1:INK 0,26:INK 1,0
100 '--- affichage alphabet
110 FOR J=1 TO 26:LOCATE XA+J,YA:PRINT CHR$(64+J):NEXT J
120 '
130 READ MOT$:MOT$=UPPER$(MOT$):IF MOT$="*" THEN END
140 L=LEN(MOT$)
150 'BPL: bien places/PD: coups perdus /LU$: lettres deja utilisees
160 BPL=0:PD=0:LU$="" ' pas d'espace entre ""
170 LOCATE XM,YM:PRINT STRING$(L,".")
180 '
190 LOCATE 2,23:INPUT "Quelle lettre ";L$:L$=UPPER$(L$)
200 IF LEN(L$)=0 THEN 190
210 P=INSTR(LU$,L$):IF P>0 THEN GOSUB 380:GOTO 310
220 '----- bonne lettre?
230 IF ASC(L$)<65 THEN 190
240 B=0
250 FOR P=1 TO L
260 IF L$=MID$(MOT$,P,1)
    THEN B=1:LU$=LU$+L$:BPL=BPL+1:GOSUB 340:GOSUB 360
270 NEXT P
280 IF B=0 THEN GOSUB 380:GOSUB 360
290 IF BPL=L THEN LOCATE 12,15:PRINT "BRAVO":FOR TP=1 TO 2000:NEXT TP:GOTO 90
300 '
310 IF PD>=8 THEN LOCATE 12,15:PRINT "PERDU .. LE MOT ETAIT: ";MOT$:FOR TP=1 TO 2000:NEXT TP:GOTO 90
320 GOTO 190
330 '----- affichage mot
340 LOCATE XM+P-1,YM:PRINT L$:RETURN
350 '----- mise a jour alphabet
360 LOCATE XA+ASC(L$)-64,YA:PRINT ".":RETURN
370 '----- coup perdu
380 PD=PD+1:PRINT CHR$(7)
390 ON PD GOSUB 410,420,430,440,450,460,470,490:RETURN
400 '
410 PLOT 20,130,1:DRAW 40,130,1:RETURN
420 PLOT 30,130,1:DRAW 30,290,1:RETURN
430 PLOT 30,285,1:DRAW 75,285,1:RETURN

```

```

440 PLOT 30,265:DRAW 50,285:RETURN
450 PLOT 75,285:DRAW 75,245:RETURN
460 PLOT 65,245:DRAW 20,0:DRAW 0,-20:DRAW -20,0:DRAW 0,20:RET
URN
470 PLOT 75,225:DRAW 75,200
480 PLOT 60,210:DRAW 90,210:RETURN
490 PLOT 75,200:DRAW 60,170:PLOT 75,200:DRAW 90,170:RETURN

```

## DICTIONNAIRE FRANÇAIS/ANGLAIS \_\_\_\_\_

Un dictionnaire français/anglais est stocké dans deux tables FR\$() et ANG\$(). Les tables sont sauvegardées sur cassette ou disquette.

| FR\$() |   | ANG\$() |
|--------|---|---------|
| LIVRE  | → | BOOK    |
| CHAT   | → | CAT     |
| CHIEN  | → | DOG     |
| MAISON | → | HOUSE   |

Un mode "recherche" donne la traduction d'un mot cherché. Vous pouvez placer dans chaque ligne des tables la liste des synonymes d'un mot.

Un mode "interrogation" propose un mot au hasard et demande la traduction. Ce mode pourrait être amélioré en enregistrant dans une table les mots mal connus de l'élève afin de les proposer en priorité.

Mode ? L

### LISTE DES MOTS

|             |                      |
|-------------|----------------------|
| LIVRE       | BOOK                 |
| CHIEN       | DOG                  |
| TRAVAIL     | WORK                 |
| CHAT        | CAT                  |
| BUREAU      | DESK                 |
| JAMBE       | LEG                  |
| BRAS        | ARM                  |
| CALCULATEUR | COMPUTER             |
| TETE        | HEAD                 |
| RECHERCHE   | SEEK/SEARCH/RESEARCH |
| ARRETER     | TO STOP/TO TERMINATE |
| CHAMBRE     | ROOM                 |

APPUYER SUR ENTER ? \*Break\*

Mode ? INT

TRAVAIL Traduction (ENTER pour fin) ? WORK

OK

RECHERCHE Traduction (ENTER pour fin) ? X

Non c'est: SEEK/SEARCH/RESEARCH

```

10 '          DICTIONNAIRE ANGLAIS/FRANCAIS
20 '
30 MODE 2:PAPER 0:PEN 1:INK 0,1:INK 1,24
40 '
50 DIM FR$(1000),ANG$(1000)
60 NFICH=0
70 '
80 INPUT "Nouveau fichier (O/N) ";R$
90 IF R$<>"O" AND R$<>"o" THEN GOSUB 390
100 '
110 CLS:PRINT "Modes:":PRINT
120 PRINT "C:  Creation"
130 PRINT "R:  Recherche"
140 PRINT "L:  Liste "
150 PRINT "INT: Interrogation"
160 PRINT "FIN: Sauvegarde "
170 PRINT:INPUT "Mode ";M$
180 M$=UPPER$(M$)
190 IF M$="C" THEN GOSUB 260
200 IF M$="FIN" THEN GOSUB 480:END
210 IF M$="L" THEN GOSUB 560
220 IF M$="R" THEN GOSUB 630
230 IF M$="INT" THEN GOSUB 720
240 GOTO 110
250 ' ===== CREATION
260 PRINT:INPUT "Mot FRANCAIS ( ENTER pour fin) ";X$
270 IF LEN(X$)=0 THEN RETURN ' fin de mode?
280 FOR RANG=1 TO NFICH
290 IF X$=FR$(RANG) THEN PRINT "Existe deja":GOTO 260
300 NEXT RANG
310 NFICH=NFICH+1
320 RANG=NFICH
330 FR$(RANG)=X$
340 '
350 INPUT "Mot ANGLAIS ";Y$
360 ANG$(RANG)=Y$
370 GOTO 260
380 '===== LECTURE FICHIER
390 OPENIN "DICO"
400 INPUT #9,NFICH
410 FOR J=1 TO NFICH
420 INPUT #9,FR$(J),ANG$(J)

```

```

430 NEXT J
440 PRINT:PRINT NFICH;"MOTS":PRINT
450 FOR TP=1 TO 2000:NEXT TP
460 RETURN
470 '===== SAUVEGARDE FICHER
480 OPENOUT "DICO"
490 PRINT #9,NFICH
500 FOR J=1 TO NFICH
510   PRINT #9,FR$(J)
520   PRINT #9,ANG$(J)
530 NEXT J
540 RETURN
550 '===== LISTE
560 PRINT:PRINT "LISTE DES MOTS":PRINT
570 FOR J=1 TO NFICH
580   PRINT FR$(J),ANG$(J)
590 NEXT J
600 PRINT:INPUT "APPUYER SUR ENTER ";X$
610 RETURN
620 '===== RECHERCHE
630 PRINT:INPUT "QUEL MOT (ou ENTER) ";X$
640 IF LEN(X$)=0 THEN RETURN
650 '
660 FOR J=1 TO NFICH
670   P=INSTR(UPPER$(FR$(J)),UPPER$(X$))
680   IF P<>0 THEN PRINT:PRINT FR$(J),ANG$(J)
690 NEXT J
700 GOTO 630
710 '===== INTERROGATION ALEATOIRE
720 X=INT(RND(1)*NFICH)+1
730 MOT$=FR$(X)
740 PRINT
750 PRINT MOT$;SPC(1);:INPUT "Traduction (ENTER pour fin) ";X$
760 IF LEN(X$)=0 THEN RETURN
770 P=INSTR(UPPER$(ANG$(X)),UPPER$(X$))
780 IF P<>0 THEN
790   PRINT:PRINT "OK" ELSE PRINT:PRINT "Non c'est: ";ANG$(X)
800 '----
810 '
820 ' Les recherches peuvent s'effectuer en majuscules ou
830 ' minuscules.
840 ' Pour eviter d'entrer en minuscules des mots
850 ' deja dans le fichier en majuscules,faire:
860 '
870 ' 290 IF UPPER$(X$)=UPPER$(FR$(RANG))
880 '     THEN PRINT "Existe deja":GOTO 260
890 '
890 ' On pourra egalement 'normaliser' les mots
900 ' en majuscules avant de les placer dans le fichier:
910 ' 330 FR$(RANG)=UPPER$(X$)

```





# CONSEILS DE LECTURE

Pour approfondir vos connaissances en BASIC Amstrad et mieux connaître le système des CPC 464, 664 et 6128, P.S.I. vous propose une palette d'ouvrages utiles.

## POUR MAITRISER LE BASIC AMSTRAD

- **BASIC Amstrad, CPC 464, 664 et 6128 - 1. Méthodes pratiques**, par Jacques Boisgontier et Bruno Césard (Editions du P.S.I.) :  
Pour ceux qui ont déjà pratiqué un BASIC, voici un ouvrage de perfectionnement au BASIC Amstrad. Un chapitre sur le CP/M 2.2 et le CP/M Plus donne les principales commandes systèmes.
- **BASIC plus 80 routines sur Amstrad**, par Michel Martin (Editions du P.S.I.) :  
Pour pousser votre Amstrad au maximum de ces capacités : 80 routines de simulation d'instructions qui n'existent pas en BASIC Amstrad.
- **Périphériques et gestion de fichiers sur Amstrad, CPC 464, 664 et 6128**, par Daniel-Jean David (Editions du P.S.I.) :  
Cet ouvrage s'adresse aux lecteurs, déjà initiés à l'Amstrad, souhaitant programmer tous les périphériques des CPC 464, 664 et 6128 et programmer des applications utilisant des fichiers sur cassette ou sur disquette.

## POUR MIEUX CONNAITRE LE SYSTEME DES CPC

- **Clefs pour Amstrad – 1. système de base**, par Daniel Martin (Editions du P.S.I.) :  
Mémento présentant synthétiquement le jeu d'instructions du Z80, les points d'entrée des routines système, les connecteurs et brochages, etc.  
Le livre de chevet du programmeur sur Amstrad.
- **Clefs pour Amstrad – 2. système disque**, par Daniel Martin et Philippe Jadoul (Editions du P.S.I.) :  
Ce deuxième tome consacré au système disque présente les points d'entrée des routines disque, les blocs de contrôle, la programmation et les brochages des circuits spécialisés... La deuxième partie du livre est aussi destinée aux possesseurs d'Amstrad 8256.

- **CP/M plus sur Amstrad**, par Yvon Dargery (Editions du P.S.I.) :  
Toutes les commandes CP/M et CP/M Plus pour maîtriser le système des 6128 et 8256 : un ouvrage de référence illustré par de nombreux programmes.
- **Le livre de l'Amstrad – Tome 1**, par Daniel Martin et Philippe Jadoul (Editions du P.S.I.)  
Ce livre, destiné aux programmeurs des CPC 464, 664 et 6128, donne une étude complète de tous les circuits internes, et analyse la structure interne du BASIC. Vous y trouverez, en outre, une étude complète des RSX, et des programmes de scrolling, de traçage de rectangles, de coloriage de surface et de manipulation vectorielle.

**Votre avis nous intéresse**

Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques sur le présent ouvrage.

— *Ce livre vous donne-t-il toute satisfaction ?*

.....

.....

.....

.....

— *Y a-t-il un aspect du problème que vous auriez aimé voir abordé ?*

.....

.....

.....

.....

Si vous souhaitez des éclaircissements techniques, écrivez-nous, nous ne manquerons pas de vous répondre directement.

**Où avez-vous acheté ce livre ?**

- |                                     |                                         |                                 |
|-------------------------------------|-----------------------------------------|---------------------------------|
| <input type="checkbox"/> cadeau     | <input type="checkbox"/> librairie      | <input type="checkbox"/> autres |
| <input type="checkbox"/> exposition | <input type="checkbox"/> boutique micro |                                 |

**Comment en avez-vous eu connaissance ?**

- |                                     |                                            |                                 |
|-------------------------------------|--------------------------------------------|---------------------------------|
| <input type="checkbox"/> publicité  | <input type="checkbox"/> catalogue         | <input type="checkbox"/> autres |
| <input type="checkbox"/> exposition | <input type="checkbox"/> conseils d'un ami |                                 |

**Avez-vous déjà acquis des livres P.S.I. ?**

Lesquels ? .....

qu'en pensez-vous ? .....

.....

Nom ..... Prénom ..... Age .....

Adresse .....

Profession .....

Centre d'intérêt .....

## **CATALOGUE GRATUIT**

Vous pouvez obtenir un catalogue complet des ouvrages PSI, sur simple demande, ou en retournant cette page remplie à votre libraire, à votre boutique micro ou aux

**Editions du PSI**  
**BP 86**  
**77402 Lagny-sur-Marne Cedex**



# **BASIC AMSTRAD CPC 464, 664 ET 6128**

## **2. PROGRAMMES ET FICHIERS**

Ce deuxième tome de "BASIC Amstrad" vous propose de mettre en pratique tout ce que vous savez du BASIC des CPC 464, 664 et 6128, grâce à :

- des programmes graphiques qui vous feront apprécier la qualité de la haute résolution ainsi que l'étendue de la gamme de couleurs de l'Amstrad ;
- des programmes de gestion de fichiers qui vous permettront de réaliser un mailing, d'éditer des étiquettes, de créer des histogrammes, etc. ;
- des jeux qui vous étonneront par leur rapidité d'exécution ;
- des programmes éducatifs, enfin, qui feront de votre Amstrad un répétiteur patient et attentif.

Tous les programmes de ce livre fonctionnent sur les trois modèles Amstrad : CPC 464, CPC 664 et CPC 6128.



9 782865 952496

**ÉDITIONS DU P.S.I.  
BP 86 - 77402 LAGNY S/MARNE CEDEX-FRANCE**

**ISBN : 2 86595-249 5**

**95 F.F.**

# **BASIC AMSTRAD CPC 464, 664 ET 6128 2. PROGRAMMES ET FICHIERS**



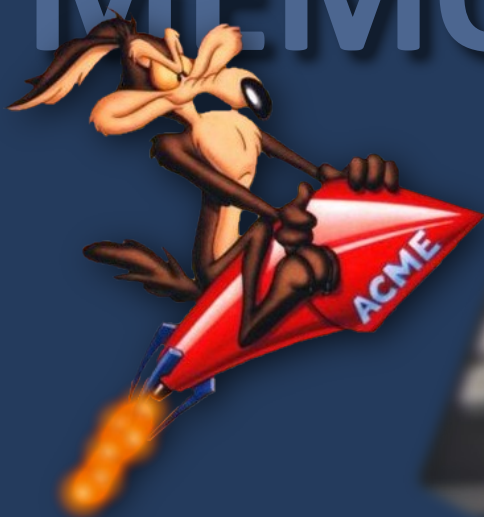


Document **numérisé**  
avec amour par :

**AMSTRAD**

CPC 

MÉMOIRE ÉCRITE



<https://acpc.me/>