

Jacky ZDUNEK
Christian JACQUIER

**EXPLOITEZ
VOTRE**

AMSTRAD

Jeux

Gestion

Education

Domestique

**Impression
Documents**

VISM'EDIT

EXPLOITEZ VOTRE AMSTRAD

J. ZDUNEK

CH. JACQUIER

COPYRIGHT © 1985, VISMEDIT

La loi du 11 mars 1985 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 ayant cause, est illicite" (alinéa 1^{er} de l'article 40). Cette représentation ou reproduction, par quelque procédé que ce soit constituerait donc une infraction sanctionnée par les articles 425 et suivants du code pénal.

TABLE DES MATIERES

PREFACE INTRODUCTION

CHAPITRE 1 : Le langage Basic

- 1 . L'ordinateur.....
- 2 . Le langage Basic.....
- 3 . La notion de programme.....
- 4 . Quelques instructions Basic.....
- 5 . Test des connaissances.....

CHAPITRE 2 : Les jeux.

- 1 . Le Loto.....
- 2 . Deviner des chiffres.....
- 3 . Le pendu.....

CHAPITRE 3 : Mathématiques

- 1 . Manipuler des degrés.....
- 2 . Résoudre des équations.....
- 3 . Statistiques.....

CHAPITRE 4 : Graphismes

- 1 . L'histogramme.....
- 2 . Une table à dessin.....

CHAPITRE 5 : la gestion de fichiers

- 1 . Les notions de données et de fichiers.....
- 2 . La gestion des fichiers.....
- 3 . Une application des fichiers.....
- 4 . Programme AGENDA.....
- 5 . Budget familial.....

CHAPITRE 6 : Comptabilité

- 1 . Facturation.....
- 2 . Comptabilité générale.....
- 3 . buletin de paye.....

PREFACE

Alors que l'informatique, après avoir envahie la quasi totalité des secteurs professionnels, entre dorénavant, sur l'initiative de l'Education Nationale, dans la formation des plus jeunes, l'ordinateur reste toujours, pour une grande majorité d'individus, un appareil complexe et mystérieux.

Pour ceux-ci, ces machines sont capables d'accomplir les meilleures prouesses, mais peuvent aussi se rendre coupables des pires méfaits, images d'ailleurs largement véhiculées par le cinéma de science fiction.

Pourtant l'apparition sur le marché, depuis quelques années, d'ordinateurs à faibles prix, permet aux non-initiés d'accéder aux mécanismes de l'informatique.

C'est une chance à ne pas laisser passer si l'on veut avoir l'espoir de comprendre un peu mieux la société moderne, et ne pas devenir, surtout face aux nouvelles générations, les ignorants de demain.

Imaginez vous donc aujourd'hui ne sachant ni lire ni écrire.

CH. J.

I N T R O D U C T I O N

Il n'était pas question pour nous de faire un ouvrage de plus sur l'apprentissage du BASIC. Il en existe tant, dont certains sont excellents. Nous avons plutôt tenté d'expliquer certaines instructions BASIC de l'AMSTRAD, au travers de programmes que nous avons écrits. Nous pensons en effet que programmer est la meilleure manière d'apprendre à programmer (c'est une évidence qui est souvent perdue de vue).

Vous disposer, avec ce livre, d'une cassette contenant tout les programmes présents dans cet ouvrage. Autant vous dire qu'ils fonctionnent donc tous sur votre AMSTRAD.

Nous avons essayé d'aborder tous les domaines de l'AMSTRAD (sauf le son qui fera l'objet d'une publication distincte). Vous trouverez des jeux, des programmes mathématiques, et une approche de la haute résolution graphique.

C'est pourtant sur la gestion des fichiers que nous avons mis l'accent, car il s'agit pour nous de la partie la plus intéressante d'un micro-ordinateur, et vous trouverez deux programmes la concernant qui méritent d'être améliorés pour pouvoir exploiter au mieux les capacités de votre AMSTRAD.

Enfin, nous avons terminé cet ouvrage avec une introduction à la comptabilité sur ordinateur, et c'est un véritable outil professionnel que nous mettons à votre disposition. A vous de les adapter à vos besoins, et vous aurez ainsi largement remboursé le prix de cet ouvrage.

J. Z. & CH. J.

CHAPITRE 1

Ce chapitre est particulièrement destiné à ceux qui n'auraient aucune connaissance en programmation, ou qui utiliseraient pour la première fois un ordinateur. Nous ne saurions trop cependant leur conseiller d'acquérir un ouvrage d'initiation au BASIC.

Les autres peuvent éluder cette introduction et se pencher sur les chapitres suivants, dédiés spécialement à l'AMSTRAD.

1) L'ordinateur

Votre AMSTRAD contient une mémoire, dont une partie recèle un certain nombre d'informations nécessaires à son fonctionnement. On y trouve par exemple le langage BASIC. Cette partie de mémoire s'appelle la ROM et si on peut la lire, il est impossible d'y écrire autre chose.

L'autre partie de mémoire, la RAM, vous est réservée et contiendra vos programmes. Cette mémoire est volatile et son contenu est perdu lorsque l'on éteint la machine.

Pour transmettre ces informations, le micro-ordinateur utilise ce que l'on appelle les périphériques.

Sur votre AMSTRAD vous trouvez :

- le clavier qui vous sert à fournir des informations à l'appareil,
- le moniteur vidéo et l'imprimante dont le rôle est d'afficher les informations issues de l'ordinateur,
- le magnétophone qui conservera sur une bande magnétique les informations contenues en ram que vous désirez garder après avoir éteint l'appareil

2) Le langage BASIC

Pour communiquer avec votre ordinateur vous devez utiliser un langage approprié.

En effet si vous tentez de converser avec lui dans votre langue maternelle, il y'a peu de chance pour que vous obteniez un résultat quelconque. Vous devrez donc utiliser le BASIC qui est un langage informatique quasi universel (mais il en existe beaucoup d'autres : Pascal, Logo, etc...), et qui,


```
A = 3 : B = 4 : PRINT "3 x 4 = ";A * B <ENTER>
```

Vous voyez que c'est déjà beaucoup mieux ainsi. Il faut cependant remarquer que :

- l'instruction PRINT affiche intégralement ce qui est entre les guillemets sans chercher à l'interpréter,
- d'autre part, le point virgule après les guillemets permet de coller le résultat de A * B au texte.

Essayez donc de changer le point virgule par une virgule simple ou par deux points dans la ligne précédente et vous verrez tout de suite la différence

Cependant un tel système est un peu lourd d'emploi pour multiplier deux valeurs entre elles. L'instruction INPUT, va vous permettre de multiplier n'importe quelles valeurs de votre choix :

```
INPUT A,B : PRINT A * B <ENTER>
```

Lorsque vous aurez tapé cette ligne à l'écran, suivie de ENTER, vous verrez apparaître un point d'interrogation. Cela signifie que l'ordinateur attend une réponse de votre part. En l'occurrence vous inscrirez en réponse, une valeur, une virgule et une autre valeur. L'ordinateur affichera immédiatement le résultat de la multiplication entre la 1ère et la 2ème valeur. vous répondrez par exemple :

```
3,4 <ENTER>
```

L'instruction INPUT permet donc d'introduire des valeurs dans le corps d'une ligne d'instructions. De la même façon vous pourrez introduire du texte :

```
INPUT A$: PRINT "BONJOUR ";A$ <ENTER>
```

Inscrivez votre prénom (toujours suivi par la frappe de la touche ENTER) et l'ordinateur va vous dire bonjour personnellement. Et toujours pour améliorer la présentation à l'écran on peut aussi, avec l'instruction INPUT, imprimer du texte, ce qui peut donner :

```
INPUT "Quel est ton prénom ";A$ : PRINT "Bonjour ";A$ <ENTER>
```

C'est tout de même plus agréable ainsi.

Malgré tout le principal inconvénient de nos petites manipulations reste qu'il faut à chaque fois retaper la ligne complète d'instructions pour obtenir un nouveau résultat.

3) La notion de programme

Nous allons reprendre l'exemple précédent en mettant devant la première instruction un numéro de ligne :

```
10 INPUT "Quel est ton prenom ";A$: PRINT "Bonjour ";AS <ENTER>
```

Lorsque vous appuyez sur la touche <ENTER> il ne se passe rien. Vous venez de créer une ligne de programme que l'ordinateur a mis dans sa mémoire. Pour pouvoir l'exécuter il faudra taper :

```
RUN <ENTER>
```

Jusqu'à présent pas beaucoup de changement. Pourtant retapez RUN à l'écran et l'ordinateur vous repose la même question, ceci chaque fois que vous le désirez, la ligne 10 étant définitivement dans la mémoire de l'ordinateur, tout au moins tant que vous ne l'éteindrez pas ou que vous ne remplacerez pas cette ligne par une autre.

Ajoutons maintenant une autre ligne à notre programme :

```
20 INPUT "Quel est ton age ";N : PRINT AS;" , tu as donc ";N;" ans" <ENTER>
```

Après avoir tapé RUN <ENTER>, l'ordinateur exécute d'abord la ligne 10 puis la ligne 20, ce qui permet de dégager la règle suivante :

- L'ordinateur exécute le programme en suivant l'ordre croissant des numéros de ligne (sauf ordre contraire).

4) Quelques instructions BASIC élémentaires

- l'instruction GOTO

Cette commande va nous permettre de diriger l'ordinateur sur une ligne de notre choix.

Rajoutons à notre précédent programme la ligne suivante :

```
30 GOTO 10 <ENTER>
```

A l'exécution, après avoir imprimé à l'écran le dernier message de la ligne no 20, lorsque la ligne 30 est rencontrée, l'instruction GOTO renvoie l'ordinateur à la ligne 10 et recommence à exécuter ce court programme.

A titre d'essai, remplacez la ligne 30 par :

```
30 GOTO 20    <ENTER>
```

La différence doit vous sauter aux yeux.

- L'instruction IF ... THEN

Elle permet de tester si une condition est réalisée ou non. Dans l'affirmative, l'ordinateur exécutera l'instruction située après THEN.

Dans notre programme, nous désirons afficher le message "tu es un adulte", dans le cas où l'âge donné est supérieur à 17 ans. Il suffit de remplacer la ligne 30 par :

```
30 IF N>17 THEN PRINT "tu es un adulte "; A$
40 END
```

Si l'âge est supérieur à 17 ans, la condition sera réalisée, et le message affiché.

Dans le cas contraire, l'ordinateur sautera à la ligne suivante et s'arrêtera. En effet l'instruction END signifie la fin du programme.

On pourrait également faire suivre THEN d'un numéro de ligne plutôt que d'une instruction :

```
30 IF N>17 THEN GOTO 40 ELSE GOTO 50
40 PRINT "tu es un adulte ";A$
50 END
```

La ligne 30 peut se traduire par : si est supérieur à 17 aller à la ligne 40, sinon aller à la ligne 50

L'instruction ELSE va diriger l'ordinateur vers une ligne précisée dans le cas où la condition n'est pas réalisée.

La combinaison de toutes ces instructions va nous permettre de créer un programme plus sophistiqué :

```
10 INPUT "Quel est ton prénom ";A$ : PRINT "Bonjour "; A$
20 INPUT "Quel est ton âge ";N : PRINT A$ ; ", tu as donc ":N:"
   ans"
30 IF N>17 THEN 70
40 IF N>11 THEN 60
50 PRINT "tu es un enfant ";A$ : GOTO 80
```

```
60 PRINT "tu es un adolescent ";AS : GOTO 80
70 PRINT "tu es un adolescent ";A$
80 END
```

Nous n'indiquons plus maintenant qu'il faut taper sur la touche ENTER à la fin de chaque ligne de programme

Vous remarquerez aux lignes 30 et 40 que si la condition n'est pas réalisée l'ordinateur sautera à la ligne suivante. Le GOTO après THEN a été supprimé car il est facultatif, de même qu'après ELSE.

Enfin, aux lignes 50 et 60, le GOTO 80 évite d'imprimer un message qui ne correspond pas à l'âge indiqué.

Il existe beaucoup d'autres instructions BASIC dans la mémoire de l'AMSTRAD. Elles sont toutes répertoriées dans votre manuel d'utilisation. La plupart sont employées dans les programmes de cet ouvrage.

5) l'utilisation du magnétophone

Avant de terminer sur cette courte introduction sur le langage BASIC, nous voudrions vous parler un peu du magnétophone.

Vous trouvez que le petit programme que vous venez de taper, et j'en suis sûr d'améliorer, n'est pas si mal que ça pour un début et vous désirez le conserver.

Ce n'est pas très difficile et même très simple. Il va falloir mettre une K7 dans le magnétophone, en prenant garde d'avoir laissé passer l'amorce, et donner un nom à votre programme. Mettons BONJOUR.

Pour enregistrer ce programme BONJOUR sur la K7, il suffit de taper :

```
SAVE "BONJOUR"      <ENTER>
```

Vous verrez alors apparaître à l'écran un message qui signifie que vous devez appuyer sur les touches REC et PLAY de votre magnétophone, ce que vous ferez. Le magnétophone va alors démarrer et enregistrer votre programme.

Un ami vient vous voir et vous voulez lui faire une démonstration de vos nouveaux talents de programmeur.

Pour retrouver votre programme, après avoir allumé évidemment l'AMSTRAD, il faut mettre la K7 dans le magnétophone, appuyer sur la touche PLAY et taper :

```
LOAD "BONJOUR"
```

ou :

```
LOAD ""
```

si vous avez oublié le nom de votre programme

A l'apparition du message READY votre programme est dans la mémoire de l'ordinateur prêt à être de nouveau exécuter.

Si ça ne marche pas demandez vous si :

- vous avez bien rembobiné la K7
- si le volume est correctement réglé

Afin de tester vos nouvelles connaissances, et pour clore ce chapitre, nous vous proposons un petit programme sur les instructions BASIC.

TEST DES CONNAISSANCES

```
10 REM -----
20 REM DICTIONNAIRE BASIC
30 REM & Test de connaissance
40 REM -----
50 CLS:INK 0,0
60 PRINT TAB(22) STRING$(30,40)
70 PRINT TAB(25) "D I C T I O N N A I R E"
80 PRINT TAB(22):STRING$(30,140)
90 PRINT : PRINT : PRINT "Quel est votre prenom ";;INPUT v$
100 i = 0: n = 0
110 READ a,a$,b$
120 b=0
130 PRINT:PRINT"Signification du mot -----> ";a$
140 PRINT:INPUT "Tapez votre reponse -----> ";c$
150 CLS
160 PRINT CHR$(23)
170 i=i+1
180 IF b$(i)<>c$ THEN 230
190 FOR s=1 TO 20
200 LOCATE 2,12: PRINT "B O N N E   R E P O N S E"
210 NEXT s: CLS
220 IF i=15 THEN 280 ELSE 110
230 FOR r=1 TO 15
240 LOCATE 2,12: PRINT "M A U V A I S E   R E P O N S E"
250 NEXT r: CLS:
260 PRINT : PRINT "La bonne reponse etait ----> ":PRINT:LOCATE 30,4:PRINT b$:PRINT
270 INPUT "Tapez sur ENTER pour continuer";x: IF x<>0 THEN 310
280 CLS : IF i>=15 THEN 310 ELSE 110
290 DATA 1,print,imprimer,2,input,entrer,3,for,pour,4,next,selon,5,step,pas,6,if
,si,7,then,alors,8,else,sinon
300 DATA 9,save,sauver,10,load,charger,11,break,arret,12,run,executer,13,new,nou
veau,14,goto,aller,15,locate,positionner,16>window,fenetre
310 FOR t=1 TO 50
320 PRINT "F I N   D E   P R O G R A M M E"
330 NEXT t:CLS
340 PRINT v$;" , vous avez ";i;" reponses justes sur 16"
350 FOR N = 1 TO 500:NEXT
360 IF i>=8 THEN 390
370 CLS:PRINT:PRINT"Veuillez reviser serieusement"
380 FOR n= 1 TO 1000 : NEXT
390 END
```

CHAPITRE 2

LES JEUX

1) Jouons au loto

Voilà un problème simple que peut se poser un programmeur amateur. Il s'agit de choisir au hasard de 6 à 10 chiffres, ce que l'ordinateur peut facilement faire grâce à l'instruction RND. Mais, pour compliquer les choses, on ne doit pas tirer un numéro déjà sorti, et, pour la beauté du programme, on va classer les numéros tirés par ordre de grandeur.

Les lignes 100 à 145 permettent de faire une présentation à l'écran et demandent combien de chiffres doivent être tirés (ligne 140).

De 180 à 310 on choisit de façon aléatoire les chiffres, avec, de 220 à 240, une vérification pour savoir si le chiffre est compris entre 1 et 49, de 250 à 290, on regarde si le numéro tiré est déjà sorti.

Le sous programme qui occupe les lignes 1000 à 1120 fait un tri par ordre croissant des nombres sortis.

Enfin les lignes 350 à 370 affichent à l'écran les valeurs triées.

Liste des variables :

no = nombre de chiffres à tirer
y = chiffre tiré
nu (z) = tableau des chiffres tirés

REMARQUES :

Les tableaux :

L'instruction DIM permet de ranger des éléments dans un tableau imaginaire et de donner ainsi à chaque élément un numéro d'ordre.

Par exemple à la ligne 50 DIM nu (10), on crée un tableau de 10 éléments (en fait 11 de 0 à 10), qui

seront rangés dans les variables nu (1) à nu (10).

Dans notre programme, si l'AMSTRAD tire les chiffres suivants, 18, 47, 2, 6, 39, 25, le tableau, après tri, se présentera de la façon suivante :

N(1)	N(2)	N(3)	N(4)	N(5)	N(6)
2	6	18	25	39	47

Avec v pour variables et d pour données.

L'instruction DIM permet de manipuler beaucoup plus facilement des variables et facilite l'écriture de programmes. Ainsi trois lignes seront suffisante (on aurait pu les mettre sur une seule ligne), de 350 à 370, pour afficher 10 variables (ou beaucoup plus selon la boucle FOR NEXT et la dimension du tableau).

Le chiffre entre parenthèse représente la dimension maximum du tableau, c'est-à-dire le nombre maximum d'éléments que ce dernier pourra recevoir. Il est cependant important de ne pas surdimensionner le tableau, car l'ordinateur, lorsqu'il rencontre l'instruction DIM, réserve en mémoire la place destinée aux variables ce qui diminue la mémoire disponible pour le reste du programme.

L'instruction FOR ... NEXT ... STEP

```
Ex : 10 FOR x = 1 to 10
      20 print x
      30 next x
```

Ce court programme peut se traduire en français par l'ordre 'imprimer x à l'écran 10 fois'. En effet l'ordinateur va imprimer la valeur de x, incrémenter x de 1, imprimer la nouvelle valeur de x, incrémenter de nouveau x et ce jusqu'à ce que x soit égal à 10.

Si on remplace la ligne 10 par

```
10 FOR X = 1 TO 10 STEP 3
```

seules les valeurs 1, 4, 7, et 10 seront affichées car x sera cette fois-ci incrémenté de 3. STEP représente le pas d'incrémentation, et la valeur qui précise ce pas peut même être comme à la ligne 1040.

LOTO

```
10 REM -----
20 REM L O T O
30 REM -----
50 DIM nu(10): BORDER 0
100 CLS : PRINT
110 PRINT "          $$$$ L O T O $$$$"
120 PRINT : PRINT : PRINT
130 PRINT "Combien de chiffres"
140 INPUT "voulez vous jouer (6 a 10) ";no
145 IF no<6 OR no>10 THEN 140
150 REM -----
160 REM tirage au sort
170 REM -----
180 FOR x=1 TO no
190 y=RND(0.1)
200 y=y*100
210 y=INT (y)
220 REM on verifie si le tirage est inferieur a 1 ou superieur a 49
230 IF y<1 OR Y>49 THEN x=x-1 : GOTO 310
240 IF x = 1 THEN 300
250 REM nombre deja sorti ?
260 FOR z=1 TO x-1 : h=0
270 IF y=nu(z) THEN x=x-1 : h=1 : z=x
280 REM on met le nombre dans le tableau
290 NEXT z : IF h=1 THEN 310
300 nu(x)=y
310 NEXT x
320 GOSUB 1000
330 REM affichage
340 PRINT
350 FOR x=1 TO no
360 PRINT nu(x),
370 NEXT x
380 PRINT "Voulez vous un autre tirage (O/N) ?"
390 a$=INKEY$ : IF a$="" THEN 390
400 IF a$="o" OR a$="O" THEN 100
410 IF a$="n" OR a$="N" THEN 420 ELSE 390
420 END
1000 REM -----
1010 REM tri des nombres sortis
1020 REM -----
1030 FOR x=1 TO no
1040 FOR y=no TO 1 STEP -1
1050 IF nu(y)>=nu(x) THEN 1100
1060 IF x>y THEN 1100
1070 a = nu(x)
1080 nu(x)=nu(y)
1090 nu(y)=a
1100 NEXT y
1110 NEXT x
1120 RETURN
```


Deviner des chiffres.

Voilà donc un programme tiré d'un jeu célèbre et dont je dois d'abord vous rappeler les règles.

Vous devez découvrir une combinaison de chiffres (4 à 9 selon le degré de difficulté), choisie de façon aléatoire dans le programme. A chaque essai, l'ordinateur vous indique quels sont les chiffres justes et bien placés et les chiffres justes mais mal placés. A vous d'en déduire la bonne combinaison.

Le programme, assez bien documenté n'appelle pas de commentaire particulier. Vous remarquerez cependant que les tableaux $f(x)$ et $t(x)$ n'ont pas été déclarés en début de programme, ce qui ne l'empêche pas de tourner normalement. Faites donc des essais pour savoir à partir de combien d'éléments on doit déclarer un tableau.

Liste des variables :

n = niveau de jeu
q = nombre d'essais autorisés
v = nombre de tentatives
c = nombre de chiffres justes et biens placés
e (x) = chiffres choisis par le joueur
s (x) = chiffres tirés par l'ordinateur
f (x) = tableau de comparaison de la combinaison du joueur
t (x) = tableau de comparaison de la combinaison de l'ordinateur

REMARQUES :

Le rôle de l'instruction VAL est de transformer une variable chaîne en un nombre.

Par exemple, à la ligne 1240, si vous tapez la touche 1, la variable a\$ sera égale au caractère "1". la ligne 1250 transforme ce caractère en chiffre et le range dans la variable a, ce qui permet de saisir des valeurs numériques directement à l'aide des instructions INKEYS et VAL (a\$).

AMSTERMIND

```
3 REM -----
5 REM DEVINER DES CHIFFRES
7 REM -----
10 MODE 1 : CLS : INK 0,0 : BORDER 0
20 DIM s(9) , e(9)
100 REM -----
110 REM Choix du niveau de jeu
120 REM -----
130 CLS
140 PRINT TAB(10)"A M S T E R M I N D"
150 PRINT : PRINT : PRINT
152 PRINT "Pour ne pas trop compliquer le jeu,": PRINT
155 PRINT "il n'y a pas 2 chiffres identiques ": PRINT
157 PRINT "dans la meme combinaison":PRINT : PRINT
160 INPUT "Niveau de jeu ( 1 a 6)";n
170 n=INT(n)
180 IF n<1 OR n>6 THEN 130
190 n=n+3
200 GOSUB 1000
207 q=n+11 : v=0
210 GOSUB 1100
215 v=v+1
217 IF v=q THEN 270
220 GOSUB 1200
230 GOSUB 1300
240 PRINT TAB(37) c, TAB(61) b
250 IF c=n THEN GOSUB 1400 : GOTO 300
255 GOTO 215
270 PRINT : PRINT "Vous avez fait ";v;" tentatives"
280 PRINT "la combinaison a decouvrir etait : ";
290 FOR x=1 TO n : PRINT s(x);: NEXT x
300 PRINT "voulez vous faire une autre partie (O/N) ?"
310 a$=INKEY$: IF a$="" THEN 310
320 IF a$="o" OR a$="O" THEN RUN
330 IF a$="n" OR a$="N" THEN 999 ELSE 310
999 END
1000 REM -----
1010 REM Tirage au sort de la combinaison
1020 REM -----
1030 FOR x=1 TO n
1040 a=RND (9)
1050 a=a * 10
1060 a=INT (a): IF x=1 THEN 1070
1065 FOR wx=1 TO x: IF a=s(wx) THEN wx=x: x=x-1
1067 NEXT wx
1070 s(x)=a
1080 NEXT x
1090 RETURN
1100 REM -----
1110 REM affichage du jeu
1120 REM -----
1125 CLS : MODE 2
1130 PRINT "      VOTRE COMBINAISON          CHIFFRES JUSTES          CHIFFRES JUSTE
"
1140 PRINT "          et bien places          et mal place
"
1150
```

```

1150 PRINT : PRINT
1160 RETURN
1200 REM -----
1210 REM saisie des caracteres
1220 REM -----
1230 FOR x=1 TO n
1240 a$=INKEY$ : IF a$="" THEN 1240
1250 a=VAL (a$)
1260 e(x)=a
1270 PRINT e(x);
1280 NEXT x
1290 RETURN
1300 REM -----
1310 REM Test des chiffres justes
1320 REM -----
1325 b=0 : c=0
1330 FOR x=1 TO n : t(x)=s(x)
1340 FOR z=1 TO n : f(z)=e(z)
1345 REM Test des chiffres justes et biens places
1347 IF f(x)=t(x) THEN f(x)=10 : t(x)=11 : c=c+1
1348 REM test des chiffres justes et mal places
1350 IF f(z)=t(x) AND z<>x THEN b=b+1 : f(z)=10 : t(x)=11 : z=n
1360 NEXT z
1370 NEXT x
1380 RETURN
1400 REM -----
1410 REM combinaison gagnante
1420 REM -----
1430 PRINT "BRAVO, vous avez trouve la solution en ";v;" essais"
1440 RETURN

```

3) Le jeu du pendu

Le but de ce jeu est de trouver un mot, en cherchant lettre par lettre, choisi par l'ordinateur dans les mots stockés en DATA.

Après 9 essais infructueux, vous serez définitivement pendu. Pendant le déroulement du programme vous verrez se dresser un élément de la potence à chacune de vos erreurs.

Le programme stocke 100 mots de vocabulaire en DATA et utilise les codes graphiques pour obtenir le dessin de la potence à l'écran. En voici le listing.

Liste des variables :

b = position du mot choisi en DATA
t = nombre d'essais justes
u = test lettre juste ou non
w = longueur du mot choisi en DATA
x = nombre d'échecs
z = total des tentatives (x + t)
p (25) = position des lettres découvertes

REMARQUES :

L'instruction DATA permet de stocker, a n'importe quel endroit du programme, des variables numériques ou alphanumériques. Vous pouvez remarquer aux lignes 2000 à 2110 que les chaînes de caractères n'ont pas besoin d'être écrites entre guillemets.

Pour lire ces variables on utilise l'instruction READ. Pour exemple READ a\$ lira le mot table. Un second READ a\$ rangera le mot cuisine dans la variable a\$. Donc l'instruction READ lit séquentiellement les mots inscrits en DATA et un pointeur se positionne sur le mot suivant le dernier lu.

L'instruction RESTORE repositionne le pointeur sur le premier mot inscrit en DATA.

Examinons ce court programme.

```
10 FOR N = 0 TO 9
20 READ a$
30 NEXT N
40 READ a$
50 PRINT a$
```

N'oubliez pas de rajouter aussi la ligne 2000 du programme.

A l'exécution vous pouvez constater que sur l'écran s'affiche le mot fenêtre, l'instruction READ de la ligne 20 lit les 10 premiers mots, celle de la ligne 40 le 11ème mot.

Rajoutez la ligne 35 RESTORE. A l'exécution s'affichera le mot table, le pointeur ayant été remis au début de la liste par l'instruction RESTORE.

Dernier essai, uniquement avec la ligne 2000 comme DATA, vous changez la ligne 10 par :

```
10 FOR N = 1 TO 20
```

Lors de l'exécution l'ordinateur vous répondra par un OUT OF DATA, ce qui veut simplement dire que vous avez essayé de lire plus de données qu'il n'y en avait rangées dans les lignes de DATA.

L'instruction CHR\$

Il existe au moins deux méthodes pour faire du graphisme à l'écran : de dessiner point par point (avec DRAW, PLOT etc ...) ce qui donne de très bon résultat mais qui est très long, d'utiliser les codes graphiques, ce qui donne un dessin beaucoup plus grossier mais qui prend très peu de place en mémoire.

C'est ici la seconde méthode que nous utilisons. Vous trouverez à la fin de votre manuel d'utilisation tous les caractères graphiques disponibles avec leur code. Pour imprimer un caractère graphique à l'écran il suffit de taper PRINT CHR\$(x) ; , x étant le code du caractère souhaité. Si vous n'avez pas vraiment compris examinez les lignes 1060 à 1220 ou, ce qui est encore mieux faites vous même des essais.

LE PENDU

```
10 REM -----
20 REM   P E N D U
30 REM -----
50 MODE 1
60 DIM p(25)
70 REM -----
80 REM tirage au sort du mot en data
90 REM -----
100 CLS
110 PRINT : INPUT "Entrez un nombre de 1 a 10 ";a;a=INT(a)
120 IF a<1 OR a>10 THEN 110
130 b=INT (RND(a)*100)
140 REM -----
150 REM lecture du mot en data
160 REM -----
170 RESTORE
180 FOR n=0 TO b
190 READ mot$
200 NEXT n
210 w=LEN (mot$)
220 REM -----
230 REM deroulement du jeu
240 REM -----
250 LOCATE 10,21
260 PRINT STRING$(w, ".")
270 x=0 : z=0 : t=0
280 WHILE x <> 9
290 u=0
300 REM -----
310 REM saisie du caractere
320 REM -----
330 a$=INKEY$:IF a$="" THEN 330
340 IF ASC (a$)<97 OR ASC(a$)>122 THEN 330
350 REM -----
360 REM comparaison du caractere avec le mot
370 REM -----
380 FOR j=1 TO w : p(j)=0
390 IF a$=MID$(mot$,j,1) THEN p(j)=j : t=t+1
400 NEXT j
410 REM -----
420 REM affichage du caractere si exact
430 REM -----
440 FOR j=1 TO w
450 IF p(j)<>0 THEN LOCATE 9+p(j),21 : PRINT a$;:u=1
460 NEXT j
470 IF u=0 THEN GOSUB 1000
480 z=z+1
490 IF t=w THEN GOSUB 1300 : x=9
500 WEND : IF t=w THEN 540
510 LOCATE 1,23
520 PRINT :PRINT "vous avez perdu apres ";z;" essais"
530 PRINT "le mot etait ";mot$;" ,vous etes perdu"
540 INPUT "une autre partie (o/n) ";a$
550 IF a$="o" OR a$="O" THEN 70
560 IF a$="n" OR a$="N" THEN 570 ELSE 540
570 END
```

L'instruction LOCATE

Evidemment si vous écrivez n'importe où vos codes graphiques vous n'obtiendrez rien de très cohérent.

Vous devez utiliser l'instruction LOCATE dont le rôle est de positionner le curseur.

En mode 1, LOCATE 10,5, mettra le curseur à la 10^{ème} colonne de la 5^{ème} ligne (regardez également l'appendice VI page 2 de votre manuel).

Si cette instruction est suivie d'un PRINT CHR\$(x), le code de x sera affiché à la position du curseur.

Le point virgule (;) après CHR\$(x) empêche le curseur de passer à la ligne suivante. Regardez à nouveau les lignes 1050 à 1220 et vous aurez tout compris.

Les instructions WEND WHILE

Examinez les lignes 280 et 500. Cela veut dire que tant que x sera différent de 9, le programme lorsque il rencontrera l'instruction WEND, recommencera l'exécution à la ligne 290. Pratique non ?

```

1000 REM -----
1010 REM affichage du dessin
1020 REM -----
1030 x=x+1
1040 ON x GOTO 1050,1070,1100,1120,1140,1160,1180,1200,1220
1050 LOCATE 14,14
1060 PRINT CHR$(140);CHR$(140);CHR$(141); CHR$(140):GOTO 1230
1070 FOR j=1 TO 6 : LOCATE 16,7+j : PRINT CHR$(133);:NEXT j
1080 LOCATE 16,7 : PRINT CHR$(135)
1090 GOTO 1230
1100 LOCATE 17,7 : PRINT CHR$(131);CHR$(131);CHR$(131);CHR$(131)
1110 GOTO 1230
1120 LOCATE 21,7 : PRINT CHR$(211)
1130 LOCATE 21,8 : PRINT CHR$(211) : GOTO 1230
1140 LOCATE 21,8 : PRINT CHR$(225)
1150 GOTO 1230
1160 LOCATE 21,9 : PRINT CHR$(202)
1170 GOTO 1230
1180 LOCATE 20,9 : PRINT CHR$(194);
1190 GOTO 1230
1200 LOCATE 22,9 : PRINT CHR$(195)
1210 GOTO 1230
1220 LOCATE 21,10 : PRINT CHR$(196)
1230 RETURN
1300 REM -----
1310 REM partie gantee
1320 REM -----
1330 LOCATE 1,23
1340 PRINT "BRAVO vous avez trouve"
1350 PRINT "apres ";z;" essais"
1360 RETURN
2000 DATA table,cuisine,chaise,fauteuil,tabouret,armoire,bibliotheque,radiateur,
tapisserie,bahut,fenetre
2010 DATA cheval,elephant,hipopotame,guopard,souris,corneille,girafe,phoque,cach
alot,crocodile,panthere
2020 DATA mysterieusement,episodiquement,curieusement,extremement,machinalement,
anticonstitutionnellement
2030 DATA platonique,gentillesse,facilite,stupidite
2040 DATA television,aspirateur,automobile,transatlantique,ordinateur,magnetosco
pe,amplificateur,motocyclette
2050 DATA poireau,courgette,safran,aubergine,aubepine,eucalyptus,jonquille,edelw
eis,ananas,mandarine
2060 DATA cinematographe,radiologie,telephone,dictionnaire,imprimerie,stomatolog
ue,ophtalmologiste
2070 DATA epistolaire,reconnaissance,reparation,respiration,coccyx,bouteille,abc
es,arcboutant,catastrophique
2080 DATA saxophone,clarinette,tuba,acordeon,contrebasse,violoncelle,trompette,h
arpe,luth,clavecin
2090 DATA araignee,fourmis,doriphore,papillon,cafard,punaise,leopard,salamandre,h
aneton,libellule
2100 DATA paysage,cumulus,aiguille,neve,peninsulaire,continental,asiatique,orien
tal,oceanique,equatorial
2110 DATA diurne,nocturne,crepusculaire,solstice,equinoxe

```


CHAPITRE 3

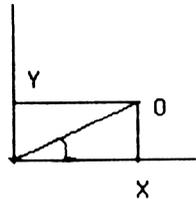
MATHÉMATIQUES

L'AMSTRAD est doté d'un certain nombre d'instructions mathématiques assez simples mais qui, combinées entre elles, permettent de résoudre des problèmes complexes. Nous vous proposons ici, à titre d'exemple, trois programmes qui utilisent ces instructions.

1) Manipuler des degrés

Vous connaissez sans doute le problème illustré par la figure 1.

Il s'agit, connaissant la valeur de l'angle \hat{a} et la longueur OP (Coordonnées polaires), de calculer x et y (coordonnées rectangulaires). x et y sont donnés par la formule :



$$\begin{aligned}x &= OP * \text{COS} (\hat{a}) \\Y &= OP * \text{SIN} (\hat{a})\end{aligned}$$

De même, connaissant x et y, on peut calculer les coordonnées polaires OP et \hat{a} avec la formule :

$$\begin{aligned}OP &= \sqrt{x^2 + y^2} \\ \hat{a} &= \tan^{-1} y/x\end{aligned}$$

Le programme qui suit, outre le calcul des valeurs, fait une présentation du problème à l'écran.

Liste des variables :

c = choix du type de calcul
ang = angle à
dis = distance OP
x = abscisse du point A
y = ordonnée du point A

COMMENTAIRES :

Examinons ce programme, il comporte un certain nombre d'instructions intéressantes.

- A la ligne 60, on met l'ordinateur en mode degrés avec l'instruction DEG. L'AMSTRAD travaille normalement en radians. Il faut donc lui préciser que l'on veut utiliser des degrés.

Cela est fait une fois pour toute en début de programme avec l'instruction DEG.

Pour revenir au calcul en radians, il suffira d'employer l'instruction RAD à l'endroit choisi

- Les lignes 120 à 260 reproduisent à l'écran le dessin de la figure 1. Si elles ne sont pas vraiment indispensables, elles améliorent considérablement la présentation du problème. vous remarquerez que la ligne 240 fait 'monter' le dessin à l'écran. Cela vous donnera peut être des idées

- lignes 330 à 380 : choix de l'option de calcul.

- ligne 390 : l'instruction ON C GOTO (ou ON C GOSUB).

Il s'agit d'une instruction particulièrement puissante puisqu'elle permet, en fonction de la valeur de C, d'accéder directement à une partie du programme.

Dans notre cas, si C est égal à 1, le programme se branchera à la ligne 400 (conversion de coordonnées polaires en coordonnées rectangulaires). Pour C égal à 2, c'est la ligne 600 qui sera exécutée.

L'instruction ON C GOSUB, s'utilise s'emploie de façon identique.

DEGRES

```
10 REM -----
20 REM transformation de coordonnees polaire en rectangulaire et l'inverse
30 REM -----
50 BORDER 0 : INK 0,0
60 CLS : DEG
100 REM -----
110 REM trace du dessin
120 REM -----
130 ORIGIN 100,100
140 DRAW 0,200
150 ORIGIN 100,100
160 DRAW 200,0
170 ORIGIN 100,100
180 DRAW 200,200
190 ORIGIN 180,100
200 DRAW 0,80
210 LOCATE 10,18 : PRINT CHR*(181);
220 LOCATE 10,20 : PRINT "x";
230 LOCATE 13,18 : PRINT "y";
240 FOR n=1 TO 12 : PRINT : NEXT n
250 LOCATE 10,9 : PRINT "A";
260 LOCATE 5,15 : PRINT "O";
330 LOCATE 1,18 : PRINT "Vous connaissez l'angle ";CHR*(181)
340 PRINT "et la distance OA (1) " : PRINT
350 PRINT "Ou vous connaissez les longueurs"
360 PRINT "rectangulaires x et y (2) " : PRINT
370 INPUT "votre choix (1 ou 2)";c
380 c=INT(c):IF c<1 OR c>2 THEN 350
390 ON c GOTO 400,600
400 REM -----
410 REM conversion coordonnees polaires a rectangulaire
420 REM -----
430 GOSUB 1000
440 LOCATE 1,18
450 INPUT "entrez l'angle en degres ";iang
460 PRINT : INPUT "quelle est la distance OA ";dis
470 x = dis*COS(ang) : y=dis*SIN(ang)
475 ZONE 3
480 PRINT:PRINT "x = ";:PRINT USING "##.##";X
485 PRINT "y = ";:PRINT USING "##.##";Y
490 GOTO 700
600 REM -----
610 REM conversion coordonnees rectangulaires a polaire
620 REM -----
630 GOSUB 1000
640 LOCATE 1,18
650 INPUT "entrez les coordonnees x,y";x,y
660 r=SQR(x^2+y^2) : ang=ATN(y/x)
675 ZONE 3
680 PRINT : PRINT "OA = ";:PRINT USING "##.##";r
685 PRINT " ";CHR*(181);" = ";:PRINT USING "##.##";ang;: PRINT " deg"
700 a$=INKEY$:IF a$="" THEN 700 ELSE 60
1000 REM -----
1010 REM effacement texte
1020 REM -----
1030 LOCATE 1,18

1040 FOR n=1 TO 7
1050 PRINT STRING$(33,32)
1060 NEXT
1070 RETURN
```

- les lignes 400 à 700 sont l'application directe des formules de calcul avec :

SIN (à) donne le sinus de l'angle à
COS (à) donne le cosinus de l'angle à
ATN (x) permet de connaître la valeur de l'angle à dont la tangente est x
SQR (x) fournit la racine carée de x ou d'une expression
La flèche vers le haut permet de calculer les puissances d'une valeur ou d'une expression.

Par ex : x ↑ 2 donnera x * x
 3 ↑ 2 " 9
 x ↑ 3 " x * x * x
 (a + b) ↑ 2 " a ↑ 2 + b ↑ 2 + 2ab

REMARQUES :

- L'instruction ZONE (ligne 475)

Sur presque tous les ordinateurs existe une tabulation automatique que l'on obtient avec PRINT et la virgule. Ex : PRINT "1","2" . Le chiffre 1 sera imprimé en début de ligne et la chiffre 2 à la 14ème position par rapport au début de ligne (ou à la colonne no 14).

l'instruction ZONE permet de faire varier cette position.

Ex : ZONE 8 : PRINT "1","2"

Le 1 sera affiché en début de ligne, et le 2 dans la colonne no 9 de la même ligne

- L'instruction STRING\$ (X,Y)

Elle affiche X fois à l'écran le caractère dont le code ASCII est Y. On peut écrire :

PRINT STRING\$ (10,65);
ou PRINT STRING\$ (10,"A");

Dans les deux cas, une série de 10 lettres A sera affichée à l'écran.

Dans notre programme, à la ligne 1050, nous employons cette instruction pour effacer du texte à l'écran, sans effacer tout l'écran, 32 étant le code ASCII d'un espace.

Résoudre des équations

L'équation du second degré, universellement connue, se présente ainsi :

$$y = ax^2 + bx + c$$

Le calcul d'un discriminant ($\text{dis} = b^2 - 4ac$) permet de déterminer dans tous les cas le type de l'équation (premier degré, pas de solution, etc...)

L'existence d'une méthode unique pour résoudre tous les cas possibles, facilite la recherche d'une solution sur ordinateur.

Le programme ne présente aucune difficulté et vous retrouverez aisément toutes les formules de calcul, au cas où vous les auriez oubliées.

Liste des variables

a = a

b = b

c = c

X1 = 1ère racine

X2 = 2ème racine

w(z) = tableau des différentes valeurs de x

y(z) = tableau des différentes valeurs de y

A titre d'amélioration de ce programme, vous pouvez essayer d'obtenir une représentation graphique de la fonction à l'écran. C'est tout à fait réalisable sur l'AMSTRAD.

Il conviendra de définir une échelle en fonction des différentes valeurs de x et de y, de dessiner un axe, et de dessiner la courbe point par point avec la fonction PLOT.

Pour vous aider dans cette tâche, vous pouvez vous reporter au programme HISTOGRAMME qui comporte un calcul d'échelle, ainsi qu'aux autres programmes graphiques du chapitre suivant.

Vous pouvez également utiliser l'instruction DEF FN, qui est une définition de fonction, en début de programme. Son utilisation est assez simple et fort pratique. Dans notre cas cela se traduirait par :

```
DEF FN y = a * x 2 + b * x + c
```

Ce qui nous permet d'obtenir la valeur de y en fonction de x à tout moment :

```
INPUT X : PRINT FN y
```

Vous pouvez donc essayer de modifier ce programme en y incluant cette instruction, et vous verrez que c'est très pratique.

Si aucune touche n'est frappée, le programme passe à l'instruction suivante et la variable a\$ est alors vide.

Combinée avec l'instruction IF, en testant le contenu de la variable A\$, cette instruction nous permet de faire des test pour savoir si une touche a été appuyée, ou pour faire une attente avec appui sur une touche, avant de poursuivre l'exécution d'un programme (c'est le cas de notre programme).

Très pratique pour les jeux.

EQUATIONS

```
10 REM -----
20 REM RESOLUTION D'EQUATION DU SECOND DEGRE "
30 REM -----
40 DIM w(20),y(20)
50 CLS : PRINT "Resolution d'equation du second degres"
60 PRINT "du type a(x*x)+bx+c=y"
70 REM -----
80 REM tests et calcul de racines
90 REM -----
100 PRINT : PRINT "entrez a,b,c"
110 INPUT "en les separant par une virgule ";a,b,c
120 IF a=0 THEN GOSUB 1000 : GOTO 510
130 dis=b^2-4*a*c
140 PRINT
150 PRINT "discriminant = ";dis
160 IF dis = 0 THEN PRINT : PRINT "Racine double " : PRINT : GOTO 190
170 IF dis<0 THEN GOSUB 1100 : GOTO 510
180 PRINT : PRINT "Racines de l'equation " : PRINT
190 x1=(-b+SQR(dis))/(2*a)
200 x2=(-b-SQR(dis))/(2*a)
210 PRINT "X1 = ";x1
220 PRINT "X2 = ";x2
230 REM -----
240 REM calcul de y en fonction de x
250 REM -----
260 PRINT : PRINT "nombre de calcul de y(0 a 20) ";:INPUT n
270 n=INT(n):IF n>20 OR n<0 THEN 260
280 IF n=0 THEN 530
290 FOR z=1 TO n
300 PRINT "entrez x";z;:INPUT w(z)
310 y(z)=a*w(z)^2+b*w(z)+c
320 NEXT z
330 REM -----
340 REM affichage des divers elements
350 REM -----
360 CLS : PRINT
370 PRINT "equation : ";a;"(x*x)";
380 IF b=0 THEN 410
390 IF b<0 THEN PRINT b;"x";:GOTO 410
400 PRINT "+";b;"x";
410 IF c=0 THEN 440
420 IF c<0 THEN PRINT c;:GOTO 440
430 PRINT "+";c;
440 PRINT "= y" : PRINT
450 PRINT "discriminant : ";dis;:PRINT
460 PRINT "racines : X1 = ";x1
470 PRINT " X2 = ";x2 : PRINT
480 FOR z=1 TO n
490 PRINT "x";z;" = ";w(z);" y";z;" = ";y(z)
500 NEXT z
510 a$=INKEY$:IF a$="" THEN 510
520 GOTO 50
530 END
540 REM -----
1000 REM -----
1010 IF b=0 THEN 1030
1020 PRINT : PRINT "equation du 1er degre " : GOTO 1040
1030 PRINT : PRINT "indetermine "
1040 RETURN
1100 REM -----
1110 PRINT : PRINT "pas de racine reelle"
1120 RETURN
```


STATISTIQUES

Ce programme de statistique utilise des instructions mathématiques déjà employées. Il affiche à l'écran, après chaque entrée de donnée, le nombre de données totales, la moyenne arithmétique, la moyenne des différences d'essais, la moyenne des différences, la somme des valeurs et la somme des carrés des valeurs. Vous trouverez dans le programme les formules qui permettent d'obtenir ces données.

Toutes ces valeurs sont présentées à l'écran avec la technique des fenêtres d'écran.

STATISTIQUES

```
100 REM -----
110 REM   S T A T I S T I Q U E S
120 REM -----
130 WINDOW 1,40,1,25
140 som=0 : do=0 : car=0 : ari=0 : ess=0 : dif=0
150 REM -----
160 REM Affichage du tableau
170 REM -----
180 CLS : x=1 : y=1 : l=17 : e=154
190 FOR n=1 TO 2
200 LOCATE x,y : PRINT CHR*(150);:GOSUB 1000: PRINT CHR*(156);
210 x=21
220 NEXT n
230 FOR x=1 TO 21 STEP 20
240 FOR y=7 TO 13 STEP 6
250 LOCATE x,y : PRINT CHR*(151);
260 GOSUB 1000 : PRINT CHR*(157);
270 NEXT y
280 NEXT x
290 l=37
300 FOR y=19 TO 25 STEP 6
310 LOCATE 1,y : PRINT CHR*(151);: GOSUB 1000:PRINT CHR*(157);
320 NEXT y
330 LOCATE 19,19 : PRINT CHR*(155);
340 LOCATE 21,19 : PRINT CHR*(155);
350 l=1 : e=149
360 FOR x=1 TO 39 STEP 38
370 FOR y=2 TO 24
380 LOCATE x,y : GOSUB 1000
390 NEXT y
400 NEXT x
410 FOR x=19 TO 21 STEP 2
420 FOR y=2 TO 18
430 LOCATE x,y : GOSUB 1000
440 NEXT y:NEXT x
450 REM -----
460 REM definition des fenetres
470 REM -----
480 WINDOW #0,2,18,2,6 : WINDOW #1,22,38,2,6
490 WINDOW #2,2,18,8,12 : WINDOW #3,22,38,8,12
500 WINDOW #4,2,18,14,18 : WINDOW #5,22,38,14,18
510 WINDOW #6,2,38,20,24
520 PRINT #0,"nombre de donnees"
530 PRINT #1," - moy. arith. -"
540 PRINT #2," - moyenne des-"
550 PRINT #2,"differ. d'essais"
560 PRINT #3," - moyenne des-";PRINT #3,"    differences"
570 PRINT #4,"somme des valeurs"
580 PRINT #5," somme des carres";:PRINT #5," des valeurs"
590 PRINT #6,TAB(13)"- donnees -"
600 PRINT #6: PRINT #6, "combien de valeurs a entrer (1 a 200)":INPUT #6,n
610 IF n<1 OR n>200 THEN 590
620 LOCATE #6,1,3 : PRINT #6,STRING*(38,32);
630 LOCATE #6,1,5 : PRINT #6,"    "
640 REM -----
650 REM saisie des donnees
```

```

660
670 FOR m=1 TO n
680 LOCATE #6,1,3 :PRINT #6,STRING$(40,32);
690 LOCATE #6,1,3 : PRINT #6,"donnee no ";m;;INPUT #6,do
700 GOSUB 1200
710 NEXT m
720 GOTO 720
1000 REM -----
1010 REM affichage des lignes du tableau
1020 REM -----
1030 FOR m=1 TO 1 : PRINT CHR$(e);: NEXT m
1040 RETURN
1100 REM -----
1110 REM position curseur et effacement
1120 REM -----
1130 LOCATE #w,2,4 : PRINT #w,STRING$(14,32);:LOCATE #w,2,4
1140 RETURN
1200 REM -----
1210 REM calcul et affichage des resultats
1220 REM -----
1230 w=0 : GOSUB 1100 : PRINT #0,m
1240 w=4 : GOSUB 1100 : som = som+do : PRINT #4,som
1250 w=5 : GOSUB 1100 : car = car+do^2 : PRINT #5,car
1260 w=1 : GOSUB 1100 : ari = som/m : PRINT #1,ari
1270 w=2:GOSUB 1100:ess=(car-(som^2)/m)/m : ess=SQR(ess):PRINT
#2,ess

1280 IF m=1 THEN 1310
1290 w=3 : GOSUB 1100 : dif=(car-(som^2)/m)/m-1
1300 IF dif>=0 THEN dif=SQR(dif):PRINT #3,dif
1310 RETURN

```

Liste des variables :

do = données
som = somme des données
dif = moyenne des différences d'essais
ari = moyenne arithmétique
car = somme des carrés des valeurs
w = canal utilisé

REMARQUES

La commande WINDOW

Vous pouvez constater que les lignes 160 à 440 dessinent 7 fenêtres à l'écran. Mais c'est en fait la commande WINDOW qui crée réellement ces fenêtres pour l'ordinateur.

L'AMSTRAD vous permet ainsi de créer 8 fenêtre d'écran numérotés respectivement de 0 à 7.

Pour définir une fenêtre, il convient d'indiquer le numéro de canal employé (0 à 7), les limites gauche et droite, ainsi que les limites supérieures et inférieures.

Ex : MODE 1 : WINDOW £1,20,40,13,25

Après avoir fait un CLS, essayez d'écrire une ligne assez longue, et vous remarquerez que vous ne pouvez plus écrire sur la partie droite de votre écran, la commande WINDOW, l'ayant "coupé" en deux.

Pour pouvoir écrire dans cette partie d'écran, il faut employer l'instruction PRINT f.

Elle va vous autoriser l'écriture dans les fenêtres définies.

Dans l'exemple précédent vous avez inscrit WINDOW # 1, suivi de coordonnées. Vous avez ainsi attribué le canal 1 à cette fenêtre. Pour y accéder il faudra préciser ce numéro de canal.

Essayez maintenant PRINT #1, "fenêtre du canal 1".

Vous verrez le message s'afficher à la droite l'écran.

De la même façon vous pourrez utiliser les

instructions :

LOCATE #1, X,Y pour positionner le curseur à la position x, y de la fenêtre (0,0 étant la position supérieure gauche de celle-ci).

INPUT #1,x affiche le point d'interrogation dans la fenêtre du canal 1, et attend l'entrée d'une donnée

LIST #1 liste le programme en mémoire dans la fenêtre du canal 1.

Vous l'avez sans doute compris, toutes les instructions d'affichage s'utilisent avec la même syntaxe, après la virgule suivant le numéro du canal.

Huit fenêtres d'affichages peuvent être ainsi définies, comme vous le montrent d'ailleurs les lignes 480 à 630.

ATTENTION n'utilisez pas comme numéro de canal une valeur supérieure à 7, le canal 9 étant destiné à la gestion du magnétophone et le 8 à celle de l'imprimante, mais nous aurons l'occasion d'en reparler.

CHAPITRE 4

LE GRAPHISME

Nous vous proposons deux programmes de graphisme qui, bien qu'ils n'utilisent pas toutes les fonctions graphiques de l'AMSTRAD, doivent vous donner par la suite les éléments nécessaires pour tirer le meilleur parti de votre ordinateur.

1) l'histogramme

Un histogramme est une représentation graphique d'un ensemble de données. Pour le représenter, nous n'utilisons pas la haute résolution mais des pavés de couleurs mis en oeuvre par la commande CHR\$.

Le problème d'une représentation par histogramme est que le programme doit être capable d'accepter des suites de valeurs fort diverses. Par exemple vous entrerez 15 valeurs inférieures à 0, puis dans un autre cas 15 valeurs supérieures à 100. Votre programme doit être capable de traiter ces deux types de données pour qu'il présente un intérêt quelconque.

Liste des variables :

n = nombre de données
d (d) = tableau des données
a = plus grand élément
hm = hauteur maximum
ec = échelle
h (b) = hauteur attribuée à chaque donnée
f = couleur des pavées
b = position sur l'axe des x des pavés
c = position sur l'axe des y des pavés

REMARQUES :

- L'instruction MODE

Vous savez sans doute que cette commande vous permet de choisir la taille de votre écran avec :

MODE 0 : 20 caractères par lignes
MODE 1 : 40 caractères par lignes
MODE 2 : 80 caractères par lignes et un affichage haute résolution de 640 x 400 points.

Sur ces trois modes, vous disposez à chaque fois de 25 lignes.

Dans notre programme on utilise le mode 0, qui nous autorise 20 colonnes de 25 lignes. Nous fixons donc à 20, la hauteur maximum de l'histogramme afin de ne pas envahir tout l'écran (ligne 260).

Après avoir déterminé le plus grand élément parmi les valeurs choisies (lignes 205 à 230), on calcule l'échelle (ligne 260).

Cette échelle permet de calculer la grandeur à l'écran de chaque valeur, afin qu'il n'y ait pas de dépassement d'écran ou de valeur trop petite (ligne 340 à 360).

vous noterez à la ligne 350, la présence de l'instruction INT. En effet on ne peut afficher des fractions de pavés CHR\$.

Pour terminer sur ce programme, les lignes 510 à 570 déterminent la couleur de chaque colonne de l'histogramme.

HISTOGRAMME

```
10 REM -----
20 REM  H I S T O G R A M M E
30 REM -----
40 MODE 1 : BORDER 0:INK 0,0 : CLS
50 DIM d(15),h(15)
100 REM -----
110 REM acquisition des donnees
120 REM -----
130 PRINT : INPUT "nombre de donnees (1 a 15)";n
140 n=INT(n):IF n<1 OR n>15 THEN 40
150 PRINT
160 FOR b=1 TO n
170 PRINT "donnee no ";b;:INPUT d(b)
180 NEXT b
190 REM -----
200 REM recherche du plus grand element
205 a=0
210 FOR b=1 TO n
220 IF d(b)>a THEN a=d(b)
230 NEXT b
240 REM -----
250 REM choix de l'echelle
260 hm=20 : ec=hm/a
300 REM -----
310 REM calcul de la hauteur de chaque element
320 REM -----
340 FOR b=1 TO n
350 h(b)=INT(ec*d(b))
360 NEXT b
400 REM -----
410 REM dessin de l'histogramme
420 REM -----
430 CLS : MODE 0 : LOCATE 2,3
440 FOR x=3 TO 24
450 LOCATE 2,x : PRINT CHR$(157);
460 NEXT x
470 LOCATE 2,24 : PRINT CHR$(147);
480 FOR x=3 TO 19
490 LOCATE x,24 : PRINT CHR$(158);
500 NEXT x
510 FOR b=1 TO n
520 FOR c=1 TO h(b)
530 LOCATE b+2,24-c
540 f=b : IF f>10 THEN f=f-10
550 IF f=5 THEN f=1
560 PEN f : PRINT CHR$(143);
570 NEXT c : NEXT b
580 GOTO 580
```


2) Une table à dessin

Le programme qui suit permet de dessiner à l'écran point par point. Avec les quatre flèches vous tracez dans les directions gauche, droite, haut et bas, et avec les touches a,s,z,x, vous dirigez le curseur selon les quatre diagonales.

Le mode haute résolution est utilisé, et les lignes coordonnées du point sont affichées en permanence dans une fenêtre préalablement définie.

Deux sous programmes permettent de tracer des lignes et des cercles.

TABLE A DESSIN

```
10 REM -----
20 REM T A B L E   A   D E S S I N
30 REM -----
100 CLS: MODE 1 : INK 0,0
102 PRINT "Utilisez les fleches pour les direction":PRINT "droite,gauche,haut et
bas": PRINT
105 PRINT "les touches A,S,Z,X pour les diagonales":PRINT
107 PRINT "et D pour effacer,C pour cercle,L pour ligne et F pour finir":PRINT
110 PRINT : PRINT "origine du dessin X,Y"
120 PRINT "avec x>=0 et x<=640"
130 PRINT "& y>=80 et y<=400"
140 INPUT x,y
150 IF x>640 OR x<0 OR y>400 OR y<80 THEN 100
160 CLS : MODE 2 : z=1
170 WINDOW #1,1,80,21,25
172 ORIGIN 0,0
175 PLOT x,y
230 a$=INKEY$:IF a$="" THEN 230
240 a=ASC(a$)
250 IF a=241 THEN GOSUB 1330 : GOTO 390
260 IF a=242 THEN GOSUB 1430 : GOTO 390
270 IF a=243 THEN GOSUB 1530 : GOTO 390
280 IF a=240 THEN GOSUB 1630 : GOTO 390
290 IF a=122 OR a=90 THEN GOSUB 1730 : GOTO 390
300 IF a=88 OR a=120 THEN GOSUB 1830 : GOTO 390
310 IF a=97 OR a=65 THEN GOSUB 1930 : GOTO 390
320 IF a=83 OR a=115 THEN GOSUB 2030 : GOTO 390
350 IF a=68 OR a=100 THEN GOSUB 1000 : GOTO 230
360 IF a=67 OR a=99 THEN GOSUB 1100 : GOTO 390
370 IF a=76 OR a=108 THEN GOSUB 1200 : GOTO 390
380 IF a=70 AND a=102 THEN 2270
390 LOCATE #1,60,1:PRINT #1,STRING$(19,32);:LOCATE #1,60,1:PRINT #1,"X = ";x;"
Y = ";y
410 GOTO 230
1000 REM -----
1010 REM effacement
1020 REM -----
1030 IF z=1 THEN z=0 ELSE z=1
1040 IF z=0 THEN LOCATE #1,1,1:PRINT #1,"effacement";
1050 IF z=1 THEN LOCATE #1,1,1:PRINT #1,STRING$(11,32);
1060 RETURN
1100 REM -----
1110 REM trace du cercle
1120 REM -----
1130 LOCATE #1,2,2 : PRINT #1,"origine du cercle"
1140 INPUT #1,"avec 0<x<640 et 80<y<400";t,u
1150 IF t>639 OR t<1 OR u>399 OR u<81 THEN 1130
1160 LOCATE #1,2,4 : INPUT #1,"rayon du cercle";r
1170 IF t+r>640 OR t-r<0 OR u+r>400 OR u-r<80 THEN 1160
1180 n=r*5 : b=6.28/n
1185 FOR j=1 TO n:m=t+r*COS(b*j):o=u-r*SIN(b*j)
1190 PLOT m,o,z
1195 NEXT j : GOSUB 2200 : RETURN
1200 REM -----
1210 REM trace de la ligne
1220 REM -----
```

```

1230 LOCATE #1,2,2 : PRINT #1,"coordonnees d'arrivee de la ligne"
1240 INPUT #1,"avec 0=<x=<640 et 80=<y=<400";x,y
1250 IF x>640 OR x<0 OR y>400 OR y<80 THEN 1230
1260 DRAW x,y
1270 GOSUB 2200
1280 RETURN
1300 REM -----
1310 REM en bas
1320 REM -----
1330 IF y=80 THEN 1350
1340 y=y-1 : PLOT x,y,z : PLOT x+1,y,z
1350 RETURN
1400 REM -----
1410 REM a gauche
1420 REM -----
1430 IF x=0 THEN 1450
1440 x=x-1 : GOSUB 2130
1450 RETURN
1500 REM -----
1510 REM a droite
1520 REM -----
1530 IF x=640 THEN 1550
1540 x=x+1 : GOSUB 2130
1550 RETURN
1600 REM -----
1610 REM en haut
1620 REM -----
1630 IF y=400 THEN 1650
1640 y=y+1 : PLOT x,y,z : PLOT x+1,y,z
1650 RETURN
1700 REM -----
1710 REM en bas a gauche
1720 REM -----
1730 IF x=0 OR y=80 THEN 1750
1740 x=x-1 : y=y-1 : GOSUB 2130
1750 RETURN
1800 REM -----
1810 REM en bas a droite
1820 REM -----
1830 IF x=640 OR y=80 THEN 1850
1840 x=x+1 : y=y-1 : GOSUB 2130
1850 RETURN
1900 REM -----
1910 REM en haut a gauche
1920 REM -----
1930 IF x=0 OR y=400 THEN 1950
1940 x=x-1 : y=y+1 : GOSUB 2130
1950 RETURN
2000 REM -----
2010 REM en haut a droite
2020 REM -----
2030 IF x=640 OR y=400 THEN 2050
2040 x=x+1 : y=y+1 : GOSUB 2130
2050 RETURN
2100 REM -----
2110 REM trace du point
2120 REM -----
2130 PLOT x,y,z : RETURN
2200 REM -----
2210 REM effacement du texte
2220 REM -----
2230 FOR j=2 TO 5
2240 LOCATE #1,1,j : PRINT #1,STRING$(40,32);
2250 NEXT j
2260 RETURN
2270 END

```

COMMENTAIRES :

- L'instruction ASC

Cette commande permet de connaître le code ASCII d'un caractère. A la ligne 240, cette instruction convertit la variable a\$ en sa valeur ASCII. En fonction de cette valeur le programme ira se brancher à la partie concernée, avec :

240 à 243	flèches
88 et 120,	
65 et 97	
90 et 122	qui sont les 8 codes ASCII majuscules
83 et 115	et minuscules de a,s,x,z
67 et 99	codes ASCII de c pour tracer un cercle
76 et 108	codes ASCII de L pour tracer une ligne
70 et 102	codes ASCII de F pour arrêter.

Contrairement aux nombreuses publicités faites sur l'AMSTRAD en FRANCE (et malheureusement aussi sur certains ouvrages récents), votre ordinateur possède, en mode 2, une résolution de 640 X 400 points. Ce qui est nettement mieux que les 640 X 200 annoncés.

Le point de coordonnées 0,0 se situe en bas à gauche de votre écran et le point de coordonnées 640,400 se situe en haut à droite de cet écran.

- L'instruction ORIGIN

Cette instruction permet de fixer l'origine à l'écran. A la ligne 172 l'origine est fixée au bord inférieur gauche de l'écran. C'est-à-dire au point de coordonnée 0,0.

Attention si vous fixez par exemple l'origine au point 320,200 (au milieu de l'écran) pour dessiner un point a cet endroit, vous devrez écrire PLOT 0,0 et non PLOT 320,200.

- L'instruction PLOT

Cette commande permet de dessiner un point à l'écran. A la ligne 2130, vous trouverez :

PLOT x,y,z

ou x et y sont les coordonnées du point à dessiner et z, la couleur du point. Examinez les lignes 1000 à

1060 pour voir comment on se sert, avec z, de l'instruction PLOT pour effacer un tracé.

- L'instruction DRAW

Cette instruction trace des lignes à l'écran, à partir de la position du curseur et jusqu'à la position précisée dans la commande DRAW.

Exemple : ORIGIN 0,0
DRAW 640,400 trace une diagonale
sur l'écran (voir les lignes 1200 à 1270).

La combinaison de ces instructions donne des possibilités illimitées de graphisme. regardez par exemple les lignes 1100 à 1195, qui définissent une véritable fonction CERCLE ou vous aurez simplement à fixer le centre du cercle et le rayon.

CHAPITRE 5

LA GESTION DE FICHIERS

1) Les notions de données et de fichier

L'intérêt d'un ordinateur réside principalement dans sa possibilité de manipuler un grand nombre de données.

On appelle donnée, toute valeur qui n'est pas déterminée dans le programme avant son exécution (ou dans le listing)

```
Ex : 10 A = 10
      20 B$ = "janvier"
      30 INPUT A$
      40 INPUT X
```

Dans cet exemple, A\$ et X sont des données fixées par l'utilisateur du programme.

L'ensemble de ces données va constituer un fichier.

Pour que votre ordinateur soit d'une quelconque utilité dans ce domaine, il doit être capable de conserver l'ensemble de ces données sur un support (K7, disquette ou autre), et de pouvoir ensuite les relire.

Si ce n'est pas le cas, vous hésitez certainement à créer des fichiers importants, si ceux-ci sont perdus lorsque vous éteignez votre appareil.

L'AMSTRAD peut conserver l'ensemble de ces données avec des instructions simples, mais organisées de façon similaire à celles de systèmes plus puissants. C'est une très bonne introduction en la matière, surtout si vous faites par la suite, comme nous vous le souhaitons, l'acquisition d'un lecteur de disquette.

2) La gestion de fichiers

Pour illustrer notre propos, prenons un exemple. Vous désirez inscrire sur votre agenda le nom, le prénom et le numéro de téléphone d'une nouvelle relation.

Je suppose que vous avez compris que l'AGENDA sera votre fichier, et les noms, prénoms et numéros de téléphone vos données.

Après avoir pris votre stylo, vous allez :

- 1 ouvrir votre AGENDA
- 2 écrire les nom, prenom et numéro de téléphone
- 3 refermer votre AGENDA

Sur l'AMSTRAD ces trois opérations se traduiront par :

- ```
100 OPENOUT "AGENDA" (ouvrir le fichier AGENDA)
110 WRITE #9, nom$,prenom$,téléph$ (écrire les données
 sur la K7)
120 CLOSEOUT
```

A la ligne 110, #9 précise le canal sur lequel on écrit les données. En l'occurrence, 9 est le numéro du canal réservé au magnétophone.

Vous souhaitez maintenant appeler votre ami (e), vous allez :

- 1 ouvrir votre AGENDA
- 2 lire les références (nom, prenom et téléphone)
- 3 refermer votre AGENDA

Ce qui se traduit en BASIC AMSTRAD par :

- ```
200 OPENIN "AGENDA" (ouvrir le fichier AGENDA)
210 INPUT #9, nom$,prenom$,téléph$ (lire les données
                                     sur la K7)
220 CLOSEIN
```

INPUT #9, transfère les données présente sur la K7, dans la mémoire de l'ordinateur.

Vous aurez sans doute compris que :

- OPENOUT et CLOSEOUT s'emploient lorsque l'on écrit sur la K7,
- OPENIN et CLOSEIN s'utilisent pour lire les données sur la K7.

Si vous ne précisez pas le nom du fichier à la ligne 200 (OPENIN ""), l'AMSTRAD chargera en mémoire le premier fichier qu'il rencontrera pourvu qu'il soit du même type.

En effet, il est important de lire le MEME TYPE et le MEME NOMBRE de données que celles qui ont été enregistrées sur la K7.

Enfin, on peut, en mettant un point d'exclamation juste avant le nom du fichier dans les instructions OPENOUT et OPENIN, supprimer les messages de manipulations du magnétophone.

Ex : OPENOUT "! AGENDA"

3) Une application directe

Reprenons l'exemple de notre AGENDA

Avec les lignes 100 à 120, nous disposons d'un fichier ne comportant qu'une seule fiche, comportant elle-même trois données, ce qui est naturellement tout à fait insuffisant.

Nous allons d'abord augmenter le nombre de données avec par exemple.

no\$	pour	nom
pr\$	"	prénom
nu\$	"	numéro de la rue
ru\$	"	nom de la rue
vi\$	"	localité
te\$	"	téléphone
pf\$	"	profession

Notre AGENDA, avec ces nouveaux éléments, comportent des fiches bien documentées dignes d'un véritable fichier client.

La transformation en BASIC ne pose aucun problème puisqu'il suffit de modifier simplement la ligne 110 :

```
110 WRITE #9, no$,pr$,nu$,ru$,vi$,te$,pf$
```

Et pour relire ces données, la ligne 210 :

```
210 INPUT #9, no$,pr$,nu$,ru$,vi$,te$,pf$
```

Notre fichier reste malgré tout insuffisant, et c'est le moins que l'on puisse dire, car il ne comporte qu'une seule fiche.

Pour y remédier, nous allons utiliser l'instruction DIM et une boucle FOR NEXT.

Ce qui donne pour 50 fiches :

```
10 DIM no$(50),pr$(50),nu$(50),ru$(50),vi$(50),te$(50),pf$(50)
```

Maintenant que nous avons créé un fichier de 50 fiches, il ne nous reste plus qu'à les enregistrer (évidemment après les avoir remplies) :

```
100 OPENOUT "AGENDA"  
105 FOR n = 1 TO 50  
110 WRITE *9, no$(n),pr$(n),nu$(n),ru$(n),etc...  
115 NEXT n  
120 CLOSEOUT
```

Nous venons ainsi de stocker, avec 5 lignes de programmes, 50 fiches.

Pour les relires, il convient d'employer le même procédé.

```
200 OPENIN "AGENDA"  
205 FOR n = 1 TO 50  
210 INPUT *9, no$(n),pr$(n),nu$(n),ru$(n),etc...  
215 NEXT n  
220 CLOSEIN
```

Pourtant tout n'est pas encore parfait, et nous allons apporter une dernière amélioration à notre exemple de gestion de fichier.

Dans le programme précédent, si l'utilisateur ne remplit que 10 fiches, 50 seront cependant enregistrées sur la K7 dont 40 seront vides. C'est à la fois une perte de temps et de place sur la bande magnétique.

Nous allons donc créer ce que l'on appelle un pointeur dont le rôle sera de nous indiquer le nombre exact de fiches remplies. La valeur de ce pointeur sera sauvée sur la K7 juste avant le fichier.

Si P contient la valeur du pointeur, notre programme devient pour sauvegarder les données :

```
100 OPENOUT "AGENDA"
```

```

102 WRITE *9, P
105 FOR N = 1 TO P
110 WRITE *9, no$(n), pr$(n), nu$(n), etc...
115 NEXT N
120 CLOSEOUT

```

A la lecture, il va falloir lire d'abord la valeur du pointeur, et ensuite les fiches enregistrées. Le programme de lecture se déduit assez facilement du précédent :

```

200 OPENIN "AGENDA"
202 INPUT *9, P
205 FOR N = 1 TO P
210 INPUT *9, no$(n), pr$(n), nu$(n), etc...
215 NEXT N
220 CLOSEIN

```

Il est tout a fait possible de procéder autrement. On peut par exemple faire des enregistrements fiche par fiche. Cela évite, comme dans le programme précédent, de recopier l'ensemble du fichier à chaque fois que l'on veut rentrer qu'une seule fiche.

Je ne vous conseille pas cette solution. Elle impose en effet beaucoup trop de manipulations de la K7 supportant les données et augmente ainsi les risques d'erreurs.

Je vous recommande par contre de donner des noms simples à vos fichiers, et de les noter.

Il est également plus sage de repérer le début et la fin de l'enregistrement, à l'aide du compteur dont vous disposez sur l'AMSTRAD.

Pour qu'une gestion de fichiers soit efficace il est indispensable d'organiser de façon très rigoureuse le K7 sur lesquelles on doit trouver ces fichiers. En particulier il vaut mieux ne pas mettre sur la même K7, le programme et le fichier.

Pour les fichiers importants, on prendra le soin d'effectuer une sauvegarde de secours sur une deuxième K7.

4) Le programme AGENDA

C'est une application directe de ce qui précède. Il possède tous les éléments indispensables à une gestion de fichier, mais peut naturellement faire l'objet d'amé-

liorations, et, à votre gré, être adapté à un usage professionnel. Nous vous fournissons l'ossature, à vous d'exercer vos talents.

Le programme est cependant tout à fait utilisable dans sa version de base.

Il vous est possible de travailler sur 50 fiches, comportant chacune 7 rubriques.

Le menu vous offre 5 options :

- créer une nouvelle fiche
- retrouver une fiche
- enregistrer le fichier sur la K7
- lire le fichier sur la K7
- lister le fichier à l'écran

Vous pourrez par exemple y'ajouter un sous programme de correction de fiches, une sortie imprimante ou tout autre option à votre gout.

Pour terminer sur ce programme, vous remarquerez que les lignes 310 à 430 permettent de remplir les fiches en utilisant les tableaux, une boucle FOR NEXT et l'instruction INPUT.

GESTION DE FICHER

```
2 REM -----
5 REM GESTION DE FICHIERS
7 REM -----
10 MODE 1 : INK 0,0 :CLS
20 DIM no$(50),pr$(50),nu$(50),ru$(50),vi$(50),te$(50),pf$(50),b(50)
30 v=0
100 REM AFFICHAGE DES OPTIONS DU PROGRAMME1
110 CLS : FOR N=1 TO 40 : PRINT CHR$(42); : NEXT N
115 PRINT
120 PRINT TAB(10)"# # M E N U # #"
130 PRINT : FOR n=1 TO 40 : PRINT CHR$(42);NEXT n
140 PRINT : PRINT : PRINT "CREER UNE NOUVELLE FICHE < 1 >"
150 PRINT : PRINT "RECHERCHER UNE FICHE < 2 >"
160 PRINT : PRINT "IMPRIMER LE FICHER SUR LA K7 < 3 >"
170 PRINT : PRINT "LIRE LE FICHER SUR LA K7 < 4 >"
175 PRINT : PRINT "LISTER LE FICHER A L' ECRAN < 5 >"
180 PRINT : PRINT : PRINT TAB(14)"votre choix ?"
190 a$=INKEY$:IF a$="" THEN 190
200 IF a$(">")"1" AND a$(">")"2" AND a$(">")"3" AND a$(">")"4" AND a$(">")"5" THEN 190
203 REM
205 REM CHOIX DU SOUS PROGRAMME SELON LA VALEUR DE A$
207 REM
208 a=VAL(a$)
210 ON a GOTO 300,500,700,900,1100
300 REM -----
302 REM CREATION DES FICHES
305 REM -----
310 v=v+1 : FOR n=v TO 50 : CLS
320 PRINT TAB(12)"Creation de fichiers"
330 PRINT : PRINT : INPUT "nom ";no$(n)
340 PRINT : INPUT "premon ";pr$(n)
350 PRINT : INPUT "No , nom de la rue ";nu$(n),ru$(n)
360 PRINT : INPUT "cde postal et ville ";vi$(n)
370 PRINT : INPUT "telephone ";te$(n)
380 PRINT : INPUT "profession ";pf$(n)
390 PRINT : PRINT "Une autre fiche ? (O/N)"
400 a$=INKEY$ : IF a$="o" OR a$="O" THEN 430
410 IF a$="n" OR a$="N" THEN 420 ELSE 400
420 v=n : n=50
430 NEXT n
440 IF v=50 THEN CLS : PRINT"fichier complet "
450 FOR n=1 TO 50:NEXT n : REM attente
460 GOTO 100 : REM retour au menu
500 REM -----
505 REM Recherche d'une fiche
507 REM -----
510 IF v=0 THEN GOSUB 3000 : GOTO 100
520 CLS : PRINT "Vous connaissez : "
530 PRINT : PRINT : PRINT "le nom < 1 >"
540 PRINT : PRINT "le prenom < 2 >"
550 PRINT : PRINT "le numero de telephone < 3 >"
560 PRINT : PRINT : PRINT TAB(14)"Votre choix ?"
570 a$=INKEY$ : a = VAL(a$) : IF a<1 OR a>3 THEN 570
580 CLS : ON a GOTO 590, 600, 610
590 INPUT "nom ";a$ : GOTO 620
600 INPUT "prenom ";a$ : GOTO 620
```

```

610 INPUT "numero de telephone ";a$
620 x=1 : FOR n = 1 TO v : b(x)=0
630 IF a$=no$(n) OR a$=pr$(n) OR a$=te$(n) THEN b(x)=n : x=x+1
640 NEXT n : IF x=1 THEN GOSUB 3500 : GOTO 100
650 FOR n=1 TO x-1 : CLS
660 PRINT : PRINT "nom : "; no$(b(n)) : PRINT : PRINT "prenom : ";pr$(b(n))
670 PRINT : PRINT "adresse : ";inu$(b(n));", ";iru$(b(n)):PRINT
675 PRINT "Ville : ";vi$(b(n)) : PRINT
680 PRINT "telephone : ";ite$(b(n)) : PRINT : PRINT "profession ";pf$(b(n))
690 a$=INKEY$ : IF a$="" THEN 690
695 NEXT n : GOTO 100
700 REM -----
705 REM impression du fichier sur cassette
707 REM -----
710 CLS : PRINT : INPUT "nom du fichier ";a$ : IF LEN (a$) >16 THEN 710
720 PRINT : PRINT "PREPAREZ VOTRE CASSETTE "
725 a$="!" +a$
730 PRINT : PRINT "ET APPUYEZ SUR LES TOUCHES REC ET PLAY"
732 PRINT : PRINT"APPUYEZ SUR UNE AUTRE TOUCHE"
735 B$=INKEY$:IF B$ = "" THEN 735
737 c$=RIGHT$(A$,LEN(A$)-1):PRINT:PRINT "Sauvegarde du fichier ";c$
740 OPENOUT a$
750 WRITE #9,v
760 FOR n=1 TO v
770 WRITE #9, no$(n),pr$(n),nu$(n),ru$(n),vi$(n),te$(n),pf$(n)
780 NEXT n
790 CLOSEOUT
800 GOTO 100
900 REM -----
905 REM lecture du fichier sur cassette
907 REM -----
910 CLS
930 PRINT : PRINT "PREPAREZ VOTRE CASSETTE"
940 PRINT : PRINT "ET APPUYEZ SUR LA TOUCHE PLAY"
945 PRINT
950 OPENIN ""
960 INPUT #9,v
970 FOR n=1 TO v
980 INPUT #9,no$(n),pr$(n),nu$(n),ru$(n),vi$(n),te$(n),pf$(n)
990 NEXT n
1000 CLOSEIN : GOTO 100
1100 REM -----
1105 REM Affichage du fichier a l'ecran
1107 REM -----
1110 IF v=0 THEN GOSUB 3000 : GOTO 100
1120 FOR n=1 TO v : CLS
1130 PRINT : PRINT "nom : ";no$(n)
1140 PRINT : PRINT "prenom : ";pr$(n)
1150 PRINT : PRINT "adresse : ";nu$(n);", ";iru$(n)
1155 PRINT : PRINT "ville : ";vi$(n)
1160 PRINT : PRINT "telephone : ";ite$(n)
1170 PRINT : PRINT "profession : ";ipf$(n)
1180 a$=INKEY$ : IF a$="" THEN 1180
1190 NEXT n : GOTO 100
3000 REM -----
3005 REM fichier vide
3007 REM-----
3010 CLS : PRINT "fichier vide, option impossible "
3020 FOR n=1 TO 500 : NEXT n
3030 RETURN

3500 REM -----
3505 REM recherche negative d'une fiche
3507 REM -----
3510 CLS : PRINT "cette fiche n'existe pas "
3520 FOR n=1 TO 500
3530 NEXT : RETURN

```

5) Le programmé compte chèque

Ce programme vous permet de gérer un compte chèque de 100 chèques avec 7 écritures par fiches.

Il comporte, par rapport au listing précédent, deux options supplémentaires :

- une possibilité de correction qui s'avère indispensable sur des programmes de fichiers un peu sophistiqués,

- une sortie imprimante qui vous permet de conserver une trace écrite de votre travail.

Vous noterez, dans le cours du programme, l'utilisation intensive de l'instruction PRINT USING, tant à l'écran que sur l'imprimante, qui améliore considérablement la présentation des chiffres, en faisant, dans le cas présent, une tabulation automatique autour de la virgule

Vous remarquerez également que pour le premier chèque, le programme vous demande le solde du compte, étant bien entendu qu'il ne peut le deviner. Pour les autres chèques, le solde est calculé par l'ordinateur, ce qui est la moindre des choses.

Nous vous laissons maintenant regarder ce listing, si le coeur vous en dit. Il ne doit poser aucune difficulté, du moins nous l'espérons, au programmeurs avertis que vous êtes devenus.

COMPTE CHEQUES

```
10 REM -----
20 REM C O M P T E   C H E Q U E S
30 REM -----
50 v=1: DIM da$(100), cr(100), de(100), so(100), nu$(100), ob$(100)
100 REM -----
110 REM choix des options
120 REM -----
130 CLS: PRINT STRING$(40, 230);
140 PRINT: PRINT TAB(10) CHR$(243); " O P T I O N S "; CHR$(242)
150 PRINT: PRINT STRING$(40, 230)
160 FOR n=2 TO 4: FOR m=1 TO 40 STEP 39
170 LOCATE m, n : PRINT CHR$(230);
180 NEXT m : NEXT n
190 PRINT: PRINT: PRINT: PRINT " Saisie de compte"; TAB(30) CHR$(243); " 1"
200 PRINT: PRINT " Correction"; TAB(30) CHR$(243); " 2"
210 PRINT: PRINT " Recherche d'un cheque"; TAB(30) CHR$(243); " 3"
220 PRINT: PRINT " Listing du fichier"; TAB(30) CHR$(243); " 4"
230 PRINT: PRINT " Sortie imprimante"; TAB(30) CHR$(243); " 5"
240 PRINT: PRINT " Enregistrement sur K7"; TAB(30) CHR$(243); " 6"
250 PRINT: PRINT " Lecture sur K7"; TAB(30) CHR$(243); " 7"
260 PRINT: PRINT: PRINT TAB(14) "Votre choix ?"
270 a$=INKEY$: IF a$="" THEN 270
280 a=VAL(a$): IF a<1 OR a>7 THEN 100
290 ON a GOTO 330, 710, 500, 1000, 1100, 1300, 1500
300 REM -----
310 REM saisie des cheques
320 REM -----
330 IF v=100 THEN GOSUB 2000 : GOTO 100
340 FOR n=v TO 100 : CLS : PRINT TAB(14) "cheque no : "; n
350 PRINT: PRINT: INPUT "date d'emission   "; da$(n)
370 PRINT: INPUT "numero de cheque   "; nu$(n)
380 PRINT: INPUT "credit             "; cr(n)
390 PRINT: INPUT "debit              "; de(n)
400 IF n>1 THEN GOTO 420
410 PRINT: INPUT "solde               "; so(n): GOTO 440
420 so(n)=so(n-1)+cr(n)-de(n)
430 PRINT: PRINT "solde               ";: PRINT USING "#####.##-"; so(n)
440 PRINT: INPUT "objet               "; ob$(n)
450 PRINT: PRINT: a=0: GOSUB 2200
460 IF a=1 THEN v=n+1: n=100
470 NEXT n : GOTO 100
500 REM -----
510 REM recherche d'un cheque
520 REM -----
530 IF v=1 THEN GOSUB 2100 : GOTO 100
540 CLS : PRINT : PRINT " vous connaissez ";: PRINT: PRINT
550 PRINT "1 --- le numero du cheque": PRINT: PRINT "2 --- le montant du cheque"
560 PRINT: PRINT "3 --- le solde"
570 PRINT: INPUT n: n=INT(n): IF n<1 OR n>3 THEN 570
580 CLS : a=0 : ON n GOTO 590, 625, 660
590 INPUT "entrez le numero du cheque"; nr$
600 FOR j=1 TO v-1
610 IF nr$=nu$(j) THEN n=j: j=v-1 : a=1
620 NEXT j : GOTO 690
625 INPUT "entrez le montant du cheque "; dr
630 FOR j=1 TO v-1
```

```

640 IF dr=de(j) THEN n=j:j=v-1 : a=1
650 NEXT j : GOTO 690
660 INPUT "entrez le solde";sr
670 FOR j=1 TO v-1
680 IF sr=so(j) THEN n=j:j=v-1 : a=1
685 NEXT j
690 IF a=0 THEN GOSUB 2300 : GOTO 100
700 CLS : GOSUB 2400 : GOTO 100
710 REM -----
720 REM correction d'un cheque
730 REM -----
740 IF v=1 THEN GOSUB 2100 : GOTO 100
750 CLS:PRINT:INPUT "numero du cheque a rectifier";ch
760 IF ch<0 OR ch>v-1 THEN GOSUB 2310 : GOTO 750
770 CLS : PRINT TAB(14)"cheque no : ";ch
780 PRINT:PRINT:PRINT "dte emis. : ";da$(ch);" ";;INPUT a$
790 da$(ch)=a$
800 PRINT:PRINT:PRINT"num.cheq. : ";nu$(ch);" ";;INPUT a$
810 nu$(ch)=a$
820 cr(0)=cr(ch)
830 PRINT:PRINT:PRINT"credit      : ";cr(ch);" ";;INPUT x
840 t=cr(0)-x:
850 cr(ch)=x
860 de(0)=de(ch)
870 PRINT:PRINT:PRINT"debit      : ";de(ch);" ";;INPUT x
880 u=de(0)-x
890 de(ch)=x
900 so(0)=so(ch)
910 so(ch)=so(0)-t+u
920 FOR z=ch+1 TO v-1 : so(z)=so(z-1)+cr(z)-de(z):NEXT z
930 PRINT:PRINT:PRINT"solde      : ";so(0);" ";;so(ch)
940 PRINT:PRINT:PRINT"objet      : ";ob$(ch);" ";;INPUT a$
950 ob$(ch)=a$
960 GOSUB 2340 : GOTO 100
1000 REM -----
1010 REM listing du fichier
1020 REM -----
1030 CLS : IF v=1 THEN GOSUB 2100
1040 FOR n=1 TO v-1:CLS
1050 LOCATE 5,24:PRINT "tapez sur enter pour sortir"
1070 PRINT TAB(3)"sur une autre touche pour continuer":LOCATE 1,1
1080 GOSUB 2400 : IF a$=CHR$(13) THEN n=v-1
1090 NEXT n : GOTO 100
1100 REM -----
1110 REM sortie imprimante
1120 REM -----
1125 CLS
1130 PRINT #8,STRING$(80,"-");PRINT #8,TAB(6)"date";TAB(17)"num.ch.";
1140 PRINT #8,TAB(30)"credit";TAB(43)"debit";TAB(56)"solde";TAB(71)"objet
1150 PRINT #8, STRING$(80,"-")
1160 FOR n=1 TO v-1 : CLS : PRINT TAB(6)"impression cheque no : ";n
1170 PRINT #8,TAB(3)da$(n);TAB(14)"I";TAB(16)nu$(n);TAB(26)"I";
1180 PRINT #8,TAB(27)" ";;PRINT #8,USING"#####.##";cr(n);
1190 PRINT #8,TAB(39)"I";TAB(40)" ";;PRINT #8,USING"#####.##";de(n);
1200 PRINT #8,TAB(52)"I";TAB(53)" ";;PRINT #8,USING"#####.##";so(n);
1210 PRINT #8,TAB(65)"I";TAB(68) ob$(n)
1220 NEXT n : GOTO 100
1300 REM -----
1310 REM enregistrement sur k7
1320 REM -----

```

```

1330 CLS:IF v=1 THEN GOSUB 2100 : GOTO 100
1340 PRINT "Enregistrement du fichier sur k7"
1350 PRINT : INPUT "nom du fichier";a$:IF LEN (a*)>16 THEN 1330
1355 a$="!"*a$
1360 PRINT:PRINT"preparez votre K7":PRINT
1362 PRINT "et appuyez sur les touches REC et play ":PRINT "de votre magnetophon
e,"
1365 PRINT "puis sur une autre touche"
1367 b$=INKEY$:IF b$="" THEN 1367
1369 PRINT : c$=RIGHT$(a$,LEN(a$)-1): PRINT "Enregistrement du fichier ";c$
1370 OPENOUT a$
1380 WRITE #9,v
1390 FOR n=1 TO v-1
1400 WRITE #9,da$(n),nu$(n),cr(n),de(n),so(n),ob$(n)
1410 NEXT n
1420 CLOSEOUT
1430 GOTO 100
1500 REM -----
1510 REM lecture sur k7
1515 REM -----
1520 CLS:PRINT:PRINT:PRINT"lecture sur K7":PRINT:PRINT"preparez votre k7":PRINT
1525 OPENIN ""
1530 INPUT #9,v
1540 FOR n=1 TO v-1
1550 INPUT #9,da$(n),nu$(n),cr(n),de(n),so(n),ob$(n)
1560 NEXT n
1565 CLOSEIN
1570 GOTO 100
2000 REM -----
2010 REM fichier cloture
2020 REM -----
2030 PRINT:PRINT"fichier complet"
2040 PRINT"vous ne pouvez prendre cette option"
2050 FOR n=1 TO 500:NEXT n
2060 RETURN
2100 REM -----
2105 REM choix des options
2110 REM -----
2120 CLS : PRINT:PRINT"fichier vide"
2130 PRINT"vous ne pouvez choisir cette option"
2140 FOR n=1 TO 500 : NEXT n
2150 RETURN
2200 REM -----
2210 REM message et test
2220 REM -----
2230 PRINT "on continue (o/n)";
2240 a$=INKEY$:IF a$="" THEN 2240
2250 b=ASC(a$):IF b=79 OR b=111 THEN 2280
2260 IF b=78 OR b=110 THEN a=1:GOTO 2280
2270 GOTO 2240
2280 RETURN
2300 REM -----
2310 REM recherche de cheque infructueuse
2320 REM -----
2330 PRINT:PRINT"Ce cheque ne figure pas dans le fichier"
2340 FOR j=1 TO 500 : NEXT j
2350 RETURN
2400 REM -----
2410 REM affichage des rubriques
2420 REM -----
2430 PRINT TAB(13)"Cheque no ";n

```

```
2440 PRINT:PRINT:PRINT"date d'emission   : ";da$(n)
2450 PRINT:PRINT"numero du cheque   : ";nu$(n)
2460 PRINT:PRINT"credit               : ";cr$(n)
2470 PRINT:PRINT"debit                : ";de$(n)
2480 PRINT:PRINT"solde                 : ";:PRINT USING "#####.##-";so$(n)
2490 PRINT:PRINT"objet                : ";ob$(n)
2500 a$=INKEY$:IF a$="" THEN 2500
2510 RETURN
```


CHAPITRE 6

COMPTABILITE GENERALE

Les trois programmes que nous vous présentons , facture, comptabilité générale, et bulletin de payes, ont été faits dans le cadre d'un projet de comptabilité générale pour petites entreprises.

Vous constaterez qu'ils demandent tous les trois à être améliorés, pour peu que vous fassiez un petit effort. Nous ne pouvions en effet, les programmes étant enregistrés sur une K7 fournie avec le livre, offrir un système complet de comptabilité pour des raisons évidentes.

Quoique ces programmes fonctionnent en l'état, nous vous indiquerons pour chacun d'entre eux les modifications qu'il convient de faire pour pouvoir disposer d'un système complet et efficace.

1) Facturation

Ce programme vous permet pour un même client d'établir une facture unique pour plusieurs produits.

Après vous avoir demandé les références propres à la facture (Code client, code représentant, date et numéro de facture), le logiciel vous demande à l'écran les références du produit. On y trouve :

- la référence du produit ou son numéro de code s'il y'en a un,
- la désignation, ou le nom du produit,
- le code de la T.V.A. , 1, 2 ou 3,
- le prix unitaire brut,
- le pourcentage de la remise accordée,
- la quantité.

Si vous avez d'autres produits à facturer pour votre client, le programme reprendra la série de questions à partir des références du produit, et non sur celles de la facture.

Lorsque vous n'avez plus de marchandise à facturer pour un même client, le programme affiche l'ensemble de la facture à l'écran sur 80 colonnes.

MODIFICATIONS :

Vous avez sans doute remarqué que nous avons choisi le mode 2 à l'affichage de la facture, afin de disposer de 80 colonnes. Beaucoup d'imprimantes sont en effet sur 80 colonnes, et si vous faites un sous programme pour obtenir la facture sur l'imprimante, vous aurez exactement la même présentation sur votre téléviseur ou sur le papier.

Pour imprimer la facture, il suffit donc simplement de remplacer les PRINT par des PRINT £8, et de recopier les lignes d'affichage écran.

Vous aurez également intérêt à sauvegarder votre facture sur une K7. Après avoir créé des tableaux de données par l'instruction DIM, vous pourrez sauvegarder votre facture. Nous vous conseillons de vous reporter au chapitre GESTION DE FICHIERS si vous éprouvez des difficultés pour réaliser cette partie.

Enfin, pour une meilleure présentation du programme à l'écran, il est souhaitable de pouvoir accéder à toutes ces options du logiciel par un menu initial qui offrira par exemple les choix suivants :

- Edition d'une facture
- Correction d'une facture
- Recherche d'une facture sur la K7
- Sauvegarde d'une facture sur la K7
- Impression d'une facture

Et bien d'autres possibilités dont vous ressentirez la nécessité.

FACTURATION

```
10 REM -----
20 REM   F A C T U R A T I O N
30 REM -----
40 DIM e(50),f$(50),t(50),pu(50),q(50)
50 DIM r(50),pn(50),qt(50),pht(50),tv1(50)
60 DIM tx2(50)
70 BORDER 0:INK 0,0
100 MODE 1
110 INPUT "nom de la societe (9 car max) ";so$
120 IF LEN(so$)>9 THEN 100
130 CLS
140 PRINT STRING$(5,202);TAB(8);" societe ' ' ";so$;" ' ' ";STRING$(5,202)
150 PRINT
160 INPUT "code client";a
170 INPUT "code representant";b
180 INPUT "date";c$
190 INPUT "n0. de facture";d
200 FOR i=1 TO 50 : CLS
210 INPUT "reference";e(i)
220 INPUT "designation";f$(i)
230 INPUT "entrez le code t.v.a ;code 1=7%,code 2=18.60%";t(i)
240 INPUT "prix unitaire brut";pu(i)
250 INPUT "entrez le pourcentage de la remise";q(i)
260 REM calcul de la remise
270 r(i)=pu(i)*q(i)/100
280 pn(i)=pu(i)-r(i)
290 INPUT "quantite";qt(i)
300 pht(i)=pn(i)*qt(i)
310 t1=0.07;t2=0.186
320 INPUT "y'a t'il d'autres marchandises pour ce client";z$
330 IF z$="o" OR z$="O" THEN 350
340 l=i;i=50
350 NEXT i
360 MODE 2
370 PRINT TAB(2);"code client ";TAB(26);"code repres.";TAB(50)"date";TAB(66);"n0
de fact."
380 PRINT
390 PRINT TAB(4);a;TAB(27);b;TAB(48);c$;TAB(68);d
400 PRINT STRING$(80,140)
410 a$="ref * designation * code * prix unit * remis * prix unit * quant * pr
ix ht *"
420 PRINT a$
420 PRINT STRING$(80,140)
440 z=0
450 x=0
460 w=0
470 y=0
490 FOR i=1 TO 1
490 PRINT e(i);TAB(9);f$(i);TAB(21);t(i);TAB(32);pu(i);TAB(42);q(i);TAB(51);pn(i)
);TAB(63);qt(i);TAB(71);pht(i)
500 IF t(i)=1 THEN tv1(i)=pht(i)*t1 : GOTO 520
510 IF t(i)=2 THEN tx2(i)=pht(i)*t2
520 w=w+pht(i)
530 z=z+tv1(i)
540 x=x+tx2(i)
550 y=w+z+x
560 NEXT i
570 PRINT STRING$(80,140)
580 PRINT TAB(40)"total ht";PRINT TAB(64) USING "#####.##";w
590 PRINT TAB(40)"tva a 7%(cde1)";TAB(64) USING "#####.##";z
600 PRINT TAB(38)"tva a 18.60%(cde2)";TAB(64) USING "#####.##";x
610 PRINT TAB(64);STRING$(8,140): PRINT
620 PRINT TAB(40)"net a payer";TAB(64) USING "#####.##";y
630 END
```


2) La comptabilité générale

La comptabilité générale offre l'éventail des écritures dont on a besoin pour réaliser sa comptabilité.

Au départ, le programme avait été présenté dans un projet de comptabilité, comme exemple de présentation sur imprimante. Il a été adapté sur AMSTRAD, et les tableaux qui jouxtent le listing ont été fait à partir d'un AMSTRAD sur une imprimante GP 500

Il permet donc d'afficher les huit tableaux que nous vous présentons, à savoir :

- Les factures fournisseurs,
- Les comptes fournisseurs individuels
- Les comptes généraux,
- Les comptes T.V.A.
- Les comptes bancaires,
- les comptes caisse,
- les comptes clients.

Pour que le programme puisse fonctionner, nous avons rangé un certain nombre de données en DATA, et avons ainsi limité le nombre d'écriture.

AMELIORATIONS :

La première des modifications à faire et évidemment de supprimer cette gestion par les DATA. Vous ne pouvez en effet passer de nombreuses écritures en tapant à chaque fois des numéros de lignes suivis de DATA, autant le faire à la main.

Il faudra donc créer un sous programme qui vous permettra de rentrer vos données à l'écran (Avec l'instruction INPUT), et comme pour le précédent programme il est également nécessaire de prévoir une sauvegarde sur K7 (ou disquette).

Il est également souhaitable de prévoir un affichage écran, identique à l'affichage imprimante, mais qui aura l'avantage d'être immédiat. Vous remarquerez donc, à l'instar du programme précédent, que nous avons choisi l'impression en 80 colonnes. Il convient donc de mettre l'affichage en mode 2, et cette fois-ci, de convertir les PRINT $\text{f}8$, en PRINT.

pourriez étendre l'usage du PRINT USING à l'ensemble de l'affichage. Nous l'avons utilisé en début de programme à titre d'exemple.

A ce propos, et pour terminer, vous remarquerez que nous utilisons qu'une seule instruction PRINT USING pour une ligne d'affichage et qu'elle définit, à elle seule, l'ensemble des paramètres d'impression.

```

10 CLS : PRINT TAB(10);"* * M E N U * *":PRINT:PRINT:PRINT
20 PRINT "1 : Afficher les fournisseurs":PRINT
30 PRINT "2 : Lecture par fournisseur":PRINT
40 PRINT "3 : Lecture par cpte H.T.":PRINT
50 PRINT "4 : Lecture par compte T.V.A.":PRINT
60 PRINT "5 : Lecture compte bancaire ":PRINT
70 PRINT "6 : Lecture compte caisse ":PRINT
80 PRINT "7 : Factures clients":PRINT
90 PRINT "8 : Clients individuels":PRINT
100 PRINT : PRINT : INPUT "tapez le numero de l'operation ";q : q=INT(q)
110 IF q>8 OR q<1 THEN 10
120 ON q GOTO 130,470,810,1080,1340,1370,130,510 : GOTO 10
130 CLS : PRINT: INPUT "Nombre d'ecriture ";m: GOSUB 310 : IF q=7 THEN 170
140 PRINT #8,TAB(11);"***** F A C T U R E S F O U R N I S S E U R S *****"
"
150 PRINT #8,STRING$(80,"*")
160 PRINT #8," ":PRINT #8," ":GOTO 210
170 PRINT #8,TAB(6);"***** F A C T U R E S C L I E N T S *****"
180 PRINT #8,STRING$(64,"*")
190 PRINT #8," ":PRINT #8," "
200 s=2 : GOTO 220
210 s=1
220 RESTORE : GOSUB 280 : FOR n=1 TO m : GOSUB 340 : IF j=8 THEN 240
230 GOSUB 320 : GOSUB 350
240 NEXT n
250 GOSUB 450 : GOSUB 390 :GOSUB 330 : GOSUB 430
260 PRINT : INPUT "tapez 1 pour le retour au menu";r: r=INT(r)
270 IF r=1 THEN 10 ELSE 160
280 PRINT #8,"num "; " date "; " code "; " libelle "; " debit "; " credit ";
290 PRINT #8," code "; " net "; " code"; " TVA "; " Echeance"
300 PRINT #8,"-----"
"-----"
"-----"
310 w=0 : x=0 : y=0 : z=0 : RETURN
320 w=w+g : x=x+h : y=y+e : z=z+f : RETURN
330 PRINT #8,x,y,z : RETURN
340 READ j,a,b#,c,a#,g,h,d,e,i,f,l : RETURN
350 PRINT #8,USING"####";a;PRINT #8,TAB(6);" ";b#;PRINT #8,TAB(11);" ";PRINT
#8,USING"####";c;
360 PRINT #8,TAB(18);a#;TAB(25)" ";
370 PRINT #8,USING"#####.## #####.## ### #####.## ### #####.## #####";g;h;d;
e;i;f;l
380 RETURN
390 REM
400 PRINT #8,"TOTAL DEBIT","TOTAL NET","TOTAL TVA"
410 PRINT #8,"-----","-----","-----"
420 RETURN
430 PRINT #8,STRING$(80,"=")
440 RETURN
450 PRINT #8,STRING$(80,"-")
460 RETURN
470 PRINT #8," "
480 PRINT #8,TAB(10);"***** F O U R N I S S E U R S I N D I V I D U E L S ****"
**"
490 PRINT #8,STRING$(80,"*")
500 PRINT #8," ":GOTO 550
510 PRINT #8,TAB(15);"***** C L I E N T S I N D I V I D U E L S *****"

```



```

1060 INPUT R : IF R<>0 THEN 1600
1070 MODE 1 : LIST
1080 PRINT #8,TAB(22);"***** C O M P T E S   T V A *****"
1090 PRINT #8,STRING$(80,"*")
1100 PRINT #8
1110 m=3
1120 CLS : INPUT "Code t.v.a. :";s
1130 PRINT : INPUT "nom ";a$
1140 PRINT #8,"nom : ";a$,"deductible a ";s: GOSUB 310
1150 GOSUB 790 : GOSUB 1190 : FOR n=1 TO m : GOSUB 340
1160 z=z+f
1170 PRINT #8,TAB(10);a;TAB(20);b$;TAB(30);c;TAB(40);a$;TAB(50);i;TAB(60);f
1180 NEXT n: GOTO 1220
1190 PRINT #8,TAB(10);" num";TAB(20);" date";TAB(30);" code";TAB(40);"libelle";
TAB(50);" code";TAB(60);"montant"
1200 PRINT #8,TAB(10);" ---";TAB(20);" ----";TAB(30);" ----";TAB(40);"-----";
TAB(50);" ----";TAB(60);"-----"
1210 RETURN
1220 GOSUB 450
1230 PRINT #8,"TOTAL TVA"
1240 PRINT #8,"-----"
1250 PRINT #8,z
1260 GOSUB 430
1270 INPUT "TAPEZ ENTER POUR RETOUR AU MENU, 1 POUR CONTINUER,2 POUR TERMINER";r
1280 r=INT(r):IF r>2 OR r<0 THEN 1270
1290 ON r GOTO 10,1120,1300
1300 CLS : PRINT : PRINT" F I N"
1310 PRINT "TAPEZ ENTER POUR LISTER"
1320 INPUT R : IF R<>0 THEN 1600
1330 MODE 1 : LIST
1340 PRINT #8,TAB(19);"***** C O M P T E   B A N C A I R E *****"
1350 PRINT #8,STRING$(80,"*")
1360 PRINT #8," ": GOTO 1400
1370 PRINT #8,TAB(21);"***** C O M P T E   C A I S S E *****"
1380 PRINT #8,STRING$(80,"*")
1390 PRINT #8," ":GOTO 1410
1400 s=560 : GOTO 1420
1410 s=565
1420 INPUT "NOMBRE D'ECRITURE : ";m : GOSUB 310
1430 GOSUB 790:GOSUB 1480:FOR n=1 TO m : GOSUB 340: IF d=s THEN 1470
1440 w=w+g : x=x+h
1450 PRINT #8,TAB(7);a;TAB(17);b$;TAB(27);c;TAB(37);a$;TAB(47);d;TAB(57);h;TAB(67);g
1460 PRINT #8
1470 NEXT n : GOTO 1530
1480 PRINT #8,TAB(10);"num";TAB(18);"date";TAB(28);"code";TAB(37);"libelle";
1490 PRINT #8,TAB(48);"code";TAB(57);"debit";TAB(67);"credit"
1500 PRINT #8,TAB(10);"---";TAB(18);"----";TAB(28);"----";TAB(37);"-----";
1510 PRINT #8,TAB(48);"----";TAB(57);"-----";TAB(67);"-----"
1520 RETURN
1530 GOSUB 450
1540 PRINT #8,"TOTAL DEBIT","TOTAL CREDIT"
1550 PRINT #8,"-----","-----"
1560 PRINT #8,X,W
1570 GOSUB 430
1580 PRINT "Tapez enter pour revenir au menu"
1590 IF r=0 THEN 10
1600 END
2000 DATA 1,3456,16no,195,gabriel,0,23456.78,610,23456.78,436,12356.78,301180
2010 DATA 1,1234,1710,195,avoirs/f23,123567.78,0,610,13456.78,436,12456.89,301180
0

```

2020 DATA2,2345,1710,901,untel,5236.45,0,710,4234.45,437,1230.45,0
2030 DATA3,0,1810,195,gabriel,802.33,0,560,0,0,0,0
2040 DATA4,0,2110,901,untel,0,5236.45,565,0,0,0,0
2050 data5,23,345,450,derni,8,3423.89,342,0,34,0,0
2060 DATA 6,1,234,873,nweod,90,1234.35,28,0,0,24,0
2070 DATA 8,2,225,342,essai,23,3456.83,90,0,0,0,45

***** FACTURES FOURNISSEURS *****

num	date	code	libelle	debit	credit	code	net	code	TVA	Echeance
3456	16no	195	gabriel	0,00	23456,78	610	23456,78	436	12356,78	301180
1234	1710	195	avoins/f23							
				123567,78	0,00	610	13456,78	436	12456,89	301180
1	234	873	nweed	90,00	1234,35	28	0,00	0	24,00	0
TOTAL DEBIT			TOTAL NET			TOTAL TVA				
24691,13			36913,56			24837,67				

***** FOURNISSEURS INDIVIDUELS *****

num de compte : 3450 nom : benoit

Num	Date	Code	Libelle	Code	Debit	Credit	Echeance
3456	16no	195	gabriel	610	0	23456,78	301180
1234	1710	195	avoins/f23	610	123567,78		
1	234	873	nweed	28	90	1234,35	0
TOTAL DEBIT					TOTAL CREDIT		
123657,78					24691,13		

***** FACTURES CLIENTS *****

num	date	code	libelle	debit	credit	code	net	code TVA	Echeance
3456	16no	195	gabriel	0.00	23456.78	610	23456.78	436 12356.78	301180
1234	1710	195	avoins/f23						
				123567.78	0.00	610	13456.78	436 12456.89	301180
1	234	873	nweed	90.00	1234.35	28	0.00	0 24.00	0
TOTAL DEBIT			TOTAL NET			TOTAL TVA			
24691.13			36913.56			24837.67			

***** CLIENTS INDIVIDUELS *****

num de compte : 2309 nom : bernie

Num	Date	Code	Libelle	Code	Debit	Credit	Echeance
3456	16no	195	gabriel	610	0	23456.78	301180
1234	1710	195	avoins/f23	610	123567.78		
						0	301180
1	234	873	nweed	28	90	1234.35	0
TOTAL DEBIT			TOTAL CREDIT				
123657.78			24691.13				

***** C O M P T E S G E N E R A U X *****

num de compte : 9032 nom : loic

num	date	code	libelle	code	montant
3456	16no	195	gabriel	610	23456.78
1234	1710	195	avoirs/f23	610	13456.78
1	234	873	nweed	28	0

TOTAL HORS TAXES

36913.56

***** C O M P T E S T V A *****

nom : Paquie deductible a 18.6

num	date	code	libelle	code	montant
3456	16no	195	gabriel	436	12356.78
1234	1710	195	avoirs/f23	436	12456.89
1	234	873	nweed	0	24

TOTAL TVA

24837.67

***** COMPTE BANCAIRE *****

<u>num</u>	<u>date</u>	<u>code</u>	<u>libelle</u>	<u>code</u>	<u>debit</u>	<u>credit</u>
3456	16no	195	gabriel	610	23456.78	0
1234	1710	195	avoins/f23	610	0	123567.78
1	234	873	nweed	28	1234.35	90
TOTAL DEBIT		TOTAL CREDIT				
24691.13		123657.78				

***** COMPTE CAISSE *****

<u>num</u>	<u>date</u>	<u>code</u>	<u>libelle</u>	<u>code</u>	<u>debit</u>	<u>credit</u>
3456	16no	195	gabriel	610	23456.78	0
1234	1710	195	avoins/f23	610	0	123567.78
1	234	873	nweed	28	1234.35	90
TOTAL DEBIT		TOTAL CREDIT				
24691.13		123657.78				

3) La fiche de paie

Encore un programme à adapter à vos besoins :
l'édition d'un bulletin de paie.

Ce dernier inspire les mêmes commentaires que les deux listings précédents. Il fait une présentation du bulletin à l'écran, et il est sans doute inutile de vous dire que ce serait tellement mieux sur un imprimante.

Pourtant une édition de bulletin de paie présente un problème un peu particulier. En effet sur ce genre de document, certaines données sont constamment modifiées, d'autres le sont plus rarement et certaines jamais ou presque.

Par exemple les noms, prénoms et qualifications sont modifiés à chaque fois que vous établissez un nouveau bulletin de salaire (surtout si vous avez beaucoup de salariés). Mais pour d'autres au contraire, employant le même personnel depuis fort longtemps, ces références sont pratiquement immuables.

Il convient donc de déterminer, selon vos besoins, quelles sont les données stables, ou celles qui ne le sont pas.

Si votre programme est bien fait, les données qui sont constamment renouvelées doivent être très facilement accessibles à l'utilisateur et le programme doit être conversationnel pour ce genre d'éléments. Par exemple on verra s'afficher à l'écran la phrase :

Nombre d'heures faites au cours de ce mois ? :

Et l'utilisateur n'aura qu'à répondre à la question posée.

Pour des données pas trop souvent modifiées, on les stockera par exemple sur K7 dans un fichier, ce qui permettra de les charger en début de programme, et évitera surtout de les retaper à l'écran à chaque bulletin de paie, ce qui est fastidieux et finit par devenir plus long qu'à la main.

Enfin pour des données fixes, on prendra soin de les déterminer dans le corps du programme (ou listing), ce qui présente l'avantage de la rapidité d'accès et de la sécuri-

té. Par contre la modification est plus complexe pour l'utilisateur qui doit, pour ce faire, rentrer dans le corps du programme.

En conclusion on peut dire que plus les données sont sujettes à des modifications plus elles doivent être faciles d'accès.

En contre partie des données invariables doivent être inaccessibles.

L'art du programmeur consiste à ne pas gêner l'utilisateur par des demandes de données inopportunes et de ne pas compliquer l'accès à des informations trop souvent modifiées.

C'est entre ces deux écueils que vous devez essayer de naviguer.

BULLETIN DE PAYE

```
5 REM -----
7 REM BULLETIN DE PAYE
9 REM -----
10 REM Edition du fichier individuel
15 REM -----
20 BORDER 0 : INK 0,0 : CLS
30 INPUT "nom " ; n$
40 INPUT "prenom " ; p$
50 INPUT "qualification " ; q$
60 INPUT "numero de securite sociale " ; ns$
70 INPUT "code " ; cd
80 INPUT "periode de paie " ; pr
90 INPUT "date de la paie " ; dp$
100 INPUT "prime " ; p
105 REM -----
110 REM Edition du bulletin de paie
115 REM -----
120 CLS : MODE 2
130 PRINT "                               BULLETIN DE PAIE"
140 PRINT STRING$(70,154)
150 PRINT TAB(5);"nom et prenom";TAB(25);"qualification";TAB(50);"numero de sec.
soc."
155 PRINT
160 PRINT TAB(5);n$;TAB(25);q$;TAB(52);ns$
170 PRINT TAB(5);p$
190 PRINT
200 PRINT TAB(0);"code";TAB(25);"periode de paie";TAB(60);"date de la paie"
210 PRINT
220 PRINT TAB(0);cd;TAB(28);pr;TAB(65);dp$
230 PRINT
240 INPUT "APPUYEZ SUR ENTER POUR VISUALISER LA SUITE " ; Z
250 r=0
260 PRINT TAB(8);STRING$(60,154)
270 PRINT TAB(0);"code";TAB(7);"elements de paie";TAB(30);"base";TAB(40);"taux";
275 PRINT TAB(50);"bases";TAB(60);"gains";TAB(68);"retenues"
280 PRINT TAB(8);STRING$(60,154)
290 c1=10 : c2=20 : c3=30 : c4=40 : c5=50
300 sb=0
310 b=186.6
320 t=25 : g=b*t
330 PRINT TAB(0);c1;TAB(7);"salaire de base";TAB(30);b;TAB(40);t;TAB(50);b$;TAB(
60);g
340 PRINT TAB(0);c2;TAB(7);"prime";TAB(60);p
350 sb=sb+g+p
360 PRINT TAB(0);c3;TAB(7);"salaire brut";TAB(60);sb
370 sss=3.5 : r1=sss+sb/100
380 PRINT TAB(0);c4;TAB(7);"s.s sur brut";TAB(40);sss;TAB(50);sb;TAB(69);r1
390 tsp = 1.5 : r2=tsp*sb/100
400 PRINT TAB(0);c5;TAB(7);"s.s sur plafond";TAB(33);tsp;TAB(50);sb;TAB(68);r1
410 c6=60 : c7=70 : c8=80 : c9=90 : c10=100
420 tvp=4.7 : r3=tv*sb/100
430 PRINT TAB(0);c6;TAB(7);"s.s.vieill./sal.plaf";TAB(40);tvp;TAB(50);sb;TAB(68);
r3
440 tc=0.84 : r4=tc*sb/100
450 PRINT TAB(0);c7;TAB(7);"chomage";TAB(40);tc;TAB(50);sb;TAB(68);r4
460 rcp=6.18 : r5=rcp*sb/100
```

```

470 PRINT TAB(0);c8;TAB(7);"ret.compl./plaf";TAB(40);rcp;TAB(50);sb;TAB(68);r5
480 rcd=1.76 : r6=rcd*sb/100
490 PRINT TAB(0);c10;TAB(7);"ret.compl des cadres";TAB(40);rcd;TAB(50);sb;TAB(68
);r6
500 ttc=r1+r2+r3+r4+r5+r6
510 PRINT TAB(0);c10;TAB(7);"total des cotisations";TAB(68);ttc
520 a1=110 : a2=120 : a3=130
530 ni=sb-ttc
540 PRINT TAB(0);a1;TAB(7);"net imposable";TAB(60);ni
550 tp=23
560 PRINT TAB(0);a2;TAB(7);"prime de transpt";TAB(60);tp
570 np=ni+tp
580 PRINT TAB(0);a3;TAB(7);"NET A PAYER";TAB(60);np
590 PRINT
600 INPUT "TAPEZ SUR ENTER POUR CONTINUER"; Z
610 CLS
620 PRINT TAB(8);STRING$(60,154)
630 cb=sb : cni=ni : br=sb : rt=ttc
640 PRINT TAB(0);"cum/bas";TAB(13);"cumbrut";TAB(28);"cumnetimp";TAB(38);"brut";
TAB(45);"retenues";TAB(55);"net.imp";TAB(65);"net a payer"
650 PRINT TAB(8);STRING$(60,154)
660 PRINT TAB(0);cbs;TAB(13);cb;TAB(28);cni;TAB(38);br;TAB(45);rt;TAB(55);ni;TAB
(65);np
670 PRINT TAB(8);STRING$(60,154)

```

Imprimerie SANDRINE 77
 77, rue Philippe-de-Girard
 75018 PARIS
 Tél. 42.08.27.96

imprimé en France

Les Editions VISMO
 84, boulevard Beaumarchais
 PARIS 75011
 N° 905718



Document **numérisé**
avec amour par :

AMSTRAD

CPC 

MÉMOIRE ÉCRITE



<https://acpc.me/>