

# Le guide du MO5

André Deledicq

Photo de couverture : Pascal SEVRAN  
Maquette et illustrations : Alain DUNOIR



CEDIC  
NATHAN

Ce volume porte la référence  
ISBN 2-7124-0522-6

*Toute reproduction, même partielle, de cet ouvrage est interdite. Une copie ou reproduction par quelque procédé que ce soit, photographie, photocopie, microfilm, bande magnétique, disque ou autre, constitue une contrefaçon passible des peines prévues par la loi du 11 mars 1957 sur la protection des droits d'auteur.*

© CEDIC 1984

CEDIC, 32, boulevard Saint-Germain, 75005 - PARIS

## Sommaire

Avant-propos .....	7
<b>Mise en route</b> .....	9
Votre ordinateur en quelques lignes .....	12

## LEÇONS

1. Le clavier et l'écran .....	15
2. Commandes d'affichage .....	21
3. Programmer .....	27
4. Faire et refaire .....	35
<b>Résumé et conseils aux apprentis-programmeurs</b> .....	41
5. Graphiques .....	45
6. Répétitions et aiguillages .....	51
7. Nombres et opérations .....	59
8. Données et résultats .....	67
9. Lettres et messages .....	73
10. Musique .....	81
11. Indices et tableaux .....	87
12. Enregistrer et charger un programme .....	95
13. Le crayon optique .....	101
14. Les fichiers .....	107
<b>Conseils de programmation</b> .....	113
<b>Correction des exercices</b> .....	123

## FICHES DE RÉFÉRENCE

### Affichage

* Le clavier .....	133
* L'écran .....	134
* COLOR, SCREEN .....	135
* ATTRB .....	138
* Ligne d'écriture (EFF, INS, CNT-X, CNT-W) .....	139
** Minuscules et accents (ACC) .....	140
** CLS, CONSOLE .....	142

** LOCATE, CSRLIN, POS, SCREEN.....	145
** Ligne de programme.....	146
*** CNT.....	148
*** Le code ASCII.....	150
*** Initialisation.....	152

#### Commande, contrôle,

##### enregistrer, exécuter, lister...

* RUN, STOP, END, CNT-C, CONT, GOTO.....	153
* LIST, DELETE, NEW.....	155
** TRON, TROFF.....	156
** ON ERROR GOTO, ERR, ERL, RESUME.....	157
*** EXEC, VARPTR.....	159

##### ...structurer, répéter, aiguiller

* FOR ... NEXT.....	161
** GOSUB ... RETURN.....	164
** ON ... GOTO..., ON ... GOSUB.....	165
** IF ... THEN ... ELSE.....	166
*** Vrai, faux, AND, OR, NOT, XOR, IMP, EQV.....	169

#### Mémoires , fonctions

##### Opérations et fonctions numériques :

* +, -, *, /, @, MOD, ^.....	172
* SIN, COS, TAN, EXP, LOG, SQR, FIX, INT, CINT, SGN, ABS.....	174
* RND.....	176

##### Opérations et fonctions alphabétiques :

* "...", +.....	177
* MID\$, LEFT\$, RIGHT\$.....	178
* LEN, INSTR.....	179
** VAL, STR\$.....	180
*** ASC, CHR\$.....	181

##### Variables et mémoires

** Types de variables.....	182
** ...\$, ...%, DEFINT, DEFSTR, DEFSNG.....	184

*** Occupation des mémoires.....	186
*** DIM, CLEAR, FRE.....	189
*** PEEK, POKE.....	191

#### Entrées-sorties/clavier-écran

##### Résultats

* PRINT, SPC, TAB.....	192
** PRINTUSING.....	194

##### Données

* INPUT, LINE INPUT, INPUT\$.....	196
** INKEY\$.....	198
** DATA, READ, RESTORE.....	199

##### Graphiques

* PSET, LINE, BOX, BOXF.....	201
** Fond et écriture, POINT.....	203
** DEFGR\$, GR\$.....	205

#### Périphériques

##### Sons

* PLAY.....	207
-------------	-----

##### Cassette

* SAVE, LOAD, RUN, MERGE, MOTORON, MOTOROFF, SKIPF.....	210
*** SAVEM, LOADM.....	213

##### Crayon optique

** INPUT PEN, INPEN, TUNE, PTRIG.....	215
---------------------------------------	-----

##### Imprimante

** "LPRT :", PRINT #, SCREENPRINT.....	217
--	-----

##### Fichiers sur cassette

*** OPEN, CLOSE, EOF, SKIPF, PRINT #, INPUT #, LINE INPUT #.....	219
---	-----

#### ANNEXES

1. Renseignements techniques MO5.....	222
2. Grilles graphiques.....	224
3. La syntaxe du Basic MO5.....	226
4. Les mots-clefs du Basic MO5.....	230

5. Conversions utiles.....	231
6. Codes des couleurs.....	232
7. Codes ASCII, caractères et touches.....	233
8. Codes des erreurs.....	235
 INDEX .....	 237

## Avant-Propos

Le MO5 peut être mis entre toutes les mains.

Ce livre aussi.

Que ce soient des mains de débutants, d'amateurs un peu débrouillés ou de programmeurs avancés.

Ce livre propose à chacun d'entre eux des entrées différentes, leur permettant d'utiliser le MO5 à leur niveau et de progresser rapidement dans la connaissance de ses merveilleux rouages.

Cependant, avant toute autre chose, n'oubliez pas qu'un bon lecteur doit consulter attentivement le sommaire et se promener un peu dans le livre...

- Les *leçons* vous guident pas à pas vers la maîtrise de votre MO5.
- Les *fiches de référence* vous renseignent, sur tous les détails relatifs à un mot ou un thème particulier (une, deux ou trois étoiles indiquent le niveau supposé du lecteur : débutant, débrouillé ou averti). Il y a 20 fiches une étoile, 20 fiches deux étoiles, 10 fiches trois étoiles.

■ **Pour les débutants**, c'est-à-dire ceux qui vont apprendre à programmer avec le MO5, la méthode est simple : laissez vous guider dans l'ordre des pages !

Étudiez les leçons l'une après l'autre, frappez les programmes proposés, faites les exercices...

Les 4 premières leçons vous habituent au maniement du clavier, à ce qui peut se passer sur l'écran, et aux premières idées sur la programmation.

Les 10 leçons suivantes vous permettent de parcourir l'éventail des possibilités de votre MO5.

Petit à petit, vous pourrez vous risquer à consulter les fiches de référence ; vous y trouverez les compléments et détails relatifs aux points du langage BASIC que vous souhaitez approfondir.

■ **Pour les amateurs un peu éclairés**, c'est-à-dire ceux qui ont déjà commencé à programmer sur une autre machine, la méthode de lecture est plus souple.

Nous vous conseillons de feuilleter d'abord les 4 premières leçons et leur résumé, ou bien, selon votre humeur, de parcourir les fiches de référence "clavier, écran, curseur" et "enregistrer, exécuter, lister". Vous y prendrez contact avec les particularités de votre MO5. Vous pourrez ensuite suivre les leçons 5 à 14, qui vous permettront d'acquérir une bonne maîtrise de la programmation de votre MO5. Et, surtout, lisez les **conseils de programmation** à la fin des leçons.

Consultez le plus souvent possible les fiches de référence ; elles vous donneront toutes précisions sur les mots et les finesses du BASIC que vous devriez maintenant connaître.

■ **Pour les programmeurs déjà avertis**, ce livre doit être conçu plutôt comme un dictionnaire ou une encyclopédie.

Vous pouvez évidemment survoler les leçons : en particulier jetez un petit coup d'œil sur les leçons 1,2 (le clavier, l'écran et le curseur), 5 (graphiques), 10 (musique) et les deux "résumés et conseils de programmation".

Votre outil principal de travail est évidemment l'ensemble des fiches de référence. Elles sont classées par thèmes ; vous pourrez ainsi très vite vous faire une idée des possibilités offertes, par exemple par les instructions graphiques ou bien le traitement alphabétique.

Vous lirez avec profit les fiches "clavier, écran, curseur", et (si vous êtes super-initiés) les fiches "types et noms de variables", "occupation des mémoires" et les annexes 3 et 7.

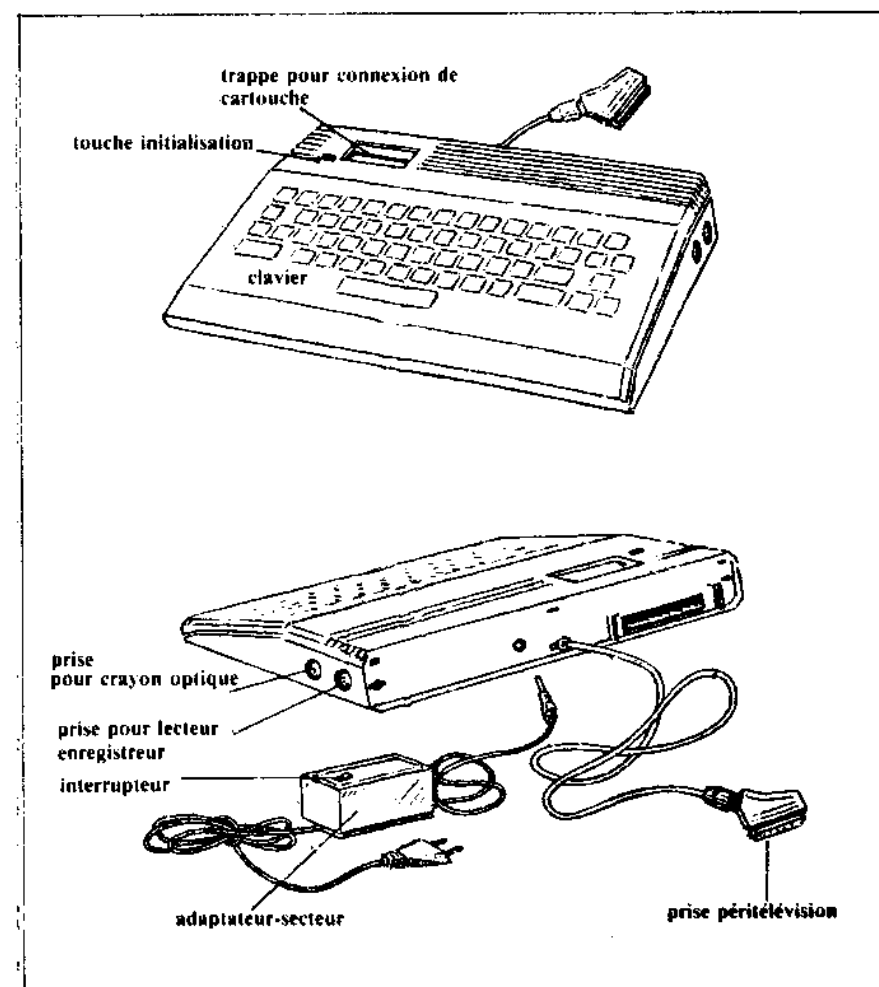
■ Cependant, pour les intimes de ce livre et du MO5, l'entrée principale sera rapidement la dernière page : elle montre tous les chemins : c'est l'INDEX.

## Mise en route

Votre système informatique de base se compose d'un micro-ordinateur MO5 et d'un récepteur de télévision.

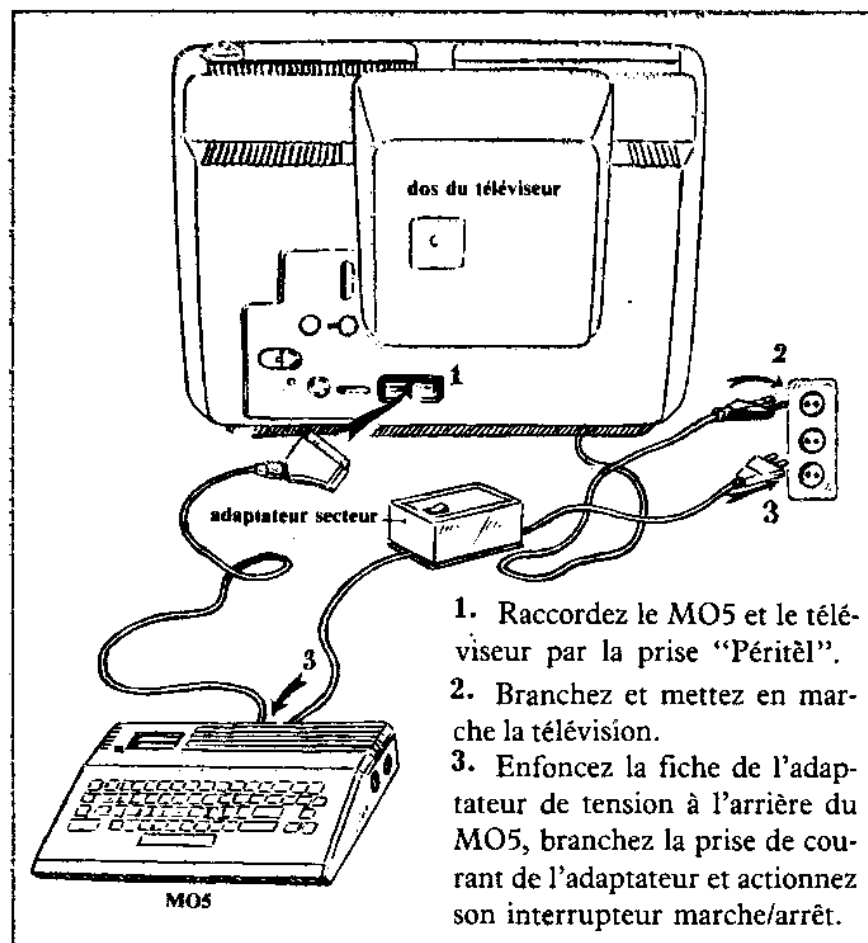
Placez le MO5 sur une surface bien dégagée (afin d'assurer un bon refroidissement), devant votre téléviseur, et à proximité d'une alimentation électrique standard 220 V - 50 Hz.

## Description extérieure



Pour plus de précisions consultez les renseignements techniques en Annexe 1.

## Pour programmer, calculer, apprendre le BASIC ...



1. Raccordez le MO5 et le téléviseur par la prise "Péritel".
2. Branchez et mettez en marche la télévision.
3. Enfoncez la fiche de l'adaptateur de tension à l'arrière du MO5, branchez la prise de courant de l'adaptateur et actionnez son interrupteur marche/arrêt.

Votre MO5 est en état de marche ; le message suivant est apparu sur l'écran :

MO5 BASIC 1.0  
(c) Microsoft 1984

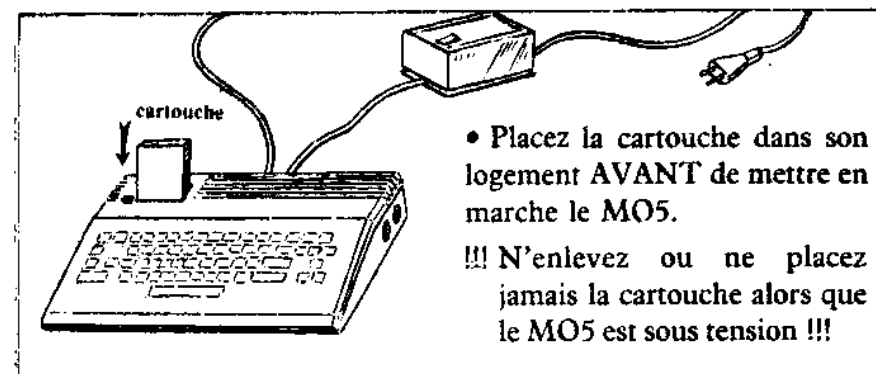
Vous pouvez maintenant :

- Régler la luminosité de votre téléviseur.
- Frapper sur les touches du clavier du MO5.

Chaque fois qu'une touche est frappée un "bip" est transmis à la voie "son" du téléviseur. Vous pouvez régler le volume de ce son en agissant sur la commande de votre téléviseur.

Le MO5 est prêt à recevoir vos ordres énoncés en BASIC !

## Pour utiliser une cartouche ...

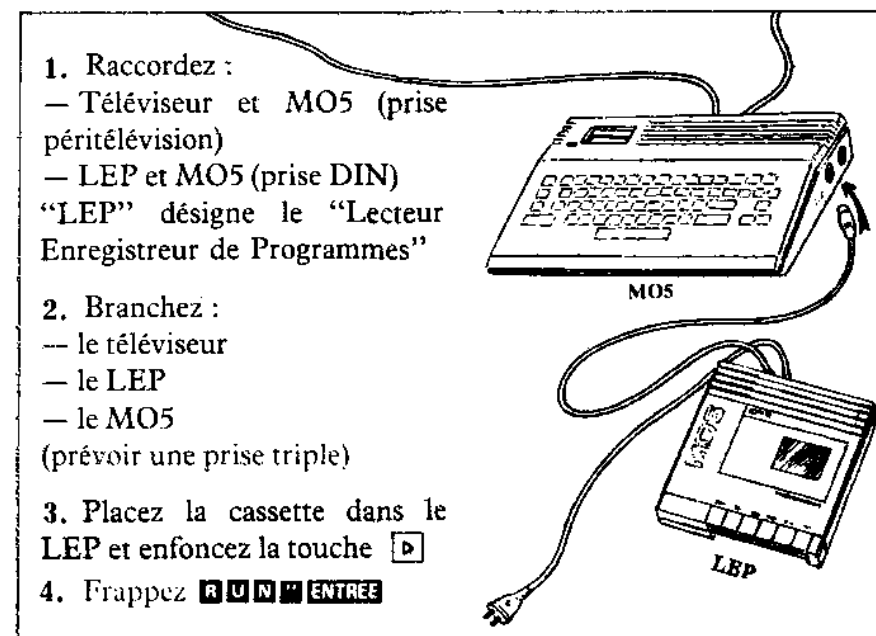


- Placez la cartouche dans son logement AVANT de mettre en marche le MO5.

!!! N'enlevez ou ne placez jamais la cartouche alors que le MO5 est sous tension !!!

Attention, cette cartouche va remplacer le BASIC résidant dans votre MO5. Il vous faudra donc suivre les indications qui vous sont données avec cette cartouche.

## Pour utiliser une cassette...

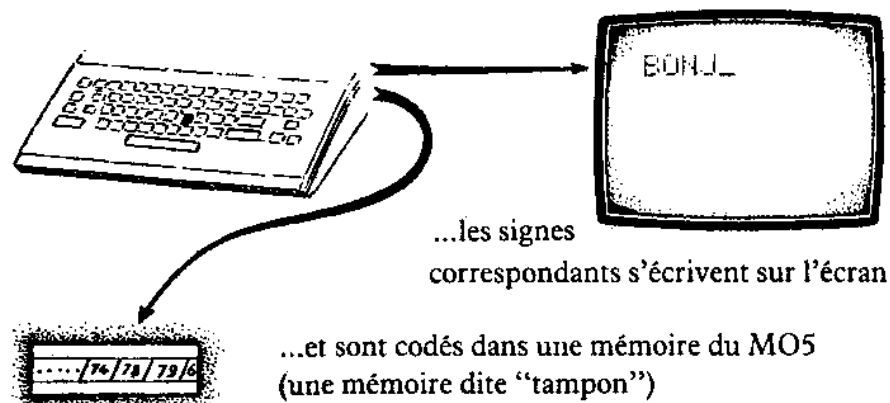


1. Raccordez :
  - Téléviseur et MO5 (prise pritélévision)
  - LEP et MO5 (prise DIN)
 "LEP" désigne le "Lecteur Enregistreur de Programmes"
2. Branchez :
  - le téléviseur
  - le LEP
  - le MO5
 (prévoir une prise triple)
3. Placez la cassette dans le LEP et enfoncez la touche **D**
4. Frappez **R U N** **ENTREE**

Attendez que le programme sur cassette entre dans la mémoire du MO5 et suivez les indications qui vous sont données avec la cassette. (Nous vous conseillons de lire la leçon 12)

## Votre ordinateur en quelques lignes

■ Vous frappez sur les touches du clavier...

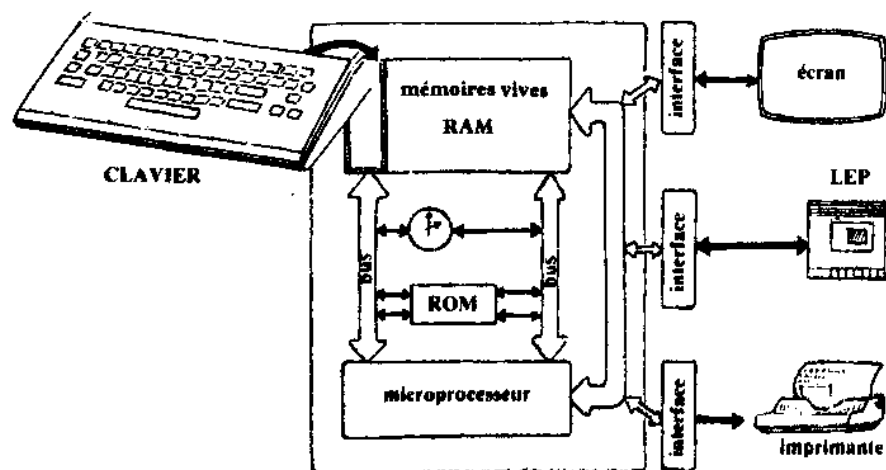


■ Lorsque vous frappez sur **ENTREE**

...Le contenu de la mémoire-tampon est "interprété"

...et le MO5 déclenche une série d'actions correspondant à ce que vous venez d'ordonner ; cela peut être, entre autres :

- un rangement ou un transfert en mémoire,
- un "traitement" alphabétique, arithmétique ou logique,
- une action d'écriture ou de dessin sur l'écran,
- une commande à un périphérique,
- un ordre d'exécution d'un programme en mémoire.



■ En fait, dès que vous frappez sur le clavier, les touches sont codées par des codes numériques sous forme binaire.

Par exemple **Q** est codée "66", c'est à dire 0 1 0 0 0 1 0.

Ces "octets" de huit "bits" circulent dans les "parties cachées" du MO5 par des circuits appelés "Bus" ; ils sont traités par le "microprocesseur" qui est en communication continue avec les autres parties du micro-ordinateur et, en particulier, avec deux sortes de mémoires :

— les mémoires *mortes* (en anglais : ROM, c'est-à-dire "Read Only Memory") qui contiennent tous les "trucs" nécessaires pour gérer l'ensemble du système et interpréter vos ordres.

— les mémoires *vives* (en anglais : RAM, c'est-à-dire "Random Access Memory") dans lesquelles vous pouvez introduire vos ordres ou vos données.

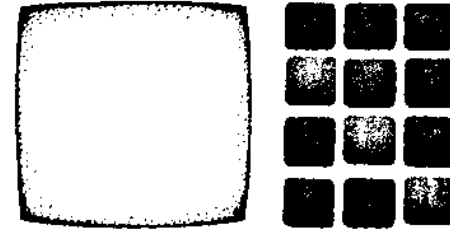
Ces bits transitent donc à grande vitesse (l'horloge interne envoie un million de tops par seconde pour régler le ballet interne du micro-ordinateur) à travers les bus, mémoires et pièces diverses. Tout cela sort, de temps en temps par une interface commandant un appareil "périphérique" montrant ces informations sous une forme humainement exploitable.

Nous n'insisterons pas sur cette fantastique cuisine électronique car vous n'avez pas besoin de savoir ce qui se passe "en réalité".

L'intérêt du micro-ordinateur est, en effet, que sa complexité vous reste cachée : nous supposons que votre préoccupation (au moins au début) est plutôt du type suivant :

Que frapper sur mon clavier pour que ce qui passe sur l'écran réponde à mes désirs ?

C'est ce que les leçons de ce livre vont tenter de vous apprendre.



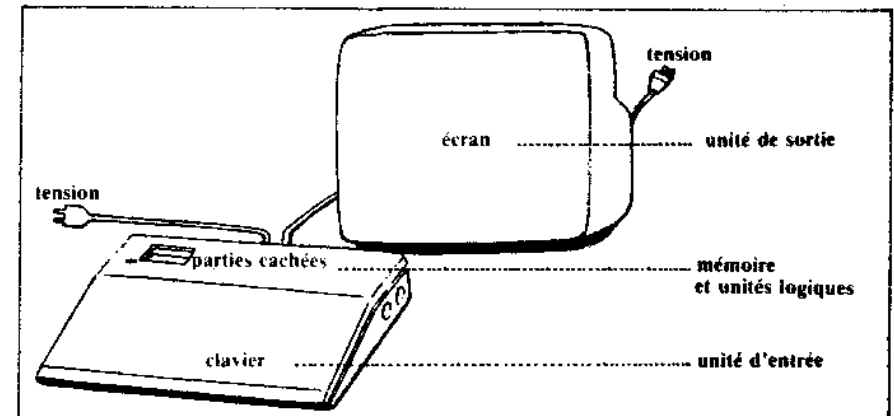
1

## Le clavier et l'écran

### ■ Les trois parties

Votre ordinateur se compose de 3 éléments : le clavier, l'écran et toutes les parties qui vous sont cachées. Ce sont évidemment ces parties cachées qui font tout le travail important !

Le clavier et l'écran vous permettent simplement de communiquer avec les parties cachées : vous "entrez" ce que vous voulez (et pouvez) par les touches du clavier... et vous observez ce qui "sort" sur l'écran (lettres, chiffres, dessins...).





## ■ Vous "commandez" votre ordinateur en frappant sur le clavier

En fait, ce que vous lui commandez, c'est de modifier telle ou telle de ses parties ; ou bien vous modifiez l'écran (et vous voyez immédiatement le résultat de ce changement), ou bien vous changez l'état des parties cachées (et il vaut mieux alors savoir ce que vous faites).

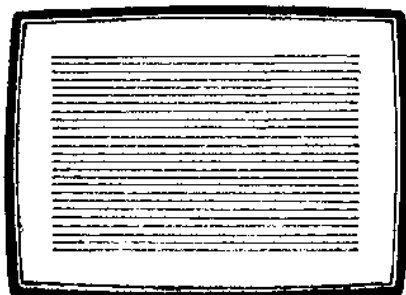
Dans tous les cas vous ne risquez pas d'abîmer votre ordinateur en frappant sur n'importe quoi ; et si d'aventure, comme le singe dactylographe réécrivant la bible, vous frappez une phrase ayant un sens pour la machine et rendant ses parties cachées inutilisables, alors vous pourriez toujours couper le courant puis le rebrancher.

## ■ Le curseur d'écriture

Ainsi s'appelle le petit trait que vous voyez sur votre écran ; il indique l'endroit où va s'écrire le prochain "caractère" frappé au clavier (un "caractère" c'est un chiffre, une lettre ou tout autre signe susceptible d'apparaître au dessus de ce petit trait, comme par exemple #, + ou ^).

Frappes sur le clavier : vous verrez alors les caractères frappés s'écrire et le curseur se déplacer.

Remarquez que vous ne pouvez pas écrire dans les coins et sur le tour de l'écran : la partie utile de l'écran est limitée à un rectangle de 40 caractères de longueur sur 25 lignes de hauteur.



## ■ Le clavier

Apprenez la position sur le clavier des différentes touches :

- Les *touches normales d'écriture* de caractères c'est-à-dire : les dix chiffres

les vingt-six lettres

les signes **~** **^** **~** **~** **~** **@**

la barre d'espacement, qui fait écrire un caractère "blanc".

Notez que : en laissant votre doigt appuyé sur une touche, vous répétez l'écriture du caractère correspondant.

- Les touches qui font *déplacer le curseur* : d'une position vers le haut



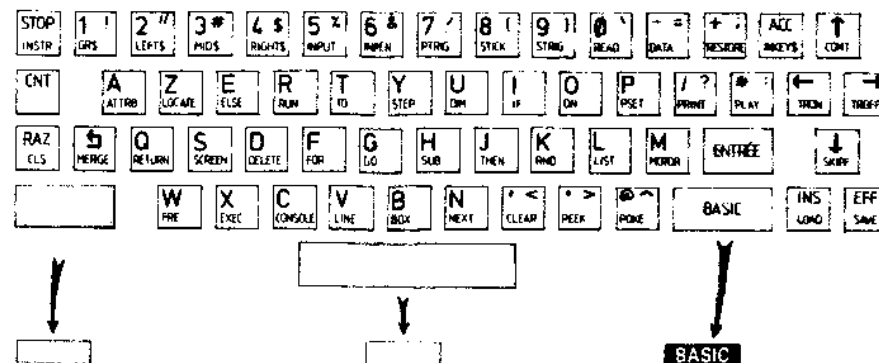
vers la gauche **←** **→** vers la droite



vers le bas

**↖** tout en haut à gauche.

- La touche **RAZ** qui efface l'écran complètement (remise à zéro).



Dans la suite du livre, ces trois touches seront ainsi notées.

- Sachez aussi que :

— les touches **INS** et **EFF** servent à corriger ce qui est déjà écrit sur l'écran (voir les fiches de référence ou essayez vous-même),  
— la touche **ACC** sert à écrire des voyelles accentuées (voir la fiche "minuscules et accents"),

- les touches **STOP** et **ENT** vous seront expliquées à la leçon 4,
- la touche **ENTREE** vous sera expliquée à la leçon 2.

• Enfin :

**ENT** En appuyant sur la touche **ENT**, située en bas à gauche, puis (en gardant cette touche enfoncée) en appuyant sur une autre touche, on commande l'écriture du signe dessiné en haut à droite de cette touche.

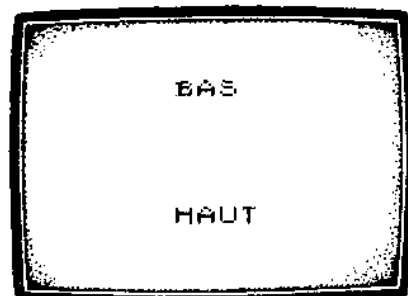
**BASIC** En appuyant sur la touche **BASIC**, située en bas à droite, puis (en gardant cette touche enfoncée) en appuyant sur une autre touche, on commande l'écriture du mot écrit dans la partie basse de cette touche.

appuyer sur	produit l'écriture de
" <b>ENT</b> "	" <b>ENT</b> "
" <b>BASIC</b> "	" <b>BASIC</b> "

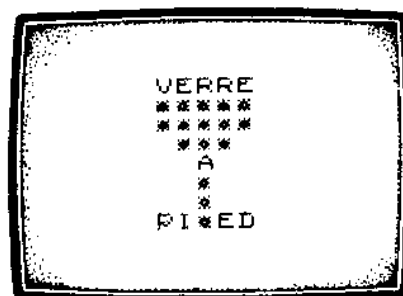
## Exercices



1. Écrire sur votre écran comme sur celui-ci :



2. Décorer votre écran avec un "dessin alphabétique" analogue à celui-ci :



Dessinez ainsi, une maison, un homme, un arbre, une carte de la France.

3. Recopiez ainsi, sur votre écran le dessin suivant ou quelque autre de votre invention qui utilise des caractères plus ou moins curieux. (Vous ne devez pas utiliser la touche **ENTREE**).

```

      ' ' + ' ' ?
    % % % % % % % ?
  / / / / / / / / ?
! ! ! ! ! ! ! ! ! !
( - * - * - ) $ $
(      &      ) $ $
(      & &      ) $
(      ~ ~ ~ ~ ~ )
(      , ;      )
(      ( ( (      )
# # # # #
: < = > :
, , , , ,

```

## Attention

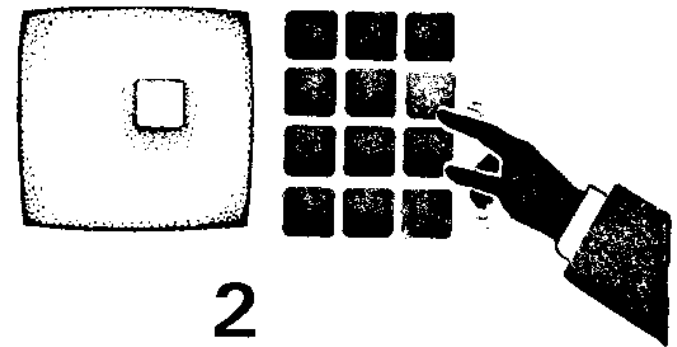
■ Ne confondez pas les touches :

**0** (zéro)

et **O** (la lettre "o")

Fiches de référence pouvant être consultées :

- \* Clavier, page 133
- \* Ligne d'écriture, page 139
- \* Minuscules et accents, page 140



## 2 Commandes d'affichages

La précédente leçon vous a peut-être laissé sur un sentiment d'insatisfaction. Car enfin vous ne disposez pas simplement d'une machine à écrire, même s'il s'agit d'écrire sur votre écran de télévision ! Soyez rassuré, les choses sérieuses vont maintenant commencer :

### ■ La touche **ENTREE**

Avec la touche marquée **ENTREE**, vous allez pouvoir donner des ordres. En effet :

Lorsque vous frappez sur cette touche, *la machine interprète tout ce que vous avez écrit précédemment sur la ligne et exécute l'ordre correspondant* (puis le curseur passe au début de la ligne suivante après avoir écrit "OK").

➤ Par exemple si vous frappez successivement :

**7 1 7 \* 3 ENTREE**

la machine interprète qu'il faut écrire le résultat de la multiplication de 17 par 3. Attention pour frapper le **\***, il faut aussi frapper sur **ENTREE** en même temps. Elle écrit alors :

51  
OK

## ■ Passez (bien) votre commande

Cependant vous ne pouvez pas écrire n'importe quoi avant de frapper la touche **ENTREE**. Vous ne pouvez écrire que des "commandes" susceptibles d'être bien interprétées, bien comprises par la machine. Pour l'essentiel, ces commandes appartiennent à une famille de langages connus sous le nom de BASIC.

Il vous faut donc apprendre comment s'expriment ces commandes et en quoi consiste leur exécution par la machine. Faute de quoi vous êtes bon pour le dialogue de sourds.

Par exemple si vous frappez :

**1 2 3 ENTREE**

la machine comprend quelque chose que vous n'apprendrez qu'à la leçon suivante et vous répond simplement par un laconique placement du curseur au début de la ligne suivante.

La situation est tout à fait semblable à celle qui consiste à commander un "bortsch" dans un restaurant vietnamien; vous vous exposez au mieux à un silence poli et, au pire, à une injure prononcée dans la langue maternelle du serveur.

Essayez :

**B O R T S C H ENTREE**

La machine répond :

Error 2

OK

(Voir les codes d'erreurs en annexe 8)

## ■ Vos premiers mots en BASIC - MO5

La suite de ces leçons va vous apprendre la plupart des mots et des phrases BASIC vous permettant d'écrire vos programmes. Vous pourrez aussi consulter très vite les fiches de référence d'abord réservées au MO-iste confirmé que vous n'allez pas tarder à devenir. (Ces fiches donnent tous les détails sur chaque commande).

Voici, pour commencer, quelques *commandes d'affichage* permettant de jouer avec les couleurs :

**C L S ENTREE**

... et l'écran s'efface complètement (tout comme si vous aviez appuyé sur la touche **RAZ**).

CLS est l'abréviation de "clear screen" ("effacer l'écran").

**C O L O R 1 ENTREE**

... et vous avez l'impression que rien ne s'est passé ! MAIS, à partir de maintenant la machine écrit en rouge.

**C O L O R 1 2 ENTREE**

... et à partir de maintenant, chaque caractère écrit le sera en rouge sur fond vert.

Vous disposez ainsi de 16 couleurs, tant pour l'"encre" d'écriture (la "forme") que pour le "papier" sur lequel vous écrivez (le "fond"). Chaque couleur correspond à un nombre de 0 à 15. Consultez la fiche "COLOR" pour connaître cette correspondance.

**S C R E E N 1 2 3 ENTREE**

... et les caractères sont alors écrits en rouge (couleur 1) sur un fond vert (couleur 2), le pourtour de votre page d'écriture étant alors colorié en jaune (couleur 3).

**Attention :** SCREEN A, B, C modifie les couleurs de l'écran au moment de son exécution et A devient la couleur d'écriture, B la couleur de tout le fond et C la couleur du pourtour.

COLOR A, B ne modifie aucune couleur déjà affichée, chaque caractère écrit *ensuite* le sera en couleur A sur fond de couleur B.

**Remarque :** A la mise sous tension de la machine et du téléviseur, les couleurs normalement choisies sont : bleu (4) pour l'écriture, bleu ciel (6) pour le fond et le pourtour (le nom "scientifique" du bleu ciel est *cyan*).

## ■ PRINT

Le mot PRINT suivi d'un nombre ou d'un calcul entraîne l'écriture de ce nombre ou du nombre résultat de ce calcul. Plusieurs nombres ou calculs peuvent être demandés à la fois, en les séparant par des virgules.

**P R I N T 8 \* 7 , 8 + 7 , 8 - 7**

... et la machine répond par écrit :

56                      15                      1  
OK

Pour des raisons de rapidité d'écriture, la suite de cinq lettres :

**P R I N T** peut être remplacée par un simple point d'interrogation ?

## Exercices



1. Expérimentez l'effet des lignes de commande suivantes :

CLS **ENTREE**  
COLOR 1 **ENTREE**  
'ROUGE **ENTREE**  
COLOR 2 **ENTREE**  
'VERT **ENTREE**  
COLOR 1+2 **ENTREE**  
'JAUNE **ENTREE**  
COLOR 4 **ENTREE**  
'BLEU **ENTREE**  
COLOR 4+1 **ENTREE**  
'VIOLET **ENTREE**

(à partir de maintenant nous écrivons les lignes de commande sans particulariser chaque caractère par un cadre ; c'est tout de même plus lisible !).



**Remarque :** Toute ligne commençant par une apostrophe ■ n'est pas "interprétée" par la machine.

2. Pour en voir de toutes les couleurs...

...essayez, selon votre humeur, des SCREEN X, Y, Z de toutes sortes.

Par exemple :

Après SCREEN 0,7,0 **ENTREE**  
écrivez 'CONDOLEANCES ! **ENTREE**  
Après SCREEN 4,6,6 **ENTREE**  
écrivez 'LE POURTOUR NE SE VOIT PAS ! **ENTREE**  
Après SCREEN 4,1,2 **ENTREE**  
écrivez 'A VOS LUNETTES ! **ENTREE**  
Après SCREEN 4,4,6 **ENTREE**  
écrivez 'C'EST SYMPATHIQUE ! **ENTREE**

Mais après être revenu de votre surprise, dépêchez-vous d'y voir plus clair en frappant successivement sur :

**ENTREE** COLOR 4,6 **ENTREE**

(Réfléchissez vous allez comprendre)

3. Sachant que vous disposez de 16 possibilités pour l'écriture, de 16 possibilités pour le fond et de 16 possibilités pour le pourtour, de combien de possibilités différentes disposez-vous au total ?

Pour avoir la réponse frappez :

? 16 \* 16 \* 16 **ENTREE**

Vous voyez que vos dons artistiques vont avoir de quoi s'exprimer !

4. Effectuez les opérations commandées ci-dessous :

? 7\*13\*11  
? 142857\*2, 142857\*3, 142857\*4  
? 142857\*5, 142857\*6, 142857\*7  
? 12\*9+3, 123\*9+4, 1234\*9+5, 12345\*9+6

## Attention

■ Dans l'écriture de "nombres-à-virgule", la partie entière et la partie décimale doivent être séparées par un *point* (et non par une virgule) !

Ainsi : 3.14 est le nombre 314/100.

3,14 est un couple de nombres (le nombre "trois" et le nombre "quatorze").

Comparer l'effet des deux instructions ? 3.14 et ? 3,14.

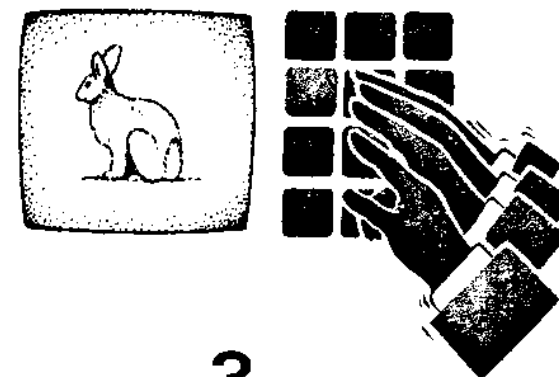
■ Le signe de division est le signe  $\div$

• Le signe  $\div$  n'est pas un signe d'opération.

• Le signe de la multiplication est le signe  $\times$  ; il ne faut pas l'oublier ! Par exemple "2 fois A" s'écrit  $2 \times A$  et non  $2 A$  qui provoquera une erreur.

### Fiches de référence pouvant être consultées :

- \* COLOR, SCREEN, page 135
- \* +, -, \*, /, @, MOD, page 172
- \*\* CLS, CONSOLE, page 142
- \* ATTRB, page 138



3

## Programmer

Est-il bien raisonnable de commander une chose à l'ordinateur qui l'exécute, puis de commander une autre chose, puis d'en commander une autre... et ainsi de suite ? Tout cela n'est pas très rapide et vous avez sans doute vu fonctionner des ordinateurs bien plus agréablement ; il se passait alors des tas de choses sans que personne n'intervienne jamais sur le clavier.

### ■ Préparez votre menu à l'avance

La rapidité d'action de la machine est en effet intéressante à condition de lui avoir *d'abord* communiqué une (longue) liste de commandes. Exactement comme vous pouvez le faire au restaurant en commandant un menu complet avant le repas : une fois la liste établie vous n'avez plus qu'à dire : "Vous pouvez servir !" et chacun des plats se succède alors sans que vous ayez à intervenir.

1. Un petit vin de pays et une carafe d'eau.

2. Foie gras frais maison.

(Ne lésinons pas, nous n'apprenons pas la programmation tous les jours)

3. Confit de canard et pommes à l'ail.

4. Salade.
5. Plateau de fromages.  
(Juste un peu, pour la tradition !)
6. Sorbet citron.
7. Café.
8. Fin.  
(Non merci, je n'ai plus faim !)

## ■ Une suite de commandes numérotées

Pour en revenir à votre machine, sachez donc ceci :

*Pour communiquer un "programme d'action" à votre ordinateur il vous suffit de numéroté chacune des commandes que vous lui donnez. Ainsi : toute ligne commençant par un numéro est interprétée par la machine comme faisant partie d'un programme.*

Lorsque vous frappez sur la touche **ENTREE** elle se contente de ranger cette ligne dans une espèce de cahier de texte : elle classe ainsi toutes les lignes numérotées que vous "entrez"... jusqu'à ce que vous lui commandiez l'exécution de toutes ces lignes les unes après les autres. La commande qui lui intime l'ordre d'exécution est : **RUN ENTREE** (le verbe "run" signifie, en anglais : "vas-y, cours !").

Pour être "branché" côté programmation vous devez enfin savoir qu'une commande faisant partie d'un programme s'appelle une *instruction*.

Il est temps maintenant de passer à un exemple "concret".

*Écriture d'un premier programme :*

— Depuis combien d'heures êtes-vous né ?

Pour répondre à cette angoissante question, vous pouvez frapper un programme qui ressemble à celui que j'ai réalisé pour un ami âgé de 39 ans 8 mois et 23 jours.

Frappez donc successivement :

```
1 ANS=39 ENTREE
2 MOIS=8 ENTREE
3 JOURS=23 ENTREE
4 HEURES=(365*ANS+30*MOIS+JOURS)*24 ENTREE
5 PRINT HEURES ENTREE
6 END ENTREE
```

Soyons juste : le résultat est approché à quelques jours près (certaines années sont bissextiles et les mois n'ont pas tous 30 jours) C'est-à-dire

à quelques centaines d'heures près ! Si vous trouvez cela grave, essayez de réaliser un meilleur programme (nous vous conseillons d'attendre pour cela quelques leçons).

### Remarques :

- Chaque fois que vous tapez sur **ENTREE**, le curseur passe au début de la ligne suivante ; la ligne précédente est alors enregistrée par la machine dans son espèce de "cahier de texte".
- L'instruction **END** signale à la machine qu'elle doit s'arrêter de travailler.
- Les "mots" **ANS**, **MOIS**, **JOURS**, **HEURES** sont des noms de *variables*. Souvenez-vous, en mathématiques, à l'école, les noms de variables n'avaient qu'une seule lettre (x, a ou y) ou bien une lettre suivie d'un chiffre (P0, R1...); cela manquait de poésie ou même d'humour. En fabriquant des programmes vous allez pouvoir vous venger en faisant preuve d'imagination !
- N'oubliez pas de distinguer les touches **0** et **O** !

### Attention

Pour entrer en mémoire une ligne de programme, il faut frapper sur la touche **ENTREE** ; sinon l'ordre est écrit sur l'écran mais il n'aura aucune chance d'être exécuté.

*N'utilisez pas les flèches pour déplacer le curseur d'une ligne à l'autre lorsque vous écrivez un programme.*

Si jamais vous désobéissez à ce conseil, vous aurez des ennuis qui se traduiront par une erreur de syntaxe qui vous paraîtra mystérieuse.

Un conseil : Si vous n'arrivez pas à détecter l'erreur signalée par :

Error 2 in 50

OK

... n'hésitez pas à supprimer l'instruction 50 de la mémoire en frappant :

50 ENTREE

Puis retaper une nouvelle instruction 50 si vous le jugez nécessaire.

## ■ Exécution du programme

Lorsque vous avez fini d'écrire un programme, il reste affiché à l'écran ; vous pouvez l'effacer de l'écran en appuyant sur la touche :

RAZ

N'ayez pas peur ! Il a disparu de l'écran, mais il est encore *en mémoire* quelque part dans la machine.

La preuve c'est que vous pouvez l'exécuter en frappant :

**RUN ENTREE**

... et la machine affiche alors le résultat attendu :

347952

OK

## ■ Lecture et modification du programme

Si vous souhaitez revoir le programme préalablement mis en mémoire, vous pouvez le faire écrire sur l'écran grâce à la commande LIST.

**Frappez :**

LIST ENTREE

... et la machine affiche sur l'écran la liste des instructions qu'elle a mémorisées (et cela dans l'ordre des numéros figurant en début de ligne).

Vous pouvez aussi *demandeur l'écriture d'une seule ligne.*

Par exemple frappez :

LIST 2 ENTREE

... et la machine affichera

2 MOIS=8

OK

Si vous voulez alors *modifier* cette ligne numéro 2, vous pouvez le faire à l'aide des touches de déplacement du curseur.

**Ainsi frappez :**

→ + → → → → → → → 4 **ENTREE**

(Ne pas oublier de frapper sur **ENTREE** , sinon l'instruction n'est pas modifiée.)

Si alors vous frappez :

LIST 2 **ENTREE**

la machine affichera :

2 MOIS=4

OK

*Pour remplacer une instruction par une autre, il suffit d'en frapper une nouvelle en utilisant le même numéro : comme sur un magnétophone, c'est la dernière chose enregistrée qui reste. Vous pouvez donc refrapper chacune des 3 premières lignes de ce programme pour l'adapter à telle ou telle de vos amies.*

Enfin pour effacer une instruction, il suffit de frapper son numéro suivi immédiatement de **ENTREE** (En fait, on remplace une instruction par une instruction vide). L'effacement a lieu dans la mémoire et non sur l'écran !

## ■ La touche **BASIC**

Cette touche permet d'écrire plus rapidement vos programmes.

Ainsi au lieu d'écrire :

**S C R E E N**

vous pouvez frapper simultanément sur

## BASIC SCREEN

Ainsi pour chaque "instruction" écrite sur la partie basse de chaque touche.

## Exercices



1. Écrire et exécuter le programme suivant :

- ```

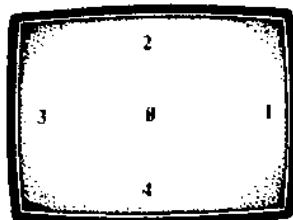
1 SCREEN 0,1,0
2 SCREEN 0,2,0
3 SCREEN 0,3,0
4 SCREEN 0,4,0
5 SCREEN 0,5,0
6 SCREEN 0,6,0
7 SCREEN 0,7,0

```

Une fois frappé n'oubliez pas de taper **R U N** **ENTREE** pour exécuter ce programme.



2. Modifier le programme précédent de manière qu'à l'exécution le fond reste le même et les caractères écrits changent de couleur.
3. Écrire un programme qui affiche en rouge le nombre 17, en bleu son carré ( $17 * 17$ ) et en noir son cube ( $17 * 17 * 17$ ).
4. Écrire un programme qui affiche ceci :



Utilisez pour cela l'instruction :

**LOCATE 20,1**

Cette instruction positionne le curseur sur la position numéro 20 de la ligne numéro 1.

Plus généralement **LOCATE X, Y** amène le curseur sur la position numéro X de la ligne numéro Y.

## Attention

■ Avant de frapper un nouveau programme, il vaut mieux vous débarrasser de celui que vous avez précédemment frappé et qui ne vous sert plus qu'à encombrer la mémoire du MO5.

La commande qui efface les programmes en mémoire est :

**N E W** **ENTREE**

Vous pouvez aussi éteindre puis rallumer votre MO5.

## Attention

■ *Frappes PUIS exécutez un programme.*

Retenez bien ceci : Pour arriver à "faire marcher" un programme il faut bien distinguer 2 étapes :

— 1<sup>re</sup> étape : *Frappes du programme.*

Dans cette étape les lignes sont frappées les unes après les autres ; chaque ligne commence par un chiffre. La machine n'exécute aucun ordre ; elle ne fait que recopier en mémoire chacune des lignes d'instruction.

— 2<sup>e</sup> étape : *Demande d'exécution du programme.*

Cette étape débute lorsque vous frappez sur **R U N** **ENTREE**. La machine exécute alors la suite des instructions enregistrées lors de la première étape.

■ Les lignes d'instructions écrites sur l'écran ne sont pas forcément les lignes d'instructions qui sont entrées en mémoire. Tout dépend de la position du curseur lorsque vous appuyez sur **ENTREE**.

Pour vérifier ce qui est réellement en mémoire, commandez **LIST** **ENTREE**.

■ Pour l'instant n'écrivez qu'une instruction par ligne.

En effet, si vous écrivez la ligne...

3 COLOR 1 PRINT 17

... la machine ne comprendra pas le morceau de phrase "PRINT 17"

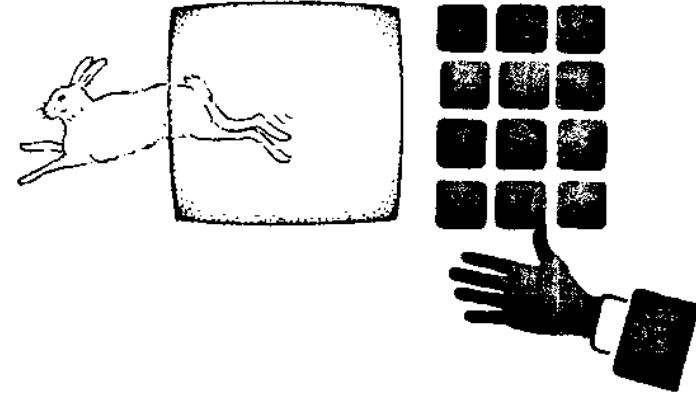
Il faut écrire :

3 COLOR 1

4 PRINT 17

## Fiches de référence pouvant être consultées :

- \* \* LOCATE, page 145
- \* LIST, DELETE, NEW, page 155
- \* \* Ligne de programme, page 146



4

## Faire et refaire

Vous voulez que votre ordinateur travaille beaucoup, mais vous ne pouvez pas passer votre vie à enregistrer des instructions dans votre micro-ordinateur !

Comme il les exécute ensuite très, très vite, son travail est vraiment dérisoire en regard du temps que vous avez passé à écrire au clavier.

### ■ GOTO...

En fait, sa rapidité d'exécution n'a de véritable intérêt que s'il exécute de très nombreuses fois une même petite suite d'instructions.

L'instruction lui intimant l'ordre d'exécuter de nouveau une certaine instruction, se présente sous la forme :

"GOTO un numéro d'instruction"

("go to" signifie "aller en").

Après ce "saut", il continue naturellement à exécuter les instructions qui suivent dans l'ordre de leurs numéros.

### ■ Un programme de clignotant

Voici un programme de clignotant éternel illustrant l'utilisation du "GOTO".

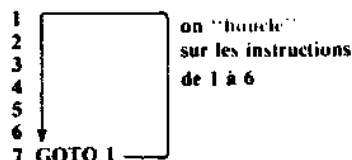
|                |                                                                                           |
|----------------|-------------------------------------------------------------------------------------------|
| 1 LOCATE 20,12 | le curseur se positionne au milieu de l'écran (position numéro 20 de la ligne numéro 12). |
| 2 COLOR 1,2    | on écrira rouge sur fond vert                                                             |
| 3 PRINT 0      | écrire le nombre zéro                                                                     |
| 4 LOCATE 20,12 |                                                                                           |
| 5 COLOR 2,2    | on écrira vert sur fond vert                                                              |
| 6 PRINT 0      | écrire le nombre zéro                                                                     |
| 7 GOTO 1       | aller refaire l'instruction numéro 1.                                                     |



**Remarque :** C'est encore plus joli si vous complétez la première instruction comme ceci :

1 LOCATE 20,12 : ATTRB 1,1  
(Voir la fiche de référence ATTRB).

Vous venez de réaliser votre première *boucle* de programme :



## ■ STOP

Évidemment le spectacle n'est pas très varié et, de plus, il n'est pas prêt de se terminer.

Vous pouvez l'arrêter en intervenant "manuellement" sur le clavier.

Dès que vous frappez la touche **STOP**, l'exécution s'arrête.

C'est là qu'une surprise vous attend : dès que vous frappez une autre touche, la machine se remet à marcher et reprend l'exécution indéfinie du programme.

Vous êtes alors désespéré, coincé dans une *boucle sans fin* que vous pouvez juste interrompre de temps en temps pour vous reposer les yeux !

Cependant, il y a un moyen de vous en sortir **sans éteindre carrément** la machine. Ce moyen est un peu curieux mais c'est ainsi : il suffit d'appuyer simultanément sur les touches **CNT** et **C** (en langage parlé : "control cé").

**CNT-C**

... et la machine interrompt définitivement l'exécution du programme et indique même le numéro de l'instruction où elle a été interrompue ; elle affiche, par exemple, le message :

Break in 5  
OK

## ■ Un programme de compteur

Refaire toujours exactement la même chose, cela n'est pas franchement excitant. Mais si un petit quelque chose n'est plus le même à chaque exécution de la boucle, alors s'ouvre devant nous une formidable variété de possibilités.

Témoin, le petit programme suivant.

Le changement consiste à augmenter de 1 la valeur d'une *variable* entre deux exécutions de la boucle.

Cette augmentation est commandée par l'instruction  $A = A + 1$ . (le fonctionnement détaillé de cette commande sera expliqué à la leçon 7).

|                |                                                         |
|----------------|---------------------------------------------------------|
| 1 A=0          | A prend la valeur 0                                     |
| 2 LOCATE 20,12 | positionnement du curseur (voir remarque du clignotant) |
| 3 PRINT A      | écriture de la valeur de A                              |
| 4 A=A+1        | A augmente sa valeur de 1                               |
| 5 GOTO 2       | refaire l'instruction numéro 2.                         |

Pour arrêter l'exécution, vous pouvez frapper la touche **STOP** quand vous le voulez. D'où l'idée d'un petit jeu de réflexe utilisant ce programme.

## ■ Un jeu de réflexe

- Enregistrez le programme précédent et frappez :

**R U N** **ENTREE**

- Essayez d'arrêter l'exécution du programme sur l'affichage d'un nombre rond (100, 200,...) en frappant sur :

**STOP**

- En frappant une autre touche, le programme redémarre et vous pouvez continuer à jouer chacun votre tour en essayant de viser de mieux en mieux.

## Exercices



1. Nous vous proposons de modifier quelque peu le programme du compteur (utiliser l'affichage de la liste des instructions — LIST — et le "curseur-pleine-page", c'est-à-dire "la possibilité, pour le curseur de se déplacer et d'écrire sur toute la page").

Pour chaque modification, essayez d'abord de prévoir le résultat de l'exécution ; ensuite déclenchez l'exécution pour vérifier vos prévisions :

- a) Remplacez l'instruction numéro 1 par :

1 A=2

- b) Remplacez l'instruction numéro 4 par :

4 A=A\*2

- c) Remplacez l'instruction numéro 4 par :

4 A=A+A

- d) Remplacez l'instruction numéro 4 par :

4 A=A\*A

- e) Supprimez l'instruction numéro 2 et remplacez l'instruction 5 par :

5 GOTO 3

2. Pour en voir de *toutes les couleurs*, analysez, frappez et exécutez le programme suivant :

```
1 I=1
2 SCREEN I-1, I+1
3 I=I+1
4 GOTO 2
```

(Évidemment après une quinzaine de changements de décor, il y a quelque chose qui ne va pas ! La leçon 6 vous sortira de ce mauvais pas).

3. Analysez, frappez et exécutez le programme suivant :

```
1 CLS
2 K=0
3 COLOR K
4 PRINT 888888
5 K=K+1
6 GOTO 3
```

4. Puisque vous commencez à comprendre ce qu'est un programme, il est temps de faire un peu d'humour avec le **programme-gag** suivant :

```
1 GOTO 1
```

**Fiche de référence pouvant être consultée :**

\* RUN, STOP, END, CNT-C, GOTO, page 153

# Résumé et conseils aux apprentis programmeurs

## ■ Frapper un programme et le faire exécuter

- *Pour enregistrer votre programme, instruction par instruction :*
  - Toute instruction doit être numérotée en début de ligne.
  - Elle est enregistrée en mémoire dès que vous frappez la touche **ENTREE**
- *Pour faire exécuter votre programme par la machine frappez :*
  - **RUN** **ENTREE**
  - Si vous voulez interrompre l'exécution frappez **STOP** ou **CNT-C**
- *Pour afficher la liste des instructions déjà enregistrées en mémoire frappez :*
  - **LIST** **ENTREE**
- *Pour modifier une instruction :*
  - Affichez cette instruction en commandant par exemple **LIST 7** **ENTREE** (et l'instruction numéro 7 s'affiche).
  - Utilisez le curseur plein-écran pour changer, ajouter ou supprimer des caractères.
  - N'oubliez pas de frapper **ENTREE** pour enregistrer la nouvelle instruction. Vous pouvez faire lister l'instruction pour vérifier.
- *Pour effacer l'instruction numéro 12 frappez :*
  - **1 2** **ENTREE**

## ■ Attention à quelques erreurs classiques

- Si vous frappez

1 RUN **ENTREE**, il ne se passera rien.

La seule chose que fera la machine, est d'enregistrer l'instruction numéro 1 dans sa mémoire.

- Si vous frappez

10+25 **ENTREE**

N'espérez pas que la machine fasse la somme de 10 et de 25. Puisque la ligne commence par un numéro, elle interprète 10 comme un numéro d'instruction et elle range alors l'instruction qui s'énonce : "+ 25". D'ailleurs elle ne comprendra rien à cette instruction.

**! Alarme !**

Après avoir frappé 10 + 25 **ENTREE**, il y a plus grave.

Si jamais, vous aviez déjà enregistré une instruction numéro 10, elle a été effacée par cette bêtise : "+ 25".

- **ATTENTION :** *Ce qui est affiché sur l'écran n'est pas forcément en mémoire et inversement.*

On oublie souvent par exemple, d'effacer une instruction dont on a modifié le numéro.

Soit par exemple, l'instruction affichée :

7 COLOR 1,2

Si on désire plutôt donner à cette instruction le numéro 9 (parce qu'elle doit suivre et non précéder l'instruction 8), alors il faut faire successivement ceci :

— Amenez le curseur sous le 7 en début de ligne

7 COLOR 1,2

— Frappez le chiffre 9

9 COLOR 1,2

— Frappez **ENTREE**

L'instruction 9 COLOR 1,2 est alors enregistrée.

**Attention !**

L'instruction numéro 7 n'est plus affichée à l'écran mais elle existe toujours en mémoire.

Il faut donc l'effacer en frappant :

7 **ENTREE**

Il n'y a plus alors d'instruction numéro 7 en mémoire, mais il y a une nouvelle instruction numéro 9.

## ■ Quelques conseils

- Pour savoir ce que fait un programme ou pour le corriger *mettez-vous à la place de la machine :*

Exécutez fidèlement les instructions les unes après les autres en imitant les actions commandées à la machine.

- N'hésitez pas à *numéroter vos instructions de 10 en 10* (ou de 5 en 5) afin de pouvoir ultérieurement insérer de nouvelles instructions intermédiaires.

- Cette habitude vous aidera aussi à *écrire d'abord l'architecture générale* de votre programme, *puis* (lorsqu'il marche bien) à le "*raffiner*" par des instructions qui améliorent ses performances ou sa présentation. Ne cherchez donc pas tout de suite à réaliser un programme sophistiqué.

Soyez, à la fois, humble et sûr de vous : il n'y a rien de mystérieux ; la machine fait exactement ce que vous lui commandez de faire !

- Utilisez la possibilité de *placer des commentaires* dans vos instructions grâce à l'apostrophe.

9 COLOR 1,2 ' rouge sur vert

En début de ligne l'apostrophe peut être remplacée par REM (début de "*remarque*"). Par exemple :

1 REM JEU DU COMPTEUR

... et l'instruction 1 n'est ni interprétée ni exécutée par la machine.

## ■ Voici les instructions et commandes que vous connaissez déjà :

CLS

LOCATE 12,17

COLOR 4

COLOR 5,12

SCREEN 2,11,8

X = X + 1

PRINT 17

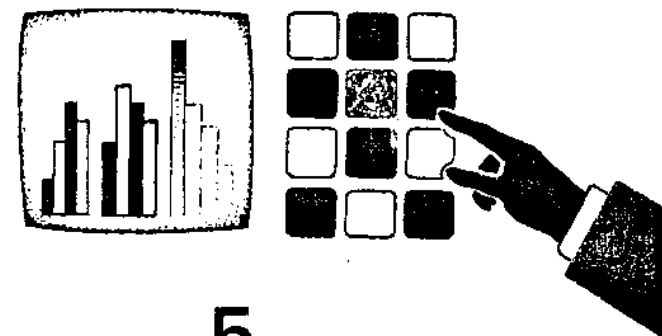
PRINT X  
GOTO 4  
RUN  
LIST  
LIST 8  
STOP  
CNT-C

Si vous vous sentez plus à l'aise, n'hésitez pas à consulter les *fiches de référence* pour avoir plus de détails sur ces instructions.

Consultez aussi l'*Index* en dernière page.

N'oubliez pas de frapper **ENTREE** pour indiquer la fin de chaque commande ou de chaque instruction. C'est la touche qui fait toute la différence entre une machine à écrire et un ordinateur !

C'est une sorte de point d'exécution terminant une phrase et ordonnant d'agir de manière appropriée.



5

## Graphiques

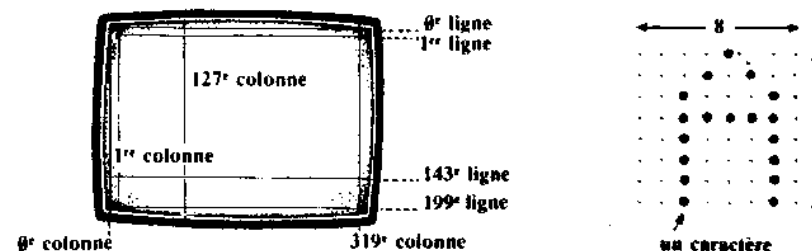
Peut-être avez vous vu chez des amis de magnifiques dessins en couleurs, affichés sur leur écran de télévision. On ne vous a pas encore parlé de la manière d'obtenir de telles œuvres d'art. Vous allez bientôt tout savoir sur le sujet.

### ■ Une feuille de dessin

Pratiquement, tout se passe comme si votre écran était une feuille de dessin sur laquelle, vous pouvez marquer des points de couleur en 64 000 endroits précis. Précisément, chaque position de caractère est constituée de 8 points en ligne et 8 points en colonne et vous disposez de 25 lignes de 40 caractères.

Votre "feuille" est donc constituée de 200 lignes et de 320 points.

Ces lignes et ces colonnes de points sont "numérotées" comme indiqué sur le schéma suivant :



## ■ Points

La commande :

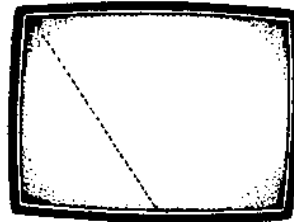
PSET (127,143) **ENTREE**

a pour effet, d'allumer le point situé sur la 127<sup>e</sup> colonne et la 143<sup>e</sup> ligne.

La couleur d'allumage est celle précédemment choisie pour l'écriture.

Voici trois programmes utilisant cette instruction et leur effet sur l'écran :

```
1 I=0 : CLS
2 COLOR 1
3 PSET (5*I,5*I)
4 I=I+1
5 GOTO 3
```



Tracé d'une "diagonale" de points rouges puis de quelques points en bas de l'écran.

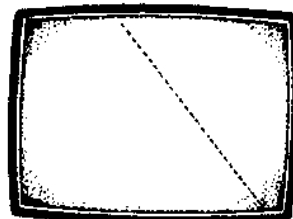
Arrêtez l'exécution en frappant **CNT-C** ; sinon après quelques temps vous obtiendrez un message d'erreur numéro 6.

```
1 I=0
2 CLS
3 COLOR 1
4 PSET (I*5, I*5)
5 I=I+1
6 GOTO 3
```



Tracé de 16 points de couleurs différentes puis arrêt sur un message d'erreur numéro 5, écrit en orange (il y a eu essai d'écrire avec la couleur numéro 16 qui n'existe pas).

```
1 I=319
2 COLOR 1
3 PSET(I, I-120)
4 I=I-1
5 GOTO 3
```



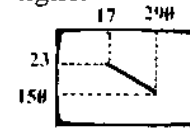
Tracé d'une "diagonale" issue du coin droit en bas et arrêt sur un message d'erreur numéro 5 (il y a eu essai de tracer un point de coordonnées négatives).

## ■ Lignes

La commande :

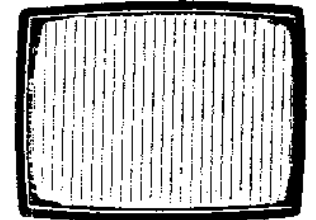
LINE (17,23)-(290,150) **ENTREE**

a pour effet de tracer un segment dont une extrémité est le point de la 17<sup>e</sup> colonne et 23<sup>e</sup> ligne, et dont l'autre extrémité est le point de la 290<sup>e</sup> colonne et 150<sup>e</sup> ligne.

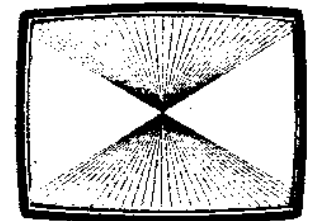


Voici trois programmes utilisant cette instruction et leur effet sur l'écran :

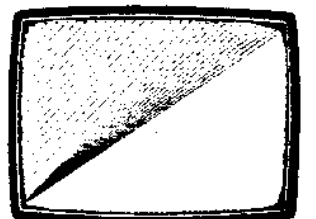
```
1 C=0
2 COLOR 1
3 LINE(C,0)-(C,199)
4 C=C+10
5 GOTO 3
```



Même programme avec :  
3 LINE (C,0)-(319-C,199)



Même programme avec :  
3 LINE (C,0)-(0,C)



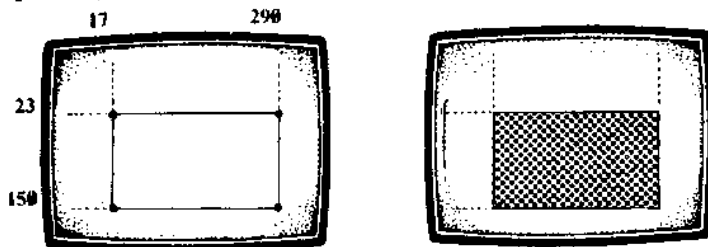


## ■ Boîtes

La commande :

BOX (17,23)-(290,150) **ENTREE**

a pour effet, de tracer le contour d'un rectangle ayant le point (17,23) et le point (290, 150) comme sommets opposés.



La commande :

BOXF(17,23)-(290,150) **ENTREE**

a presque le même effet mais l'intérieur du rectangle est colorié (de la couleur choisie précédemment pour l'écriture).



**Remarque :**

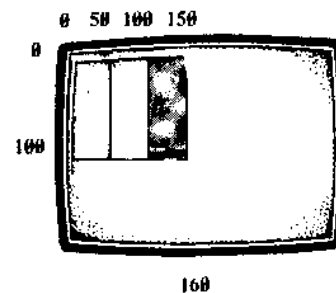
BOX signifie "boîte" et F est la première lettre de full ("pleine")

## ■ Drapeaux et emboîtements

Voici trois programmes utilisant cette instruction et leur effet sur l'écran :

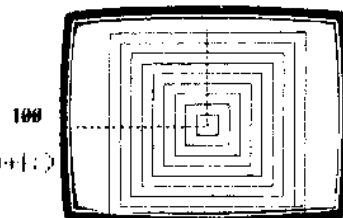
### drapeaux

```
1 COLOR 4
2 BOXF(0,0)-(50,100)
3 COLOR 7
4 BOXF(50,100)-(100,0)
5 COLOR 1
6 BOXF(100,0)-(150,100)
```



### carrés centrés

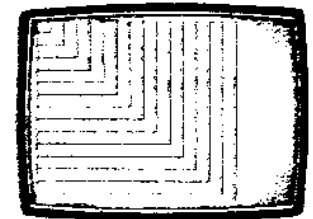
```
1 B=0
2 COLOR 1
3 BOX(160-B,100-B)-(160+B,100+B)
4 B=B+10
5 GOTO 3
```



essayez le même programme avec :

2 COLOR B/10

5 GOTO 2



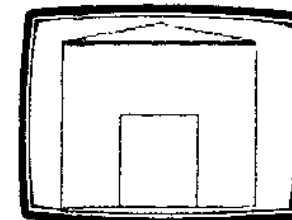
### carrés en coin

```
1 B=15 : LOCATE 0,20
2 COLOR B
3 BOXF(0,0)-(10*B,10*B)
4 B=B-1
5 GOTO 2
```

## Exercices



1. Écrire un programme qui dessine une maison :

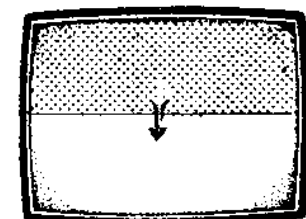
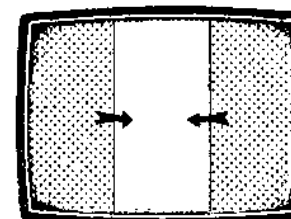


(Vous ne saurez colorier l'intérieur du triangle constituant le toit en rouge qu'après la leçon suivante).

2. Écrire un programme qui "ferme le rideau"

comme ceci :

ou comme cela :



3. Écrire un programme dessinant des boîtes pleines emboîtées.



4. Que dessine le programme suivant ?

```
1 CLS
2 COLOR 0
3 LINE(100,100)-(100,50)
4 LINE -(200,50)
5 LINE -(150,20)
6 LINE -(100,50)
7 LINE -(200,100)
8 LINE -(200,50)
9 LINE -(100,100)
10 LINE -(200,100)
```

Sachez que l'instruction `LINE - (X, Y)` a pour effet de tracer une ligne depuis le dernier point dessiné jusqu'au point (X, Y).

### Attention

■ Lorsque vous listez et lorsque vous commandez l'exécution d'un programme, il y a du texte écrit sur l'écran. Si l'exécution du programme provoque le tracé de dessins, il peut y avoir superposition du texte et des dessins et ce n'est pas joli du tout.

Il vaut donc mieux commencer tout programme de dessin par l'instruction qui *Chasse La Saleté* :

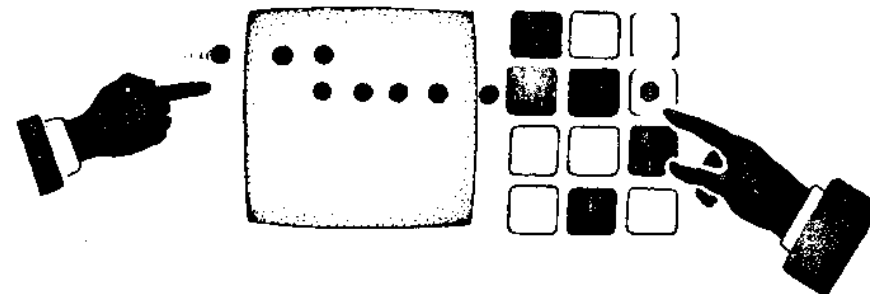
1 CLS (en anglais : *Clear Screen*)

■ Lorsque vous demandez le tracé sur l'écran blanc d'un segment rouge puis d'un segment bleu coupant le rouge, il y a une petite bavure au voisinage de l'intersection. (Les fiches de référence \*\* vous en expliqueront les causes).

### Fiches de références pouvant être consultées :

\* PSET, LINE, BOX, BOXF, page 201

\*\* Fond et écriture, page 203



## 6

# Répétitions et aiguillages

Dans la leçon précédente, quelques programmes donnés en exemple s'arrêtent sur un message d'erreur.

Il doit pourtant bien avoir un moyen de refaire quelque chose un certain nombre de fois, au lieu d'avoir à le refaire soit indéfiniment soit jusqu'à ce qu'un incident vienne interrompre le cours des événements.

## ■ Répétez 7 fois

Précisément : Pour répéter 7 fois une suite d'instructions (ici numérotées de 11 à 19), on écrit :

```
10 FOR K= 1 TO 7
```

```
11 '
```

```
12 '
```

```
13 '
```

```
14 '
```

```
15 '
```

```
16 '
```

```
17 '
```

```
18 '
```

```
19 '
```

```
20 NEXT K
```

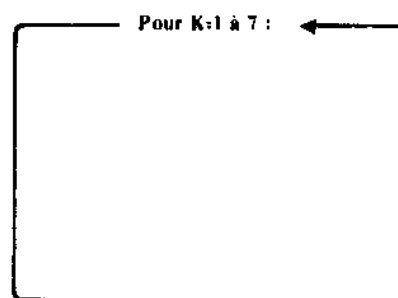
10 Pour  $K = 1$  à 7 exécuter des instructions : 11, 12, 13, 14, 15, 16, 17, 18, 19

20 Revenir en 10 avec la valeur de K suivante (c'est-à-dire augmentée de 1) ; mais lorsque K vaut  $7 + 1$ , passer à l'instruction suivante.

La traduction en français des deux instructions "FOR..." et "NEXT..." est donnée ci-dessus en couleur.

On peut aussi imaginer que le programme précédent est équivalent à la suite d'instructions :

```
K = 1
suite des instructions 11 à 19
K = 2
suite des instructions 11 à 19
K = 3
suite des instructions 11 à 19
K = 4
suite des instructions 11 à 19
K = 5
suite des instructions 11 à 19
K = 6
suite des instructions 11 à 19
K = 7
suite des instructions 11 à 19
```



On dit que l'on exécute une "boucle" pour K variant de 1 à 7. Dans la leçon 4, nous avons déjà rencontré une "boucle" sans fin. Maintenant nous savons boucler un certain nombre de fois précisé par l'instruction FOR...

FOR... indique le début de la boucle ; NEXT indique la fin de la boucle c'est-à-dire l'endroit où l'exécution est renvoyée en début de boucle si la variable n'a pas atteint sa valeur maximum.



## ■ Les 16 couleurs

Vous pouvez alors reprendre chacun des programmes donnés dans la leçon précédente en remplaçant le brutal 6 GOTO 2 par quelque chose du type :

```
2 FOR C=0 TO 15
:
:
6 NEXT C
```

La variable C représente la couleur d'écriture. Pour rendre les programmes plus lisibles, on peut d'ailleurs lui donner un nom de 7 lettres au lieu d'une. Comme ceci :

```
2 FOR COULEUR=0 TO 15
:
:
6 NEXT COULEUR
```

Pour l'exemple, voici deux programmes de démonstrations des 16 couleurs du MO 5.

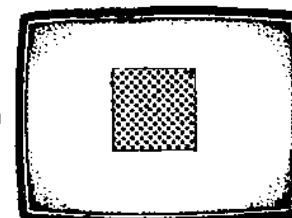
```
1 CLS
2 FOR COULEUR=0 TO 15
3 BOXFC(16*COULEUR,0)-(16*COULEUR+15,199)
:COULEUR
4 NEXT COULEUR
5 END
```

(Les 16 bandes colorées)



```
1 CLS
2 FOR COULEUR=0 TO 15
3 BOXFC(100,40)-(220,160),COULEUR
4 NEXT COULEUR
5 GOTO2
```

(pulsation d'un carré de couleur)



## ■ Si... alors... sinon...

Les boucles commandées par GOTO... équivalent à :

RÉPÉTER quelque chose A L'INFINI.

Les boucles commandées par FOR... NEXT équivalent à :

RÉPÉTER quelque chose UN CERTAIN NOMBRE DE FOIS.

Mais il arrive que l'on ne sache pas à l'avance le nombre de répétitions que l'on désire. Par exemple, on peut désirer continuer JUSQU'A CE QU'une certaine condition soit réalisée (ou un certain objectif atteint) ou bien TANT QU'une certaine condition reste vraie (par exemple, on dispose encore de carburant). Ou bien encore, on peut vouloir faire telle ou telle chose SELON la valeur d'une certaine variable.

Le langage BASIC dispose d'une instruction de "branchement conditionnel" pour s'adapter aux différents désirs énoncés ci-dessus.

Cette instruction se présente ainsi :

IF X=3 THEN GOTO 7 ELSE PRINT X

A l'exécution, la machine calcule si la condition énoncée après IF est vraie ou fausse. (ici "X est-il égal à 3"?).

Si la condition est vraie *alors* elle exécute l'instruction écrite après THEN (ici elle va en 7 suivre son destin).

Si la condition est fausse (ELSE signifie "sinon"), elle exécute l'instruction écrite après ELSE (ici elle écrit la valeur de X), puis elle passe à l'instruction suivante.

## ■ Un programme de carrés géométriquement emboîtés

Le programme suivant dessine un carré puis, à l'intérieur, un carré de côté égal à 90 % du précédent et ainsi de suite. On arrête de dessiner lorsque le carré devient trop petit (précisément lorsqu'il devient de la taille d'un caractère : son côté vaut alors 8 points).

```
5 CLS
10 COTE=100
15 BOX(0,0)-(COTE,COTE)
20 COTE=0.90*COTE
25 IF COTE<8 THEN GOTO 30 ELSE GOTO 15
30 LOCATE 0,15
35 END
```



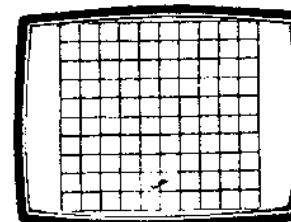
### Remarques :

- L'instruction IF... THEN... ELSE... admet de multiples variantes en particulier pour simplifier son écriture. Mais il vaut mieux attendre d'avoir une certaine habitude de la programmation pour les utiliser (voir la fiche de référence).
- Dans l'écriture décimale des nombres, notre virgule française est remplacée par un point anglo-saxon.

## Exercices



1. Dessiner un quadrillage régulier sur l'écran.



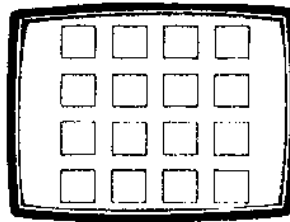
Il vous faudra pour cela utiliser 2 boucles successives du genre :

```
[ FOR X=0 TO 10
  ...
NEXT X

[ FOR Y=0 TO 10
  ...
NEXT Y
```

2. Analyser et expérimenter le programme suivant :

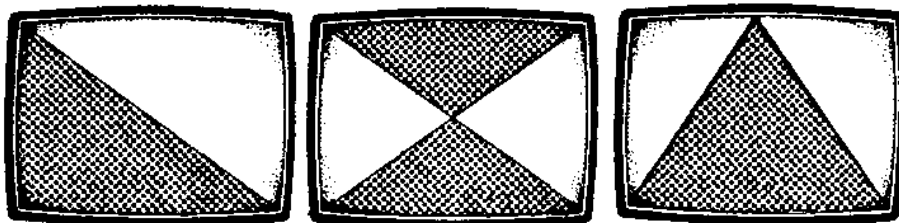
```
1 CLS
2 FOR I=0 TO 3
3   FOR J=0 TO 3
4     COLOR I+4*J
5     BOXF(10+30*I,10+30*J)
      -(30+30*I,30+30*J)
6   NEXT J
7 NEXT I
```



Ici les 2 boucles utilisées sont "imbriquées"

```
FOR I=0 TO 3
  FOR J=0 TO 3
    ...
    ...
  NEXT J
NEXT I
```

3. Dessiner les figures suivantes :



4. Comparer l'effet du programme des "carrés géométriquement emboîtés" avec l'effet du suivant :

```
10 CLS
20 C=100
30 FOR K=1 TO 10
32   BOX(0,0)-(C,C)
34   C=C-10*K
36 NEXT K
40 LOCATE 0,15
50 END
```

5. Reprendre les programmes de la leçon 5 en transformant toutes les boucles commandées par un GOTO en boucles commandées par FOR... NEXT, de manière que l'exécution s'arrête sur une instruction END.

## Attention

■ Sur les listes de vos programmes repérez les boucles FOR... NEXT par un signe schématisé du genre suivant, cela vous aidera à mieux voir la structure de vos programmes :

```
FOR K = .....
.....
.....
NEXT K
```

Vous pouvez aussi décaler les instructions internes à une boucle comme nous l'avons fait dans les exercices 2 et 4

■ L'instruction END n'est pas obligatoire dans un programme. Mais cela fait plus "propre"... et évite les ennuis ultérieurs.

■ L'instruction  
IF C<8 THEN GOTO 30 ELSE GOTO 15  
peut être abrégée en  
IF C<8 THEN 30 ELSE 15

En fait, vous verrez en consultant la fiche de référence que l'instruction IF... THEN... ELSE peut prendre des formes très diverses.

Dans un premier temps nous vous conseillons de n'utiliser que la forme abrégée ci-dessus.

Et si, comme la chèvre de monsieur Seguin, vous vous aventurez trop loin en solitaire, il se peut que vous ayez à lutter une nuit entière contre les "bugs" qui risquent de vous harceler.

### Fiches de références pouvant être consultées :

- \* FOR... NEXT, page 161
- \*\* IF... THEN... ELSE..., page 166
- \*\* ON... GOTO, page 165
- \* RUN, STOP, END, CNT-C, CONT, GOTO, page 153



7

## Nombres et opérations

Vous pensiez peut-être que les ordinateurs étaient faits pour manipuler des chiffres et cela n'a pas été tellement le cas jusqu'ici. Il est temps de combler cette lacune.

### ■ Le signe =

Nous avons déjà rencontré des instructions permettant "d'affecter" des valeurs numériques à des variables représentées par une ou plusieurs lettres.

Ainsi par exemple :

`ANS=39`

`A=0`

`I=1`

`C=100`

Nous avons aussi rencontré et utilisé des "instructions d'affectation de valeurs" un peu plus complexes :

`HEURES=(365*ANS+30*MOIS+JOURS)*24`

`A=A+1`

`B=B+10`

`C=0.90*C`

leçon 3

leçon 4

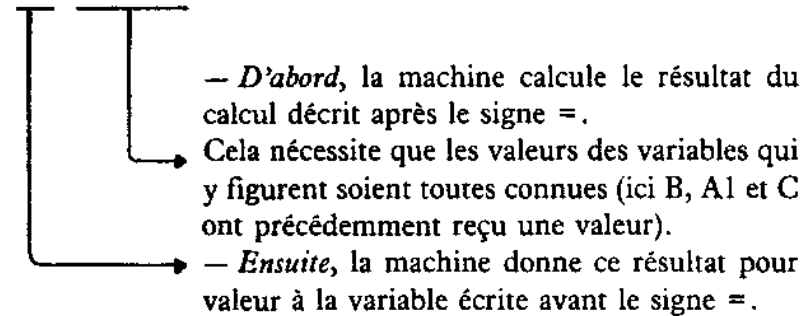
leçon 5

leçon 6

Il est temps maintenant d'expliciter le comportement de la machine dans l'exécution d'une telle instruction :

Pour exécuter l'instruction :

**A1=B+A1+C**



Remarquez bien que, dans ce processus, rien ne s'oppose à ce que le même nom de variable figure avant et après le signe =. En fait cette opportunité est exactement ce qui fait la puissance de la plupart des programmes...



## ■ Un programme de tables numériques

Les opérations et les fonctions "classiques" sont disponibles sur le MO 5 (voir, pour le détail, la fiche de référence).

A titre d'exemple, voici un programme très simple qui édite une table des carrés et des cubes des nombres entiers.

```
5 CLS
10 N=1
20 PRINT N, N*N, N*N*N
30 N=N+1
40 GOTO 20
50 END
```

A l'exécution, la table se met à défiler très vite, on peut la consulter en frappant sur **STOP** au moment opportun.

## Remarques

- On peut aussi ralentir l'exécution en faisant exécuter une "boucle vide" comme ceci :

```
25 FOR K =1 TO 500
26 NEXT K
```

Essayez aussi en changeant 500 en 100 ou en 1000.

- En remplaçant la ligne 20 par :

```
20 PRINT N, LOG(N)
```

on édite une table de logarithmes.

- En remplaçant la ligne 20 par :

```
20 PRINT N, COS(N), SIN(N)
```

on édite une table trigonométrique (attention à l'unité d'angle : voir la fiche de référence des fonctions).

## ■ Un programme pour pousser les divisions aussi loin qu'on le veut

Rappelez-vous la technique de la division en usage chez nous.

$$\begin{array}{r} 59 \overline{) 347} \\ 3 \overline{) 59} \end{array}$$

$$\begin{array}{r} A \overline{) B} \\ R \overline{) Q} \end{array}$$

Le "programme" de son exécution n'est-il pas le suivant ?

- Soit à diviser A par B
  - Écrire le quotient (entier) de A par B ; appelons-le QUOTIENT
- Le signe de la division entière est le signe @.
- Soustraire B × QUOTIENT de A ; on obtient le RESTE.
  - Abaisser un zéro c'est-à-dire "multiplier le reste par 10".
  - Faire comme si RESTE était le nouveau nombre à diviser par B (autrement dit traiter RESTE comme A).
- et ainsi de suite...

Ce programme se traduit très bien en BASIC (quasiment mot à mot).

```
1 CLS
2 A=365
3 B=7
4 QUOTIENT=A@B
5 PRINT QUOTIENT
6 RESTE=A-B*QUOTIENT
7 RESTE=10*RESTE
8 A=RESTE
9 GOTO 4
```

**Remarque :** L'affichage du résultat est très spectaculaire si on termine l'instruction 5 par un point virgule (;) qui indique à la machine de ne pas aller à la ligne après avoir écrit la valeur de Q.

Voici le résultat de l'exécution de ce programme :

```

RUN  ENTREE
52  1  4  2  8  5  7  1  4  2  8  5  7
1   4  2  8  5  7  1  4  2  8  5  7  1
4   2  8  5  7  1  4  2  8  5  7  1  4
2   8  5  7  1  4  2  8  5  7  1  4  2
8   5  7  1  4  2  8  5  7  1  4  2  8
5   7  1  4  2  8  5  7  1  4  2  8  5 ...

```

## ■ Tirer des nombres au hasard

La machine que vous utilisez offre une possibilité à laquelle vous ne vous attendiez pas :

Elle peut tirer des nombres au hasard.

Ainsi lorsque vous commandez l'instruction :

A=RND

... vous êtes strictement incapable de savoir quelle sera la valeur de A.

*En fait A peut prendre n'importe quelle valeur (décimale) entre 0.000000 et 0.999999.*

Pour vous en persuader, exécutez le programme suivant :

```

10 FOR I =1 TO 10
20 A=RND
30 PRINT A
40 NEXT I

```

Et vous obtenez :

```

.594972
.512679
.58108E-02  nombre égal à 0.0158108 (voir page 182)
.428832
.16661
.62274
.906189
.430424
4.96011E-02
9.94663E-02

```

Le nom RND est l'abréviation de l'anglais *random* qui signifie "hasard".

## ■ Un programme simulant le jet d'un dé

Cela peut vous permettre, par exemple, de "simuler" le jet d'un dé à 6 faces par la machine.

Pour cela il faudrait que RND ait une valeur qui ne soit pas comprise entre 0 et 1 mais soit égal à l'un des 6 nombres entiers : 1, 2, 3, 4, 5 ou 6.

Heureusement nous disposons de la fonction INT qui prend la "partie entière" d'un nombre (par exemple : INT (4.356) vaut 4). Pour obtenir, au hasard, le numéro de la face d'un dé, il suffit alors de s'apercevoir que :

RND est un nombre compris entre 0 et 0.999999

6 \* RND est un nombre compris entre 0 et 5.99999

1 + 6 \* RND est un nombre compris entre 1 et 6.99999

INT (1 + 6 \* RND) est un des nombres 1, 2, 3, 4, 5 ou 6.

Voici donc un programme simulant le jet d'un dé et son tournoiement au centre de l'écran.

Lorsque vous frappez sur STOP, il s'arrête de "rouler".

```

1 CLS
2 ATTRB 1,1
3 LOCATE 20,12
4 D=INT(1+6*RND)
5 PRINT D
6 GOTO 2

```

**Note :** L'instruction ATTRB 1, 1 commande l'écriture de caractères de grandeur double par rapport à la grandeur habituelle. Voir la fiche de référence.



## Exercices



1. L'inflation étant de 10 % par an et la baguette de pain coûtant 2 F en 1981, vérifier que le programme suivant affiche le prix de la baguette jusqu'à l'an 2000.

```
1 CLS
2 INFLATION=0.10
3 PRIX=2
4 ANNEE=1981
5 FOR N=1 TO 20
6 PRINT ANNEE, PRIX
7 ANNEE=ANNEE+1
8 PRIX=PRIX*(1+INFLATION)
9 NEXT N
```



**Remarque :** La variable N ne sert ici que pour "compter" de 1 à 20 et n'est pas utilisée dans les calculs.

2. Quel rapport le programme suivant a-t-il avec l'astronomie ?

```
1 CLS
2 X=319#RND
3 Y=199#RND
4 PSET(X,Y)
5 GOTO 2
```

3. Le programme suivant provoque l'affichage d'une table de multiplication des nombres de 4 à 9.

```
1 CLS
10 FOR I=4 TO 9
20 FOR J=4 TO 9
30 PRINT I*J
40 NEXT J
50 PRINT
60 NEXT I
```

Analysez les effets des instructions 50 et 30 et comparez avec l'instruction 6 du programme de l'exercice 1. Quels peuvent être les rôles de la virgule et du point virgule ?

## Attention

■ Le même signe "=" est utilisé dans les instructions comme...

```
P=2
A=A+1
IF X=0 THEN 20 ELSE 80
FOR N=1 TO 20
```

... avec des significations bien différentes.

Relisez attentivement le premier paragraphe de cette leçon.

■ Lorsque vous employez une variable (par exemple PRIX, ANS, I ou N), pensez qu'il doit y avoir une instruction où cette variable prend une valeur numérique pour la première fois.

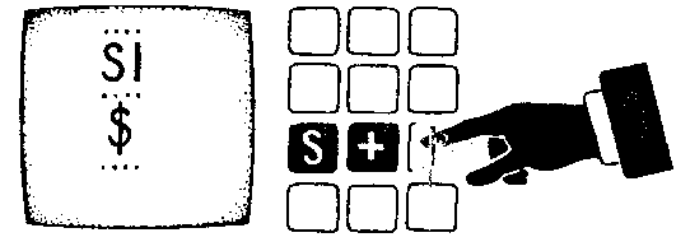
*Par exemple pour le programme de l'exercice 1 :*

Dans les instructions 1 à 4, les variables prennent leurs valeurs "initiales".

Puis, dans les instructions 7 et 8, certaines variables changent de valeur ; INFLATION reste constante pendant toute l'exécution alors que N varie de 1 à 20 pour servir de "compteur".

### Fiches de référence pouvant être consultées :

- \* +, -, \*, /, @, MOD, ^, page 172
- \* RND, page 176
- \* SIN, COS..., page 174
- \* PRINT, page 192



## 8

# Données et résultats

Jusqu'à maintenant, le programme une fois lancé, l'ordinateur ne vous demande plus jamais rien : vous ne pouvez plus intervenir sur les calculs et, il n'accepte pas de s'interrompre en cours de travail pour entamer le moindre échange bilatéral d'information. Mettons un terme à cette anomalie.

### ■ Un programme pour calculer l'inflation

Reprenons, par exemple, l'exercice de la leçon précédente sur l'évolution du prix de la baguette de pain.

Nous voulons faire un programme plus général.

Supposons que l'utilisateur de notre programme veuille connaître le prix d'un certain objet dans quelques années, connaissant le prix de cet objet aujourd'hui et le taux d'inflation à prévoir. L'idéal serait que l'exécution de notre programme se passe comme ceci :

- RUN **ENTREE** (c'est parti !)
- La machine demande que l'on frappe au clavier le taux d'inflation prévu.

- La machine demande que l'on frappe le prix de l'objet qui nous intéresse.
- La machine calcule et donne le prix de l'objet pour chacune des années suivantes.

Nous vous donnons ce programme, avant d'analyser le fonctionnement des nouvelles instructions qu'il utilise. Le voici :

```
10 CLS
20 INPUT "TAUX D'INFLATION PREVU (EN POURCENTAGE)"; I
30 INPUT "PRIX DE L'OBJET AUJOURD'HUI (EN FRANCS)"; PRIX
40 ANNEE=0
50 ANNEE=ANNEE+1
60 PRIX=PRIX*(1+I/100)
70 PRINT "DANS ";ANNEE;" ANS, L'OBJET COUTERA ";PRIX;"FRANCS"
80 GOTO 50
```

**Note :** sur le clavier du MO5 le caractère guillemet **■** est situé sur la même touche que le chiffre **2**. Remarquez que tout a été fait pour vous simplifier l'écriture : les guillemets ouvrant et fermant sont les mêmes !

(Voir aussi l'exercice 1 de la leçon précédente)

## ■ INPUT

INPUT signifie "entrée" en anglais.

C'est donc une instruction permettant d'entrer des données dans la machine. Précisément, l'exécution de l'instruction 20 provoque le comportement suivant :

- tout ce qui est écrit entre guillemets est intégralement réécrit sur l'écran (le point virgule après les guillemets est obligatoire).
- puis la machine affiche un point d'interrogation et s'arrête.
- la machine reprend la suite de l'exécution lorsque l'utilisateur frappe **ENTREE** ; normalement avant de frapper sur **ENTREE** l'utilisateur a dû frapper la suite des chiffres d'un nombre.

C'est ce "nombre frappé" qui est donné comme valeur à la variable indiquée après le point virgule.

L'exécution du programme précédent donne donc lieu au dialogue suivant (les frappes de l'utilisateur sont indiquées touche par touche).

**R U N** ENTREE

TAUX D'INFLATION PREVU (EN POURCENTAGE)?

**1 4** ENTREE

PRIX DE L'OBJET AUJOURD'HUI (EN FRANCS)?

**8 3** ENTREE

DANS 1 ANS, L'OBJET COUTERA 94.62  
FRANCS


DANS 2 ANS, L'OBJET COUTERA 107.867  
FRANCS

DANS 3 ANS, L'OBJET COUTERA 122.968  
FRANCS

DANS 4 ANS, L'OBJET COUTERA 140.184  
FRANCS

DANS 5 ANS, L'OBJET COUTERA 159.809  
FRANCS

**STOP**

 **Note :** Si le message "Redo" apparaît sur l'écran, c'est que l'utilisateur n'a pas frappé la suite de chiffres attendue par la machine.

## ■ Le dialogue à l'entrée

En fait, l'instruction INPUT est une instruction d'arrêt. La machine s'arrête et attend la valeur d'une (ou plusieurs) variable(s) dont la liste des noms est donnée dans la suite de l'instruction.

Les indications entre guillemets n'ont rien de nécessaire mais elles aident l'utilisateur du programme à se souvenir de la signification des nombres introduits.

Ainsi, dans le programme précédent, on pourrait remplacer les instructions 20 et 30 par la seule instruction :

```
15 INPUT I, PRIX
```

L'exécution ressemblerait alors à :

**R U N** ENTREE

?

**1 4 . 8 3** ENTREE

DANS 1 ANS, L'OBJET COUTERA 94.62  
FRANCS

La simplification de l'écriture oblige à ne pas oublier le nombre et l'ordre d'entrée des données ; ici le taux d'inflation et le prix (qui doivent être séparés par une virgule).

## ■ Le dialogue à la sortie

De la même manière l'affichage de commentaires avec les résultats permet de soulager sa mémoire.

Il est plus agréable de lire :

DANS 3 ANS L'OBJET COUTERA 122.968

FRANCS

plutôt que de devoir traduire un laconique

122.968

Inversement, cela oblige à réfléchir un peu dans l'écriture des instructions PRINT : la manipulation des ■ et des ■ n'est, en effet, pas aussi triviale qu'on le voudrait. Témoin l'instruction suivante qui donne aussi une bonne information :

```
60 PRINT "ANS="; ANNEE; "PRIX="; PRIX
```

## Exercices



1. En utilisant l'instruction IF... THEN... ELSE, modifiez le programme "inflation" de manière à éviter la faute d'orthographe : "1 ans" dans l'affichage des résultats.
2. Reprenez le programme de la 3<sup>e</sup> leçon, calculant le nombre d'heures vécues en faisant entrer, par une instruction INPUT, l'âge d'une personne.
3. Voici un programme de tracé d'un triangle dont les sommets sont définis par l'utilisateur :

```
10 INPUT "SOMMET NUMERO 1"; X1, Y1
20 INPUT "SOMMET NUMERO 2"; X2, Y2
30 INPUT "SOMMET NUMERO 3"; X3, Y3
40 CLS
50 LINE(X1, Y1)-(X2, Y2)
60 LINE -(X3, Y3)
70 LINE -(X1, Y1)
80 END
```

Observez l'effet des séparateurs ■ ■ et ■

La syntaxe du MO 5 est très précise !

4. Réalisez un programme coloriant en vert un rectangle dont l'utilisateur définit deux sommets opposés et traçant en rouge ses diagonales. Pour éviter les "bavures" de couleur, vous pouvez colorier le rectangle en couleur de fond, grâce à un numéro de couleur négatif (voir les fiches sur les instructions graphiques).

Exemple : `EOXF(X1, Y1)-(X2, Y2), -3`

5. Réalisez un programme dessinant un drapeau à 3 bandes dont les codes des 3 couleurs sont entrés par l'utilisateur à l'aide d'une instruction "INPUT".

## Attention

■ Lorsque le message :

?Redo

apparaît sur l'écran c'est que la machine attend la valeur d'une variable dont le nom figure dans une instruction INPUT et que la suite de touches que vous venez de frapper est incompréhensible pour elle.

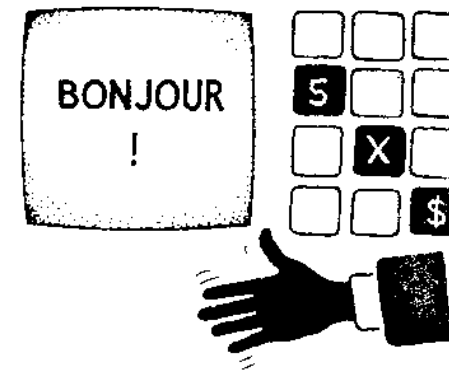
Il faut alors tout simplement frapper sur une bonne suite de touches ; par exemple une suite de chiffres suivie de **ENTREE**.

Si vous ne comprenez pas du tout ce qui se passe, vous pouvez toujours interrompre le programme en frappant :

**CNT-C**

## Fiches de référence pouvant être consultées

- \* INPUT, page 196
- \* PRINT, page 192
- \*\* PRINTUSING, page 194



## 9

# Lettres et messages

Vous avez sûrement déjà tenu avec certains ordinateurs de véritables “dialogues” et vous voudriez bien maintenant pouvoir traiter de véritables textes et messages. Voici le moment venu.

### ■ Un programme de bienvenue

Vous savez déjà que :

Dans une instruction PRINT ou INPUT, tout ce qui est écrit entre guillemets est affiché tel quel lors de l'exécution du programme.

Il est possible d'aller plus loin en donnant un nom à un message (ou à un morceau de message) tout comme on donne un nom à une valeur numérique.

La seule chose à savoir pour l'instant est que le nom d'un message doit se terminer par le caractère “\$”. (Eh oui ! tout se paye en dollars aujourd'hui !).

Voici par exemple un petit programme de bienvenue.

```
1 CLS
2 A$="BONJOUR, MES"
3 B$="DAMES !"
4 C$="SIEURS !"
5 PRINT A$;B$
6 PRINT A$;C$
7 END
```

En informatique, on parle plutôt de "chaîne de caractères" au lieu de "suite de caractères". Ainsi "DAMES !" est une chaîne de caractères de longueur 7. Et B\$ est une "variable-chaîne".

## ■ Entrez votre nom !

Dans le programme précédent, la machine est seule à "parler". La possibilité d'interroger l'utilisateur par une instruction INPUT ouvre la voie du dialogue.

On peut faire réagir la machine, par exemple par la répétition indéfinie de la réponse de l'utilisateur.

A défaut de signification, c'est au moins spectaculaire :

```
1 CLS
2 X$="BONJOUR JE M'APPELLE MOI ET VOUS ?"
3 PRINT X$
4 INPUT V$
5 PRINT "SUR CET ECRAN ";V$;" J'ECRIS TOUJOURS VOTRE NOM"
6 GOTO 5
7 END
```

**Note :** Dans l'instruction 4, la machine attend une suite de caractères ; lorsque l'utilisateur frappera sur **ENTREE**, elle donnera le nom V\$ à la suite de caractères qu'il aura préalablement frappés.

## ■ Vers un dialogue (truqué)

Pour progresser vers un véritable dialogue, il y a au moins deux moyens :

— afficher des questions et des réponses stéréotypées (voir le programme suivant jusqu'à l'instruction 50)

— varier la réaction de l'ordinateur en fonction de la réponse de l'utilisateur grâce aux instructions du type :

IF... THEN... ELSE...

(voir le programme suivant à la suite de l'instruction 50 et particulièrement l'instruction 80)

```
10 CLS
20 X$="BONJOUR JE M'APPELLE MOI ET VOUS ?"
30 PRINT X$
40 INPUT V$
50 PRINT "J'AI CONNU UN ";V$
60 PRINT "MAIS IL ETAIT MOINS SYMPA QUE VOUS !"
70 INPUT "VOULEZ-VOUS ALLER PLUS LOIN AVEC MOI ?";R$
80 IF R$="OUI" THEN A$="AH ! MON CHER " : B$="ALLONS FAIRE" ELSE A$="DOMMAGE !" : B$="AURIONS FAIT"
90 FOR I=0 TO 15
100 FOR J=0 TO 15
110 COLOR I,J
120 ATTRIB 1,1
130 PRINT A$;V$;
140 ATTRIB 0,0
150 PRINT " NOUS ";B$;" DE "
160 PRINT "GRANDES ET BELLES CHOSES ENSEMBLE" : PRINT
170 NEXT J
180 NEXT I
190 END
```

Nous aurions pu nous contenter des instructions 130, 150 et 160 mais, finalement, un spectacle coloré ne coûte pas cher sur le MO 5.

## ■ Les mots bout à bout

Il est possible de faire des "opérations" sur les mots comme il est possible de faire des "opérations" sur les nombres.

La plus simple des opérations est la mise bout à bout. Le signe de cette opération est le signe +.

En voici une utilisation simplette :

```
10 P$="BI"
20 Q$="POLY"
30 A$="CYCLE"
40 B$="NOME"
50 X$=Q$+A$
60 Y$=P$+A$
70 Z$=Q$+B$
80 T$=P$+B$
90 PRINT X$,Y$
100 PRINT Z$,T$
```

## ■ Découper les mots

Une "opération" plus délicate consiste à découper les mots en sous-mots.

On peut ainsi prendre les N derniers caractères d'un mot X\$ grâce à la fonction RIGHT\$(X\$, N) ; ou bien les N premiers grâce à la fonction LEFT\$(X\$, N) ; ce qui permet de jolis effets ; surtout à l'aide de la fonction LEN(X\$) qui donne la longueur du mot X\$.

Voici un exemple :

```
1 CLS
2 INPUT "QUEL EST VOTRE NOM " ; NOM$
3 LONGUEUR=LEN(NOM$)
4 FOR N=1 TO LONGUEUR
5 P$=LEFT$(NOM$,N)
6 PRINT P$
7 NEXT N
```

Voici un effet possible de ce programme :

R U N ENTREE

QUEL EST VOTRE NOM ?

UNTEL

U

UN

UNT

UNTE

UNTEL

Remarque : En remplaçant l'instruction 4 par l'instruction

4 FOR N=LONGUEUR TO 1 STEP -1  
on peut obtenir comme résultat :

```
ABRACADABRA
BRACADABRA
RACADABRA
ACADABRA
CADABRA
ADABRA
DABRA
ABRA
BRA
RA
A
```

Vous constatez qu'il est possible de choisir le "pas" de la "variable de boucle" en ajoutant STEP PAS à une instruction FOR. (voir la fiche de référence FOR... NEXT).

## Exercices



1. Faire un programme qui écrive successivement :

```
ANDRE
ANDR
AND
AN
A
```

2. Faire un programme qui écrive successivement :

HELENE  
ELENEH  
LENEHE  
ENEHEL  
NEHELE  
EHELEN

3. Faire un programme qui demande un mot à l'utilisateur et qui l'écrive

V  
E  
R  
T  
I  
C  
A  
L  
E  
M  
E  
N  
T

puis en

D  
I  
A  
G  
O  
N  
A  
L  
E

4. Écrire un programme qui met les mots au pluriel :

— si le mot se termine par S, X ou Z, l'afficher tel quel.

— si le mot se termine par AU, EU, EAU, lui ajouter un X.

Pour un programme plus performant, voyez le corrigé (après avoir consulté la fiche de référence DATA, READ) dans le chapitre conseils de programmation.

## Attention

■ Les guillemets ont une grande importance !

**17** est une valeur numérique, **89** aussi.

L'exécution de l'instruction :

PRINT 17+89

provoque l'impression de :

106

Par contre **"17"** et **"89"** sont des suites de caractères.

L'exécution de l'instruction :

PRINT "17+89"

provoque l'impression de :

17+89

et l'exécution de l'instruction :

PRINT "17"+"89"

provoque l'impression de :

1789

Tout ce qui figure entre guillemets n'est pas "traité" par la machine mais simplement mémorisé ou recopié.

### Fiches de référence pouvant être consultées :

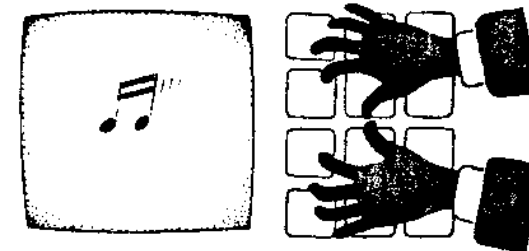
\* "...", +, page 177

\* MID\$, LEFT\$, RIGHT\$, page 178

\* LEN, INSTR, page 179

\*\* VAL, STR\$, page 180





# 10

## Musique

Il est temps maintenant de jouer au compositeur électronique.

### ■ La gamme

La commande

`PLAY"DOREMIFASOLASI" ENTRÉE`

a pour effet de faire jouer à la machine la suite de 7 notes constituant la gamme de DO majeur.

Remarquez que :

- chaque note est nommée par un mot de deux lettres (SOL est donc appelée seulement SO).
- Un ensemble de notes se présente exactement comme une chaîne de caractères. Ainsi le programme :

```
1 A$="DOFALA"  
2 PRINT A$  
3 PLAY A$  
4 END
```

a pour effet de faire imprimer le mot DOFALA puis de jouer les notes correspondantes. En effet :

La commande PRINT déclenche l'écriture d'une suite de caractères sur l'écran.

La commande PLAY déclenche l'émission d'une suite de sons par le haut parleur.

Évidemment, une commande comme

PLAY "MOZART"

déclenche l'affichage d'une erreur puisque ni MO, ni ZA, ni RT ne sont interprétés comme des notes jouables !

## ■ Les octaves

En fait, la machine émet les notes correspondant à une octave particulière.

Et, donc, si l'on commande :

PLAY "DOREMIFASOLASI DO"

Le premier et le dernier DO seront les mêmes.

Si l'on veut que le DO suivant le SI soit une octave plus haut que le DO initial, il faut commander un changement d'octave, par exemple comme ceci :

PLAY "DOREMIFASOLASI O5DO"

Vous disposez, sur le MO5, de 5 octaves numérotées de 1 à 5, le choix de l'une d'entre elles s'effectuant en insérant dans la suite des notes un "O" suivi d'un chiffre de 1 à 5 :

O1 fait choisir l'octave la plus grave.

O4 fait choisir la même octave que celle qui est implicitement choisie à l'allumage de la machine.

O5 fait choisir l'octave la plus aiguë.

En guise d'exemple, voici un programme de montée et de descente des gammes très impressionnant :

1 B\$="DOREMIFASOLASI" : C\$="SILASOFAMIREDO"

2 PLAY "O1"+B\$+"O2"+B\$+"O3"+B\$+"O4"+B\$+"O5"+B\$

3 PLAY C\$+"O4"+C\$+"O3"+C\$+"O2"+C\$+"O1"+C\$

4 END

En remplaçant B\$ et C\$ par une simple note au lieu d'une gamme entière on entend très bien le changement d'octave. Essayez par exemple de remplacer la ligne 1 par :

1 B\$="LA" : C\$="LA"

## ■ La durée des notes

Il est aussi possible de choisir la longueur pendant laquelle une note est jouée. Pour cela on insère dans la suite des notes un "L" suivi d'un nombre compris entre 1 et 96 :

L96 fait jouer des rondes :



L48 fait jouer des blanches :



L24 fait jouer des noires :



(C'est la longueur implicitement choisie lors de la mise en fonctionnement de la machine).

L12 fait jouer des croches :



L6 fait jouer des doubles croches :



L3 fait jouer des triples croches :



Mais toute longueur intermédiaire est jouable. Ainsi L18 fait jouer une "croche pointée" :

Voici par exemple la traduction des 4 premières mesures de "Au clair de la lune, mon ami Pierrot !".



1 PLAY "DOOOODORE"

2 PLAY "L48NIRE"

3 PLAY "L24DOMIRERE"

4 PLAY "L96DO"

5 PLAY "L24PPP" : GOTO 1



**Remarque :** A la ligne 5, avant de rejouer le même air, nous avons commandé 4 "pauses" indiquées chacune par la lettre P.

L24P ne dure que le temps d'un "soupir" !

## ■ Le tempo

Indépendamment des durées relatives de chaque note, on peut jouer un même air plus ou moins vite.

La vitesse d'exécution peut être choisie en insérant dans la suite des notes un "T" suivi d'un nombre compris entre 1 et 255. Bien que cela soit un peu approximatif, on peut risquer une correspondance que vous voudrez bien corriger selon votre humeur :

T1 fait jouer Prestissimo

T2 fait jouer Presto

T4 fait jouer Allegro

T5 fait jouer Allegretto

(C'est le tempo implicitement choisi à l'allumage de la machine).

T8 fait jouer Moderato

T12 fait jouer Andantino

T16 fait jouer Andante

T32 fait jouer Adagio

T48 fait jouer Lento

T64 fait jouer Largo

T256 fait jouer très très très très lentement.

Pour vous rendre compte de l'effet du tempo, jouez par exemple des séries de gammes sur des tempos différents.

1 B\$="DOREMIFASOLAS1"

2 PLAY "T1"+B\$+"T2"+B\$+"T3"+B\$+"T4"+B\$+"T5"+B\$

Ou bien séparez l'émission d'une note par des pauses :

1 PLAY "T1LAPT2LAPT4LAPT8LAPT16LAPT32LAPT64LAPT128LAPT255LA"

## ■ Complément

• Les dièses et les bémols s'écrivent naturellement après le nom des notes. Exemples :

PLAY"DO#FASO#"

PLAY"DObMIbSOb"

• Pour choisir une "attaque" particulière, on insère dans la suite des notes un "A" suivi d'un nombre de 0 à 255.

A0 fait jouer un son continu.

A200 fait jouer une note très rapidement amortie de sorte qu'un silence précède l'émission de la note suivante. Le mieux est de faire vos propres expériences !

• On peut insérer des espaces entre les notes si l'on veut rendre la chaîne plus lisible.

## Exercices

1. Traduire l'air suivant sur votre MO5 :



2. Quelle tragédie évoque pour vous le programme suivant ?

1 PLAY "L96SISO"

2 GOTO 1

3. Parmi ces deux programmes, l'un d'eux évoque le ronflement d'un moteur, l'autre des tirs de fusée.

Reconnaissez-les en faisant jouer :

1 PLAY "T1L2A100100"

2 GOTO 1

10 A\$="DOOQ#ERE#MIFAF#SOSO#LALA#SI"

20 PLAY "T1L4A404"+A\$+"05"+A\$+"PP"

30 GOTO 20

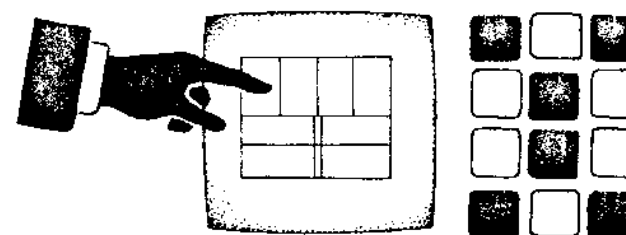
4. Faites un programme jouant des notes au hasard mais affichant simultanément leur nom écrit en noir sur un fond de couleur différente selon la note (DO sur rouge, RE sur vert, MI sur jaune, etc.)

## Attention

■ Dans un programme jouant de la musique, l'appui sur **CNT-C** ne semble pas toujours arrêter l'exécution ; c'est que la machine ne "comprend" l'appui sur **CNT-C** qu'entre deux instructions. Si vous appuyez seulement pendant l'exécution de notes, rien ne se passe ; il faut donc garder les deux touches **CNT** et **C** appuyées au moins jusqu'à la fin de l'émission d'une suite de notes commandée par un ordre **PLAY**.

Fiche de référence pouvant être consultée :

\* **PLAY**, page 207



11

## Indices et tableaux

### ■ Statistiques

Faire des statistiques c'est traiter des tableaux de nombres pour en extraire des renseignements.

Au niveau familial, par exemple, on peut vouloir faire des statistiques permettant de mieux contrôler les dépenses. Imaginons un possesseur de MO5, M. Dupont qui note la nature de chacune de ses dépenses ; ainsi sur son carnet de chèques note-t-il :

le chiffre 1 pour les dépenses d'alimentation.

le chiffre 2 pour les dépenses d'habillement.

le chiffre 3 pour les dépenses de transports.

le chiffre 4 pour les dépenses courantes diverses.

le chiffre 5 pour les dépenses d'investissements (meubles, appareils...).

le chiffre 6 pour les dépenses de logement (loyer, électricité...).

le chiffre 7 pour les impôts et les assurances.

Voici le "tableau" représentant la répartition de ces dépenses au mois de janvier :

| 1      | 2      | 3     | 4     | 5     | 6      | 7      |
|--------|--------|-------|-------|-------|--------|--------|
| 2130 F | 1100 F | 320 F | 710 F | 800 F | 2650 F | 1800 F |

## ■ Tableaux et indices

Pour effectuer des calculs avec ses dépenses, M. Dupont est obligé de leur donner un nom.

Pour cela il est pratique d'utiliser une "variable" avec un "indice".

Ainsi :

DEPENSE (1) représente les dépenses d'alimentation de M. Dupont (ici DEPENSE (1) vaut 2130 F)

DEPENSE (2) représente les dépenses d'habillement de M. Dupont (ici DEPENSE (2) vaut 1100 F)

DEPENSE (I) représente les dépenses relatives au "poste" n° I.

I est un *indice* ; DEPENSE est un nom de *tableau* et DEPENSE (I) vaut une certaine valeur numérique.

## ■ Programme de M. Dupont

• Pour entrer ses différentes dépenses de janvier dans le MO5, M. Dupont écrit alors le programme suivant :

```
10 FOR I=1 TO 7
11 INPUT DEPENSE(I)
12 NEXT I
```

Et chaque fois que MO5 affiche un point d'interrogation, M. Dupont frappe successivement 2130, 1100, 320, 710, 800, 2650 et 1800.

• Pour connaître le total de ses dépenses, M. Dupont complète son programme par les instructions suivantes :

```
20 SOMME=0
21 FOR I=1 TO 7
22 SOMME=SOMME+DEPENSE(I)
23 NEXT I
24 PRINT "Ce mois-ci j'ai dépensé" (SOMME)
  "Francs"
```

Et, à l'exécution, la machine affichera :

Ce mois-ci, j'ai dépensé 9510 Francs

• Pour comparer sa manière de vivre avec celle de ses amis, M. Dupont voudrait connaître les pourcentages relatifs à chacun de ces postes de dépenses. Les instructions suivantes effectuent ce travail :

```
30 FOR I=1 TO 7
31 PRINT "Poste numéro" (I)
32 P=INT(100*DEPENSE(I)/SOMME+0.5)
33 PRINT " : (P) %"
34 NEXT I
```

Et, à l'exécution, la machine affichera :

```
Poste numéro 1 : 22 %
Poste numéro 2 : 12 %
Poste numéro 3 : 3 %
Poste numéro 4 : 7 %
Poste numéro 5 : 8 %
Poste numéro 6 : 28 %
Poste numéro 7 : 19 %
```

*Note :* Pour effectuer ce dernier travail, l'ensemble des instructions de 10 à 34 doit être présent dans la machine.

## ■ Remarques

• Les instructions 20 à 24 valent la peine d'être bien comprises ; elles montrent comment *calculer la somme des éléments d'un tableau* dans le cas général :

— Donner un nom à cette somme (ici SOMME) et lui affecter la valeur : 0

— Ajouter successivement à SOMME chacun des éléments du tableau grâce à une boucle commandée par un "indice" parcourant tous les numéros du tableau.

— A la fin de la boucle, SOMME vaut exactement la somme des éléments du tableau.

• Pour bien comprendre les affichages prévus dans les instructions 31 à 33, relisez la fiche de référence PRINT précisant le rôle des guillemets et des points virgules.

- Si M. Dupont veut affiner l'analyse de ses dépenses, il décidera peut-être de considérer plus de 10 postes. Il faut alors savoir ceci : Lorsque l'indice d'un tableau doit dépasser 10, il est nécessaire de l'indiquer au MO5 pour qu'il prévoit le rangement d'une longue suite de nombres.

Précisément l'instruction :

```
1 DIM D(20)
```

placée en tête de programme, vous permet de parler des variables D(0), D(1), D(2), D(3), ...D(19) et D(20), dont il pourra être question par la suite. Remarquez bien que le MO5 réserve alors 21 mémoires.

En l'absence de l'instruction 1, l'exécution de l'ordre INPUT D(11) entraînerait l'affichage d'un message d'erreur.

- L'instruction 32 permet classiquement d'arrondir *un résultat à l'entier le plus proche*.

Le résultat intéressant est ici  $R = 100 * \text{DEPENSE}(I) / \text{SOMME}$

L'instruction  $P = \text{INT}(R + 0,5)$  permet de se débarrasser des décimales.

- Avec un peu d'habitude, vous vous lancerez certainement dans une *représentation graphique* des dépenses de M. Dupont...

## ■ Tableaux à 2 dimensions

Si Monsieur Dupont tient ses statistiques tous les mois de l'année, il dispose en fin d'année d'un tableau à 2 dimensions représentant l'analyse de ces dépenses :

| poste<br>mois | 1     | 2     | 3   | 4     | 5     | 6     | 7     |
|---------------|-------|-------|-----|-------|-------|-------|-------|
| 1             | 2 130 | 1 100 | 320 | 710   | 800   | 2 650 | 1 800 |
| 2             | 1 840 | 720   | 340 | 820   | 1 560 | 2 500 | 1 320 |
| 3             | 2 050 | 810   | 310 | 1 010 | 0     | 2 710 | 1 320 |
| ...           | ...   | ...   | ... | ...   | ...   | ...   | ...   |
| 12            | 2 110 | 380   | 250 | 1 820 | 0     | 2 850 | 2 910 |

Pour donner un nom à chacune des dépenses, il est évidemment pratique d'utiliser une variable avec 2 indices. Ainsi :

DEPENSE (4, 3) représente les dépenses courantes diverses (indice 4) de M. Dupont pendant le mois de mars (3<sup>e</sup> mois).

DEPENSE (I, J) représente les dépenses relatives au poste numéro I pendant le mois numéro J.

- L'entrée des données oblige Monsieur Dupont à frapper 84 nombres ; c'est évidemment ennuyeux mais il n'y a pas d'autres moyens que de les frapper au moins une fois. (La saisie des données est le goulot d'étranglement classique de l'utilisation de l'informatique). Le programme n'est pourtant pas complexe :

```
1 DIM DEPENSE(7,12)
9 FOR J=1 TO 12
10 FOR I=1 TO 7
11 INPUT DEPENSE(I,J)
12 NEXT I
19 NEXT J
```

- Mais M. Dupont (ou vous-même) pourra s'en donner à cœur joie en effectuant des totaux par mois comme précédemment, mais aussi par poste sur une année grâce à des morceaux de programmes ressemblant à celui ci :

```
39 FOR I=1 TO 7
40 SOM=0
41 FOR J=1 TO 12
42 SOM=SOM+DEPENSE(I,J)
43 NEXT J
44 PRINT "EN UN AN, POUR LE POSTE";I;"J"
45 DEPENSE : "/SOM;"Frands"
49 NEXT I
```

Et vous pourrez aussi calculer tous les pourcentages qui vous intéressent, déterminer les moyennes de pourcentages, le mois le plus dépensier, le mois le plus "normal", les écarts de dépenses trop importants...

## Exercices



1. En 1975 il y avait en France :

- 4,9 millions de "ménages" de 1 personne.
- 5,5 millions de "ménages" de 2 personnes.
- 3,4 millions de "ménages" de 3 personnes.
- 2,7 millions de "ménages" de 4 personnes.
- 1,5 millions de "ménages" de 5 personnes.
- 1,3 millions de "ménages" de 6 personnes ou plus.

Écrire un programme permettant d'afficher successivement :

- le nombre total de personnes vivant en France.
- le nombre de ménages.
- les pourcentages de ménages correspondant à chacune des catégories.

2. Écrire un programme permettant de :

- rentrer un nombre : N
- rentrer une suite de N notes (c'est-à-dire, ici de N nombres compris entre 0 et 20)
- afficher la moyenne de ces notes.

3. Écrire un programme, utilisable à la caisse d'un magasin en permettant de :

- rentrer les prix unitaires de 10 produits numérotés de 1 à 10.
- rentrer les quantités achetées de chacun de ces dix produits.
- afficher le prix total à payer.

Aide : l'une des instructions du programme ressemblera sûrement à celle-ci :  $S = S + P(I) * Q(I)$

4. Que fait le programme suivant ?

```

1 FOR I=1 TO 6
2 N(I)=0
3 NEXT I
10 FOR J=1 TO 100
11 D=INT(6*RND+1)
12 N(D)=N(D)+1
13 NEXT J
20 FOR I=1 TO 6
21 PRINT "SUR 100 JETS DE DES, LE";I;"ES
T SORTI";N(I);"FOIS"
22 NEXT I
99 END
    
```

## Attention

■ Nous pouvons parler entre nous du "tableau DEPENSE" mais la machine ne "comprend", elle, que des noms de variables : DEPENSE(4) vaut quelque chose. Si I vaut 3, DEPENSE(I) vaut DEPENSE(3)... Mais une instruction telle que "PRINT DEPENSE" provoquera l'affichage de quelque chose qui n'aura rien à voir avec le tableau DEPENSE.

■ Si, au lieu de "SOMME", on essaie d'appeler la variable "TOTAL", une erreur sera affichée. En effet, il existe des *mots clefs* dans le langage BASIC : ce sont les mots interprétés comme faisant partie d'une instruction : ainsi PRINT, FOR, TO, NEXT, IF... Aucun nom de variable ne doit commencer par l'un de ces mots (Voir la liste de ces mots en annexe 4) et TOTAL commence par TO !.

Fiche de référence pouvant être consultée :

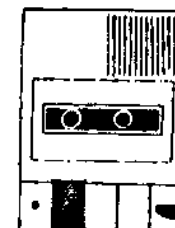
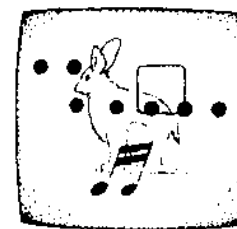
\*\*\* DIM, page 189

Attention ! Lorsque vous souhaitez enregistrer sur cassette un de vos programmes (voir p. 98) ou de vos fichiers (voir p. 109), n'utilisez que des cassettes neuves ou préalablement effacées ; pour effacer, il suffit de faire défiler toute la bande en mode d'enregistrement après avoir frappé MOTORON (voir p. 212) au clavier.

Une fois la bande effacée, frappez MOTOROFF (voir p. 212).

Remarque : il est cependant possible d'enregistrer un nouveau programme ou un nouveau fichier par-dessus un ancien sans l'effacer au préalable, à condition de se positionner bien avant le début de celui-ci.

Pour ce faire, il est recommandé de laisser de larges blancs entre vos différents programmes (MOTORON, MOTOROFF).



## 12

# Enregistrer et charger un programme

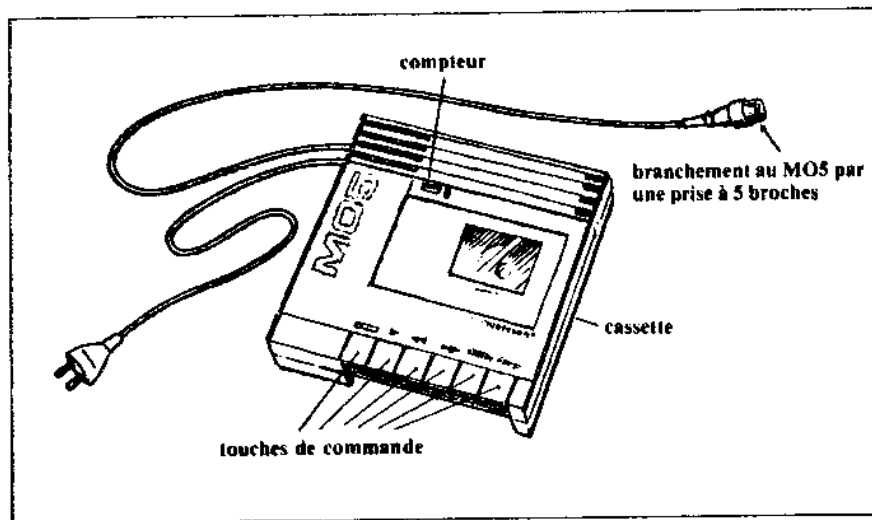
### ■ Le lecteur-enregistreur de programmes

Vous commencez maintenant à écrire des programmes un peu longs. Malheureusement, lorsque vous éteignez la machine, le programme enregistré est effacé et vous êtes obligé de le refrapper le lendemain si vous voulez encore l'utiliser.

Il est donc temps d'apprendre à transférer un programme depuis la mémoire du MO5 vers un support plus sûr (par exemple une bande magnétique) et sur lequel il se conservera.

L'appareil le moins cher permettant cette conservation est une sorte de magnétophone nommé *lecteur-enregistreur de programmes* spécifique au MO5 fonctionnant avec les mêmes cassettes magnétiques que celles sur lesquelles vous enregistrez de la musique. (N'utilisez cependant pas de cassettes de durée supérieure à C60 et choisissez les cassettes dites "qualité-ordinateur".)





Avant de vous lancer dans son utilisation, il y a de nombreux points à bien étudier :


- Les branchements
- Les touches de commande
- Le placement de la cassette.


Si vous n'avez pas l'habitude, suivez bien l'ordre et les conseils ci-dessous, cela vous évitera quelques crises de nerfs.


## ■ Les branchements





- Mettez le lecteur-enregistreur (en abrégé LEP) sous tension.
- Connectez le LEP au MO5 grâce au fil et à la prise DIN 5 broches prévue à cet effet.

## ■ Les touches

La touche , *stop/eject.*, permet d'ouvrir le couvercle afin de placer ou d'enlever la cassette. Elle sert aussi à interrompre l'effet de n'importe quelle touche.

La touche , *enreg.*, sera enfoncée lorsqu'on voudra enregistrer quelque chose sur la cassette (c'est une touche dangereuse puisqu'elle efface le contenu précédent de la cassette).

La touche  sera enfoncée lorsque l'on voudra lire quelque chose (préalablement enregistré) sur la cassette.


Les touches  et  permettent de faire défiler rapidement la bande de la cassette soit pour la rembobiner () , soit pour la faire avancer () dans le sens de la lecture.

## ■ Le placement de la cassette

**Attention :** Il y a deux côtés (et donc deux pistes d'enregistrement possibles) sur une cassette ! Commencer par repérer ces deux côtés en les marquant très lisiblement ; par exemple A d'un côté, B de l'autre. Choisissez le côté qui vous intéresse et...


... *placer la cassette* dans le LEP (la bande doit passer devant la tête de lecture du LEP, c'est-à-dire vers l'avant).



... *rembobiner la cassette* (touche ) complètement.

(une cassette est en début de lecture lorsque la partie droite est vide, c'est-à-dire prête à enrouler de la bande).

... *mettre le compteur à zéro*. Cela vous permettra de repérer les numéros où commencent et où finissent les divers enregistrements de la cassette.

... *faire un peu avancer la bande* (touche ) de 1 ou 2 numéros. En effet le début de la bande (dite "amorce") est absolument inutilisable pour l'enregistrement.

Votre bande, votre LEP, et vous-même êtes maintenant prêts pour l'enregistrement (ou la lecture) d'un programme.

## ■ Pour enregistrer ("sauver") un programme

- Assurez-vous d'abord de sa présence en mémoire ; par exemple, commander **LIST** **ENTREE**

- Choisissez-lui un nom ; par exemple : TRUC.

- Appuyez simultanément sur les touches **[C]** et **[D]** du LEP.

Commandez sur le clavier du MO5 :

**SAVE"TRUC"** **ENTREE**

Le LEP se met alors à faire avancer la bande de la cassette. Votre programme est en train de s'enregistrer sur la bande. Lorsque la bande s'arrête, le message OK apparaît sur l'écran.

Nous vous conseillons alors de :

- noter le numéro du compteur correspondant à la fin de l'enregistrement du programme "TRUC".

- rembobiner la cassette avant de l'éjecter du LEP.

## ■ Pour charger ("lire") un programme dans le MO5

- Placer dans le LEP la bande sur laquelle est enregistré le programme qui vous intéresse (n'oubliez pas de la rembobiner si elle ne l'est pas déjà).

- Appuyez sur la touche "lecture" ; **[D]** du LEP.

- Si le programme que vous voulez charger s'appelle "TRUC", commandez sur le clavier du MO5 :

**LOAD"TRUC"** **ENTREE**

Le LEP se met alors à faire avancer la bande de la cassette. Le MO5 affiche alors :

Searching

Si d'autres programmes sont enregistrés avant TRUC, ils ne sont pas lus mais simplement "parcourus" et le MO5 affiche leur nom, par exemple :

Skip: TOTO BAS

("skip" veut dire "sauté" et "BAS" est l'abréviation de "BASIC".)

Lorsque la partie où est enregistrée le programme "TRUC" vient devant la tête de lecture, le message suivant est affiché :

Found: TRUC BAS

Laissez faire : votre programme est en train de se charger dans la mémoire du MO5.

Lorsque la bande s'arrête, le message OK apparaît sur l'écran : vous pouvez vérifier que TRUC est maintenant présent en mémoire en commandant LIST ou RUN.

## ■ Quelques conseils pour gérer vos cassettes

Lorsque vous avez enregistré quelques-uns de vos programmes et recopié quelques programmes de vos nombreux amis, vous risquez de vous mélanger les bandelettes ! Un seul moyen de vous en sortir : soyez très strict sur les méthodes de classement.

Pour vous aider, voici réunis quelques conseils que, bien sûr, vous ne suivrez pas... avant d'avoir compris que vous ne pouvez pas faire autrement...

- Numérotez ou donnez un nom à vos cassettes et marquez leurs deux côtés.

Attention : Notez ce nom *sur* la cassette (avec un autocollant blanc) et non sur la boîte (les boîtes sont interchangeables)

- Tenez un cahier où vous notez pour chaque cassette (repérée par son nom), la suite des enregistrements et les numéros-compteurs correspondants.

- Rembobinez toujours une cassette avant de la sortir du LEP.

- Enregistrez vos programmes sur 2 cassettes : l'une dont vous vous servez, l'autre que vous stockez par précaution.

- Placez vos cassettes dans des boîtes.

- Régulièrement : récupérez les cassettes contenant des enregistrements (de travail) dont vous ne vous servez plus et enregistrez du "blanc".

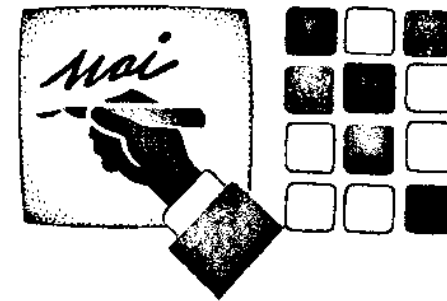
- Si vous tenez à un programme, enregistrez le deux fois sur une même cassette et une autre fois sur une cassette à laquelle vous touchez le moins possible.

- Si des erreurs se produisent, gardez votre calme et réfléchissez.

- Nettoyez régulièrement la tête de lecture de votre LEP avec un tampon imbibé d'alcool.

### Fiche de référence pouvant être consultée

\* SAVE, LOAD, RUN", MERGE, SKIPF,  
MOTORON MOTOROFF, page 210



13

## Le crayon optique

Jusque-là, vous ne pouviez rentrer des données dans votre MO5 que par le clavier (ou le LEP). Si vous disposez de l'extension "crayon optique", vous allez pouvoir utiliser l'écran comme périphérique d'entrée (et non plus seulement comme sortie d'image). Le crayon optique peut, en effet, détecter le passage d'un spot lumineux sur l'écran de télévision. Voici, par exemple, un premier programme utilisant cette possibilité.

### ■ L'oursin optique

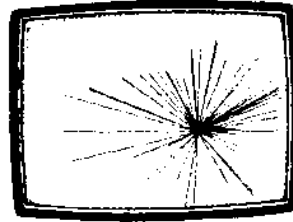
```
10 CLS
20 INPUT PEN X,Y
30 LINE (160,100)-(X,Y)
40 GOTO 20
```

Après **R U N** **ENTRÉE**, prenez votre crayon optique et appuyez le bien perpendiculairement quelque part sur l'écran.

Comme l'instruction 30 le demande, une ligne se trace alors, joignant le centre de l'écran au point où vous avez appuyé. En appuyant à dif-

férents endroits de l'écran vous obtenez alors un magnifique "oursin" étoilé du plus bel effet.

Comment cela fonctionne-t-il ?



## ■ L'instruction INPUT PEN X, Y

Comme l'instruction INPUT A, l'instruction INPUT PEN X, Y est une instruction d'arrêt : l'exécution est interrompue et la machine attend un appui sur l'extrémité du crayon.

Remarquez en effet que l'extrémité noire du crayon cède à la pression comme un interrupteur.

Lorsque cette pression est exercée, le rayon lumineux qui passe devant l'extrémité du crayon est intercepté ; le MO5 est alors capable de calculer les coordonnées, sur l'écran, du point où ce rayon a été intercepté. Ces coordonnées sont affectées aux variables X et Y indiquées dans l'instruction INPUT PEN et l'exécution continue à l'instruction suivante.

La manière dont le MO5 calcule les coordonnées du point d'appui du crayon est intéressante : vous savez peut-être que l'image télévision est fabriquée par un pinceau lumineux qui parcourt l'écran 25 fois par seconde. Ce pinceau parcourt donc chacun des 64 000 points de l'écran en 1/25 de secondes.

Au temps  $t = t_0$ , le pinceau lumineux est en haut à gauche de l'écran.



Au temps  $t$ , le pinceau lumineux est quelque part sur l'écran.

Au temps  $t = t_0 + 1/25$ , le pinceau lumineux est en bas à droite de l'écran.

Il suffit donc, pour repérer la position du crayon, de repérer l'instant où le pinceau lumineux part du haut de l'écran et l'instant où il passe devant le crayon : (une horloge du MO5, synchronisée avec le balayage, suffit à ce repérage) ; un simple règle de trois et quelques calculs permettent alors de retrouver la ligne et la colonne où le crayon a capté le pinceau lumineux.

## ■ Quelques sources d'erreur

- Le pinceau lumineux ne sera pas reçu par le crayon lorsque la luminosité de l'image n'est pas suffisante, mais aussi lorsque l'endroit où vous pointez est rouge ou noir (ces couleurs sont trop "opaques"). Les variables X et Y prennent alors la valeur -1 et une erreur se produit lorsque vous essayez de tracer une figure utilisant ces coordonnées, puisqu'elles sont négatives.

Pour pallier cette difficulté il vaut mieux s'assurer une bonne lecture en bouclant sur l'instruction INPUT PEN tant que X et Y n'ont pas de valeur positive. Par exemple comme ceci :

```
20 INPUTPEN X,Y : IF X<0 THEN 20
```

- Il n'est pas recommandé de faire suivre (sans le vouloir explicitement) deux instructions INPUT PEN sans s'assurer qu'un temps suffisant s'est écoulé entre leurs exécutions respectives. Un bon moyen consiste à faire suivre chaque lecture du crayon par l'émission d'une note de musique (ou d'un petit air plus long). L'utilisateur a alors l'impression que la lecture est accompagnée d'un bruit et on est assuré qu'elle a bien eu lieu...

Voici un petit programme nécessitant une telle précaution pour dessiner 15 rectangles de toutes les couleurs.

```
10 CLS:SCREEN 0,7,7
20 FOR COUL=1 TO 15
30 INPUT PEN A,B:IF A<0 THEN 30:IF B<0 THEN 30
31 PLAY "D0"
40 INPUT PEN C,D:IF C<0 THEN 40:IF D<0 THEN 40
41 PLAY "M1"
50 BOXF(A,B)-(C,D),COUL
60 NEXT COUL
99 END
```

Essayer le même programme après avoir supprimé les instructions 31 et 41 ! Ce qui ne marche pas alors est dû au fait que la machine effectue *deux* lectures du crayon optique alors que l'utilisateur appuie sur l'un des sommets du rectangle désiré... et le rectangle défini par ces "deux" lectures est ridiculement petit.

## ■ Un crayon ! deux organes !

En fait le crayon optique est un double organe des sens pour le MO5. Nous avons vu en effet qu'il était capable de "sentir" deux phénomènes physiques : l'appui sur son extrémité (c'est donc un organe de toucher) et le passage d'un rayon lumineux (c'est donc un organe de la vue).

Pour plus de détails voir la fiche de référence.

Et, pour votre plaisir, voici un programme qui fait de votre crayon optique un véritable stylo sur écran, puisque le MO5 trace des traits qui suivent exactement le mouvement de votre crayon.

**Un programme pour dessiner sur l'écran :**

```
5 PRINT "ANNONCEZ LA COULEUR"
10 INPUT C:CLS:SCREEN C,7,7
20 INPUTPEN X,Y:IF X<0 THEN 20:IF Y<0 THEN 20
30 PSET(X,Y)
40 INPUTPEN X,Y:IF X<0 THEN 40:IF Y<0 THEN 40
50 LINE -(X,Y)
60 GOTO 40
```

(Avec l'instruction 40 INPEN..., vous pouvez même dessiner en "pointant" votre crayon jusqu'à environ 20 cm de l'écran).



## Exercices

1. Que fait le programme-gag suivant ?

```
10 CLS
20 INPUTPEN X,Y
30 LOCATE X/8,Y/8
40 PRINT "Aie"
50 GOTO 20
```

Il est surtout "gag" lorsque vous appuyez le crayon sur la partie droite de l'écran, vers la 39<sup>e</sup> position !

2. Faites un programme qui écrive "BONJOUR" à partir de l'endroit où l'utilisateur pointe le crayon. Arrangez-vous pour qu'aucune erreur de lecture ne soit possible.

3. Faites un programme dessinant des carrés dont l'utilisateur pointe successivement le centre puis un sommet.

4. Faites un programme qui choisit une ligne au hasard entre 0 et 39 puis qui répond "plus bas SVP" ou "plus haut SVP" chaque fois que l'utilisateur pointe son crayon à une certaine hauteur sur l'écran.

## Attention

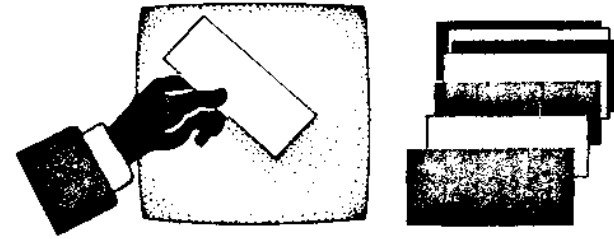
■ Tenez bien compte du paragraphe "Quelques sources d'erreur".

Réglez la luminosité de votre téléviseur à un niveau suffisant.

Tenez bien votre crayon perpendiculairement à l'écran l'inter-rupteur étant au dessus du petit trou "capteur" du rayon.

## Fiche pouvant être consultée :

\*\* TUNE,INPUT PEN,INPEN,PTRIG, page 215

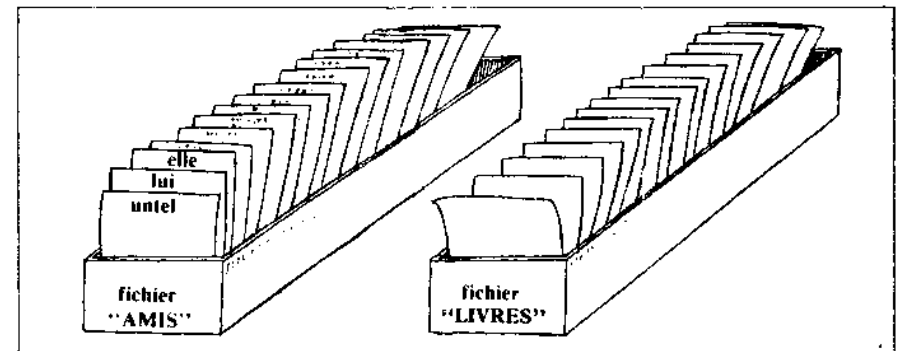


14

## Les fichiers

### ■ Un ensemble de fiches

Un “fichier” de données est une suite de données rangées les unes derrière les autres comme des “fiches” dans une boîte.



Chacune des “fiches” contient un certain nombre d’informations relatives à l’“article” correspondant.

Il peut s’agir, par exemple, du fichier de vos amis, chaque fiche contenant son adresse, son numéro de téléphone et sa date de naissance ; ou bien du fichier des clients d’une entreprise, chaque fiche contenant ses caractéristiques, l’état de ses commandes et de son compte ; ou bien du fichier des livres d’une bibliothèque... C’est en tout cas un

ensemble de fiches destiné à être ou bien feuilleté ou bien modifié (addition ou soustraction d'une fiche). Pour consulter un fichier ou pour écrire dedans, il faut successivement :

- Trouver la bonne boîte (grâce à son "nom", par exemple) et "l'ouvrir".
- Feuilletter les fiches jusqu'à ce que l'on trouve celle qui nous intéresse.
- Prendre la fiche et l'amener devant celui qui doit la traiter (par le "canal" approprié)
- Traiter les informations contenues dans la fiche ou la modifier ; puis la remettre dans le fichier.
- Ne pas oublier de "fermer" la boîte (sinon les couvercles se mélangeraient.)

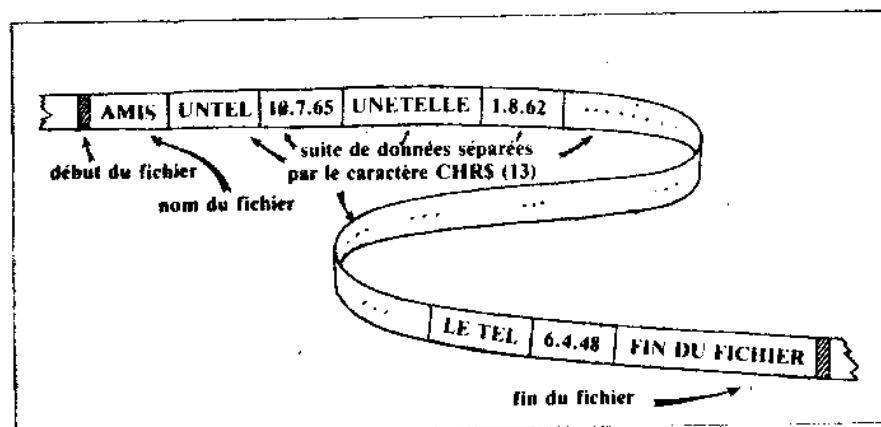
## ■ Fichiers sur cassettes

Les fichiers traités par un ordinateur se constituent et se consultent un peu comme les ensembles de fiches dans des boîtes ; leur "support" physique est, en général, une bande ou une disquette magnétique.

Grâce au LEP, il est ainsi possible de stocker sur une cassette, à partir du MO5, des "fichiers" ou des listes de données destinées à être relues plus tard les unes après les autres.

Pratiquement un fichier sur cassette est constitué :

- d'un premier enregistrement contenant le nom du fichier
- d'une liste de données alphabétiques ou numériques
- d'un code signalant la fin des enregistrements de ce fichier.



Les fichiers ici décrits sont dits *séquentiels*.

Cela signifie qu'il n'est pas possible d'atteindre une donnée quelque part au milieu du fichier sans avoir fait défiler la suite de toutes les données précédentes. En effet, aucune des données n'a de "nom" permettant de l'"appeler" ; pour consulter un tel fichier il faut avoir bien noté la structure des enregistrements au moment de sa constitution afin de "reconnaître" chacune des données au fur et à mesure de leur défilement. Dans l'exemple du schéma précédent, chaque article du fichier AMIS est constitué de : un nom, une suite de chiffres représentant la date de naissance.

## ■ Pour enregistrer un fichier sur une cassette, il faut :

1. S'assurer que le lecteur-enregistreur est en position d'enregistrement avec la cassette positionnée à l'endroit où vous désirez que s'effectue l'enregistrement.
2. Ouvrir un canal, en sortie, pour permettre le passage des données du MO5 vers le lecteur-enregistreur.

L'instruction

`OPEN "O", #1, "AMIS"`

ouvre le fichier de nom "AMIS" en Output (c'est-à-dire que l'on va écrire dessus), le canal réservé à sa manipulation par le MO5 portant le numéro 1.

3. Commander le passage des données sur la cassette.

L'instruction :

`PRINT #1, A$, T$`

commande l'envoi des variables A\$ et T\$ par le canal 1 (lequel a été affecté à un certain nom de fichier par l'instruction OPEN)

4. Ne pas oublier de fermer le canal après son utilisation.

L'instruction `CLOSE #1` commande cette fermeture.

Voici un exemple d'enregistrement de données dans un fichier sur cassette. On y voit qu'une suite d'instructions "DATA", dans un programme, n'est pas autre chose qu'un fichier séquentiel. Ce fichier est consulté grâce à l'instruction "READ" (Le "pointeur-des-DATA" joue le même rôle que la tête de lecture du LEP : à un instant donné, il est prêt à lire la donnée suivante dès que l'ordre lui en

sera communiqué). Notez que "DATA" signifie "données".

```
10 DATA UNTEL,345 26 33,LUI,235 26 47,MO
1,010 37 48
11 DATA ELLE,720 14 10,PERSONNE,456 78 9
0
19 DATA Z,0
100 '
101 ' *** ENREGISTREMENT ***
110 OPEN"O",#1,"TELEPHON"
120 READ A$,T$
130 IF A$="Z" THEN 190
140 PRINT #1,A$,T$
150 GOTO 120
190 CLOSE #1
199 STOP
```

*Note* : les données de l'instruction 19 permettent de repérer la fin de la liste des données.

Évidemment, avant d'exécuter ce programme, il faut que le LEP soit en position d'enregistrement avec une bande magnétique dans le lecteur. Vous devez avoir bien repéré le début de l'enregistrement qui va avoir lieu.

## ■ Pour lire un fichier, d'une cassette vers la mémoire du MO5, il faut :

— s'assurer que le lecteur-enregistreur est en position de lecture avec la cassette positionnée avant l'endroit où sont enregistrés les fichiers qui seront lus.

— ouvrir un canal, en entrée, pour permettre le passage des données du lecteur-enregistreur vers le MO5.

L'instruction

OPEN" I",#1,"AMIS"

ouvre le fichier "AMIS" en INPUT (c'est-à-dire que l'on va lire dessus), le canal réservé à sa manipulation portant le numéro 1.

— commander la lecture des données enregistrées sur la cassette :

INPUT#1,A\$,B\$

— ne pas oublier de fermer le canal après son utilisation :

CLOSE#1

```
1 CLEAR 600
2 DIM A$(30),T$(30)
200 '
201 ' ***** LECTURE *****
210 OPEN" I",#1,"TELEPHON"
215 I=1
220 IF EOF(1) THEN 290
230 INPUT #1,A$(I),T$(I)
240 I=I+1 : GOTO 220
290 CLOSE #1
299 STOP
```

Voici, maintenant un exemple de lecture de fichier.

Au fur et à mesure de leur lecture, les données se voient nommées et rangées comme les éléments de deux tableaux, ce qui permettra ensuite de les rappeler (l'instruction 1 réserve 30 places en mémoire pour chacun de ces tableaux).

La variable EOF (1) prend la valeur VRAI lorsque la dernière donnée du fichier traitée par le canal 1 vient d'être lue (EOF est l'abréviation de *End Of File* et signifie "fin de fichier").

*Note* : Évidemment, avant l'exécution de ce programme, la bande doit être positionnée de manière à ce que le fichier "TELEPHON" soit après la tête de lecture du LEP.

Voici enfin un programme permettant de sortir sur imprimante le fichier lu par le programme précédent.

```
300 ' ***** IMPRESSION *****
310 OPEN"O",#2,"LPRT:"
320 FOR K=1 TO I
330 PRINT #2,"NON : ";A$(K),"TELEPHONE :
";T$(K)
340 NEXT K
350 CLOSE #2
399 END
```

### Fiches de référence pouvant être consultées :

- \* OPEN, CLOSE, EOF, PRINT #, INPUT #, LINE INPUT #, page 219
- \* "LPRT :", PRINT #, page 217