

**La découverte  
de l'ALICE  
et du TRS-MC/10**

La collection « **MATERIELS** » s'intéresse à l'utilisation de tel ou tel type d'ordinateur, depuis la première prise de contact jusqu'à l'utilisation la plus pointue. Elle est aujourd'hui constituée de 41 titres.

- Exercices pour Apple II — Frédéric Lévy
- La découverte de l'Apple II — Dominique Schraen et Frédéric Lévy
- La pratique de l'Apple II — volumes I et II — Nicole Bréaud-Pouliquen
- La pratique de l'Apple II — volume III — Nicole Bréaud-Pouliquen et Daniel-Jean David
- La découverte de l'Atari — Daniel-Jean David
- La découverte du CBM — Daniel-Jean David
- La pratique du CBM — volumes I et II — Daniel-Jean David
- La découverte du Commodore 64 — Daniel-Jean David
- La pratique du Commodore 64 — Daniel-Jean David
- La découverte du Dragon — Frédéric Lévy et Dominique Schraen
- Dragon tout feu tout flammes — Trévor Toms et John Phipps — traduit par Olivier Arnaud
- La découverte du FX-702P — Jean-Pierre Richard
- La découverte du Goupil — Jean-Yves Michel
- Programmer HP-41 — Philippe Deschamps et Jean-Jacques Dhénin
- La pratique du MZ-80 A et K — Jean-Pierre Lhoir
- La découverte du PB-100 — Pierrick Moigneau
- La découverte du PC-1251 — Jean-Pierre Richard
- La découverte du PC-1500 — Jean-Pierre Richard
- Le petit livre du Spectrum — Trévor Toms — traduit par Allan Keil
- La pratique du Spectrum — tome 1 — Xavier Linant de Bellefonds
- La pratique du Spectrum — tome 2 — Marcel Henrot
- Exercices pour Spectrum — Julien Lévy
- La pratique du TRS-80 — tomes 1, 2 et 3 — Pierre Giraud et Alain Pinaud
- Exercices pour TRS-80 — Frédéric Lévy
- Les graphiques sur TRS-80 — Donn Inman — traduit par Alain Pinaud
- La découverte de la TI-57 — Xavier de la Tullaye
- La découverte du TI-99/4A — Frédéric Lévy et Dominique Schraen
- Exercices pour TI-99/4A — Frédéric Lévy
- La découverte du VIC — Daniel-Jean David
- La pratique du VIC — tome 1 — Daniel-Jean David
- Le petit livre du ZX-81 — Trévor Toms — traduit par Ghislaine Lapeyre
- La pratique du ZX-81 — tome 1 — Xavier Linant de Bellefonds
- La pratique du ZX-81 — tome 2 — Marcel Henrot
- La découverte de l'Oric — Daniel-Jean David
- La découverte du T07 — Dominique Schraen et Maurice Charbit
- Exercices pour Goupil — Yves Martin
- La découverte de l'Alice et MC/10 — Dominique Schraen

Autre ouvrage relatif à l'Alice et MC/10 :

- 102 programmes pour Alice et MC/10 — Jacques Deconchat

#### RAPPELS

##### Les séries :

En fait, il faudrait parler de niveaux, puisque la couleur attachée à chaque ouvrage permet de situer la « force » de celui-ci selon le code suivant :

*Série VERTE* : ouvrage d'initiation ne nécessitant que des connaissances de base.

*Série BLEUE* : suppose une connaissance élémentaire du sujet traité.

*Série ROUGE* : ouvrage d'approfondissement, niveau de complexité moyen.

*Série NOIRE* : ouvrage d'approfondissement, niveau de complexité élevé.

##### Les collections :

Les ouvrages d'Édition du PSI, actuellement une centaine, sont répartis en collections :

« **LANGAGES** », « **MATERIELS** », « **PROGRAMMES** », « **GUIDES PRATIQUES** », « **MENTOS** », « **UTILISATIONS DE L'ORDINATEUR** », « **LOGI GUIDE** » et pour l'initiation, outre quelques livres hors collection, « **... POUR TOUS** ».

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

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

# **La découverte de l'ALICE et du TRS-MC/10**

**par  
Maurice Charbit**



Editions du P.S.I.  
1984

# PRESENTATION

*Cet ouvrage s'adresse à tous ceux qui souhaitent s'initier à la programmation d'ALICE ou du MC/10.*

*Ce micro-ordinateur possède un langage BASIC qui offre de multiples possibilités d'affichage graphique couleur.*

*L'apprentissage du BASIC est présenté à la façon d'un cours dans lequel les notions à acquérir et les difficultés sont rencontrées de manière progressive. De nombreux exemples et programmes facilitent la compréhension des notions nouvelles. Des exercices corrigés, en fin de chaque chapitre, aident à l'assimilation et à la pratique des méthodes de programmation.*

*La lecture de ce livre ne nécessite aucune connaissance préalable en informatique.*

# SOMMAIRE

	Pages
PRESENTATION	5
CHAPITRES	
1 - L'INSTRUCTION PRINT	9
. Le clavier	10
. L'instruction PRINT	12
. Effacement d'un caractère ou d'une ligne	15
. Utilisation de la touche CONTROL	16
. Ecriture abrégée de PRINT	16
2 - LES VARIABLES NUMERIQUES ET ALPHANUMERIQUES	17
. Les nombres	18
. Les opérations arithmétiques	18
. Les variables numériques	20
. Tableaux de nombres	22
. Tableaux à plusieurs dimensions	24
. Chaînes de caractères et les variables alphanumériques	25
. L'instruction CLEAR	26
. Comparaisons	27
. Opérateurs logiques : AND, OR, NOT	30
3 - LES PREMIERS PROGRAMMES	33
. Les ordres LIST et RUN	34
. L'instruction END	36
. Effacement d'une ligne de programme	36
. Les instructions CSAVE, CLOAD, SKIPF	37
. L'instruction INPUT, l'ordre NEW	38
. Les instructions READ, DATA et RESTORE	42
. Les instructions CSAVE*, CLOAD*	45
. Les instructions CLS, PRINT TAB et PRINT@	47
. L'instruction REM	49
4 - TROIS INSTRUCTIONS FONDAMENTALES	51
. L'instruction GOTO	52
. L'instruction ON GOTO	55
. L'instruction IF THEN	56
. L'instruction FOR...TO...STEP...NEXT	59

## LA DECOUVERTE DE L'ALICE ET DU MC/10

5 - LES SOUS-PROGRAMMES	63
. Les instructions GOSUB et RETURN	66
. L'instruction ON..GOSUB	67
6 - INSTRUCTION POUR LA MISE AU POINT DES PROGRAMMES	71
. Plusieurs instructions sur une même ligne	72
. Les instructions STOP et CONT	73
. Interruption par clavier	74
. La mémoire d'Alice	75
7 - LES FONCTIONS NUMERIQUES	79
. RND : nombre au hasard	80
. ABS : valeur absolue	80
. SGN : signe	81
. INT : valeur entière	81
. SQR : racine carrée	82
. SIN, COS et TAN : SINus, COSinus et TANgente	82
. EXP et LOG : EXPonentielle et LOGarithme népérien	82
8 - LE TRAITEMENT DES CHAINES DE CARACTERES	85
. LEN : longueur d'une chaîne de caractères	86
. Les instructions LEFT\$, RIGHT\$ et MID\$	86
. La concaténation	88
. L'instruction STR\$	89
. L'instruction VAL	89
. Les instructions ASC et CHR\$	90
9 - LES INSTRUCTIONS D'INTERACTIVITE	95
. L'instruction CLS	96
. L'instruction INKEY\$	96
. Affichage de caractères graphiques	97
. L'instruction POINT	101
. L'instruction SOUND	102
ANNEXES	
1 - Signification des messages d'erreur	104
2 - Mots réservés	106
3 - Caractéristiques graphiques	
Codes des couleurs	107
4 - Codes ASCII	108
5 - Résultat des exercices	110
6 - Index	126

## **CHAPITRE 1**

### L'INSTRUCTION PRINT

- Le clavier
- L'instruction PRINT
- Effacement d'un caractère  
ou d'une ligne
- Control

## LA DECOUVERTE DE L'ALICE ET DU MC/10

### LE CLAVIER

Une fois le récepteur de télévision connecté, il suffit pour mettre en marche ALICE d'actionner l'interrupteur situé à droite de l'appareil.

Sur l'écran apparaît alors un message vous indiquant que ALICE est en marche, ainsi que le mot OK qui signifie que le Basic est à votre disposition. Au-dessous apparaît un petit carré noir clignotant appelé curseur qui indique l'endroit où s'affichera le prochain caractère tapé au clavier.

N'hésitez pas à jouer avec les touches du clavier : ALICE n'encourt aucun risque de détérioration.

En plus des lettres de l'alphabet, des chiffres et des caractères typographiques, le clavier d'ALICE comporte d'autres touches dont voici la signification et le fonctionnement :

**SHIFT** SHIFT signifie CHANGER, DECALER. Cette touche, analogue à celle du clavier de machine à écrire, permet d'afficher la seconde définition figurant sur certaines touches du clavier.

- Exemple :**
- tapez sur la touche **%** **5** : 5 s'affiche sur l'écran.
  - en maintenant la touche **SHIFT** enfoncée, tapez sur la même touche : % s'affiche sur l'écran.
  - si vous faites **SHIFT** **[** **8** , le crochet ouvert apparaît sur l'écran.
  - si vous faites **SHIFT** **X** **■** , un petit rectangle semblable à celui inscrit sur la touche apparaît sur l'écran : il s'agit d'un caractère graphique. Il y a 16 caractères graphiques sur ALICE. Ils seront étudiés en détail au chapitre 9.
  - si vous faites **SHIFT** **∅** , vous passez du mode normal (ou mode majuscule) au mode inverse (ou mode minuscule). Sur l'écran, les caractères s'affichent en clair sur fond noir (vidéo inverse) et sur votre imprimante, ils s'inscrivent en minuscule.



## LA DECOUVERTE DE L'ALICE ET DU MC/10

Pour revenir au mode normal il suffit de refaire SHIFT  $\emptyset$  .

ENTER

ENTER signifie ENTRER. Cette touche permet, entre autres, de positionner le curseur au début de la ligne suivante mais son rôle, beaucoup plus complexe, sera détaillé dans la suite de ce chapitre.

BREAK

BREAK signifie BRISER, INTERROMPRE. L'utilisation de cette touche est expliquée au chapitre 6.

CONTROL

CONTROL signifie COMMANDE. Cette touche permet, elle aussi, de dédoubler certaines touches du clavier.

- en maintenant la touche CONTROL enfoncée, tapez sur la touche  $\begin{matrix} J \\ 9 \end{matrix}$  : le mot PRINT

apparaît sur l'écran. Les mots que l'on peut obtenir rapidement de cette façon sont indiqués au-dessus des touches correspondantes. Ces mots s'appellent des instructions Basic et vous allez les découvrir tout au long des chapitres qui suivent.

Le symbole @ se lit ARROBAS en français, AT en anglais.

Ne confondez pas :

- O qui est la lettre O.
- $\emptyset$  qui est le chiffre zéro, qui s'affiche sur l'écran 0 mais que dans la suite de cet ouvrage nous écrirons toujours  $\emptyset$ .

Nous sommes maintenant prêts à explorer les possibilités d'ALICE et pour commencer, voyons s'il peut effectuer un calcul très simple :

tapez 7+5  
puis appuyez sur la touche ENTER .

ALICE ne répond pas.  
tapez PRINT 7+5  
puis tapez sur ENTER  
réponse 12  
OK

Cette première expérience montre qu'il n'est pas aussi facile de "communiquer" avec ALICE qu'avec une calculatrice. En effet, les touches du clavier telles que

## LA DECOUVERTE DE L'ALICE ET DU MC/10

+, -, etc., ne servent pas à faire exécuter l'opération indiquée mais uniquement à l'écrire : c'est en appuyant sur la touche ENTER que vous demandez l'exécution de ce qui a été écrit.

De plus, ALICE obéit à un ensemble de règles d'écriture très strictes en dehors desquelles il est incapable de "comprendre" ce que vous voulez lui faire faire.

### L'INSTRUCTION PRINT

PRINT signifie IMPRIMER.

Avec PRINT vous venez de rencontrer une première règle : *pour obtenir sur l'écran le résultat d'une opération, il est nécessaire de la faire précéder de PRINT.*

L'ensemble de toutes les instructions compréhensibles par un ordinateur constitue son langage. Pour ALICE, c'est le langage Basic. Dans les chapitres suivants, nous allons en étudier toutes les instructions.

De même que l'utilisation d'une langue exige le respect de règles de grammaire et d'orthographe, l'écriture en langage Basic suit des règles de syntaxe strictes.

Chaque fois que vous ferez une entorse à cette syntaxe, l'ordinateur indiquera qu'il est incapable de comprendre en affichant le message d'erreur : **? SN ERROR** qui signifie **ERREUR** de **SyNtaxe**.

#### *Exemples*

```
tapez          PRINY 7+5
puis tapez sur 
réponse       ?SN ERROR
              OK
```

Voyons en détail l'instruction PRINT :

- Affichage des nombres entiers.

```
tapez          PRINT 5
tapez sur     
réponse       5
              OK
tapez          PRINT 764521
tapez sur     
réponse       764521
              OK
```

- Affichage des nombres décimaux.

## LA DECOUVERTE DE L'ALICE ET DU MC/10

**Remarque** : dans la notation anglo-saxonne, le point remplace la virgule pour séparer la partie entière de la partie décimale.

```
tapez      PRINT 44.7
tapez sur  [ENTER]
réponse    44.7
           OK
tapez      PRINT 0.93
tapez sur  [ENTER]
réponse    .93
           OK
tapez      PRINT -0.1596
tapez sur  [ENTER]
réponse    -.1596
           OK
```

Pour les nombres décimaux compris entre -1 et +1 (partie entière nulle), l'ordinateur n'affiche que la partie décimale précédée du point, vous pouvez en faire autant :

```
tapez      PRINT .75
tapez sur  [ENTER]
réponse    .75
           OK
```

**Remarque** : désormais nous n'indiquerons plus systématiquement "tapez sur [ENTER]" pour obtenir l'exécution d'une instruction, ni l'affichage du message OK.

Essayons maintenant d'afficher des lettres et des mots :

```
tapez      PRINT MICRO
réponse    Ø
```

Nous expliquerons ce résultat surprenant mais voici d'abord la marche à suivre pour afficher un mot :

```
tapez      PRINT "MICRO"
réponse    MICRO
```

vous avez remarqué qu'*il faut écrire le mot entre guillemets*.

Que se passe-t-il avec une seule lettre ?

```
tapez      PRINT A
réponse    Ø
tapez      PRINT "A"
réponse    A
```

A est interprété comme une variable numérique ayant la valeur Ø et "A" est compris comme un caractère.

De même, MICRO était compris comme une variable numérique ayant la valeur Ø et "MICRO" comme une suite de caractères appelée **chaîne de caractères**.

## LA DECOUVERTE DE L'ALICE ET DU MC/10

Les variables et les chaînes de caractères sont définies et étudiées au chapitre suivant.

Avec une seule instruction PRINT, il est possible d'afficher plusieurs nombres et plusieurs chaînes de caractères :

```
tapez      PRINT "MICRO";"/";"ORDINATEUR"
réponse    MICRO/ORDINATEUR
```

*Le point virgule associé à PRINT juxtapose les chaînes de caractères :*

```
tapez      PRINT "LA";"DECOUVERTE";"D'";"ALICE"
réponse    LADECOUVERTED'ALICE
```

Faisons la même expérience avec des nombres :

```
tapez      PRINT 362;34
réponse    362 34
```

```
puis tapez PRINT 362;-34
réponse    362 -34
```

Dans ces deux exemples, vous remarquez que sur l'écran

- le premier nombre s'affiche à partir de la deuxième position de la ligne,
- il existe deux espaces entre 362 et 34 et un seul entre 362 et -34.

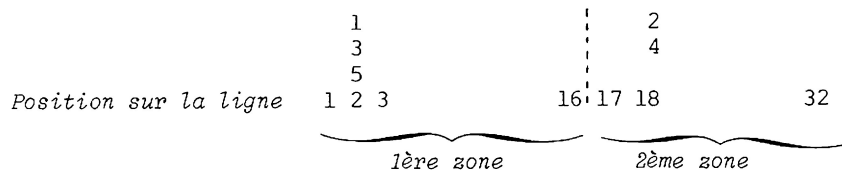
En effet, un nombre affiché sur l'écran est précédé d'un espace s'il est positif et du signe (-) s'il est négatif ; il est toujours suivi d'un espace.

*La virgule associée à l'instruction PRINT permet de réaliser une tabulation qui divise la ligne de 32 caractères en 2 zones de 16 caractères :*

```
tapez      PRINT 1,2,3,4,5
réponse    1           2
           3           4
           5
```

Le schéma ci-dessous indique précisément les positions occupées par les chiffres 1,2,3,4 et 5 sur leur ligne d'écran :

1, 3 et 5 à la 2e position  
2 et 4 à la 18e position



## LA DECOUVERTE DE L'ALICE ET DU MC/10

```
Maintenant, tapez      PRINT "A","B","C","E"
réponse A              B
                   C    D
                   E
```

A, C et E s'affichent à la 1ère position sur la ligne,  
B et D s'affichent à la 17e position.

### Exemple

```
tapez      PRINT "NOM","PRENOM","---","-----","DUPONT",
           "PIERRE","MARTIN","PAUL"
```

**Remarque :** lorsque la longueur de l'instruction dépasse la longueur de la ligne (32 caractères), le curseur saute automatiquement au début de la ligne suivante, et vous n'appuyez sur la touche **ENTER** qu'à la fin de l'instruction (ici, après "PAUL").

ALICE accepte des instructions dont la longueur n'excède pas 127 caractères.

### Autre exemple

```
tapez      PRINT "MICROORDINATEUR/COULEUR","ALICE"
reponse    MICROORDINATEUR/COULEUR
           ALICE
```

Dans cet exemple, le premier mot ayant un nombre de caractères supérieur à la largeur de la première zone, le mot suivant est affiché dans la première zone de la ligne suivante.

## EFFACEMENT D'UN CARACTERE OU D'UNE LIGNE

**CONTROL** **Q** permet de ramener le curseur d'une position vers la gauche en effaçant le dernier caractère frappé. En répétant cette opération autant de fois que nécessaire, il est possible de corriger n'importe quelle erreur de frappe sur la ligne où on se trouve.

### Exemple


```
tapez      PRINT "BONHOUR"
```


Avant d'avoir tapé sur **ENTER**, vous constatez la faute de frappe.

- Pour corriger l'erreur sans retaper toute la ligne, procédez comme suit :


```
Appuyez 5 fois sur CONTROL Q en même temps.
retapez  JOUR"
tapez    ENTER
```

### LA DECOUVERTE DE L'ALICE ET DU MC/10

- Pour effacer toute une ligne, vous devrez utiliser les touches **CONTROL** **A** 

L'inscription **L.DEL** inscrite au-dessus de la touche **A**  est l'abréviation de Line **DE**lete qui signifie "effacer une ligne".

tapez PRINT "CETTE LIGNE EST INUTILE"  
Avant d'avoir tapé **ENTER**, vous constatez que cette ligne est inutile.

tapez **CONTROL** **A**   
La ligne complète est effacée.

### UTILISATION DE LA TOUCHE **CONTROL**

L'instruction PRINT peut être aussi affichée par la touche **CONTROL** **.**

tapez **CONTROL** **)** **9** "ALICE"

Vous voyez apparaître l'instruction PRINT "ALICE"  
réponse ALICE

### ECRITURE ABREGEE DE PRINT

Dans l'écriture d'une instruction PRINT, vous pouvez remplacer le mot PRINT par le point d'interrogation :

tapez ? "FIN DU PREMIER CHAPITRE"  
réponse FIN DU PREMIER CHAPITRE

## **CHAPITRE 2**

# LES VARIABLES NUMERIQUES ET ALPHANUMERIQUES

- Les nombres
- Les opérations arithmétiques
- Les variables numériques
- LET
- Les tableaux de nombres
- DIM
- Les chaînes de caractères  
et variables alphanumériques
- L'instruction CLEAR
- Les comparaisons de nombres  
et de chaînes de caractères
- Les opérateurs logiques AND,  
OR, NOT





### LA DECOUVERTE DE L'ALICE ET DU MC/10

Nous pouvons additionner plusieurs nombres :  
tapez PRINT 5 + 6 + 7 + 8  
réponse 26

#### LA SOUSTRACTION

tapez PRINT 230-152  
réponse 78

#### LA MULTIPLICATION

Elle utilise le symbole \* (touche 

*
:

 )  
tapez PRINT 11 \* 7  
réponse 77

#### LA DIVISION

Elle utilise le symbole / (touche 

?
/

 )  
tapez PRINT 10 / 3  
réponse 3.33333333

#### ELEVATION A UNE PUISSANCE

Pour effectuer l'opération suivante  $3 * 3 * 3 * 3$  :

tapez PRINT 3\*3\*3\*3  
réponse 81

mais il est plus simple d'utiliser l'opération d'élevation à la puissance ↑ (touches 

CONTROL
---------

 et 

Z
---

 )

tapez PRINT 3 ↑ 4  
réponse 81

$3 \uparrow 4$  se lit : 3 à la puissance 4 et correspond à la notation mathématique  $3^4$ .

#### Exemple

tapez PRINT 3+6\*8  
réponse 51

Les nombres négatifs peuvent s'écrire sans parenthèses

tapez PRINT 2 \* -5  
réponse -10

Lorsque vous écrivez une expression très compliquée, il est toujours possible d'éviter des erreurs en utilisant des parenthèses, mais ceci a l'inconvénient de ralentir l'exécution de l'opération par l'ordinateur.

## LA DECOUVERTE DE L'ALICE ET DU MC/10

Pour éviter de mettre trop de parenthèses, il est donc bon de connaître l'ordre d'exécution des opérateurs :

- 1 - l'élevation à la puissance : †
- 2 - la multiplication et la division : \* et /
- 3 - l'addition et la soustraction : + et -

### Exemple

```
tapez      PRINT 100/5 † 2
réponse    4
L'opération 5 † 2 est exécutée en premier
tapez      PRINT 100/10*2
réponse    20
```

Dans le deuxième exemple, les opérateurs ayant même niveau de priorité, les opérations sont exécutées de la gauche vers la droite : d'abord  $100/10=10$  puis  $10*2=20$ .

## LES VARIABLES NUMERIQUES

Pour calculer la distance parcourue pendant un temps donné par un véhicule dont la vitesse est connue, on utilise la formule :

$$\text{DISTANCE} = \text{VITESSE} * \text{TEMPS}$$

ou celle-ci plus simple

$$D = V * T$$

Dans ces deux formules, DISTANCE, VITESSE, TEMPS, D, V et T sont des variables numériques.

Pour appliquer cette formule, il faut donner une valeur numérique aux variables V et T et la valeur numérique de D en est alors déduite.

Pour faire effectuer un tel calcul par l'ordinateur, nous allons procéder de la même façon, c'est-à-dire, en lui donnant la formule indiquant la méthode de calcul et les valeurs numériques des variables.

Voyons comment se fait en langage Basic l'affectation d'une valeur à une variable numérique :

Par exemple, pour affecter à A la valeur 7 :

```
tapez      A=7
N'oubliez pas 
puis tapez PRINT A
réponse    7
```

A = 7 est la simplification de l'instruction "LET A = 7" qui signifie "Soit A = 7".

En tapant LET A=7, on obtient donc exactement le même résultat que précédemment.

## LA DECOUVERTE DE L'ALICE ET DU MC/10

Dans le choix des noms des variables, certaines règles sont à respecter :

- *Le nom peut comprendre plusieurs dizaines de caractères mais seuls les deux premiers sont significatifs.*

```
tapez      TEMPS = 76.28
tapez      TENEUR = 889
tapez      PRINT TEMPS
réponse    889
tapez      PRINT TERRAIN
réponse    889
```

Pour l'ordinateur, il n'existe qu'une seule variable : TE.

### *Autre exemple*

```
tapez      A18 = 9.652
tapez      PRINT A14
réponse    9.652
```

Ici la seule variable identifiée est A1.

- *Le nom peut comporter des lettres et des chiffres à l'exclusion de tout autre caractère typographique.*
- *Le premier caractère est nécessairement une lettre.*
- *Certains noms sont interdits : les noms utilisés par le langage Basic et les noms contenant un de ces noms (voir Annexe 2 : mots réservés).*

### *Exemples de noms valides*

```
ALICE
N
BØ
V5B869
```

### *Exemple de noms interdits*

```
PRINTEMPS
TOTO
RAYON
SCORE
```

Une variable B peut se faire affecter la valeur d'une autre variable H,

### *Exemple*

```
tapez      B = -8
tapez      H = 255
tapez      PRINT B
réponse    -8
tapez      B = H
tapez      PRINT B
réponse    255
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

On peut également affecter à une variable la valeur d'une expression arithmétique :

### Exemple

```
tapez      AZ = 3
tapez      QW = 7
tapez      H3 = AZ * 2 + 10 - QW * 8
tapez      PRINT H3
réponse    -37
```

**Remarque importante :** dans les lignes qui précèdent, le signe égal (=) est une instruction qui affecte à la variable écrite à gauche la valeur de l'expression écrite à droite.

Donc l'expression  $A = A + 5$  est parfaitement autorisée en Basic alors qu'elle est absurde en arithmétique.

### Exemple

```
tapez      A = 2
tapez      A = A + 5
tapez      PRINT A
réponse    7
```

Maintenant :

```
tapez      PRINT T
réponse    Ø
tapez      PRINT V
réponse    Ø
```

**Toute variable numérique non utilisée a en effet la valeur Ø.**

Pour remettre toutes les variables à Ø :

```
taper      CLEAR
```

### Exemple

```
taper      BK = -1893
tapez      CLEAR
tapez      PRINT BK
réponse    Ø
```

**CLEAR** signifie effacer, son rôle sera précisé dans la suite de ce chapitre.

## TABLEAUX DE NOMBRES

Pour présenter clairement des résultats numériques, on utilise couramment des tableaux. En voici un exemple, donnant les populations de la Communauté Européenne en 1980.

**LA DECOUVERTE DE L'ALICE ET DU MC/10**

<i>PAYS</i>	<i>HABITANTS (en million)</i>
Allemagne	61
Belgique	10
Danemark	5
France	53
Irlande	3
Italie	56
Luxembourg	0,4
Pays-Bas	14
Royaume-Uni	56

Pour manipuler une série de nombre comme celle-ci, au lieu d'utiliser un nom différent pour chaque variable, on utilise un seul nom avec un indice. On parle alors de **tableaux**.

Appelons HA ce tableau à 9 éléments :

```
tapez      HA(0) = 61
           HA(1) = 10
           HA(2) = 5
           HA(3) = 53
           HA(4) = 3
           HA(5) = 56
           HA(6) = 0.4
           HA(7) = 14
           HA(8) = 56
```

HA(3) est l'élément d'indice 3 (c'est le quatrième élément du tableau). Plus généralement HA(I) est l'élément d'indice I.

**Exemple**

```
tapez      I = 4
tapez      PRINT HA(I)
réponse    3
tapez      PRINT HA(I+1)
réponse    56
tapez      PRINT HA(I/2)
réponse    5
```

Si nous voulons compléter ce tableau, par exemple pour l'entrée de la GRECE (9 millions d'habitants), de l'ESPAGNE (36) et du PORTUGAL (10), nous procéderons de la même façon :

```
tapez      HA(9) = 9
tapez      HA(10) = 36
tapez      HA(11) = 10
réponse    ?BS ERROR
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

ALICE refuse l'entrée d'un douzième élément dans tout tableau dont le nombre maximal d'éléments n'a pas été précisé au préalable, à l'aide de l'instruction DIM (dimensionnement).

Avant de créer un tableau, il est donc utile d'évaluer le nombre d'éléments qu'il contiendra et de l'indiquer à l'ordinateur.

Par exemple, pour créer un tableau de nom TA comportant 31 éléments :

```
tapez      DIM TA(30)
puis tapez TA(11)=33.98
tapez      PRINT TA(11)
réponse    333.98
tapez      PRINT TA(23)
réponse    Ø
tapez      PRINT TA(3Ø)
réponse    Ø
tapez      PRINT TA(31)
réponse    ?BS ERROR
```

**Remarques :** tout élément d'un tableau auquel aucune valeur n'a été affectée est égal à Ø. Un tableau dimensionné à 3Ø contient 31 éléments d'indice Ø à 3Ø.

Essayons d'utiliser l'instruction DIM pour ajouter à HA le deuxième élément :

```
tapez      DIM HA(11)
réponse    ?DD ERROR
```

Ce message d'erreur apparaîtra chaque fois que vous essaieriez d'utiliser l'instruction DIM pour un tableau déjà dimensionné.

Le tableau HA avait été dimensionné automatiquement à 1Ø par l'ordinateur dès l'écriture du premier élément.

Pour redimensionner correctement HA, il est nécessaire de l'effacer à l'aide de l'instruction CLEAR :

```
tapez      CLEAR
puis tapez DIM HA(11)
```

L'ordinateur est prêt maintenant à accepter douze éléments dans le tableau HA.

## TABLEAUX A PLUSIEURS DIMENSIONS

Le tableau HA de la page précédente ne comporte qu'un

**LA DECOUVERTE DE L'ALICE ET DU MC/10**

seul indice (donc qu'une seule colonne de nombres) :  
c'est un tableau à une dimension.

Le tableau ci-dessous donne les populations, les densités et les superficies des douze pays européens.

<i>Pays</i>	<i>Population (millions)</i>	<i>Densité</i>	<i>Superficie (milliers km)</i>
Allemagne	61,5	247	249
Belgique	10	319	31
Danemark	5	118	43
Espagne	36	71	505
France	53	97	547
Grèce	9	69	132
Irlande	3	45	70
Italie	56,3	187	301
Luxembourg	0,4	138	3
Pays-Bas	13,8	339	41
Portugal	9,5	103	92
Royaume-Uni	56	228	244

*(Chiffres provenant du "Petit-Atlas" Larousse 1973)*

Pour traiter tous les nombres contenus dans ce tableau, il est possible de les considérer comme des éléments d'un seul tableau HB.

Le repérage de chaque élément nécessite deux indices, un pour la ligne et un pour la colonne ; le tableau ainsi défini est dit à "deux dimensions". La taille des tableaux n'est limitée que par la capacité de la mémoire d'ALICE :

tapez DIM HB(11,2)

Cette instruction dimensionne le tableau HB à 12 lignes et 3 colonnes.

tapez HB(10,2)=92

tapez PRINT "SUPERFICIE DU PORTUGAL:";HB(10,2)

réponse SUPERFICIE DU PORTUGAL : 92

**CHAINES DE CARACTERES**  
**ET VARIABLES ALPHANUMERIQUES**

Les variables numériques sont utilisées pour les nombres et les variables alphanumériques pour les chaînes

## LA DECOUVERTE DE L'ALICE ET DU MC/10

de caractères (alphanumérique est la fusion de "alpha-bétique" et "numérique").

Pour les distinguer des variables numériques, les noms des variables alphanumériques se terminent toujours par le symbole \$.

### Exemple

```
A$  
ALICE$  
K8$
```

A l'exception de la convention ci-dessus, toutes les règles d'écriture des noms de variables numériques s'appliquent aux variables alphanumériques.

```
tapez      D$="ALICE"  
tapez      PRINT D$  
réponse    ALICE  
tapez      K8$="DATA"  
tapez      FIN$="LIMITED"  
tapez      PRINT D$;K8$;FIN$  
réponse    ALICEDATALIMITED
```

Un chapitre entier est consacré aux manipulations des chaînes de caractères dans la suite de ce manuel.

### Remarque :

```
tapez      A$=""  
tapez      PRINT "*" ; A$ ; "*"   
réponse    * *  
tapez      A$=" "  
tapez      PRINT "*" ; A$ ; "*"   
réponse    * *
```

L'exemple montre la différence existant entre la chaîne de caractères vide "" et la chaîne de caractères égale à un espace " ". De même que les variables numériques sont a priori toutes égales à zéro, les variables alphanumériques sont a priori toutes vides.

## L'INSTRUCTION CLEAR

Les caractères constituant les variables alphanumériques sont enregistrés dans la mémoire d'ALICE. Le nombre de caractères que peuvent contenir ces variables est donc limité. Il est fixé a priori à 100 caractères maximum, mais peut être modifié grâce à l'instruction CLEAR.



## LA DECOUVERTE DE L'ALICE ET DU MC/10

- l'instruction CLEAR N
- remet à zéro toutes les variables numériques,
  - affecte à toutes les variables alphanumériques la chaîne vide,
  - annule l'effet de toutes les instructions DIM qui précèdent,
  - réserve la place mémoire pour N caractères pour les variables alphanumériques. Si N est omis, 100 caractères sont réservés automatiquement.

### **TABLEAUX DE CHAINES DE CARACTERES**

Ils sont analogues aux tableaux de nombres ;

```
tapez      DIM PI$(11)
           PI$(0)="ALLEMAGNE"
           PI$(1)="BELGIQUE"
           PI$(2)="DANEMARK"
           PI$(3)="ESPAGNE"
           PI$(4)="FRANCE"
           PI$(5)="GRECE"
           PI$(6)="IRLANDE"
           PI$(7)="ITALIE"
           PI$(8)="LUXEMBOURG"
           PI$(9)="PAYS-BAS"
           PI$(10)="PORTUGAL"
           PI$(11)="ROYAUME-UNI"

puis tapez I=2
tapez      PRINT PI$(I)
réponse   DANEMARK
tapez      PRINT PI$(3*I+1)
réponse   ITALIE
```

**Remarque :** on peut également créer des tableaux alphanumériques à deux dimensions.

### **COMPARAISONS**

En plus des opérations arithmétiques, ALICE peut effectuer sur les nombres et sur les expressions numériques les comparaisons suivantes :

```
égal à ..... =
différent de ..... < >
supérieur à ..... >
inférieur à ..... <
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

supérieur ou égal .... > =  
inférieur ou égal .... < =

Lorsqu'on écrit "A > B", cette déclaration, ou proposition peut être soit vraie soit fausse. Si on demande à l'ordinateur d'examiner cette proposition, voyons comment il répondra dans chacun de ces deux cas :

### 1er cas

tapez            A = 6  
tapez            B = 4  
tapez            PRINT A>B  
réponse        -1

### 2e cas

tapez            A = 3  
tapez            B = 15  
tapez            PRINT A>B  
réponse        Ø

Dans le premier cas, la proposition "A > B" est vraie, l'ordinateur répond -1.

Dans le second cas, elle est fausse, l'ordinateur répond Ø.

L'ordinateur répondra ainsi (-1=vrai), (Ø=faux) chaque fois qu'il examinera une telle proposition :

### **Exemple**

tapez            PRINT 3 < > 5  
réponse        -1  
tapez            PRINT 3 > = 3  
réponse        -1  
tapez            PRINT 13 < > 13  
réponse        Ø

Revenons sur l'égalité :

tapez            PL=3=5  
tapez            PRINT PL  
réponse        Ø  
tapez            PL=6=6  
tapez            PRINT PL  
réponse        -1

Ces deux exemples montrent *les deux significations possibles du signe égal (=) dans le langage Basic :*  
- affectation d'une valeur à une variable,  
- comparaison.

Dans la ligne PL=3=5, ces deux opérations sont effectuées :

1- examen de la proposition 3=5 qui donne Ø (résultat faux),

## LA DECOUVERTE DE L'ALICE ET DU MC/10

2- affectation de ce résultat à PL

De la même façon, dans la ligne PL=6=6, PL prend la valeur -1.

### **Autres exemples**

```
tapez      KV = 129.7
tapez      KV = KV = KV
tapez      PRINT KV
réponse    -1
tapez      KV = 129.7
tapez      KV = KV = KV = KV
tapez      PRINT KV
réponse    Ø
```

Ces résultats s'expliquent comme précédemment.

Il est également possible de *comparer des chaînes de caractères* :

```
tapez      HJ$="POMME"
tapez      PRINT HJ$="POIRE"
réponse    Ø
tapez      PRINT HJ$ <> "ORANGE"
réponse    -1
```

### ***Attention : Ne pas confondre deux chaînes de caractères égales avec deux chaînes de caractères de même longueur.***

Deux chaînes de caractères sont égales si chaque caractère de l'une est identique au caractère de même rang de l'autre.

Les opérateurs de comparaison des nombres <, >, <= et >= permettent également de comparer deux chaînes de caractères.

```
tapez      PRINT "ALICE" > "ACHILLE"
réponse    -1
tapez      PRINT "ALICE" > "PRINCESSE"
réponse    Ø
tapez      PRINT "ALICE" > "ALICE"
réponse    Ø
```

Dans les tests précédents, on compare le mot ALICE' à d'autres mots, suivant l'ordre alphabétique. La réponse est -1 si le mot ALICE est après l'autre dans l'ordre alphabétique ; la réponse est Ø dans le cas contraire.

Lorsque les chaînes de caractères comportent des chiffres ou des symboles, le classement s'effectue suivant l'ordre dans lequel sont codés tous les caractères dans le micro-ordinateur (cf instruction ASC du chapitre VIII et annexe 3).

## LA DECOUVERTE DE L'ALICE ET DU MC/10

Voici quelques exemples :

```
tapez      PRINT "7253" < "ALICE"
réponse    -1
tapez      PRINT "12" < "2"
réponse    -1
tapez      PRINT "%%%" > "'%$"
réponse    Ø
tapez      PRINT "AF$23" <= "VE%24"
réponse    -1
```

Vous constatez que le classement s'effectue comme dans un dictionnaire.

### OPERATEURS LOGIQUES : AND, OR, NOT

L'ordinateur répond par Ø ou par -1 lorsqu'on lui soumet une proposition pour exprimer qu'elle est fausse ou qu'elle est vraie.

Il est également possible de lui soumettre plusieurs propositions articulées par AND (ET), OR (OU) et NOT (NON).

#### **Exemple**

```
tapez      H = 15
           K = 1ØØ
           L = 3
           PRINT H>L AND H<K
réponse    -1
```

L'ordinateur répond par -1 car les deux propositions "H>L" et "H<K" sont vraies simultanément.

Maintenant,

```
tapez      PRINT H>L AND H>K
réponse    Ø
```

L'ordinateur répond Ø car l'une au moins des deux propositions est fausse.

```
tapez      PRINT H>L OR H>K
réponse    -1
```

Ici la réponse est -1 car l'une au moins des deux propositions est vraie.

```
tapez      PRINT H<L OR H>K
réponse    Ø
```

La réponse est Ø car les deux propositions sont fausses.

LA DECOUVERTE DE L'ALICE ET DU MC/10

En résumé :

- la combinaison Proposition 1 AND Proposition 2 est vraie dans le seul cas où les deux propositions sont vraies, fausse dans tous les autres cas.
- la combinaison Proposition 1 OR Proposition 2 est fausse dans le seul cas où les deux propositions sont fausses, vraie dans tous les autres cas.

On peut écrire des combinaisons de plusieurs propositions utilisant des AND et des OR :

```
tapez      PRINT H<L OR H>K AND K=H OR L=K
réponse    Ø
```

NOT exprime la négation d'une proposition :

**Exemple**

```
tapez      PRINT H<K
réponse    -1
tapez      PRINT NOT(H < K)
réponse    Ø
```

Voici l'ordre de priorité d'exécution des opérateurs logiques et arithmétiques :

1	( )
2	NOT
3	†
4	* /
5	+ -
6	> < = >= <= <>
7	AND
8	OR

LA DECOUVERTE DE L'ALICE ET DU MC/10

EXERCICES SUR LE CHAPITRE II

**Exercice 2.1 :** Trouver l'écriture habituelle avec neuf chiffres significatifs des nombres suivants écrits en notation exponentielle :

$$\begin{aligned} &7.28354957E+9 \\ &-3.84163524E-04 \end{aligned}$$

**Exercice 2.2 :** Calculer, à l'aide de votre ALICE, le résultat des opérations suivantes :

$$150-92$$

$$(89-18)+(25 \times 2)-13$$

$$\frac{15 \times 3}{2}$$

$$\frac{15}{2 \times 3}$$

$$\frac{15+3}{2}$$

$$\frac{15}{2} + 3$$

$$\frac{1736}{(1.3)^6}$$

$$\left(\frac{1736}{1.3}\right)^6$$

**Exercice 2.3 :** Combien d'éléments comprend un tableau dimensionné à 14 ?

Combien de dimensions et d'éléments comprend un tableau dimensionné à (10,6) ?

## CHAPITRE 3

### LES PREMIERS PROGRAMMES

- LIST
- RUN
- END
- EFFACEMENT D'UNE LIGNE  
DE PROGRAMME
- CSAVE
- CLOAD
- SKIPF
- INPUT
- NEW
- READ
- DATA
- RESTORE
- CSAVE\*
- CLOAD\*
- CLS
- PRINT - TAB
- PRINT @
- REM

## LA DECOUVERTE DE L'ALICE ET DU MC/10

ALICE joue aux échecs, fait de la musique, imprime des lettres, contrôle des appareils électriques. Toutes ces tâches très différentes résultent toutefois d'un même processus : l'exécution d'un programme préalablement enregistré dans la mémoire de l'ordinateur.

Un exemple va nous permettre d'aborder cette notion importante : un hôtelier calcule le montant de ses recettes. Aujourd'hui, il a loué 7 chambres à F 125, 15 à F 85 et 12 à F 50. Il fait donc le calcul suivant :

```
- chambres catégorie 1 : 125 * 7 = 875
- chambres catégorie 2 : 85 * 15 = 1275
- chambres catégorie 3 : 50 * 12 = 600
Total : 875+1275+600 = 2750
```

Tous les jours, il effectue la même suite d'opérations seul change le nombre de chambres louées dans chaque catégorie. Il lui est possible d'enregistrer dans la mémoire de l'ordinateur cette suite d'opérations pour l'utiliser chaque fois qu'il en aura besoin.

Il écrit donc un programme. La suite des opérations est la suivante :

```
C1 = 125 * 7
C2 = 85 * 15
C3 = 50 * 12
T = C1+C2+C3
```

Pour enregistrer ce programme dans la mémoire d'ALICE, il suffit d'écrire chacune de ces opérations précédées d'un nombre compris entre 0 et 63999 et suivies de l'enfoncement de la touche ENTER. On obtient ainsi quatre lignes de programme. Une ligne de programme peut contenir jusqu'à 127 caractères.

Quand vous arrivez en fin de ligne d'écran, n'appuyez pas sur ENTER. ALICE positionne le curseur au début de la ligne suivante. L'action sur la touche ENTER ne doit se faire que pour provoquer l'enregistrement de la ligne frappée dans la mémoire d'ALICE.

### LES ORDRES LIST ET RUN

Ecrivons la première ligne :  
tapez            1 C1 = 125 \* 7



### LA DECOUVERTE DE L'ALICE ET DU MC/10

Vérifions à l'aide de l'ordre LIST qu'elle a été enregistrée :

```
tapez      LIST
réponse    1 C1 = 125 * 7
```

Ecrivons les lignes suivantes :

```
tapez      2 C2 = 85 * 15
tapez      3 C3 = 50 * 12
tapez      4 T = C1+C2+C3
puis tapez  LIST
réponse    1 C1 = 125 * 7
           2 C2 = 85 * 15
           3 C3 = 50 * 12
           4 T = C1+C2+C3
```

*L'ordre LIST permet de "lister" le programme, c'est-à-dire de l'afficher.*

Pour commander l'exécution de ce programme :

```
tapez      RUN
réponse    OK
```

Le message OK signifie que le programme a été exécuté. Vérifions-le en demandant à l'ordinateur d'afficher T :

```
tapez      PRINT T
réponse    2750
```

Il serait pratique d'inclure cette instruction dans le programme !

Pour cela, ajoutons une cinquième ligne :

```
tapez      5 PRINT T
puis tapez  LIST
réponse    1 C1 = 125 * 7
           2 C2 = 85 * 15
           3 C3 = 50 * 12
           4 T = C1+C2+C3
           5 PRINT T
```

Maintenant,

```
tapez      RUN
réponse    2750
           OK
```

Cette réponse correspond au total calculé par le programme. Complétons l'affichage de ce nombre par un texte qui le rendra plus compréhensible ; pour cela changeons la ligne 5 : il suffit de la réécrire.

```
tapez      5 PRINT "TOTAL=";T
```

Vérifions que le changement a été effectué en listant seulement la ligne qui nous intéresse :

```
tapez      LIST 5
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

réponse 5 PRINT "TOTAL=";T  
puis tapez RUN

*L'ordre LIST suivi d'un numéro affiche la ligne correspondante.*

Cet ordre permet également en utilisant le signe (-) :  
- de lister toutes les lignes comprises entre deux numéros :

tapez LIST 2-4  
réponse 2 C2 = 85 \* 15  
3 C3 = 50 \* 12  
4 T = C1+C2+C3

- de lister toutes les lignes à partir d'un numéro donné :

tapez LIST 3-  
réponse 3 C3 = CASSETTE)  
4 T = C1+C2+C3  
5 PRINT "TOTAL=";T

- de lister toutes les lignes jusqu'à un numéro donné :

tapez LIST -4  
réponse 1 C1 = 125 \* 7  
2 C2 = 85 \* 15  
3 C3 = 50 \* 12  
4 T = C1+C2+C3

### L'INSTRUCTION END

Complétons le programme de l'hôtelier par une dernière ligne :

tapez 6 END

*Cette instruction END (FIN en français) provoque la fin de l'exécution du programme.*

### EFFACEMENT D'UNE LIGNE DE PROGRAMME

Nous avons vu que pour modifier une ligne il suffit de la réécrire ; *pour supprimer une ligne, il suffit de taper son numéro suivi de **ENTER*** .

tapez 5 **ENTER**  
tapez LIST  
réponse 1 C1 = 125 \* 7  
2 C2 = 85 \* 15

### **LA DECOUVERTE DE L'ALICE ET DU MC/10**

```
3 C3 = 50 * 12
4 T = C1+C2+C3
6 END
```

**Remarque :** Ne pas confondre

- effacer une ligne d'écran, ce qui s'obtient en tapant **CONTROL A**
- et effacer une ligne de programme, ce qui s'obtient en tapant son numéro de ligne suivi de **ENTER**.

### **LES INSTRUCTIONS CSAVE, CLOAD, SKIPF**

Si nous éteignons ALICE, le programme que nous venons d'écrire sera perdu. Il est facile de le vérifier mais il est prudent auparavant de sauvegarder ce programme sur une cassette.

Rappelons d'abord la manière de raccorder le lecteur de cassette à ALICE : utiliser le fil de raccordement muni à une extrémité d'une prise DIN qui se connecte à l'arrière d'ALICE (dans la prise située au-dessus de l'inscription CASSETTE) et à l'autre extrémité de trois jacks qui se branchent à votre magnétophone de la manière suivante :

- le petit jack à la prise MIC (micro),
- l'autre jack gris à l'entrée AUX,
- le jack noir à la sortie (EAR).

#### **Enregistrement d'un programme : CSAVE**

- Positionnez la bande à l'endroit à partir duquel vous voulez enregistrer le programme.
- Réglez le niveau de volume sonore de votre magnétophone à 5.
- Mettez le magnétophone en position d'enregistrement.
- Tapez CSAVE "HOTELIER" puis sur la touche **ENTER** .
- Le message OK et la réapparition du curseur clignotant vous indiquent que l'enregistrement du programme est terminé.
- Appuyez sur la touche STOP du magnétophone.

L'ordre CSAVE "HOTELIER" a sauvegardé le programme qui se trouve dans la mémoire d'ALICE sous le nom HOTELIER que vous utiliserez pour retrouver ce programme lorsque vous voudrez le charger de nouveau dans la mémoire d'ALICE.

## LA DECOUVERTE DE L'ALICE ET DU MC/10

### Chargement d'un programme : CLOAD

- Positionnez la bande magnétique avant le début de l'enregistrement.
- Tapez au clavier l'ordre CLOAD "HOTELIER", puis sur la touche `ENTER`.
- Enfoncez la touche lecture du magnétophone, l'écran est effacé et la lettre S (abréviation de Searching) indique qu'ALICE recherche le programme HOTELIER.
- L'apparition des messages F HOTELIER (abréviation de Found HOTELIER) puis OK signifie que le programme HOTELIER a été trouvé puis chargé en mémoire.

L'ordre CLOAD "HOTELIER" charge dans la mémoire d'ALICE le programme précédemment enregistré par l'ordre CSAVE "HOTELIER". Le nom qui suit CSAVE contient au plus 8 caractères significatifs.

**Remarques** : L'ordre CLOAD seul charge le premier programme rencontré. L'ordre CSAVE seul sauvegarde le programme en mémoire sans lui donner de nom.

### Positionnement en fin d'un enregistrement : SKIPF

L'ordre SKIPF "HOTELIER" positionne la bande magnétique à la fin de l'enregistrement du programme HOTELIER. L'ordre SKIPF seul positionne la bande magnétique à la fin de l'enregistrement du premier programme rencontré.

### L'INSTRUCTION INPUT, L'ORDRE NEW

Le lendemain, le nombre de chambres louées dans chaque catégorie a changé.

Voyons comment l'hôtelier va utiliser le même programme en modifiant les trois premières lignes :

```
tapez      1 C1 = 125 * 2
           2 C2 =  85 * 10
           3 C3 =  50 * 16
puis       LIST
           RUN
réponse    TOTAL = 1900
           OK
```

Il n'est pas très pratique pour l'hôtelier de devoir réécrire ces lignes tous les jours. Il existe dans le langage Basic une instruction qui permet de modifier la valeur d'une ou plusieurs variables pendant l'exécution du programme : c'est l'instruction **INPUT**.

### LA DECOUVERTE DE L'ALICE ET DU MC/10

Nous allons écrire un nouveau programme en utilisant cette instruction. Au préalable, il est prudent d'effacer le programme existant en mémoire à l'aide de l'ordre **NEW**.

tapez NEW  
puis tapez LIST

Vous constatez que le programme a bien été effacé.

Appelons N1, N2 et N3 le nombre de chambres louées pour chacune des 3 catégories :

tapez  
1 INPUT N1  
2 INPUT N2  
3 INPUT N3  
4 T=125\*N1+85\*N2+50\*N3  
5 PRINT "TOTAL=",T  
6 END

tapez RUN  
réponse ?

Le point d'interrogation signifie que l'ordinateur attend que vous donniez la valeur de N1.

tapez 2  
réponse ?

L'ordinateur attend la valeur de N2.

tapez 10  
réponse ?

L'ordinateur attend la valeur de N3.

tapez 16  
réponse TOTAL=1900

Ce même programme peut donc être utilisé quelles que soient les valeurs numériques prises par les variables N1, N2 et N3. Essayez-le de nouveau avec d'autres valeurs.

**INPUT signifie "Mettre dans", "Entrer". Cette instruction permet donc d'"Entrer" les valeurs de certaines variables à partir du clavier pendant l'exécution du programme.**

Voyons maintenant un exemple d'utilisation de l'instruction INPUT pour l'entrée des valeurs de variables alphanumériques. (Désormais nous numérotions les lignes de 10 en 10, ce qui facilite l'insertion éventuelle de nouvelles lignes).

tapez NEW

### LA DECOUVERTE DE L'ALICE ET DU MC/10

puis le programme suivant :

```
10 PRINT "QUEL EST VOTRE PRENOM ?"  
20 INPUT P$  
30 PRINT  
40 PRINT "MOI JE M'APPELLE ALICE "  
50 PRINT "AU REVOIR, "P$;" !!"  
60 END
```

```
tapez      RUN  
  
          QUEL EST VOTRE PRENOM ?  
          ? LAURENT  
  
          MOI JE M'APPELLE ALICE,  
          AU REVOIR,LAURENT!  
          OK
```

Il est possible de supprimer l'instruction PRINT "QUEL EST VOTRE PRENOM?" en intégrant cette question dans l'instruction INPUT.

Ceci est vrai aussi bien pour les variables numériques que pour les variables alphanumériques.

N'oubliez pas de taper NEW avant d'écrire le programme suivant :

```
10 INPUT "QUEL EST VOTRE PRENOM ";P$  
20 INPUT "QUEL EST VOTRE AGE ";A  
30 PRINT  
40 PRINT "MOI JE M'APPELLE ALICE,"  
50 PRINT "ET J'AI BIEN MOINS DE";A;"ANS !!"  
60 PRINT "AU REVOIR, ",P$;" !"  
70 END
```

Vous constatez que le signe ; (point virgule) est indispensable dans l'exécution de l'instruction INPUT lorsqu'il y a un texte à afficher :

```
tapez      RUN  
  
          QUEL EST VOTRE PRENOM ? ANNE  
          QUEL EST VOTRE AGE ? 38  
  
          MOI JE M'APPELLE ALICE,  
          ET J'AI BIEN MOINS DE 38 ANS !!  
          AU REVOIR, ANNE !
```

### LA DECOUVERTE DE L'ALICE ET DU MC/10

Voici un autre exemple :

```
10 INPUT "QUEL EST VOTRE PRENOM " :P$
20 INPUT "TAPEZ UN ADJECTIF QUALIFICATIF " :Q$
30 INPUT "TAPEZ UN ADVERBE " :B$
40 PRINT
50 PRINT
60 PRINT P$:" EST UN " :Q$:" APPRENTI"
70 PRINT "EN PROGRAMMATION"
80 PRINT
90 PRINT P$:" APPREND TRES " :B$
100 END
```

et une utilisation de ce programme :

```
QUEL EST VOTRE PRENOM ? SANDRINE
TAPEZ UN ADJECTIF QUALIFICATIF ? BON
TAPEZ UN ADVERBE ? VITE

SANDRINE EST UN BON APPRENTI
EN PROGRAMMATION

SANDRINE APPREND TRES VITE !!
```

Enfin, il est également possible d'entrer plusieurs variables, numériques et alphanumériques, avec une seule instruction INPUT, ceci en séparant chaque variable par une virgule.

#### *Exemple*

```
10 INPUT "TAPEZ VOTRE PRENOM, VOTRE AGE " :P$,A
20 PRINT
30 PRINT
40 PRINT "SE SERAIT-ON DOUTE."
50 PRINT "IL Y A" :A-1:"ANS QUE"
60 PRINT P$:" CE JOYEUX BAMBIN."
70 PRINT "APPRENDRAIT UN JOUR A PROGRAMMER ALICE ?!?! "
80 END
```

```
tapez      RUN
           TAPEZ VOTRE PRENOM, VOTRE AGE ?
           BERNARD, 19

           SE SERAIT-ON DOUTE
           IL Y A 18 ANS QUE
           BERNARD CE JOYEUX BAMBIN,
           APPRENDRAIT UN JOUR A PROGRAMMER
           ALICE ?!?!
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

A une instruction INPUT suivie de plusieurs variables, vous devez donc donner les réponses en les séparant par une virgule, comme dans l'exemple précédent, ou comme dans l'exemple suivant :

```
10 INPUT A,B$,B,A$
20 PRINT A,B$,B,A$
30 END
```

Ce programme demande, dans l'ordre, l'entrée d'un nombre réel, d'une chaîne de caractères, puis de nouveau d'un nombre réel et d'une chaîne de caractères.

Essayons d'entrer des données non conformes aux attentes du programme :

```
tapez      RUN
réponse    ?
tapez      645.87,12,ALICE,CHATEAU
réponse    ?REDO
```

Ce message signifie : recommencer. Il s'affichera chaque fois que vous essaieriez d'entrer une chaîne de caractères à la place d'un nombre et vous indiquera que l'ordinateur reprend complètement l'exécution de l'instruction INPUT.

Comme vous le demande l'ordinateur, essayez de nouveau, cette fois en respectant les types de variables.

## LES INSTRUCTIONS READ, DATA ET RESTORE

*L'instruction READ associée à l'instruction DATA fournit également des données à un programme mais à la différence d'INPUT, ces données proviennent du programme lui-même et non du clavier.* En général, il s'agit de données constantes.

Dans le cas du programme de l'hôtelier, ce sont, par exemple, les prix des chambres inchangés pendant une période de temps relativement longue.

Tapez le programme suivant :

```
10 READ P1,P2,P3
20 INPUT N1,N2,N3
30 T=N1*P1+N2*P2+N3*P3
40 PRINT "TOTAL ="/T
50 END
60 DATA 125.85,50
```



### LA DECOUVERTE DE L'ALICE ET DU MC/10

```
puis      RUN
réponse   ?
tapez     1,5,8
réponse   TOTAL = 950
```

A la ligne 30, les variables numériques P1, P2 et P3 ont été remplacées par les valeurs particulières 125, 85 et 50.

C'est l'instruction READ de la ligne 10 qui affecte à ces variables leurs valeurs particulières en lisant la liste des nombres de l'instruction DATA de la ligne 60.

Il sera plus facile pour l'hôtelier de modifier cette liste dans l'instruction DATA que dans la ligne du programme qui effectue les calculs. Ceci sera d'autant plus vrai que le nombre de valeurs à modifier sera élevé et leur emploi fréquent dans un même programme.

```
Essayez de modifier ces valeurs :
tapez     60 DATA 150,100,60
puis tapez RUN
réponse   ?
tapez     1,5,8
réponse   TOTAL=1130
```

**Remarques :** READ signifie Lire et DATA Donnée. L'instruction DATA peut être placée à une ligne quelconque du programme et être utilisée plusieurs fois dans un même programme.

Modifiez le programme précédent de la manière suivante :

```
tapez
10 READ P1
15 READ P2,P3
35 DATA 150,100
60 DATA 60
```

puis tapez LIST pour voir le programme modifié dans son ensemble :

```
10 READ P1
15 READ P2,P3
20 INPUT N1,N2,N3
30 T=N1*P1+N2*P2+N3*P3
35 DATA 150,100
40 PRINT 'TOTAL =';T
50 END
60 DATA 60
```

### LA DECOUVERTE DE L'ALICE ET DU MC/10

Vous vérifiez que ce programme est équivalent au précédent :

```
tapez      RUN
réponse    ?
tapez      1,5,8
réponse    TOTAL = 1130
```

Donc toutes les instructions DATA sont vues comme une seule liste, lue dans l'ordre de ces instructions.

Les données écrites dans l'instruction DATA peuvent être des nombres et également des chaînes de caractères.

```
tapez      NEW
```

puis tapez le programme suivant :

```
10 READ A,C(4),M$,E$,M$(2),P$(2)
20 PRINT "A=";A
30 PRINT "C(4)=";C(4)
40 PRINT "M$=";M$,"E$=";E$
50 PRINT "M$(2)=";M$(2)
60 PRINT "P$(2)=";P$(2)
70 END
80 DATA 34.76,4,"ALICE",PSI,"DATA LIMITED",FRANCE
```

```
puis tapez      RUN
réponse         A=34.76
                C(4)=4
                M$=ALICE  E$=PSI
                M$(2)=DATA LIMITED
                P$(2)=FRANCE
```

Une même liste peut comporter tous les types de variables et de tableaux dans un ordre quelconque. De plus les chaînes de caractères peuvent être inscrites sans les guillemets à la condition que celles-ci ne comprennent ni la virgule, ni les deux points et qu'elles ne commencent pas par des espaces.

Dans le programme ci-dessus, il y a autant de nombres dans l'instruction READ que d'éléments dans l'instruction DATA. Que se passe-t-il lorsque ceci n'est pas vérifié ?

Modifiez la ligne 80 de la manière suivante :

```
tapez      80 DATA 1,2
puis tapez      RUN
réponse        ?OD ERROR IN 10
                OK
```

Ce message d'erreur signifie qu'il n'y a pas assez de

## LA DECOUVERTE DE L'ALICE ET DU MC/10

données pour affecter une valeur à toutes les variables de l'instruction READ.

**Remarque :** Il n'y a aucun problème en revanche lorsque le nombre d'éléments contenus dans les instructions DATA dépasse le nombre d'appels des instructions READ.

Associée aux instructions READ et DATA, l'instruction RESTORE donne la possibilité de se replacer au début de la liste des données.

Tapez le programme suivant :

```
10 DATA 1
20 READ A,B,C,D
30 PRINT A;B;C;D
40 PRINT
50 READ U,V,W,X,Y,Z
60 DATA 2,3
70 PRINT U;V;W;X;Y;Z
80 END
90 DATA 4,5,6,7
```

```
puis tapez      RUN
réponse       1 2 3 4
              ?OD ERROR IN 50
              OK
```

Ajoutez l'instruction RESTORE (restaurer) à la ligne 45 :

```
tapez      45 RESTORE
puis       LIST
et         RUN
réponse   1 2 3 4
          1 2 3 4 5 6
```

Grâce à l'exécution de l'instruction RESTORE à la ligne 45, la lecture des données effectuée par l'instruction READ de la ligne 50 recommence à partir de la première donnée de la première instruction DATA du programme qui, cette fois, s'exécute correctement jusqu'au bout.

## LES INSTRUCTIONS CSAVE\*, CLOAD\*

Nous avons vu que pour conserver un programme de façon à pouvoir le réutiliser plus tard, nous disposons des instructions CSAVE et CLOAD. Dans le cas où nous

## LA DECOUVERTE DE L'ALICE ET DU MC/10

voudrions conserver des données de façon à les réutiliser plus tard dans un programme, nous disposons des instructions CSAVE\* et CLAOD\*.

```
tapez CLEAR
      T(Ø) = .3439
      T(1) = 6.784
      T(2) = 8
      T(3) = 942.5
```

### Enregistrement d'un tableau : CSAVE\*

- Positionnez la bande à l'endroit à partir duquel vous voulez enregistrer le tableau.
- Réglez le niveau de volume sonore de votre magnétophone à 5.
- Tapez CSAVE\* T, "DONNEES" mais ne tapez pas sur ENTER.
- Mettez le magnétophone en position d'enregistrement.
- Tapez sur ENTER  
Le message OK et la réapparition du curseur clignotant vous indiquent que l'enregistrement est terminé.
- Appuyez sur la touche STOP de votre magnétophone.

**Remarque :** Le tableau T n'a pas été dimensionné car il contient moins de 11 éléments.

### Lecture d'un tableau : CLOAD\*

- Tapez CLEAR
- Tapez PRINT A(Ø), A(1), A(2), A(3)  
Réponse Ø Ø  
Ø Ø

Vous vérifiez ainsi que le tableau A a été remis à zéro.

- Positionnez la bande.
- Tapez au clavier l'instruction CLOAD\* A, "DONNEES".
- Tapez ENTER.
- Enfoncez la touche lecture de votre magnétophone.

Successivement vous obtenez :

- L'effacement de l'écran.
- L'apparition du S (S comme Searching qui signifie recherche).
- L'apparition de F DONNEES (F comme Found qui signifie trouvé).
- OK qui indique que l'enregistrement a été trouvé puis chargé en mémoire.
- Appuyez sur la touche STOP de votre magnétophone.
- Tapez PRINT A(Ø),A(1),A(2),A(3)  
Réponse .3439 6.784  
8 942.5

## LA DECOUVERTE DE L'ALICE ET DU MC/10

### Remarques :

- A peut être de type numérique ou alphanumérique.
- A doit être un tableau dimensionné de façon implicite s'il contient moins de 11 éléments - ici cela a été obtenu par l'exécution de l'instruction PRINT A(0), A(1),A(2),A(3) - ou explicite par l'instruction DIM.

La taille du tableau A doit être supérieure ou égale aux nombres de données lues sur la cassette, sinon le message d'erreur OM est affiché. Si le tableau a une dimension supérieure au nombre de valeurs lues sur la cassette, il est complété par des zéros.

A la différence de CLOAD, **CLOAD\* ne détruit pas le programme qui se trouve dans la mémoire d'ALICE lors du chargement du tableau de données lu sur la cassette.** Aussi CLOAD\* peut être utilisé par un programme pour travailler sur des données enregistrées sur cassette (cf. exercice 5.2).

### LES INSTRUCTIONS CLS, PRINT TAB ET PRINT @

*L'instruction PRINT TAB permet de réaliser une tabulation sur une ligne et l'instruction CLS de vider l'écran de tout caractère.* Pour illustrer le fonctionnement de ces deux instructions, tapez le programme suivant :

```
10 CLS
20 INPUT "VALEUR DE TAB";T
30 PRINT TAB(T);"A"
40 END
```

L'instruction CLS de la ligne 10 a pour effet de vider l'écran de tout caractère et de positionner le curseur au début de la première ligne de l'écran.

```
tapez      RUN
réponse   VALEUR DE TAB ?
```

L'ordinateur attend une valeur numérique positive entière ou décimale (si la valeur entrée est décimale, les chiffres suivant la virgule seront ignorés)

```
tapez      18
```

L'ordinateur affiche la lettre A à la colonne 18 de la ligne (celles-ci sont numérotées de 0 à 31), ce qui correspond donc à la 19e position.

*LA DECOUVERTE DE L'ALICE ET DU MC/10*

Ainsi le curseur est positionné horizontalement par TAB à la position calculée en fonction du nombre ou de l'expression qui suit cette instruction. Essayez le programme précédent en tapant des valeurs différentes.

Dans une instruction PRINT, on peut utiliser plusieurs instructions TAB.

L'instruction PRINT TAB permet de tabuler une ligne mais pas de "revenir en arrière", ni de tabuler l'écran.

*L'instruction PRINT @ permet d'afficher des caractères à tout endroit choisi sur l'écran.*

Tapez le programme suivant :

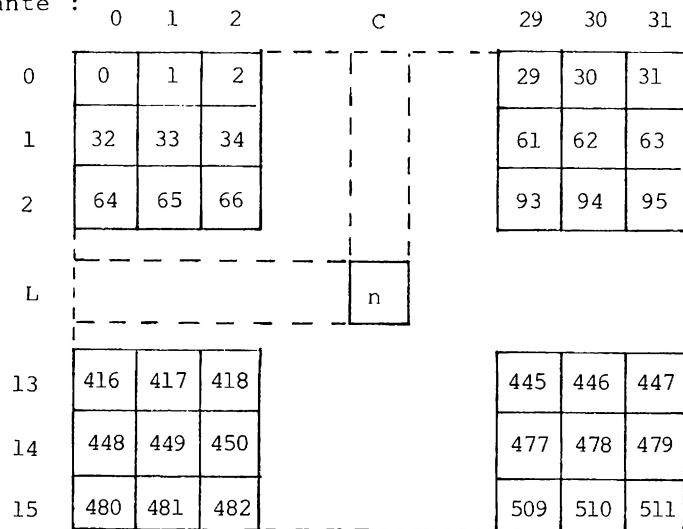
```
10 INPUT "VALEUR (0-511)";V
20 PRINT @V,"COUCOU"
30 END
```

tapez            RUN  
réponse        VALEUR (0-511) ?

L'ordinateur attend une valeur comprise entre 0 et 511.

tapez            56

L'ordinateur affiche COUCOU à partir du 56e caractère de l'écran, donc sur la position 24 de la deuxième ligne. La position d'affichage est calculée de la manière suivante :



$$n = 32 \times L + C$$

## LA DECOUVERTE DE L'ALICE ET DU MC/10

L'écran est divisé en 16 lignes numérotées de 0 à 15, comprenant chacune 32 colonnes numérotées de 0 à 31.

L'instruction PRINT @ n, affiche toute expression numérique ou alphanumérique qui suit la virgule à partir de la nième position de l'écran. Cette position de l'écran se trouve à la ligne L et à la colonne C : L est le quotient de la division de n par 32 et C son reste.

### L'INSTRUCTION REM

REM est l'abréviation de REMark qui signifie "Commentaire". *Cette instruction permet d'inclure des explications en langage clair dans la liste des lignes du programme (ou listing en anglais). Ces instructions sont ignorées par l'ordinateur lors de l'exécution du programme.*

#### *Exemple*

```
10 REM DEBUT DU PROGRAMME
20 REM AFFICHAGE D'UNE LIGNE
30 PRINT "ESSAI DE REM"
40 REM FIN DU PROGRAMME
50 END
```

Il est conseillé d'utiliser cette instruction pour inclure des commentaires qui permettront une lecture plus facile de vos programmes par d'autres utilisateurs ou par vous-même.

Voici un exemple d'utilisation de l'instruction REM :

```
10 REM PROGRAMME DE L'HOTELIER
20 REM ENTREE DES DONNEES
30 INPUT "CATEGORIE 1":N1
40 INPUT "CATEGORIE 2":N2
50 INPUT "CATEGORIE 3":N3
60 REM CALCUL DU TOTAL
70 T=125*N1+85*N2+50*N3
80 REM AFFICHAGE DU TOTAL
90 PRINT "TOTAL=";T
100 REM FIN DU PROGRAMME
110 END
```

**LA DECOUVERTE DE L'ALICE ET DU MC/10**

**EXERCICES SUR LE CHAPITRE III**

**Exercice 3.1 :** *Ecrire un programme qui vous demande le rayon d'un cercle, qui en calcule le périmètre et la surface et qui affiche les résultats.*

**Exercice 3.2 :** *Ecrire un programme qui vous demande le nom d'un pays, sa superficie et sa population, qui en calcule ensuite la densité et qui affiche les résultats. Tester ensuite ce programme en utilisant le tableau du chapitre 2 donnant ces informations pour dix pays.*

**Exercice 3.3 :** *Modifier le programme de l'exercice 3.2 de manière à afficher les résultats comme dans l'exemple ci-dessous :*

PAYS	DENSITE
FRANCE	96.892139

**Exercice 3.4 :** *Voici cinq lignes de programme :*

```
300 DATA ALLEMAGNE,BELGIQUE,DANEMARK
310 DATA FRANCE,IRLANDE,ITALIE
320 DATA LUXEMBOURG,PAYS-BAS,ROYAUME-UNI
330 DATA 61.498,9.889,5.073,52.915,3.162
340 DATA 56.323,0.358,13.825,55.928
```

*Ecrire un programme (contenant les lignes ci-dessus) qui crée un tableau de chaînes de caractères et un tableau numérique et qui les garnit des données lues aux lignes 300 à 340.*



## **CHAPITRE 4**

### TROIS INSTRUCTIONS FONDAMENTALES

- GOTO et ON... GOTO
- IF... THEN
- FOR... TO... STEP.. NEXT

## LA DECOUVERTE DE L'ALICE ET DU MC/10

Le programme du chapitre III réduit le travail quotidien de l'hôtelier à une entrée de nombres au clavier de l'ordinateur qui se charge ensuite d'effectuer les calculs. Pour éviter tout risque d'erreur dans le résultat final, il est souhaitable de procéder à une vérification des nombres entrés (nous dirons plus brièvement, les entrées) avant leur traitement.

Le schéma ci-après donne la suite des actions nécessitées par une telle vérification :

- 1- Lire les entrées.
- 2- Répondre à la question : les entrées sont-elles correctes ?
- 3- Si la réponse est OUI : passer à la suite du traitement,  
si la réponse est NON : revenir au début du programme.

Nous connaissons déjà les instructions qui permettent de réaliser les actions n°1 et n°2. L'action n°3 demande à l'ordinateur d'analyser la réponse donnée et de "revenir en arrière" dans le programme en cas de réponse négative.

### L'INSTRUCTION GOTO

Voyons d'abord l'instruction qui permet de "revenir en arrière" et plus généralement d'"aller à" une ligne quelconque du programme : l'instruction GOTO (en anglais, GO TO signifie : ALLER A).

Ecrivez le programme suivant :

```
10 PRINT "LIGNE 10"  
20 PRINT "LIGNE 20"  
30 PRINT "LIGNE 30"  
40 END
```

```
tapez      RUN  
réponse   LIGNE 10  
           LIGNE 20  
           LIGNE 30
```

Modifions le programme en intercalant la ligne 15 :

```
tapez      15 GOTO 30  
puis      LIST
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

réponse

```
10 PRINT "LIGNE 10"  
15 GOTO 30  
20 PRINT "LIGNE 20"  
30 PRINT "LIGNE 30"  
40 END
```

tapez        RUN  
réponse     LIGNE 10  
            LIGNE 30

L'ordinateur a sauté directement de la ligne 15 à la ligne 30 en ignorant la ligne 20.

Maintenant, effaçons la ligne 15 :

en tapant    15  
            puis     LIST

```
10 PRINT "LIGNE 10"  
20 PRINT "LIGNE 20"  
30 PRINT "LIGNE 30"  
40 END
```

*RAPPEL : Pour supprimer une ligne d'un programme, il suffit d'en taper le numéro suivi, évidemment, d'un appui sur la touche ENTER .*

Ecrivons une nouvelle ligne :

tapez        25 GOTO 10  
            puis     LIST

```
10 PRINT "LIGNE 10"  
20 PRINT "LIGNE 20"  
25 GOTO 10  
30 PRINT "LIGNE 30"  
40 END
```

tapez        RUN

L'ordinateur exécute la ligne 10, la ligne 20 puis la ligne 25 qui le renvoie à la ligne 10 et ceci sans arrêt. Pour l'arrêter, tapez sur la touche BREAK.

réponse     BREAK IN 20 (ou 10, ou 25)

Ce message indique à quelle ligne s'est arrêtée (BREAK) l'exécution du programme.

Ces deux exemples montrent que l'*exécution d'une instruction GOTO* provoque un "*branchement*" (un saut) à la

## LA DECOUVERTE DE L'ALICE ET DU MC/10

*ligne indiquée.* Si la ligne n'existe pas, l'ordinateur affiche un message d'erreur.

Ajoutez au programme précédent la ligne suivante :

```
tapez      5 GOTO 50
puis tapez  RUN
réponse    ?UL ERROR IN 5
```

Ce message est l'abréviation de UNDEFINED LINE ERROR IN 5 et signifie que la ligne 50 n'est pas définie.

**GOTO** peut s'utiliser également en mode direct pour lancer l'exécution d'un programme à partir d'une ligne donnée :

```
tapez      NEW
puis le programme suivant :
```

```
10 A=23.7
20 B=54
30 PRINT "B/A=";B/A
40 END
```

```
tapez      RUN
réponse    B/A=2.27848101
Maintenant
tapez      GOTO 20
réponse    B/A=2.27848101
```

On peut également utiliser l'instruction RUN suivie d'un numéro de ligne, mais le résultat est différent :

```
tapez      RUN 20
réponse    B/A=
           ?/0 ERROR IN 30
```

Ce message d'erreur signifie que l'ordinateur ne peut exécuter une division par zéro (/0) et s'affiche ici parce que l'instruction RUN 20 remet toutes les variables et en particulier A à zéro.

**En revanche, l'exécution de l'instruction GOTO ne modifie pas la valeur des variables.**

L'introduction d'une instruction GOTO dans un programme ne modifie que le déroulement séquentiel de l'exécution (l'ordre croissant des instructions) du programme mais ce déroulement reste totalement déterminé.

Voyons comment l'instruction ON...GOTO permet à un programme d'avoir plusieurs déroulements différents.

*LA DECOUVERTE DE L'ALICE ET DU MC/10*

L'INSTRUCTION ON...GOTO

Tapez le programme suivant :

```
10 REM AFFICHAGE DU MENU
20 CLS
30 PRINT "ALICE CALCULATEUR"
40 PRINT
50 PRINT
60 PRINT "1-ADDITION"
70 PRINT "2-SOUSTRACTION"
80 PRINT "3 MULTIPLICATION"
90 PRINT "4-DIVISION"
100 PRINT
110 REM ENTREES
120 INPUT "TAPEZ LE NUMERO CORRESPONDANT A L'OPERATION
          A EFFECTUER";T
130 PRINT
140 INPUT "ENTREZ LES DEUX NOMBRES";A,B
150 ON T GOTO 170,210,250,290
160 GOTO 360
170 REM ADDITION
180 A#="+"
190 R=A+B
200 GOTO 330
210 REM SOUSTRACTION
220 A#="-"
230 R=A-B
240 GOTO 330
250 REM MULTIPLICATION
260 A#="*"
270 R=A*B
280 GOTO 330
290 REM DIVISION
300 A#="/"
310 R=A/B
320 GOTO 330
330 REM AFFICHAGE DU RESULTAT
340 PRINT
350 PRINT A;A#;B;"=";R
360 END
```

```
puis tapez      RUN
                ALICE CALCULATEUR
                1- ADDITION
                2- SOUSTRACTION
                3- MULTIPLICATION
                4- DIVISION
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

```
TAPEZ LE NUMERO CORRESPONDANT A
L'OPERATION A EFFECTUER ? 2
ENTREZ LES DEUX NOMBRES ? 129,34.7
129 - 34.7 = 94.3
OK
```

Les lignes 60 à 90 permettent l'affichage du "Menu" c'est-à-dire la liste des options possibles pour la suite de l'exécution du programme.

A la ligne 150, l'instruction ON T GOTO 170,210,250,290 utilise la valeur de T entrée à la ligne 120 de la façon suivante : le branchement s'effectue vers la ligne dont la place dans la liste est égale à la valeur de T. Par exemple, si T est égal à 3, l'exécution du programme continue à la ligne 250 (puisque 250 est la troisième valeur dans la liste 170,210,250,290 qui suit GOTO).

### Remarque :

- Si T est égal à 0 ou est supérieur au nombre de numéros de ligne de la liste de valeurs qui suit GOTO (en restant inférieur à 255), le programme ignore l'instruction de branchement et continue à la ligne suivante.
  - Si T est négatif ou supérieur à 256, l'ordinateur affiche le message d'erreur ?FC ERROR IN 150 indiquant que la valeur est incorrecte.
  - Si T est un nombre décimal, l'instruction ON T GOTO ne tient compte que de la partie entière de T.
- On peut également écrire une expression numérique, au lieu d'une variable, entre ON et GOTO.

Avec l'instruction GOTO, nous avons résolu une partie du problème posé par la vérification. Reste à donner à l'ordinateur le moyen de tester la réponse à la question "les entrées sont-elles correctes ?", pour se brancher à la ligne voulue.

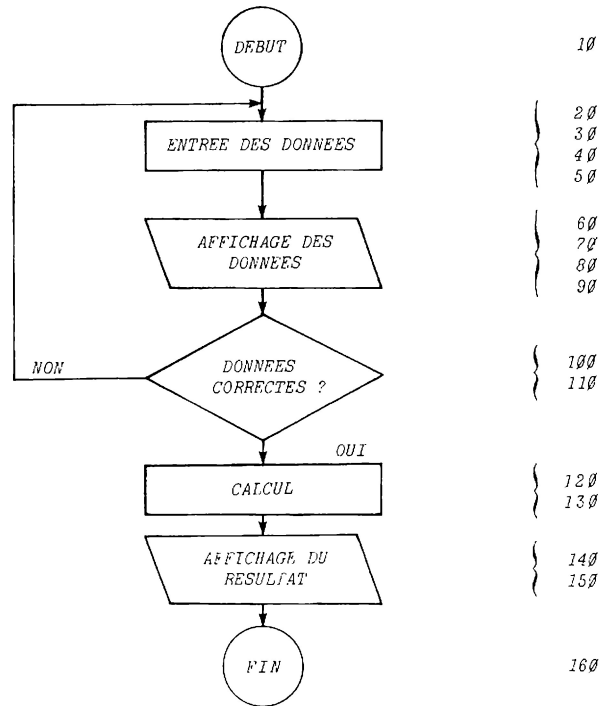
## L'INSTRUCTION IF...THEN

*C'est l'instruction IF...THEN qui permet de tester cette réponse.*

Voyons comment cette instruction résout le problème posé. On représente la suite des opérations grâce à un schéma appelé organigramme qui se lit à partir de la case DEBUT en suivant les flèches.

LA DECOUVERTE DE L'ALICE ET DU MC/10

En face de l'organigramme, figurent les numéros des lignes du programme correspondant à chaque étape :



Voici le programme correspondant à l'organigramme précédent :

```

10 REM PROGRAMME DE L'HOTELIER
20 REM ENTREE DES DONNEES
30 INPUT "CATEGORIE 1";N1
40 INPUT "CATEGORIE 2";N2
50 INPUT "CATEGORIE 3";N3
60 REM VERIFICATION
70 PRINT "CATEGORIE 1 :";N1
80 PRINT "CATEGORIE 2 :";N2
90 PRINT "CATEGORIE 3 :";N3
100 INPUT "NOMBRES CORRECTS (OUI/NON) ";R$
110 IF R$<>"OUI" THEN GOTO 20
120 REM CALCUL
130 T=N1*125+N2*85+N3*50
140 REM AFFICHAGE DU RESULTAT
150 PRINT "TOTAL =";T
160 END
  
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

### Explications

Ligne 100 : dans le texte de l'instruction INPUT, on indique à l'utilisateur les deux seules réponses qu'il peut donner.

Ligne 110 : l'exécution de l'instruction IF..THEN.. s'effectue de la façon suivante :

- SI R\$ <> "OUI" (R\$ différent de "OUI"), ALORS l'exécution du programme reprend à la ligne 20.
- SI R\$ = "OUI", ALORS l'exécution du programme continue à la ligne suivante.

Maintenant, lancez l'exécution du programme en essayant plusieurs réponses :

```
CATEGORIE 1? 3
CATEGORIE 2? 15
CATEGORIE 3? 9
CATEGORIE 1: 3
CATEGORIE 2: 15
CATEGORIE 3: 9
NOMBRES CORRECTS (OUI/NON)? NON
CATEGORIE 1? 5
CATEGORIE 2? 13
CATEGORIE 3? 9
CATEGORIE 1: 5
CATEGORIE 2: 13
CATEGORIE 3: 9
NOMBRES CORRECTS (OUI/NON)? OUI
TOTAL = 2180
OK
```

En anglais, IF..THEN.. signifie SI..ALORS.. Dans un programme, on rencontre cette instruction sous la forme : IF (proposition) THEN (instruction). Elle fonctionne de la manière suivante :

- *Si la proposition écrite entre IF et THEN est vraie, l'instruction écrite après THEN est exécutée.*
- *Si la proposition est fausse, l'exécution du programme continue à la ligne suivante.*

Voici quelques exemples d'utilisation de cette instruction (ce qui suit n'est pas un programme) :

```
11 IF A#=B# THEN GOTO 110
19 IF A#=B# THEN 110
24 IF A#=B# GOTO 110
40 IF A>B THEN A=B
55 IF B THEN A=100
60 IF NOT B THEN A=105
66 IF A=36 AND B=6 THEN PRINT "EXACT"
72 IF D#="OUI" GOTO 110
74 IF C3=3 AND A#="NON" THEN F=1
82 IF (C=3 OR A=1) AND (B=1 AND C=3) THEN T=0
```



## LA DECOUVERTE DE L'ALICE ET DU MC/10

### Explications

Lignes 19,24 et 27 : ces trois lignes sont équivalentes.

Ligne 55 : la proposition "B" est vraie si  $B <> \emptyset$ , fausse si  $B = \emptyset$ .

Ligne 56 : la proposition "NOT B" est vraie si  $B = \emptyset$ , fausse si  $B <> \emptyset$ .

### L'INSTRUCTION FOR..TO..STEP.. NEXT

Aux lignes 30, 40 et 50 du programme de l'hôtelier (voir ci-dessus au paragraphe concernant l'instruction IF..THEN), la même opération est programmée trois fois : il en est de même aux lignes 70, 80 et 90. Avec un hôtel comprenant 10 catégories, ces deux opérations devraient l'être chacune dix fois.

Le programme suivant évite cette répétition pour cinq catégories :

```
10 DIM N(5)
20 REM BOUCLE D'ENTREE
30 FOR I=1 TO 5 STEP 1
40 PRINT "CATEGORIE";I;
50 INPUT N(I)
60 NEXT I
70 PRINT
80 REM VERIFICATION
90 FOR I=1 TO 5
100 PRINT "CATEGORIE";I;": ";N(I)
110 NEXT
120 INPUT "NOMBRES CORRECTS (OUI/NON) ";R$
130 IF R$<>"OUI" THEN 20
140 REM CALCUL
150 T=N(1)*150+N(2)*125+N(3)*100+N(4)*75+N(5)*50
160 REM AFFICHAGE
170 PRINT "TOTAL =";T
180 END
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

```
tapez      RUN
           CATEGORIE 1? 2
           CATEGORIE 2? 0
           CATEGORIE 3? 6
           CATEGORIE 4? 19
           CATEGORIE 5? 6

           CATEGORIE 1: 2
           CATEGORIE 2: 0
           CATEGORIE 3: 6
           CATEGORIE 4: 19
           CATEGORIE 5: 6
           NOMBRES CORRECTS (OUI/NON) ? OUI
           total = 2625
           OK
```

### Explications

Les lignes 30 à 60 constituent une "boucle" ainsi que les lignes 90 à 110.

Examinons la première boucle :

- A la ligne 30, on trouve l'instruction :  
FOR I=1 TO 5 STEP 1 (DE I=1 A 5 PAS 1)
- et à la ligne 60, l'instruction :  
NEXT I (I SUIVANT)

Ces deux instructions définissent la boucle qui fonctionne de la manière suivante : l'exécution de la ligne 30 met le "compteur" I à sa valeur initiale (ici 1) puis les lignes suivantes sont exécutées normalement.

A la ligne 60, le compteur I est incrémenté (augmenté) du PAS (ici 1). Si sa valeur est inférieure ou égale à la valeur finale (ici 5), il y a branchement à la ligne 40, sinon l'exécution du programme continue par l'instruction suivante.

### **Remarques :**

A la ligne 90, première ligne de la seconde boucle, le pas n'a pas été précisé. Dans ce cas, il est pris égal à 1 par défaut. Il aurait donc pu également être omis à la ligne 30.

A la ligne 110, la variable I de la boucle a été omise après l'instruction NEXT. Dans ce cas l'incrémentation et le test de sortie se feront sur la boucle dont l'instruction FOR..TO..STEP est la plus proche, qui précède. De la même façon à la ligne 60, I aurait pu être omis.

### LA DECOUVERTE DE L'ALICE ET DU MC/10

Les valeurs initiale et finale du compteur ainsi que le pas peuvent être des nombres positifs ou négatifs, ou définies par des expressions numériques.

**A l'intérieur d'une boucle, il est possible d'écrire une ou plusieurs boucles. On parle alors de boucles imbriquées.**

En voici un exemple :

```
10 FOR I=1 TO 10
20 FOR J=1 TO 7
30 PRINT "I =",I,"J =",J
40 NEXT J
50 NEXT I
60 END
```

Il est possible de cette manière d'imbriquer jusqu'à 25 boucles.

Voici un contre-exemple :

```
10 FOR I=1 TO 10
20 FOR J=1 TO 7
30 PRINT "I =",I,"J =",J
40 NEXT I
50 NEXT J
60 END
```

Ici, les deux boucles se chevauchent et le programme ne peut pas s'exécuter normalement. L'ordinateur affiche alors le message :

```
?NF ERROR IN 50
```

qui signifie qu'il a rencontré une instruction NEXT (N) à laquelle ne correspond pas d'instruction FOR..TO (F).

**Remarque :** Dans le programme correct des boucles imbriquées, il est possible de remplacer les instructions NEXT J et NEXT I des lignes 40 et 50 par une seule instruction NEXT J,I où les variables I et J doivent obligatoirement être écrites dans l'ordre J,I.

```
10 FOR I=1 TO 10
20 FOR J=1 TO 7
30 PRINT "I =",I,"J =",J
40 NEXT J,I
60 END
```

*LA DECOUVERTE DE L'ALICE ET DU MC/10*

EXERCICES SUR LE CHAPITRE IV

**Exercice 4.1 :** *Ecrire un programme réalisant l'entrée de deux nombres A et B et leur comparaison ainsi que l'affichage du résultat de cette comparaison.*

**Exercice 4.2 :** *Reprendre le programme de l'exercice 3.4 en utilisant cette fois l'instruction FOR..TO..STEP.*

**Exercice 4.3 :** *A partir des données des programmes précédents, écrire un programme affichant les pays dans l'ordre croissant de leurs populations avec alignement des nombres sur le point décimal.*

**Exercice 4.4 :** *Ecrire un programme qui permet :*  
- *d'entrer la longueur d'une liste de mots*  
- *d'entrer les mots de cette liste*  
- *de les ranger par ordre alphabétique*  
- *d'afficher la liste ordonnée sur l'écran.*

## **CHAPITRE 5**

### LES SOUS-PROGRAMMES

- GOSUB
- RETURN
- ON... GOSUB

### LA DECOUVERTE DE L'ALICE ET DU MC/10

Enhardi par la réussite de son programme de gestion des chambres de l'hôtel, l'hôtelier souhaite compléter celui-ci par la gestion du restaurant qui possède également cinq catégories de menus.

Ceci entraîne de nouveau une entrée de données, une vérification et un calcul. Pour écrire le programme complet de gestion des chambres et des repas, il suffit donc de rajouter des instructions semblables aux précédentes, excepté pour les calculs, et l'on obtient le programme suivant :

```
10 DIM N(5)
20 PRINT "CHAMBRES"
30 FOR I=1 TO 5
40 PRINT "CATEGORIE";I;
50 INPUT N(I)
60 NEXT I
70 PRINT
80 FOR I=1 TO 5
90 PRINT "CATEGORIE";I;":";N(I)
100 NEXT I
110 INPUT "EXACT (OUI/NON) ";R$
120 IF R$="NON" THEN 20
130 T1=N(1)*250+N(2)*200+N(3)*150+N(4)*100+N(5)*50
140 PRINT "REPAS"
150 FOR J=1 TO 5
160 PRINT "CATEGORIE";J;
170 INPUT N(J)
180 NEXT J
190 PRINT
200 FOR I=1 TO 5
210 PRINT "CATEGORIE";I;":";N(I)
220 NEXT I
230 INPUT "EXACT (OUI/NON) ";R$
240 IF R$="NON" THEN 140
250 T2=N(1)*150+N(2)*125+N(3)*100+N(4)*75+N(5)*50
260 PRINT "RESULTATS : "
265 PRINT
270 PRINT "CHAMBRES :";T1
280 PRINT "REPAS      :";T2
290 PRINT "-----"
300 PRINT "TOTAL      :";T1+T2
310 END
```

Vous constatez que les lignes 150 à 240 sont identiques aux lignes 30 à 120. Elles sont constituées du même groupe d'instructions effectuant une tâche précise.

*LA DECOUVERTE DE L'ALICE ET DU MC/10*

On peut extraire ce groupe d'instructions du programme principal pour en faire un sous-programme qui sera appelé chaque fois qu'il faudra effectuer cette tâche. On écrit alors le programme suivant.

tapez :

```
10 DIM N(5)
20 PRINT "CHAMBRES"
30 GOSUB 200
40 T1=N(1)*250+N(2)*200+N(3)*150+N(4)*100+N(5)*50
50 PRINT "REPAS"
60 GOSUB 200
70 T2=N(1)*150+N(2)*125+N(3)*100+N(4)*75+N(5)*50
80 PRINT "RESULTATS :"
90 PRINT
100 PRINT "CHAMBRES :")T1
110 PRINT "REPAS      :")T2
120 PRINT "-----"
130 PRINT "TOTAL      :")T1+T2
140 END
200 REM
210 REM SOUS PROGRAMME
220 REM
230 FOR I=1 TO 5
240 PRINT "CATEGORIE")I)
250 INPUT N(I)
260 NEXT I
270 PRINT
280 FOR I=1 TO 5
290 PRINT "CATEGORIE")I) " )N(I)
300 NEXT I
310 INPUT "EXACT (OUI/NON)"):R#
320 IF R#="NON" THEN 200
330 RETURN
```

puis	tapez	RUN
	réponse	CHAMBRES
		CATEGORIE 1? 2
		CATEGORIE 2? 5
		CATEGORIE 3? 4
		CATEGORIE 4? 8
		CATEGORIE 5? 7
		CATEGORIE 1: 2
		CATEGORIE 2: 5
		CATEGORIE 3: 4
		CATEGORIE 4: 8
		CATEGORIE 5: 7
		EXACT (OUI/NON) ? OUI

## LA DECOUVERTE DE L'ALICE ET DU MC/10

```
REPAS
CATEGORIE 1? 0
CATEGORIE 2? 2
CATEGORIE 3? 5
CATEGORIE 4? 1
CATEGORIE 5? 1

CATEGORIE 1; 0
CATEGORIE 2: 2
CATEGORIE 3: 5
CATEGORIE 4: 1
CATEGORIE 5: 1
EXACT (OUI/NON) ? OUI

RESULTATS
CHAMBRES : 3250
REPAS    : 875
-----
TOTAL    : 4125
```

### Explications

Les lignes 10 à 140 constituent le programme principal, terminé par l'instruction END.

Les lignes 200 à 330 constituent le sous-programme, terminé par l'instruction RETURN.

Aux lignes 30, 60 et 330, deux nouvelles instructions, GOSUB et RETURN, apparaissent.

### LES INSTRUCTIONS GOSUB ET RETURN

*GOSUB, abréviation de GO SUBroutine, signifie ALLER au SOUS-programme. L'instruction GOSUB 200 provoque le branchement à la ligne 200 puis l'exécution des lignes suivantes jusqu'à la ligne 330 où l'instruction RETURN indique la fin du sous-programme et provoque un RETOUR au programme principal.*

L'exécution reprend alors par l'instruction qui suit immédiatement l'instruction qui a provoqué l'appel du sous-programme.

*Un sous-programme peut également appeler un autre sous-programme qui lui-même peut en appeler d'autres.*

Remarque : l'instruction GOSUB peut également s'écrire GO SUB.



## LA DECOUVERTE DE L'ALICE ET DU MC/10

### L'INSTRUCTION ON..GOSUB

Le fonctionnement de cette instruction est tout à fait analogue à celui de ON..GOTO. Il permet donc de sélectionner l'appel d'un sous-programme parmi plusieurs, suivant la valeur de la variable ou de l'expression écrite entre ON et GOSUB.

Voici un exemple d'utilisation :

```
10 CLS
20 PRINT "-----"
30 PRINT "ALICE CALCULATEUR"
40 PRINT
50 PRINT "1  ADDITION"
60 PRINT "2  SOUSTRACTION"
70 PRINT "3  MULTIPLICATION"
80 PRINT "4  DIVISION"
90 PRINT
100 PRINT "0  FIN DU PROGRAMME"
110 PRINT
120 INPUT "TAPEZ LE NUMERO DE L'OPERATION A EFFECTUER":T
130 PRINT
140 IF T<=0 OR T>4 THEN 190
150 INPUT "ENTREZ LES DEUX NOMBRES SEPARES PAR UNE VIRGULE ":A,B
160 ON T GOSUB 300,400,500,600
170 GOSUB 700
180 GOTO 20
190 PRINT "FIN DU PROGRAMME"
200 END
300 REM
310 REM SOUS-PROGRAMME D'ADDITION
320 REM
330 A#= "+"
340 R=A+B
350 RETURN
400 REM
410 REM SOUS-PROGRAMME DE SOUSTRACTION
420 REM
430 A#="-"
440 R=A-B
450 RETURN
500 REM
510 REM SOUS-PROGRAMME DE MULTIPLICATION
520 REM
530 A#="*"
540 R=A*B
550 RETURN
```

### LA DECOUVERTE DE L'ALICE ET DU MC/10

```
600 REM
610 REM SOUS-PROGRAMME DE DIVISION
620 REM
630 A$="/"
640 R=A/B
650 RETURN
700 REM
710 REM SOUS-PROGRAMME D'AFFICHAGE DU RESULTAT
720 REM
730 PRINT
740 PRINT A;A$;B;" = ";R
750 PRINT
760 RETURN
```

#### Explications

Les lignes 50 à 100 constituent un "Menu" qui donne les réponses possibles à la question posée par l'instruction de la ligne 120.

Les lignes 10 à 200 constituent le programme principal.

Les lignes 300 à 350 , 400 à 450 , 500 à 550 , 600 à 650 et 700 à 760 constituent cinq sous-programmes.

**LA DECOUVERTE DE L'ALICE ET DU MC/10**

**EXERCICE SUR LE CHAPITRE V**

**Exercice 5.1 :** *Ecrire un programme qui permet de classer les pays de l'Europe, suivant différents critères : population, superficie et densité.*

**Indications :** *Utiliser les données des chapitres précédents. Ecrire un sous-programme de lecture et un sous-programme de classement.*

**Exercice 5.2 :** *Ecrire un programme qui permet la mise à jour d'un répertoire de personnes enregistré sur cassette. Ce programme doit afficher et traiter le menu suivant :*

- 1- *Lecture du fichier sur cassette.*
- 2- *Affichage du répertoire sur l'écran.*
- 3- *Insertion d'une personne dans le fichier.*
- 4- *Suppression d'une personne dans le fichier.*
- 5- *Ecriture du fichier sur cassette.*
- Ø- *Fin.*

**Indications :** *Pour les éléments 1 et 5 du menu proposé, utiliser les instructions CLOAD\* et CSAVE\*.*

## **CHAPITRE 6**

# INSTRUCTION POUR LA MISE AU POINT DES PROGRAMMES

- Plusieurs instructions sur la même ligne
- STOP et CONT
- Interruption par clavier  
BREAK, SHIFT @
- La mémoire d'Alice
  - PEEK
  - POKE
  - MEM
  - VARPTR

## LA DECOUVERTE DE L'ALICE ET DU MC/10

Dans ce chapitre, nous examinons les possibilités offertes par ALICE pour l'écriture et la mise au point des programmes.

### PLUSIEURS INSTRUCTIONS SUR UNE MEME LIGNE

Jusqu'à présent, nous avons toujours écrit une instruction par ligne, ce qui a facilité la lecture des programmes. Mais il est possible d'écrire plusieurs instructions dans une même ligne en les séparant par le signe ":" (deux points).

Tapez le programme suivant :

```
INPUT "A (1-5)":P
IF A<1 AND A<2 THEN PRINT "A EST DIFFERENT DE 1
                                ET 2" END
20 ON H GOSUB 100,200
40 INPUT "VOULEZ-VOUS CONTINUER (OUI/NON)":R#
50 IF R#="OUI" THEN G0
60 END
100 PRINT "UN":RETURN
200 PRINT "DEUX":RETURN
```

puis	tapez	RUN
	réponse	A (1-5) ?
	tapez	1
	réponse	UN
		VOULEZ-VOUS CONTINUER (OUI/NON) ?
	tapez	OUI
	réponse	A (1-5) ?
	tapez	3
	réponse	A EST DIFFERENT DE 1 ET 2

Vous constatez que l'écriture des lignes 20, 100, 200 ne change rien à l'exécution du programme. Cependant elle permet d'accélérer son exécution et de réduire sa place en mémoire. En contrepartie, le listage (ou listing) sera plus difficile à déchiffrer.

D'autre part, si vous prévoyez un branchement à une instruction, celle-ci doit être nécessairement au début d'une ligne car les branchements s'effectuent obligatoirement à un numéro de ligne.

Enfin, lors de l'utilisation de l'instruction IF..THEN (ligne 20), si la proposition testée par IF est fautive, l'exécution du programme passe à la ligne suivante

## LA DECOUVERTE DE L'ALICE ET DU MC/10

(ligne 30) ; les différentes instructions séparées par ":" qui suivent THEN ne sont, elles, exécutées que si la proposition testée est vraie.

### LES INSTRUCTIONS STOP ET CONT

L'instruction **STOP** met fin à l'exécution du programme en affichant un message indiquant le numéro de ligne d'arrêt. Elle est surtout utile lors de la mise au point d'un programme.

tapez NEW  
puis le programme :

```
10 INPUT "A (1-5)";A
20 ON A GOTO 100,200,300,400,500
30 END
100 STOP
200 STOP
300 STOP
400 STOP
500 STOP
```

```
tapez RUN
réponse A(1-5)?
tapez 2
réponse BREAK IN 200

tapez RUN
réponse A(1-5)?
tapez 5
réponse BREAK IN 500
```

L'instruction **CONT** permet la reprise de l'exécution d'un programme interrompu, par exemple par une instruction STOP. Modifiez le programme précédent pour obtenir le programme suivant :

```
10 INPUT "A (1-5)";A
20 ON A GOTO 100,200,300,400,500
30 END
100 STOP:GOTO 10
200 STOP:GOTO 10
300 STOP
400 STOP
410 GOTO 10
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

```
500 STOP
510 GOTO 100
```

```
puis tapez      RUN
réponse        A (1-5)?
tapez          1
réponse        BREAK IN 100

puis tapez      CONT
réponse        A (1-5)?
tapez          3
réponse        BREAK IN 300

puis tapez      CONT
réponse        BREAK IN 400
```

Après la première instruction CONT, l'exécution du programme reprend à l'instruction GOTO 10 de la ligne 100. Ainsi vous constatez que l'exécution du programme reprend à l'instruction qui suit STOP.

### INTERRUPTIONS PAR CLAVIER

Il est possible d'interrompre un programme avant la fin normale de son exécution, en utilisant soit la touche **BREAK** soit **SHIFT** et **@** simultanément.

```
tapez          NEW
et le programme suivant :
```

```
10 FOR I=1 TO 10000000
20 PRINT "IT EST ECH. A " I
30 PRINT
40 NEXT I
50 END
```

```
puis tapez      RUN
Ce programme mettrait plusieurs jours pour arriver à son terme !
```

#### Arrêt par BREAK

Pour l'arrêter, tapez sur la touche **BREAK**. Le programme s'arrête seulement après la fin de l'exécution de l'instruction en cours, avec l'affichage d'un message d'interruption.

## LA DECOUVERTE DE L'ALICE ET DU MC/10

Il est possible de relancer l'exécution du programme à l'aide de l'instruction CONT.

Le défilement d'un listage peut également être interrompu par `BREAK`.

### Arrêt par SHIFT @

Il est possible d'arrêter le programme en appuyant simultanément sur les touches `SHIFT` et `@`. Dans ce cas, aucun message n'apparaît sur l'écran et l'exécution reprend dès la pression d'une touche quelconque, à l'exception de la touche `BREAK`.

Le défilement d'un listage peut également être interrompu par les touches `SHIFT @` puis repris par la pression d'une touche quelconque.

**Remarque :** Il est toujours possible d'interrompre l'exécution d'un programme ou le défilement d'un listage en tapant sur le bouton INIT situé à l'arrière de l'appareil.

Cette interruption brutale que vous pouvez provoquer lors d'une opération quelconque (exécution d'un programme, entrée au clavier, etc.) remet ALICE dans l'état initial, sans toutefois provoquer l'effacement du programme qui pourrait se trouver en mémoire.

## LA MEMOIRE D'ALICE

La mémoire d'ALICE est constituée d'un certain nombre d'emplacements appelés cases-mémoires ; chacune d'elles peut conserver un nombre compris entre 0 et 255. Ces nombres sont utilisés pour représenter les caractères alphanumériques (cf Annexe 3 : code ASCII) ainsi que les valeurs numériques. Les quatre instructions qui suivent permettent de réaliser directement certaines opérations en mémoire.

**L'INSTRUCTION PEEK** permet de lire le contenu d'une case-mémoire dont on précise l'adresse :

tapez            PRINT PEEK (190000)

La réponse est un nombre compris entre 0 et 255. Sa valeur exacte dépend de l'état du micro-ordinateur au moment où l'instruction est exécutée.



## LA DECOUVERTE DE L'ALICE ET DU MC/10

Le nombre obtenu peut être affecté à une variable A en écrivant l'instruction :

```
A = PEEK (19000)
```

**L'INSTRUCTION POKE** permet de modifier le contenu d'une position mémoire dont on précise l'adresse et le nouveau contenu.

```
tapez      POKE 20000,113
```

Cette instruction place la valeur 113 dans la position mémoire 0. On peut le vérifier immédiatement grâce à l'instruction PEEK.

```
tapez      PRINT PEEK (20000)
réponse    113
```

Les deux instructions PEEK et POKE ont une très grande variété d'applications dépendant des cases-mémoires qu'elles utilisent.

**L'INSTRUCTION MEM** permet de connaître la place disponible en mémoire vive.

```
tapez      ? MEM
```

La réponse est un nombre qui dépend de la capacité de votre micro-ordinateur et de son état et représente le nombre de cases-mémoires disponibles pour l'écriture ou l'utilisation du programme et des variables numériques et alphanumériques.

**L'INSTRUCTION VARPTR** permet de connaître l'adresse en mémoire de la première case-mémoire d'une variable numérique ou alphanumérique.

```
tapez      CLEAR
           A = 2
           B = 1
           PRINT VARPTR(A), VARPTR(B).
```

La réponse est deux nombres dont la différence est égale à 7

*LA DECOUVERTE DE L'ALICE ET DU MC/10*

EXERCICE SUR LE CHAPITRE VI

**Exercice 6.1 :** *Ecrire un programme qui affiche le contenu de la mémoire d'ALICE dont l'adresse est comprise entre 49221 et 49479. Cette zone-mémoire correspond à l'emplacement de la table des instructions exécutables par ALICE.*

## **CHAPITRE 7**

# LES FONCTIONS NUMERIQUES

- RND
- ABS
- SGN
- INT
- SQR
- SIN, COS et TAN
- EXP et LOG

## LA DECOUVERTE DE L'ALICE ET DU MC/10

### RND : NOMBRE AU HASARD

RND est l'abréviation de RaNDom qui signifie "hasard". RND est utilisé essentiellement pour introduire un élément aléatoire dans les programmes de jeux et de graphisme en donnant la possibilité d'utiliser des nombres tirés au hasard.

```
tapez          PRINT RND(10)
```

La réponse sera un nombre entier compris entre 1 et 10 (inclus). Essayez plusieurs fois cette instruction pour constater que le nombre est chaque fois différent. Cependant, chaque fois qu'ALICE est remis dans son état initial (en éteignant l'ordinateur), la séquence des nombres fournis par RND est identique.

Il est également possible d'obtenir des nombres décimaux aléatoires, compris entre 0 (inclus) et 1 (non inclus) :

```
tapez          PRINT RND(0)
```

RND peut être utilisée dans toute expression numérique comme le montre l'exemple ci-dessous :

```
10 FOR I=1 TO 10
20 A=RND(100)+10^RND(3)
30 PRINT A
40 NEXT I
50 END
```

### ABS : VALEUR ABSOLUE

*Cette instruction fait correspondre à tout nombre ou expression numérique sa valeur ABSolue.*

#### *Exemple*

```
tapez          PRINT ABS(-5)
réponse       5
tapez          PRINT ABS(38.275)
réponse       38.275
tapez          PRINT ABS(0)
réponse       0
tapez          PRINT ABS(-4.9+1.34)
réponse       3.56
tapez          T=25.002
tapez          PRINT ABS(3*T-103)
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

réponse 27.994

ABS peut être utilisée dans toute expression numérique.

### SGN : SIGNE

*SGN fait correspondre à tout nombre ou expression numérique :*

-1 si sa valeur est négative

1 si sa valeur est positive

Ø si sa valeur est nulle.

#### *Exemple*

```
tapez      PRINT SGN(-5)
réponse    -1
tapez      PRINT SGN(38.275)
réponse    1
tapez      PRINT SGN(Ø)
réponse    Ø
tapez      XN=76.152
tapez      PRINT SGN(3-2*XN)
réponse    -1
```

SGN peut être utilisée dans toute expression numérique.

### INT : VALEUR ENTIERE

*INT fait correspondre à tout nombre ou expression numérique l'entier immédiatement inférieur, soit sa partie entière pour les nombres positifs et sa partie entière plus un pour les nombres négatifs.*

```
tapez      PRINT INT(3.99999)
réponse    3
tapez      PRINT INT(3.ØØØØ1)
réponse    3
tapez      PRINT INT(-5.1)
réponse    -6
```

INT peut être utilisée dans toute expression numérique.

## LA DECOUVERTE DE L'ALICE ET DU MC/10

### SQR : RACINE CARREE

*SQR est l'abréviation de Square Root qui signifie Racine Carrée.*

Cette instruction fait correspondre à tout nombre ou expression numérique positive sa racine carrée :

```
tapez      PRINT SQR(16)
réponse    4
tapez      PRINT SQR(65536)
réponse    256
tapez      PRINT SQR(56.25)
réponse    7.5
```

SQR peut être utilisée dans toute expression numérique.

### SIN, COS ET TAN : SINus, COSinus et TANgente

*Ces instructions font correspondre à tout angle donné en radian la valeur de la fonction trigonométrique correspondante :*

```
tapez      PI=3.141592654
tapez      PRINT SIN(PI/2)
réponse    1
tapez      PRINT SIN(PI)
réponse    0
tapez      PRINT COS(PI/4)
réponse    .707106781
tapez      PRINT COS(PI)
réponse    -1
tapez      PRINT TAN(PI)
réponse    0
```

SIN, COS et TAN peuvent être utilisées dans toute expression numérique.

### EXP ET LOG : EXPonentielle et LOGarithme népérien

*Ces instructions font correspondre à tout nombre ou expression numérique la valeur de ces fonctions calculée pour ce nombre ou cette expression :*

```
tapez      PRINT EXP(1)
réponse    2.71828183
```

*LA DECOUVERTE DE L'ALICE ET DU MC/10*

tapez        PRINT LOG(2.71828183)  
réponse     1

**EXP** et **LOG** peuvent être utilisées dans toute expression numérique.

LA DECOUVERTE DE L'ALICE ET DU MC/10

EXERCICE SUR LE CHAPITRE VII

**Exercice 7.1 :** Tracer, sur l'écran, la courbe de la fonction  $f(x) = \text{Sin } Cx$ .

**Indications**

- l'axe des  $x$  est supposé vertical et au milieu de l'écran .

**Exercice 7.2 :** Ecrire un programme qui calcule le quotient et le reste de la division de deux entiers positifs.

**Exercice 7.3 :** La fonction LOG peut être utilisée pour aligner, sur le point décimal, une colonne de nombres. Appliquer cette technique au programme donné comme solution de l'exercice 4.3.

**Exercice 7.4 :** Ecrire un programme permettant de calculer les racines réelles d'une équation du second degré.



## **CHAPITRE 8**

### LE TRAITEMENT DES CHAINES DE CARACTERES

- LEN
- LEFT\$
- RIGHT\$
- MID\$
- CONCATE NATION
- STR\$
- VAL
- ASC
- CHR\$

## LA DECOUVERTE DE L'ALICE ET DU MC/10

Les ordinateurs conçus à l'origine pour effectuer des opérations sur les nombres ont été très vite utilisés dans le traitement de texte.

Le Basic d'ALICE permet de manipuler les chaînes de caractères et les variables alphanumériques.

### LEN : LONGUEUR D'UNE CHAÎNE DE CARACTÈRES

*L'instruction LEN donne le nombre de caractères d'une chaîne (sa longueur). LEN est donc une fonction qui, à une chaîne de caractères, fait correspondre un nombre. LEN est l'abréviation de LENgth (longueur).*

#### *Exemple*

```
tapez      PRINT LEN("ABCDE")
réponse    5
tapez      A$="POURQUOI PAS ?"
tapez      PRINT LEN(A$)
réponse    14
```

La longueur de la chaîne de caractères vide "" est 0 :

```
tapez      PRINT LEN("")
réponse    0
```

Le programme suivant montre comment on peut utiliser cette instruction pour cadrer un titre (de longueur inférieure à 33 caractères) au milieu d'une ligne sur l'écran :

```
10 INPUT "TITRE  ";T$
20 CLS
30 PRINT TAB(16-LEN(T$)/2) T$
40 END
```

C'est la ligne 30 qui réalise le cadrage du titre entré à la ligne 10 : LEN(T\$) donne le nombre de caractères du titre et l'expression 16-LEN(T\$)/2 donne la position d'affichage du premier caractère du titre. (Rappelons qu'une ligne comprend 32 caractères).

### LES FONCTIONS LEFT\$, RIGHT\$ ET MID\$

(LEFT signifie Gauche, RIGHT droite et MIDdle milieu)

## LA DECOUVERTE DE L'ALICE ET DU MC/10

Ces trois fonctions permettent de créer de nouvelles chaînes de caractères à partir d'une chaîne de caractères donnée :

```
tapez      PRINT LEFT$( "ALICE", 3)
réponse    ALI
tapez      A$ = "PIQUE, COEUR, CARREAU, TREFLE"
tapez      PRINT LEFT$(A$, 5)
réponse    PIQUE
tapez      B$=LEFT$(A$, 10)
tapez      PRINT B$
réponse    PIQUE, COEU
```

**LEFT\$(A\$,n) donne la chaîne de caractères constituée des n premiers caractères de A\$.**

De même, **RIGHT\$(A\$,n) donne la chaîne de caractères constituée des n derniers caractères de A\$ :**

```
tapez      PRINT RIGHT$(A$, 6)
réponse    TREFLE
tapez      D=3
tapez      PRINT RIGHT$(A$, D)
réponse    FLE
```

maintenant

```
tapez      PRINT MID$(A$, 13, 7)
réponse    CARREAU
tapez      H=3
tapez      K=4
tapez      PRINT MID$("1234567890", H, K)
réponse    3456
```

**La fonction MID\$(A\$,i,n) donne la chaîne de caractères constituée de n caractères de la chaîne A\$ pris à partir du ième caractère.**

**Remarque :** Si n est plus grand (tout en restant inférieur à 255) que le nombre de caractères à la droite du ième, MID\$ donne la fin de la chaîne de caractères :

```
tapez      PRINT MID$("1234567890", 3, 100)
réponse    34567890
```

**MID\$(A\$,i) donne également la chaîne constituée de la fin de A\$ à partir du ième caractère :**

```
tapez      PRINT MID$("1234567890", 3)
réponse    34567890
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

### LA CONCATENATION

Les instructions précédentes permettent, à partir d'une chaîne de caractères donnée, d'en fabriquer de nouvelles en les tronquant.

*La concaténation est l'opération qui permet de fabriquer des chaînes de caractères à partir de plusieurs chaînes de caractères données ; elle utilise le symbole + .*

```
tapez      PRINT "MICRO"+"ORDINATEUR"
réponse    MICROORDINATEUR
tapez      A$ = "MICROORDINATEUR"
puis       B$ = " "
et         C$ = "UN"+B$+A$+B$+"EST"+B$+"UN"+B$+
           "PETIT"+B$+MID$(A$,6)
et         PRINT C$
réponse    UN MICROORDINATEUR EST UN PETIT ORDI-
           NATEUR
```

Le programme suivant illustre le fonctionnement des cinq opérations sur les chaînes de caractères (LEN,LEFT\$,RIGHT\$,MID\$,+) que nous venons de voir.

```
10 CLS
20 INPUT "VITESSE (10-10000) ";VIT:NV=10000/VIT
30 INPUT "DONNEZ UN MOT":A$:L=LEN(A$)
40 FOR I=1 TO L
50 GOSUB 200
60 PRINT LEFT$(A$,I)
70 NEXT:PRINT
80 FOR I=1 TO L
90 GOSUB 200
100 PRINT MID$(A$,I,1)
110 NEXT:PRINT
120 FOR I=1 TO L
130 GOSUB 200
140 PRINT RIGHT$(A$,I)
150 NEXT:PRINT
160 FOR I=1 TO 32-L
170 GOSUB 200
175 A$=" "+A$
180 PRINT A$
185 NEXT
190 END
200 FOR K=1 TO NV:NEXT:RETURN
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

**Remarque :** Le sous-programme ligne 200 constitue une boucle de temporisation permettant de ralentir le défilement des messages affichés sur l'écran.

### L'INSTRUCTION STR\$

**STR** est l'abréviation de **STRing** qui signifie chaîne (de caractères).

**STR\$** convertit un nombre en une chaîne de caractères qui sera constituée des chiffres et éventuellement des autres caractères utilisés dans la représentation de ce nombre (**.,E,+,-**).

```
tapez      J$=STR$(137.19)
puis       PRINT J$
réponse    137.19
tapez      J$=STR$(SIN(LOG(4*25.2)))
tapez      PRINT J$
réponse    -.995078699
tapez      PRINT LEN(J$)
réponse    11
tapez      PRINT MID$(J$,3,5)
réponse    99507
```

Ainsi, **STR\$** facilite dans certains cas (tabulation, par exemple) la manipulation des nombres en permettant l'utilisation de toutes les instructions de traitement des chaînes de caractères.

Par exemple, le programme suivant donne immédiatement le nombre de chiffres de tout entier compris entre -999999999 et +999999999.

```
10 INPUT "DONNEZ UN NOMBRE ENTIER " : A
20 PRINT "NOMBRE DE CHIFFRES : " ; LEN(STR$(A))-1
30 PRINT
40 GOTO 10
```

### L'INSTRUCTION VAL

**VAL** est l'abréviation de **VALue** qui signifie valeur.

**VAL** effectue l'opération inverse de celle réalisée

## LA DECOUVERTE DE L'ALICE ET DU MC/10

par STR\$: elle convertit une chaîne de caractères en un nombre.

```
tapez      D = VAL("543.82")
tapez      PRINT D+10
réponse    553.82
```

Dans l'exemple ci-dessus, VAL opère sur une chaîne de caractères pouvant être interprétée complètement comme un nombre.

Voyons d'autres cas :

```
tapez      PRINT VAL("AOUT")
réponse    0
tapez      PRINT VAL("AOUT 1983")
réponse    0
tapez      PRINT VAL("536YYSHDJE")
réponse    536
tapez      PRINT VAL("53+8193")
réponse    53
tapez      PRINT VAL("1E+5")
réponse    100000
```

Explications :

- dans l'interprétation de la chaîne de caractères, VAL prend en compte à partir de la gauche tous les caractères représentant un nombre.
- si le premier caractère, à l'exclusion de l'espace, ne peut pas entrer dans la représentation d'un nombre, la "VALEUR" de la chaîne est prise égal à 0.
- les espaces sont ignorés quelle que soit leur place :

```
tapez      PRINT VAL("6 8 . 7 5 D W")
réponse    68.75
```

Voici d'autres exemples :

```
tapez      A$ = "253FD7261"
tapez      PRINT VAL(A$)
réponse    253
tapez      PRINT VAL(RIGHT$(A$,2))
réponse    61
```

## LES INSTRUCTIONS ASC et CHR\$

Pour bien comprendre ces instructions, il est nécessaire de rappeler que le système binaire n'utilise pour

## LA DECOUVERTE DE L'ALICE ET DU MC/10

représenter les nombres que deux chiffres : 0 et 1.

Ainsi les premiers nombres s'écrivent :

<i>Dans le système binaire</i>	<i>Dans le système décimal</i>
0	0
1	1
10	2
11	3
100	4
101	5
...	.
10000000	128
.....	...
11111111	255

Tous les chiffres et caractères du clavier sont codés par un nombre binaire à 8 chiffres. Le code utilisé par ALICE est le code ASCII (American Standard Code for Information Interchange).

Dans ce code, la lettre C, par exemple, est représentée par le nombre binaire 01000011 dont l'équivalent décimal est 67. On dira que 67 est le code ASCII de la lettre C.

En langage Basic, l'instruction ASC permet de retrouver ce code et de l'utiliser :

```
tapez      PRINT ASC("C")
réponse    67
```

### L'INSTRUCTION ASC

Elle donne la valeur du code ASCII du premier caractère d'une chaîne :

```
tapez      PRINT ASC("BINAIRE")
réponse    66
tapez      CODE$="ASCII"
tapez      PRINT ASC(CODE$)
réponse    65
```

### L'INSTRUCTION CHR\$

CHR est l'abréviation de CHaRacter = Caractère.

Elle donne le caractère correspondant à une valeur du code ASCII.

*LA DECOUVERTE DE L'ALICE ET DU MC/10*

```
tapez      PRINT CHR$(67)
réponse    C
tapez      PRINT CHR$(40)
réponse    (
tapez      PRINT CHR$(50)
réponse    2
```

**Remarque :** Le code ASCII est donné à l'annexe 3. Notez que la représentation des lettres est donnée par des nombres décimaux qui se suivent. D'autre part, la différence entre le code ASCII d'une lettre minuscule et le code ASCII de la même lettre en majuscule est toujours égale à 32.



**LA DECOUVERTE DE L'ALICE ET DU MC/10**

**EXERCICES SUR LE CHAPITRE VII**

**Exercice 8.1 :** *Ecrire un programme recherchant, dans un texte quelconque, les mots dans lesquels entre un groupe de lettres donné (clé).*

**Indications :** *écrire le texte sur lequel se fait la recherche, sous forme d'une liste de mots dans une ou plusieurs instructions DATA.*

**Exercice 8.2 :** *Dans le programme précédent, améliorer l'affichage de la liste des mots de la manière suivante : le mot trouvé est affiché en minuscule, à l'exclusion de la clé de recherche.*

*Exemple : si la clé est "IT", le mot VOITURE s'affiche : voITure.*

**Exercice 8.3 :** *"Le Mot le plus long". Ecrire un programme simulant le tirage aléatoire de 9 lettres en précisant, pour chacune, s'il s'agit d'une consonne ou d'une voyelle.*

**Exercice 8.4 :** *Ecrire un programme qui affiche sur l'écran les caractères dont le code ASCII est compris entre 32 et 255.*

## **CHAPITRE 9**

### LES INSTRUCTIONS D'INTERACTIVITE

- CLS
- INKEY\$
- Affichage de caractères  
graphiques  
avec PRINT  
avec SET et RESET
- POINT
- SOUND

## LA DECOUVERTE DE L'ALICE ET DU MC/10

### L'INSTRUCTION CLS

*Cette instruction permet de définir la couleur de l'écran.*

Le programme suivant permet de voir les différentes couleurs disponibles :

```
10 CLS
20 FOR I=1 TO 8
30 CLS I
40 FOR T=1 TO 1000:NEXT T
50 NEXT I
60 GOTO 20
```

Les codes des couleurs sont donnés dans le tableau ci-dessous :

<i>Code</i>	<i>Couleur</i>
Ø	noir
1	vert
2	jaune
3	bleu
4	rouge
5	blanc
6	cyan
7	magenta
8	orange

Si le code couleur est omis ou a une valeur numérique non comprise entre Ø et 8, cette instruction efface l'écran dans la couleur verte qui est la couleur de fond pour affichage des caractères alphanumériques.

### LA FONCTION INKEY\$

Cette instruction est utilisée lors de l'exécution d'un programme. Elle permet d'affecter à une variable alphanumérique un caractère tapé au clavier. Ce caractère est le dernier caractère qui précède l'exécution de INKEY\$. Si aucun caractère n'a été frappé, INKEY\$ affecte une chaîne vide. Dans tous les cas, le déroulement du programme exécute l'instruction qui suit INKEY\$.

## LA DECOUVERTE DE L'ALICE ET DU MC/10

Voici deux exemples d'utilisation :

Tapez le programme suivant :

```
10 P=0
20 PRINT "TAPEZ SUR UNE TOUCHE QUECONQUE"
30 P=P+1:IF INKEY$="" THEN 30
40 PRINT P
50 GOTO 10
```

A la ligne 30, si aucune touche n'a été frappée, INKEY\$ est égal à la chaîne vide et le programme reprend à cette même ligne et augmente à chaque passage la variable P d'une unité. Lors de l'appui sur une touche quelconque, le programme exécute la ligne 40 et affiche ainsi une valeur de P qui dépend de votre temps de réaction.

Dans certains cas, la sortie de la boucle qui utilise INKEY\$ doit s'effectuer par appui sur une touche particulière.

En voici un exemple :

tapez :

```
10 PRINT "TAPEZ SUR 0 POUR SORTIR"
20 FOR I=1 TO 5000
30 REP$=INKEY$
40 IF INKEY$="0" THEN 60
50 NEXT I
60 PRINT "SORTIE DE LA BOUCLE A I =";I
70 GOTO 10
```

En plus de ses possibilités de calcul numérique et de traitement des chaînes de caractères, ALICE peut réaliser des dessins sur l'écran du téléviseur.

## AFFICHAGE DE CARACTERES GRAPHIQUES

### *AVEC L'INSTRUCTION PRINT*

En plus des caractères standards, vous disposez sur ALICE d'un jeu de 128 caractères graphiques répartis en huit groupes de seize caractères de même couleur (voir annexe 4).

### LA DECOUVERTE DE L'ALICE ET DU MC/10

Le programme suivant vous permet d'afficher ces caractères :

```
10 CLS
20 INPUT "ENTREZ UN CODE DE COULEUR (1-8)";C
25 IF C<1 OR C>8 THEN END
30 CLS
40 FOR I=128 TO 143
50 V=I+16*(C-1)
60 PRINT @2*I-31,CHR$(V)
70 PRINT:PRINT
80 NEXT I
90 GOTO 20
```

En tapant pour le code un nombre compris entre 1 et 8, vous obtenez l'affichage des 16 caractères graphiques de la couleur qui correspond à ce code.

#### Explications :

- Lignes 10 et 30 : l'instruction CLS remplit l'écran de la couleur verte.
- Lignes 50 et 60 : affichage du caractère graphique dont le code a pour valeur  $V=I+16*(C-1)$ .

Si  $C=1$ , cette valeur varie entre 128 et 143 et permet d'obtenir les 16 caractères graphiques de couleur verte.

Si  $C=2$ , cette valeur varie entre 144 et 159 et permet d'obtenir les 16 caractères graphiques de couleur jaune...

Enfin, si  $C=8$ , cette valeur varie entre 240 et 255 et permet d'obtenir les 16 caractères graphiques de couleur orange.

#### **AVEC LES INSTRUCTIONS SET ET RESET**

*L'instruction SET permet d'afficher un pavé de couleur à une position donnée de l'écran.*

Tapez le programme suivant :

```
10 CLS 0
20 INPUT "COULEUR (1-8)";C
25 IF C<1 OR C>8 THEN END
30 INPUT "COLONNE (0-63)";X
35 IF X<0 OR X>63 THEN END
40 INPUT "LIGNE (0-31)";Y
```

LA DECOUVERTE DE L'ALICE ET DU MC/10

```

45 IF Y<0 OR Y>31 THEN END
50 SET(X,Y,C)
60 PRINT @0,;
70 GOTO 20

```

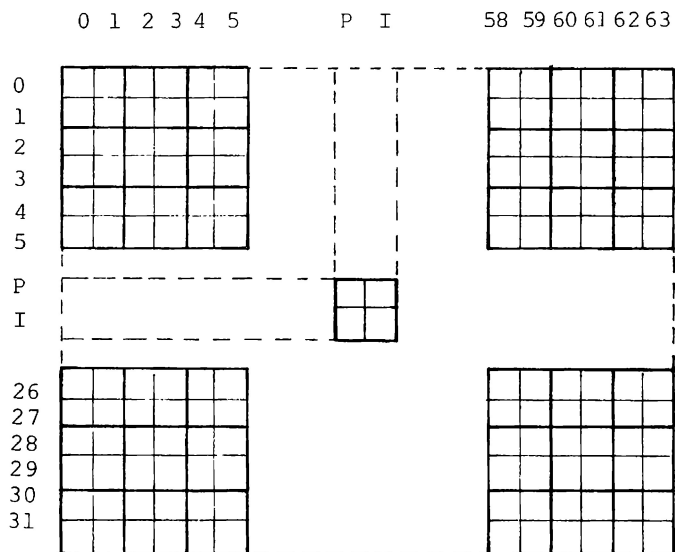
```

tapez      RUN
réponse    COULEUR (1-8)?
tapez      3
réponse    COLONNE (0-63)?
tapez      30
réponse    LIGNE (0-31)?
tapez      15

```

Un pavé de couleur bleue s'affiche sur l'écran à la position définie par les coordonnées 30 et 15.

En mode graphique, l'écran comprend 32 lignes (numérotées de 0 à 31) et 64 colonnes (numérotées de 0 à 63). La position de chaque pavé est repérée par un numéro de ligne et un numéro de colonne. L'écran comprend donc 2048 pavés.



*P signifie que le nombre est pair*  
*I signifie que le nombre est impair*  
 Les quatre pavés encadrés en gras correspondent à la zone d'affichage d'un caractère.

## LA DECOUVERTE DE L'ALICE ET DU MC/10

La couleur du pavé est définie par un code dont voici le tableau de correspondance :

vert	1	blanc	5
jaune	2	cyan	6
bleu	3	magenta	7
rouge	4	orange	8

*L'instruction SET (X,Y,C) affiche un pavé de couleur de code C à l'intersection de la colonne X et de la ligne Y.*

*L'instruction RESET (X,Y) efface le pavé se trouvant à l'intersection de la colonne X et de la ligne Y.*

### Remarque 1

```
tapez      CLS
tapez      SET(45,26,5)
puis tapez SET(44,27,3)
```

Vous constatez que l'instruction SET(44,27,3) affiche un pavé bleu de coordonnées 44 et 27 et modifie la couleur du pavé précédemment affiché. En effet, deux pavés qui appartiennent à la même position d'affichage d'un caractère sur l'écran, sont obligatoirement de la même couleur. Les positions d'affichage d'un caractère (numérotées de 0 à 511) sont formées de 4 pavés voisins. Elles ont été étudiées avec l'instruction PRINT @ au chapitre 3.

```
tapez      SET(43,27,5)
```

Le pavé de coordonnées 43,27 appartenant à un caractère voisin, a une couleur indépendante de celle des pavés précédemment affichés.

Pour calculer la position P d'un caractère dans la zone d'affichage de l'écran, à partir du numéro de colonne X et du numéro de ligne Y d'un pavé, utilisez la formule :

$$P = \text{INT}(X/2) + 32 * \text{INT}(Y/2)$$

### Exemple

```
tapez      CLS
            SET(32,23,5)
            P = INT(42/2) + 32 * INT(23/2)
            PRINT P
            PRINT @ P, "A"
```

## LA DECOUVERTE DE L'ALICE ET DU MC/10

### Remarque 2

```
tapez      CLS 1
puis      SET (45,26,5)
```

Vous constatez que l'instruction SET(45,26,5) a coloré en blanc les 4 pavés repérés par les couples respectifs (numéro de colonne, numéro de ligne), (44,26), (44,27), (45,26) et (45,27) de la position 438 de la zone d'affichage des caractères. Cet exemple montre que les deux instructions CLS 1 et CLS ne sont pas totalement identiques.

CLS 1 colore en vert tous les pavés de l'écran en les considérant comme des caractères graphiques. En conséquence, lors d'une redéfinition de la couleur d'un pavé, quatre pavés sont modifiés puisque quatre pavés, d'une même position d'affichage, sont nécessairement de la même couleur.

En revanche, CLS ou CLS C (où C est une valeur numérique non comprise entre 0 et 8) efface l'écran dans la couleur verte, en la considérant comme la couleur de fond des caractères alphanumériques ; lors de l'affichage d'un pavé par l'instruction SET, le fond du pavé est de couleur noire.

```
Maintenant tapez  RESET(45,27)
```

Vous constatez que l'instruction RESET éteint (couleur noire) le pavé repéré par le couple (45,27).

```
Enfin      tapez  SET(45,26,3)
```

### L'INSTRUCTION POINT

Vous savez à présent afficher un pavé de couleur donnée à une position particulière de l'écran. Inversement, *L'instruction POINT permet de savoir si, à une position particulière de l'écran, il y a un pavé et si oui, quelle est sa couleur.* Cette instruction est très utile dans les programmes de jeux pour traiter, par exemple, les problèmes de collision entre deux objets qui se déplacent sur l'écran.

```
tapez      CLS
tapez      SET(25,28,4)
tapez      PRINT POINT(23,28)
réponse    -1
tapez      PRINT POINT(24,28)
réponse    Ø
```



### LA DECOUVERTE DE L'ALICE ET DU MC/10

tapez        PRINT POINT(25,28)  
réponse     4

L'instruction POINT(X,Y) prend comme valeur :

- (-1) si le pavé pointé par le numéro de colonne X et le numéro de ligne Y, est de la couleur verte correspondant au fond des caractères,
- Ø    si le pavé est éteint (couleur noire),
- C    si le pavé est de couleur dont le code est C.

#### Remarque

tapez        CLS  
tapez        SET(25,12,1)  
tapez        PRINT POINT(25,12)  
réponse     1

La réponse 1 correspond à la couleur verte du pavé (25, 12), mais cette couleur est considérée ici comme la couleur des caractères graphiques et non plus comme la couleur de fond des caractères alphanumériques.

#### L'INSTRUCTION SOUND

Votre micro-ordinateur ALICE sait encore faire de la musique.

tapez        SOUND 137,8

*L'instruction SOUND F,D provoque l'émission d'un son d'autant plus aigu que F est grand, et de durée d'autant plus longue que D est grand.*

F doit prendre une valeur numérique comprise entre 1 et 255.

D doit prendre une valeur numérique comprise entre 1 et 255.

Essayez vous-même d'autres valeurs.

*LA DECOUVERTE DE L'ALICE ET DU MC/10*

EXERCICES SUR LE CHAPITRE IX

**Exercice 9.1 :** *Ecrire un programme qui trace un cadre de couleur donnée sur un fond de couleur donnée.*

**Exercice 9.2 :** *Ecrire un programme qui fait se déplacer une balle dans le rectangle de l'exercice précédent. Les rebonds sur les bords du rectangle, seront accompagnés d'un bip musical.*

**Exercice 9.3 :** *Ecrire un programme qui place des pavés de couleur aléatoirement dans une zone choisie de l'écran.*

**Exercice 9.4 :** *A partir du programme précédent, écrire un programme qui permet de déplacer une "souris" du bord gauche au bord droit de l'écran en évitant les obstacles sur son chemin.*

**Exercice 9.5 :** *Ecrire un programme qui simule une machine à sous : Celle-ci est constituée de trois rouleaux sur lesquels sont dessinées trois sortes de figures ; lors d'un tirage, la machine lance les trois rouleaux qui s'arrêtent ensuite aléatoirement sur une position.*

**Exercice 9.6 :** *Ecrire un programme qui entre un mot de moins de 20 caractères et qui l'affiche de façon clignotante au milieu de l'écran.*

**Exercice 9.7 :** *Ecrire un programme simulant le jeu de "Pile ou Face".*

## ANNEXE 1

# SIGNIFICATION DES MESSAGES D'ERREUR

<i>Message</i>	<i>Signification</i>
BS	<b>Bad Subscript</b> : Indice hors des limites.
CN	<b>Can't contiNue</b> : Impossibilité de continuer l'exécution du programme.
DD	<b>Duplicate Definition</b> : Tentative de dimensionnement d'un tableau déjà dimensionné.
FC	<b>Function Call</b> : Argument incorrect dans l'appel d'une fonction.
FM	<b>File Mode</b> : Erreur de type de fichier.
ID	<b>Illegal Direct</b> : Instruction interdite en mode direct.
IO	<b>Input Output</b> : Erreur d'entrée sortie.
LS	<b>String too long</b> : Chaîne de caractères trop longue (plus de 255 caractères).
NF	<b>Next without For</b> : Instruction Next sans instruction For associée.
OM	<b>Out of Memory</b> : Dépassement de capacité de la mémoire.
OD	<b>Out of Data</b> : Epuisement des données dans les instructions Data.
OS	<b>Out of string Space</b> : Manque de place en mémoire pour les chaînes de caractères.

*LA DECOUVERTE DE L'ALICE ET DU MC/10*

<i>Message</i>	<i>Signification</i>
<b>OV</b>	<b>Overflow</b> : Dépassement de capacité par valeur supérieure.
<b>RG</b>	<b>Return without Gosub</b> : Instruction RETURN rencontrée sans exécution d'une instruction GOSUB.
<b>SN</b>	<b>Syntax error</b> : Erreur de syntaxe.
<b>ST</b>	<b>String formula Too complex</b> : Expression utilisant des chaînes de caractères trop complexes.
<b>TM</b>	<b>Type Mismatch</b> : Erreur de correspondance de type.
<b>UL</b>	<b>Undefined Line</b> : Numéro de ligne inexistant.
<b>/Ø</b>	<b>Division par zéro.</b>

## ANNEXE 2

# MOTS RESERVES

FOR	GOTO	GOSUB
REM	IF	DATA
PRINT	ON	INPUT
END	NEXT	DIM
READ	LET	RUN
RESTORE	RETURN	STOP
POKE	CONT	LIST
CLEAR	NEW	CLOAD
CSAVE	LLIST	LPRINT
SET	RESET	CLS
SOUND	EXEC	SKIPF
TAB(	TO	THEN
NOT	STEP	OFF
OR	SGN	AND
ABS	USR	INT
SQR	LOG	RND
SIN	COS	EXP
PEEK	LEN	TAN
VAL	ASC	STR\$
LEFT\$	RIGHT\$	CHR\$
POINT	VARPTR	MID\$
MEM		INKEY\$

## ANNEXE 3

# CARACTERISTIQUES GRAPHIQUES CODES DES COULEURS



128+C



129+C



130+C



131+C



132+C



133+C



134+C



135+C



136+C



137+C



138+C



139+C



140+C



141+C



142+C



143+C

### CODE DE COULEUR

Vert	C=0
Jaune	C=16
Bleu	C=32
Rouge	C=48
Blanc	C=64
Cyan	C=80
Magenta	C=96
Orange	C=112

On a ainsi  $16 \times 8 = 128$  caractères graphiques allant des codes ASCII 128 à 255.

## ANNEXE 4

### CODE ASC II

Code		Code		Code	VI
00		42	*	85	U
01		43	+	86	V
02		44	,	87	W
03	touche BREAK	45	-	88	X
04		46	.	89	Y
05		47	/	90	Z
06		48	Ø	91	[
07		49	1	92	/
08	(Recul arrière) <b>CONTROL</b> <b>Q</b>	50	2	93	]
09	<b>CONTROL</b> <b>S</b>	51	3	94	†
10	<b>CONTROL</b> <b>W</b>	52	4	95	↑
11		53	5	96	,
12		54	6	97	a
13	touche ENTER	55	7	98	b
14		56	8	99	c
15		57	9	100	d
16		58	:	101	e
17		59	;	102	f
18		60	<	103	g
19	(ARRET clavier) <b>SHIFT</b> <b>@</b>	61	=	104	h
20		62	>	105	i
21	(Eff. Ligne) <b>CONTROL</b> <b>A</b>	63	?	106	j
22		64	@	107	k
23		65	A	108	l
24		66	B	109	m
25		67	C	110	n
26		68	D	111	o
27		69	E	112	p
28		70	F	113	q
		71	G	114	r

**LA DECOUVERTE DE L'ALICE ET DU MC/10**

Code		Code		Code		VI
29		72	H	115	s	S
30		73	I	116	t	T
31		74	J	117	u	U
32	Espace	75	K	118	v	V
33	!	76	L	119	w	W
34	"	77	M	120	x	X
35	#	78	N	121	y	Y
36	\$	79	O	122	z	Z
37	%	80	P	123	{	[
38	&	81	Q	124		\
39	'	82	R	125	}	]
40	(	83	S	126	~	†
41	)	84	T	127		←

*VI : Caractères affichés sur l'écran en Vidéo Inverse.*



## ANNEXE 5

### RESULTAT DES EXERCICES

#### *Exercice 2.1*

```
7283549570  
-0.0003841635
```

#### *Exercice 2.2*

Pour obtenir les résultats corrects, il faut taper :

```
PRINT 150-92  
PRINT 89-18+25*2-13  
PRINT 15*3/2  
PRINT 15/(2*3)  
PRINT (15+3)/2  
PRINT 15/2+3  
PRINT 1736/1.3+6  
PRINT (1736/1.3)+6
```

#### *Exercice 2.3*

15 éléments  
2 dimensions et 77 éléments (11x7)

#### *Exercice 3.1*

```
PI=3.1416  
20 CLS  
30 INPUT "RAYON DU CERCLE " : R  
40 P=2*PI*R  
50 S=PI*R^2  
60 PRINT "PERIMETRE " : P  
70 PRINT "SURFACE " : S  
80 END
```

LA DECOUVERTE DE L'ALICE ET DU MC/10

**Exercice 3.2**

```
10 CLS
20 REM ENTREE DES DONNEES
30 INPUT "NOM DU PAYS ";P$
40 INPUT "SUPERFICIE ";S
50 INPUT "POPULATION ";P
60 REM CALCUL DE LA DENSITE
70 D=P/S
80 REM AFFICHAGE
90 PRINT P$;" : DENSITE  ";D
100 END
```

**Exercice 3.3**

```
10 CLS
20 REM ENTREE DES DONNEES
30 INPUT "NOM DU PAYS ";P$
40 INPUT "SUPERFICIE ";S
50 INPUT "POPULATION ";P
60 REM CALCUL DE LA DENSITE
70 D=P/S
80 REM AFFICHAGE
90 PRINT "PAYS";TAB(14);"DENSITE"
100 PRINT P$;TAB(13);D
110 END
```

**Exercice 3.4**

```
10 REM LECTURE DES DONNEES
20 READ N$(1),N$(2),N$(3),N$(4),N$(5),N$(6),N$(7),N$(8),
                                     (6)N$
30 READ H(1),H(2),H(3),H(4),H(5),H(6),H(7),H(8),H(9)
40 REM AFFICHAGE
50 CLS
60 PRINT "PAYS.....POPULATION"
70 PRINT "-----"
80 PRINT N$(1),H(1),N$(2),H(2),N$(3),H(3)
90 PRINT N$(4),H(4),N$(5),H(5),N$(6),H(6)
100 PRINT N$(7),H(7),N$(8),H(8),N$(9),H(9)
110 PRINT "-----"
200 DATA ALLEMAGNE,BELGIQUE,DANEMARK,FRANCE,IRLANDE
210 DATA ITALIE,LUXEMBOURG,PAYS-BAS,ROYAUME-UNI
220 DATA 61.498,9.889,5.073,52.915,3.162,56.323,1.358
230 DATA 13.825,55.928
```

LA DECOUVERTE DE L'ALICE ET DU MC/10

*Exercice 4.1*

```
10 CLS
20 REM ENTREE DES DONNEES
30 PRINT
40 INPUT "NOMBRE A";A
50 PRINT
60 INPUT "NOMBRE B";B
70 REM TESTS
80 PRINT
90 IF A>B THEN 130
100 IF A<B THEN 150
110 IF A=B THEN 170
120 REM AFFICHAGE
130 PRINT "A EST SUPERIEUR A B"
140 GOTO 180
150 PRINT "A EST INFERIEUR A B"
160 GOTO 180
170 PRINT "A EST EGAL A B"
180 PRINT
190 INPUT "VOULEZ-VOUS CONTINUER (OUI/NON)";RE#
200 IF RE#="OUI" THEN 20
210 IF RE#<>"NON" THEN 190
220 END
```

*Exercice 4.2*

```
10 REM DIMENSIONNEMENT DES TABLEAUX
20 REM
30 DIM N$(15),H(15)
40 REM
50 REM LECTURE DES DONNEES
60 REM
70 FOR I=1 TO 9
80 READ N$(I)
90 NEXT I
100 FOR I=1 TO 9
110 READ H(I)
120 NEXT I
130 REM AFFICHAGE
140 CLS
150 PRINT "PAYS          POPULATION"
160 PRINT "-----"
170 FOR I=1 TO 9
180 PRINT N$(I),H(I)
190 NEXT I
200 PRINT "-----"
```

**LA DECOUVERTE DE L'ALICE ET DU MC/10**

```
210 END
500 DATA ALLEMAGNE,BELGIQUE,DANEMARK,FRANCE
510 DATA IRLANDE,ITALIE,LUXEMBOURG
520 DATA PAYS-BAS,ROYAUME-UNI
530 DATA 61.498,9.889,5.073,52.915,3.162
540 DATA 56.323,.358,13.825,55.928
```

**Exercice 4.3**

```
10 REM DIMENSIONNEMENT DES TABLEAUX
20 REM
30 DIM N$(15),H(15)
40 REM
50 REM LECTURE DES DONNEES
60 REM
70 FOR I=1 TO 9
80 READ N$(I)
90 NEXT I
100 FOR I=1 TO 9
110 READ H(I)
120 NEXT I
130 REM CLASSEMENT PAR ORDRE CROISSANT
140 FOR I=1 TO 9
150 P=1
160 FOR J=1 TO 9
170 IF H(I)>H(J) THEN P=J+1
180 NEXT J
190 Z(P)=I
200 NEXT I
210 REM AFFICHAGE
220 CLS
230 PRINT "PAYS" TAB(16)"POPULATION"
240 PRINT "-----"
250 FOR I=1 TO 9
260 PRINT N$(Z(I));
270 K=2
280 IF H(Z(I))>=1 THEN K=1
290 IF H(Z(I))>=10 THEN K=0
300 PRINT TAB(19+K)H(Z(I))
310 NEXT I
320 PRINT "-----"
330 END
500 DATA ALLEMAGNE,BELGIQUE,DANEMARK,FRANCE
510 DATA IRLANDE,ITALIE,LUXEMBOURG
520 DATA PAYS-BAS,ROYAUME-UNI
530 DATA 61.498,9.889,5.073,52.915,3.162
540 DATA 56.323,.358,13.825,55.928
```

LA DECOUVERTE DE L'ALICE ET DU MC/10

**Exercice 4.4**

```
10 CLR
20 DIM NM$(20)
30 PRINT
40 INPUT "NOMBRE DE NOMS A CLASSER (MAXIMUM 20)";NB
50 IF NB<? OR NB>20 THEN 40
50 PRINT
70 REM ENTREE DES DONNEES
80 FOR K=1 TO NB
90 PRINT "NOM";K;
100 INPUT NM$(K)
110 NEXT K
120 REM RECHERCHE DU MIN.
130 FOR J=1 TO NB-1
140 MN#=NM$(J)
150 FOR I=J+1 TO NB
160 IF MN#<NM$(I) THEN 210
170 REM PERMUTATION DE NM$(I) ET DE MN# PAR L'INTERME-
DIAIRE DE AX#
180 AX#=MN#
190 MN#=NM$(I)
200 NM$(I)=AX#
210 NEXT I
220 REM RANGEMENT DE MN# A SA PLACE DANS LE TABLEAU NM#
230 NM$(J)=MN#
240 NEXT J
250 REM AFFICHAGE DU TABLEAU CLASSE
260 PRINT
270 PRINT "LISTE ALPHABETIQUE : "
280 PRINT
290 FOR L=1 TO NB
300 PRINT TAB(8)NM$(L)
310 NEXT L
320 END
```

**Exercice 5.1**

```
10 DIM H$(11),C$(2),D(11,2),Z(11)
20 REM LECTURE DES DONNEES
30 GOSUB 500
40 REM AFFICHAGE DU MENU ET CHOIX
50 GOSUB 1000
60 REM CLASSEMENT
70 GOSUB 1500
80 REM AFFICHAGE DU TABLEAU
90 GOSUB 2000
100 REM CHOIX ARRETER, CONTINUER
```

**LA DECOUVERTE DE L'ALICE ET DU MC/10**

```

110 INPUT "ON CONTINUE (OUI/NON)";A$
120 IF A$("<" "NON") THEN 40
130 END
500 REM SOUS-PROGRAMME DE LECTURE
510 READ C$(0),C$(1),C$(2)
520 FOR I=0 TO 11
530 READ N$(I)
540 NEXT I
550 FOR J=0 TO 2
560 FOR I=0 TO 11
570 READ D$(I,J)
580 NEXT I
590 NEXT J
600 RETURN
1000 REM SOUS-PROGRAMME D'AFFICHAGE DU MENU
1010 CLS
1020 PRINT "0-POPULATIONS CROISSANTES"
1030 PRINT "1-SUPERFICIES CROISSANTES"
1040 PRINT "2-DENSITES CROISSANTES"
1050 INPUT "QUELLE OPTION ";J
1060 IF J<0 THEN 1000
1070 IF J>2 THEN 1000
1080 RETURN
1500 REM SOUS-PROGRAMME DE CLASSEMENT
1510 FOR I=0 TO 11
1520 P=0
1530 FOR K=0 TO 11
1540 IF D$(I,J)<D$(K,J) THEN 1560
1550 P=P+1
1560 NEXT K
1570 Z(P)=I
1580 NEXT I
1590 RETURN
2000 REM SOUS-PROGRAMME D'AFFICHAGE
2010 CLS
2020 PRINT "PAYS",C$(J)
2030 PRINT "-----"
2040 FOR I=0 TO 11
2050 PRINT N$(Z(I)),D$(Z(I),J)
2060 NEXT I
2070 PRINT "-----"
2080 RETURN
5000 DATA POPULATION,SUPERFICIE,DENSITE
5010 DATA ALLEMAGNE,BELGIQUE,DANEMARK,ESPAGNE,FRANCE
5020 DATA GRECE,IRLANDE,ITALIE,LUXEMBOURG
5030 DATA PAYS-BAS,PORTUGAL,ROYAUME-UNI
5040 DATA 61.498,9.889,5.073,35.971,52.915,9.165,3.162
5050 DATA 56.323,0.358,13.825,9.449,55.928
5060 DATA 249,31,43,505,547,132,70,301,3,41,92,244
5070 DATA 247,319,118,71,97,69,45,187,138,339,103,228

```

LA DECOUVERTE DE L'ALICE ET DU MC/10

Exercice 5.2

```
10 CLEAR: DIM N(100)
15 CLS
20 PRINT "1- LECTURE DU FICHIER"
30 PRINT "2- AFFICHAGE DU TABLEAU"
40 PRINT "3- INSERTION "
50 PRINT "4- SUPPRESSION"
60 PRINT "5- ECRITURE DU FICHIER"
70 PRINT "0- FIN"
80 PRINT
90 INPUT "CHOISISSEZ (0-5) "; R
100 IF R<=0 OR R>5 THEN END
110 ON R GOSUB 1000,1100,1200,1300,1400
120 GOTO 15
130 END
1000 REM
1010 REM LECTURE DE LA CASSETTE
1020 INPUT "LA CASSETTE EST-ELLE POSITIONNEE"; R$
1030 IF R$<>"OUI" THEN RETURN
1040 CLOAD* N, "FICDAT"
1050 RETURN
1100 REM
1110 REM AFFICHAGE
1115 IF N(0)=0 THEN 1150
1120 FOR I=1 TO N(0)
1130 PRINT N(I)
1140 NEXT I
1150 INPUT "POUR REVENIR AU MENU TAPÉZ M "; M$
1160 IF M$<>"M" THEN 1150
1170 RETURN
1200 REM
1210 REM INSERTION
1215 N(0)=N(0)+1
1220 INPUT "QUELLE VALEUR"; N(N(0))
1240 RETURN
1300 REM
1310 REM SUPPRESSION
1320 INPUT "NUMERO DE LA VALEUR A SUPPRIMER"; K
1330 IF K<0 OR K>N(0) THEN 1375
1340 N(0)=N(0)-1
1350 FOR J=K TO N(0)
1360 N(J)=N(J+1)
1365 NEXT J
1370 RETURN
1375 PRINT "SUPPRESSION IMPOSSIBLE"
1380 FOR H=1 TO 400
1385 NEXT H
1390 RETURN
```

*LA DECOUVERTE DE L'ALICE ET DU MC/10*

```
1400 REM
1410 REM ECRITURE DE N SUR CASSETTE
1420 INPUT "LA CASSETTE EST-ELLE POSITIONNEE";R#
1430 IF R#<>"OUI" THEN RETURN
1440 CSAVE* N,"FICDAT"
1450 RETURN
```

*Exercice 6*

```
10 FOR I=49221 TO 49479
15 A=PEEK(I)
20 IF A>128 THEN F=1:A=A-128
25 PRINT CHR$(A);
30 IF F=1 THEN PRINT " ";F=0
35 NEXT I
```

*Exercice 7.1*

```
10 CLS
20 INPUT "QUELLE VALEUR DE C ";C
30 FOR I=1 TO 17
40 X=I/30
50 Y=10*SIN(C*X)+15
60 PRINT TAB(Y);"*"
70 NEXT I
80 END
```

*Exercice 7.2*

```
10 REM QUOTIENT ET RESTE
20 CLS
30 INPUT "TAPEZ 2 NOMBRES ENTIERS SEPARES PAR UNE
          VIRGULE";A,B
40 IF B<=0 OR A<0 OR A-INT(A)<>0 OR B-INT(B)<>0 THEN 30
50 Q=INT(A/B)
60 R=A-INT(A/B)*B
70 PRINT A;" = ";B;"*";Q;" + ";R
80 INPUT "VOULEZ-VOUS CONTINUER (OUI/NON) ";R#
90 IF R#="OUI" THEN 30
100 END
```



*LA DECOUVERTE DE L'ALICE ET DU MC/10*

**Exercice 7.3**

```
10 L=LOG(10)
20 REM LECTURE DES DONNEES
30 FOR I=1 TO 9
40 READ N%(I)
50 NEXT I
60 FOR I=1 TO 9
70 READ H%(I)
80 NEXT I
90 REM AFFICHAGE
100 CLS
110 PRINT "PAYS          POPULATION"
120 PRINT "-----"
130 FOR I=1 TO 9
140 PRINT N%(I)TAB(17-INT(LOG(H%)/L))H%(I)
150 NEXT I
160 PRINT "-----"
170 END
180 DATA ALLEMAGNE,BELGIQUE,DANEMARK,FRANCE,IRLANDE
190 DATA ITALIE,LUXEMBOURG,PAYS-BAS,ROYAUME-UNI
200 DATA 61.498,9.889,5.073,52.915,3.162,56.323
210 DATA .358,13.825,55.928
```

**Exercice 7.4**

```
10 CLS
20 PRINT "RESOLUTION DE L'EQUATION DU SECOND DEGRE"
30 REM
40 PRINT
50 PRINT TAB(10)"AX2+BX+C=0"
60 INPUT "A =":A
70 INPUT "B =":B
80 INPUT "C =":C
90 DELTA=B*B-4*A*C
100 IF DELTA=0 THEN 500
110 IF DELTA<0 THEN 600
120 REM
130 REM DEUX RACINES REELLES DISTINCTES
140 REM
150 X1=(-B-SQR(DELTA))/2/A
160 X2=(-B+SQR(DELTA))/2/A
170 PRINT
180 PRINT TAB(15)"X1      ":X1
190 PRINT TAB(15)"X2      ":X2
200 END
500 REM
```

*LA DECOUVERTE DE L'ALICE ET DU MC/10*

```
510 REM UNE RACINE DOUBLE
520 REM
530 PRINT
540 X=-B/2/A
550 PRINT "CETTE EQUATION A UNE RACINE DOUBLE"
560 PRINT TAB(15)"X = ";X
570 END
600 REM
610 REM PAS DE RACINE REELLE
620 PRINT "CETTE EQUATION N'A PAS DE RACINE REELLE"
630 END
```

*Exercice 8.1*

```
10 CLS
20 INPUT "QUELLE CLE ";C$
30 C=LEN(C$)
40 PRINT
50 REM RECHERCHE
60 F=0
70 READ A$
80 IF A$="FIN" THEN 150
90 FOR I=1 TO LEN(A$)-C+1
100 IF MID$(A$,I,C)<>C$ THEN 130
110 PRINT TAB(16-I)A$
120 F=F+1
130 NEXT I
140 GOTO 70
150 REM AFFICHAGE DU RESULTAT
160 PRINT
170 PRINT "LA CLE : ";C$
180 PRINT "A ETE TROUVEE";F;"FOIS"
190 END
1000 DATA LA,VOITURE,EST,ARRIVEE,QUAND,LE,TRAIN,ETAIT
1010 DATA EN,GARE,LE,BUREAU,DANS,LA,VOITURE,SOUS,LE
1020 DATA TOIT,AVAIT,ETE,VOLE,LA,RIVIERE
1030 DATA A,DEBORDE,PENDANT,LA,NUIT
1040 DATA FIN
```

*Exercice 8.2*

```
10 CLS
20 INPUT "QUELLE CLE ";C$
30 C=LEN(C$)
40 PRINT
50 REM RECHERCHE
60 F=0
```

**LA DECOUVERTE DE L'ALICE ET DU MC/10**

```
70 READ A$
80 REM TEST DE FIN DES DONNEES
90 IF H$="FIN" THEN 200
100 REM
110 IF LEN(A$) <> 0 THEN 70
120 FOR I=1 TO LEN A$ :C=C+1
130 IF MID$(A$,I,C) > D$ THEN 170
140 GOSUB 300
150 PRINT TAB(16-I)AFF$
160 F=F+1
170 NEXT I
180 GOTO 70
200 REM
210 REM AFFICHAGE DU RESULTAT
220 PRINT
230 PRINT "LE MOT : " ; C$
240 PRINT "A ETE TROUVEE" IF "YAY"
250 END
300 REM PASSAGE EN MAJE MINUSCULE
310 AFF$=""
320 IF I=1 THEN 400
330 FOR J=1 TO F-I
340 AFF$=AFF$+CHR$(ASC(MID$(A$,J,I))+32)
350 NEXT J
400 AFF$=AFF$+C$
410 IF LEN(A$)=I+C-1 THEN 450
420 FOR J=I+1 TO LEN A$
430 AFF$=AFF$+CHR$(ASC(MID$(A$,J,I)+32)
440 NEXT J
450 RETURN
1000 DATA LA,VOTURE,EST,ARRIVEE,QUAND,LE,TRAIN,LE,1917
1010 DATA EN,CARE,LE,BUREAU,DANS,LA,VOTURE,SOUS,LE
1020 DATA TOIT,AVAIT,ETE,VOLE,LA,RIVIERE
1030 DATA A,DEBORDE,PENDANT,LA,NUIT
1040 DATA FIN
```

**Exercice 8.3**

```
10 CLS
20 C$="BCDEFGHIJKLMNOPQRSTUVWXYZ"
30 V$="AEIOUY"
40 PRINT "POUR SORTIR UNE CONSONNE TAPEZ C"
50 PRINT "POUR SORTIR UNE VOYELLE TAPEZ V"
55 PRINT
60 FOR I=1 TO 9
70 PRINT "LETTRE NUMERO " ; I ;
80 INPUT R$
90 IF R$ <> "C" AND R$ <> "V" THEN 80
```

*LA DECOUVERTE DE L'ALICE ET DU MC/10*

```
100 IF R#="C" THEN A#=A#+ " "+MID$(C0$,RND(20),1)
110 IF R#="V" THEN A#=A#+ " "+MID$(V0$,RND(6),1)
120 PRINT TAB(16-LEN(A#))A#
130 PRINT
140 NEXT I
150 END
```

*Exercice 8.4*

```
10 REM
20 REM AFFICHAGE DES CARACTERES
30 REM ASCII DE CODE COMPRIS ENTRE 32 TO 127
40 REM
50 FOR I=32 TO 126 STEP 2
60 PRINT I;CHR$(I);I+1;CHR$(I+1)
70 REM
80 REM TEMPORISATION
90 REM
100 FOR T=1 TO 200:NEXT T
110 REM
120 NEXT I
130 END
```

*Exercice 9.1*

```
10 REM DESSIN D'UN CADRE DE COULEUR
15 CLEAR
20 INPUT "COULEUR DU FOND";CF
25 IF CF<0 OR CF>8 THEN 20
30 INPUT "COULEUR DU CADRE";CC
40 IF CC<0 OR CC>8 THEN 30
50 CLS CF
60 REM BARRES HORIZONTALES
70 FOR I=0 TO 63
80 SET (I,0,CC);SET (I,31,CC)
90 NEXT I
100 REM BARRES VERTICALES
110 FOR I=0 TO 31
120 SET (0,I,CC);SET (63,I,CC)
130 NEXT I
135 RESET(0,0)
140 GOTO 140
```

LA DECOUVERTE DE L'ALICE ET DU MC/10

Exercice 9.2

```
10 REM DESSIN D'UN CADRE DE COULEUR
15 CLEAR
50 CLS 5:CC=4
60 REM BARRES HORIZONTALES
70 FOR I=0 TO 63
80 SET (I,0,CC):SET (I,31,CC)
90 NEXT I
100 REM BARRES VERTICALES
110 FOR I=0 TO 31
120 SET (0,I,CC):SET (63,I,CC)
130 NEXT I
140 REM
150 X=5
160 Y=10
170 DX=1
180 DY=1
190 SET(X,Y,5):SET(X+DX,Y+DY,3)
200 X=X+DX:Y=Y+DY
210 IF X>59 OR X<3 THEN DX=-DX:SOUND 180,3:GOTO 190
220 IF Y>27 OR Y<3 THEN DY=-DY:SOUND 180,3:GOTO 190
230 GOTO 190
```

Exercice 9.3

```
10 REM
20 INPUT "NIVEAU DE DIFFICULTE (1-5)";NV
30 IF NV<1 OR NV>5 THEN 20
40 CLS 6
42 PRINT @32*15,:PRINT
45 PRINT @0," ":PRINT
50 FOR I=1 TO 10*NW
60 X=RND(22)+8:Y=RND(12)+1
70 PRINT @X+32*Y,CHR$(255);
80 NEXT I
90 GOTO 90
```

Exercice 9.4

```
10 REM
20 INPUT "NIVEAU DE DIFFICULTE (1-5)";NV
30 IF NV<1 OR NV>5 THEN 20
35 NC=1
40 CLS 6
42 PRINT @32*15,:PRINT
```

LA DECOUVERTE DE L'ALICE ET DU MC/10

```
45 PRINT @0,SC:PRINT @29,NC
50 FOR I=1 TO 10*NV
60 X=RND(22)+8:Y=RND(12)+1
70 PRINT @X+32*Y,CHR$(255);
80 NEXT I
90 X=0:Y=RND(12)+1
100 A$=INKEY$
110 IF A$="" THEN 200
120 IF A$="A" THEN DY=-1:GOTO 200
130 IF A$="P" THEN DY=1:GOTO 200
140 IF ASC(A$)=32 THEN DY=0
200 IF (Y<3 AND DY=-1) OR (Y>12 AND DY=1) THEN DY=0
205 X=X+1:Y=Y+DY
210 IF POINT(2*X,2*Y)>6 THEN 500
220 PRINT @X+32*Y,CHR$(175);:PRINT @X-1+32*(Y-DY),
CHR$(223);
230 IF X<31 THEN GOSUB 1000:GOTO 100
235 NC=NC+1
240 PRINT @X+32*Y,CHR$(223);
250 PRINT @14,"GAGNE";:SOUND 100,6
255 IF NC=10 THEN PRINT @14,"FIN":END
260 SC=SC+NV
270 NV=NV+1:GOTO 40
500 PRINT @X-1+32*(Y-DY),CHR$(223);
505 NC=NC+1
510 PRINT @14,"PERDU";:SOUND 128,5
515 IF NC=10 THEN PRINT @14,"FIN":END
520 NV=NV-1:IF NV<1 THEN NV=1
530 GOTO 40
1000 REM TEMPORISATION
1010 FOR T=1 TO 100/NV
1020 NEXT T
1030 RETURN
```

Exercice 9.5

```
10 CLS @
20 FOR I=24 TO 37
30 FOR J=8 TO 23
40 SET(I,J,6)
50 NEXT J
60 NEXT I
70 FOR I=26 TO 34 STEP 4
80 FOR J=10 TO 11
90 RESET(I,J)
100 NEXT J
110 NEXT I
120 FOR I=173 TO 177 STEP 2
```

LA DECOUVERTE DE L'ALICE ET DU MC/10

```
124 PRINT @I,CHR$(150);
127 NEXT I
130 PRINT @450,"POUR JOUER FRAPPEZ UNE TOUCHE";
140 IF INKEY$="" THEN 140
150 FOR I=1 TO 3:N(I)=RND(3):NEXT I
160 REM TESTS
170 G=-20
180 IF N(1)=1 AND N(2)=1 AND N(3)=1 THEN G=100
190 IF N(1)=1 AND N(2)=2 AND N(3)=1 THEN G=20
200 IF N(1)=1 AND N(2)=1 AND N(3)=2 THEN G=10
210 IF N(1)=2 AND N(2)=1 AND N(3)=1 THEN G=10
220 REM AFFICHAGE
230 FOR I=173 TO 177 STEP2
240 FOR T=1 TO 4
250 SOUND 140,2
260 P=(I-173)/2+1
270 FOR J=1 TO 3
280 PRINT @I,CHR$(134+16*J);
290 NEXT J
300 NEXT T
310 FOR J=1 TO N(P)
320 PRINT @I,CHR$(134+16*J);
330 NEXT J
340 NEXT I
350 PRINT @429,G;
360 GT=GT+G
370 PRINT @55,GT;
380 GOTO 130
```

Exercice 9.6

```
10 REM
20 REM AFFICHAGE CLIGNOTANT
30 REM
40 INPUT "ENTREZ UN NOM AYANT MOINS DE 20 CARACTERES";M$
50 L=LEN(M$):IF L>20 THEN 40
60 CLS 3
70 P=272-L/2
80 FOR I=1 TO L
90 M$=M$+CHR$(159)
100 NEXT I
110 PRINT @P,M$;
120 GOSUB 200
130 PRINT @P,M$;
140 GOSUB 200
150 GOTO 110
200 REM TEMPORISATION
205 FOR T=1 TO 100
```

LA DECOUVERTE DE L'ALICE ET DU MC/10

```
210 NEXT T
220 RETURN
```

*Exercice 9.7*

```
10 REM
20 REM JEU DE PILE OU FACE
30 REM
40 REM INITIALISATIONS
50 CLS 5
60 PRINT @137,"POUR CHOISIR:";
70 PRINT @201,"PILE TAPEZ P:";
80 PRINT @233,"FACE TAPEZ F:";
85 PRINT @265,"FIN TAPEZ X:";
90 P=P+1:A$=INKEY$:IF A$="" THEN 90
100 IF A$="X" THEN PRINT @0,;:END
110 IF A$<>"P" AND A$<>"F" THEN 90
120 PRINT @335,A$;
130 FOR T=1 TO 500:NEXT T
140 P=P-2*INT(P/2)
150 P$="F":IF P=0 THEN P$="P"
160 M$="PERDU":IF P$=A$ THEN M$="GAGNE":NG=NG+1
170 PRINT @397,M$;:SOUND 100,4
180 FOR T=1 TO 500:NEXT T
190 PRINT @397," ";
195 PRINT @335," ";
200 NC=NC+1
210 PRINT @67,NG;:PRINT @90,NC;
220 GOTO 90
```



# ANNEXE 6

## I N D E X

ABS	80	LIST	34
Addition	18	Listage	49-72-75
Affectation	28	Lister	35
Aléatoire	80	LOG	82
Alphanumérique	25	Mantisse	18
AND	30	MEM	76
ASC	91	Menu	56-64-68
ASCII	91	MID\$	86
Basic	10-12	Minuscules	10
Boucle	60-89	Multiplication	19
Branchement	53	Musique	102
Break	53-74	NEW	38
BS Error	23-24	NEXT	59
Caractères graphiques	10-97	NF Error	61
Cassette	37-45	Noms de variables	21
Chaîne de caractères	13-25-86	NOT	30
Chargement d'un programme	37	Notation scientifique	18
Chargement de données	45	OD Error	44
CHR\$	90	ON GOSUB	66
CLEAR	24-26	ON GOTO	55
CLOAD	37	OR	30
CLOAD*	45	Organigramme	57
CLS	47-96-100	PEEK	75
Comparaisons'	27	POINT	101
Concaténation	88	POKE	76
CONTROL	11-16-37	PRINT	12-14-16-47-97
COS	82	Programme	34-52-65
CSAVE	37	READ	42
CSAVE*	45	REM	49
Curseur	10-34	RESET	98
		RESTORE	42

LA DECOUVERTE DE L'ALICE ET DU MC/10

DATA	42	RETURN	66
DD Error	24	RIGHT\$	86
DIM	24	RND	80
Division	19	RUN	35
Effacement	15-36	SET	98
Egal	20-28	SGN	81
Élévation à une puissance	19	SHIFT	10-74
END	36-66	SIN	82
Enregistrement d'un programme	37	SKIPF	37
Enregistrement de données	45	sous-programme	65
ENTER	11-12-15-34-36	soustraction	19
EXP	82	SN Error	12
Exposant	18	SOUND	102
FC Error	56	SQR	82
FOR..TO	59	STEP	59
GOSUB	66	STOP (instruction)	73
GOTO	52-54	STR\$	89
Guillemets	13-44	Supérieur	27
IF..THEN	56-72	TAB	47
Inférieur	27	Tableaux	22-27
Initialisation	75	TAN	82
INKEY\$	96	UL Error	54
INPUT	38	VAL	89
INT	81	Valeur absolue	80
Interrupteur	10	Valeur entière	81
Interruption programme (touche)	53-74-75	Variables alphanumériques	25-86
LEFT\$	86	Variables numériques	20
LEN	86	VARPTR	76
LET	20	Zéro	11

Achévé d'imprimer en mars 1984  
sur les presses de l'imprimerie Laballery et C<sup>o</sup>  
58500 Clamecy  
Dépôt légal : mars 1984

N<sup>o</sup> d'impression : 402090  
N<sup>o</sup> d'édition : 86595-133-1  
ISBN : 2-86595-133-2