

hors série

# Led

# MICRO

APPRENDRE • La programmation, cours de Claude Polgar  
 • Le Microprofessor MPF 1B, cours pratique de microprocesseur par Philippe Duquesne. • Les fortiches : un éditeur de tableau. MAGAZINE • Choisir son imprimante  
 • La contre-mesure du Laser 3000 • Compte rendu SICOB.

## DE L'INITIATION A LA PRATIQUE DE L'INFORMATIQUE

COURS  
**N°13**



### LE SX 64 PORTABLE DE COMMODORE

ISSN 0757-6889



# MICROPROCESSEURS

**COMPRENDRE**  
leur fonctionnement

**CONCEVOIR - RÉALISER**  
vos applications



Z 80  
R 6502  
6809

## MPF-1 B

- MICROPROCESSEUR Z-80®, haute performance, répertoire de base de 158 instructions.
  - 4 Ko ROM (moniteur + mini interpréteur BASIC). 2 Ko RAM.
  - Clavier 36 touches dont 19 commandes. Accès aux registres. Programmable en langage machine.
  - 6 afficheurs L.E.D. Interface K7.
  - Options : 4 Ko EPROM ou 2 Ko RAM, CTC et PIO.
- Le MICROPROFESSOR MPF-1 B est parfaitement adapté à l'initiation de la micro-informatique. Matériel livré complet, avec alimentation, prêt à l'emploi, manuels d'utilisation (en français), applications et listing.

Prix TTC, port inclus - 1 495 F



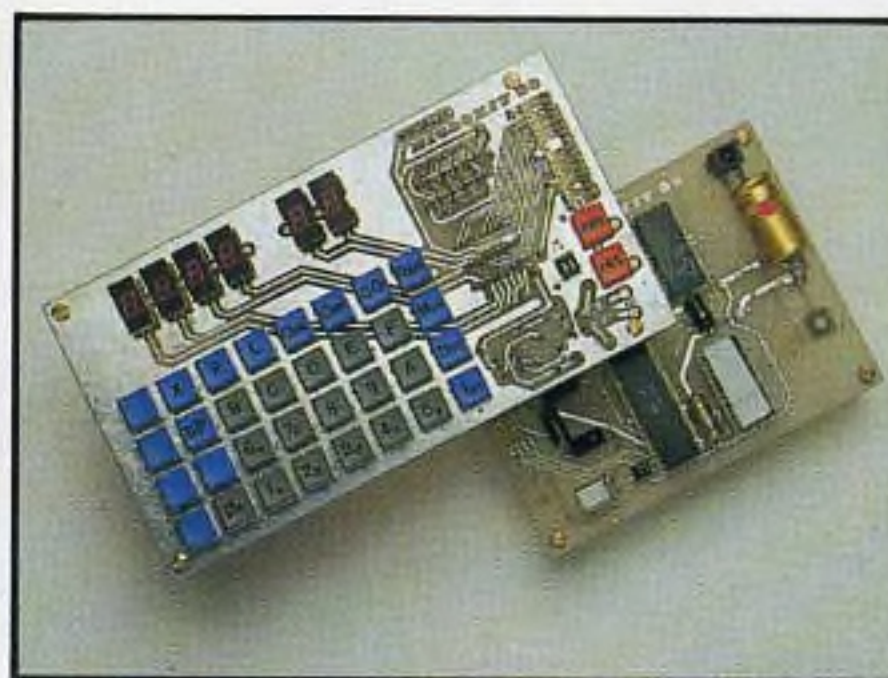
## MPF-1 PLUS

- MICROPROCESSEUR Z-80®, 8 Ko ROM, 4 Ko RAM (extensible).
  - Clavier QWERTY, 49 touches mécaniques avec « Bip ».
  - Affichage alphanumérique 20 caractères (buffer d'entrée de 40 caractères). Interface K7, connecteur de sortie.
  - ÉDITEUR, ASSEMBLEUR, DEBUGGER résidents (pointeurs, messages d'erreurs, table des symboles, etc.).
  - Options : 8 Ko ROM-BASIC, 8 Ko FORTH.
  - Extensions : 4 Ko ou 8 Ko EPROM, 8 Ko RAM (6264).
- Le MICROPROFESSOR MPF-1 PLUS est à la fois un matériel pédagogique et un système de développement souple et performant. Matériel livré complet, avec alimentation, notice d'utilisation et d'application en français, listing source du moniteur.

Prix TTC, port inclus - 1 995 F.

## MODULES COMPLÉMENTAIRES POUR MPF-1B ET MPF-1 PLUS

- PRT-MPF B ou PLUS, imprimante thermique
- SSB-MPF B ou PLUS, synthétiseur de paroles.
- SGB-MPF B ou PLUS, synthétiseur de musique.
- EPB-MPF-1B/PLUS, programmeur d'EPROMS.
- TVB-MPF-1 PLUS, interface vidéo pour moniteur TV.
- I.O.M. - MPF-1 PLUS, carte entrée/sortie et mémoire (6 Ko).



## MICROKIT 09

- MICROPROCESSEUR 6809, haut de gamme, organisation interne orientée 16 bits. Compatible avec 6800, programme source 2 Ko EPROM (moniteur). 2 Ko RAM. Clavier 34 touches. Affichage 6 digits. Interface K7. Description et applications dans LED.
- Le MICROKIT 09 est un matériel d'initiation au 6809, livré en pièces détachées.

## MPF-1/65

- MICROPROCESSEUR 6502, haute performance, bus d'adresses 16 bits, 56 instructions, 13 modes d'adressage. 16 Ko ROM. 64 Ko RAM Dynamiques. Clavier 49 touches avec 153 codes ASCII distincts. Affichage sur moniteur ou TV : 24 lignes de 40 caractères.
  - ÉDITEUR, ASSEMBLEUR, DEBUGGER résidents.
  - Interface K7 à 1 000 bps. Connecteurs pour imprimante et extension.
- Matériel livré complet avec alimentation (+ 5V, - 5V et 12V). Notice d'utilisation et listing source. Prix TTC, port inclus - 2 995 F.

LES MICROPROFESSORS SONT GARANTIS 1 AN PIÈCES ET MAIN-D'ŒUVRE

**MICROPROFESSOR EST UNE MARQUE DÉPOSÉE MULTITECH**

SI VOUS VOULEZ EN SAVOIR PLUS : TÉL. : 16 (4) 458.69.00

BON DE COMMANDE À RETOURNER À Z.M.C. B.P. 9 - 60580 COYE-LA-FORET

- |  |  |
|--|--|
| <input type="checkbox"/> MPF-1 B - 1 495 F TTC       | <input type="checkbox"/> IOM AVEC RAM - 1 795 F TTC  |
| <input type="checkbox"/> MPF-1 PLUS - 1 995 F TTC    | <input type="checkbox"/> TVB PLUS - 1 695 F TTC  |
| <input type="checkbox"/> MPF-1/65 - 2 995 F TTC      | <input type="checkbox"/> OPTION B BASIC PLUS - 400 F TTC   |
| <input type="checkbox"/> PRT B OU PLUS - 1 095 F TTC | <input type="checkbox"/> OPTION FORTH PLUS - 400 F TTC   |
| <input type="checkbox"/> EPB B/PLUS - 1 795 F TTC    |  |
| <input type="checkbox"/> SSB B OU PLUS - 1 595 F TTC | DOCUMENTATION DÉTAILLÉE  |
| <input type="checkbox"/> SGB B OU PLUS - 1 095 F TTC | <input type="checkbox"/> MPF-1 B <input type="checkbox"/> MPF-1/65 <input type="checkbox"/> MPF-1 PLUS |
| <input type="checkbox"/> IOM SANS RAM - 1 495 F TTC  | <input type="checkbox"/> MICROKIT - LISTE ET TARIF   |

NOM : \_\_\_\_\_

ADRESSE : \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Ci-joint mon règlement  
(chèque bancaire ou C.C.P.)

Signature et date : \_\_\_\_\_



LO... ROUES D'AUJOURD'HUI

**hors série**

# LED

# MICRO

OCTOBRE 84

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

**LED MICRO**

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

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

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

**Rédaction**  
**Rubriques pédagogiques**  
**Chef de rubriques :**  
 Claude Polgar,  
 Ont collaboré à ce numéro :  
 Philippe Duquesne, Bruno  
 Lilamand

**Rubriques Magazine**  
**Chef de rubrique :**  
 Claude-Hélène Roze  
 Ont collaboré à ce numéro :  
 Charles-Henry Delaleu, P.F.,  
 Claude Polgar

Secrétaire de Rédaction  
 Chantal Cauchois  
 assistée de  
 Marianne Bergère  
 Réalisation graphique  
 Serge Fayol

**Publicité, à la revue**  
 Tél. : 607.01.97  
 Chef de publicité  
 Frédérique Rotelli  
 Secrétaire responsable  
 Annie Perbal

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

**Réalisation**  
 Composition-Photogravure  
 Edi'Systèmes  
 Montage  
 Valérie Martineau  
 Impression  
 Berger-Levrault - Nancy

**RUBRIQUES PEDAGOGIQUES**



**7**

**COURS DE PROGRAMMATION  
 EN BASIC**

Initiation progressive à l'informati-  
 que  
 par **Claude Polgar**

**30**

**COURRIER DES LECTEURS**

Claude Polgar répond...

**32**

**COURS PRATIQUE DE MICRO-  
 PROCESSEUR**

avec le Microprofessor MPF 1B  
 par **Philippe Duquesne**

**44**

**LE COIN DES FORTICHES**

Un éditeur de tableau  
 par **Bruno Lilamand**

**48**

**LIBRES PROPOS**

Réflexions sur la micro-informatique  
 par **Charles-Henry Delaleu**

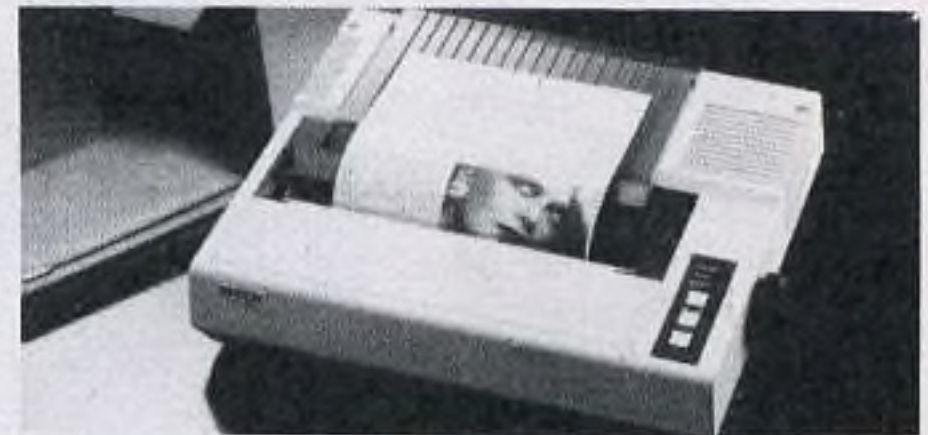
*3 63-13  
 20/24*

**RUBRIQUES MAGAZINE**

**50**

**LES IMPRIMANTES**

Le choix dépend de l'usage  
 par **Claude-Hélène Roze**



**54**

**LES CONTRE-MESURES**

Le Laser 3000 5' 1/4  
 par **Charles-Henry Delaleu**

**59**

**COMPTE RENDU SICOB**

par **Claude-Hélène Roze**



**64**

**BIBLIOGRAPHIE**

A lire  
 par **Philippe Faugeras**

**NOTRE COUVERTURE :** Le SX64 portable de Commodore, le clavier professionnel détachable et l'écran couleur.





# L'incroyable TI-66 programmable. Des performances exceptionnelles à un prix exceptionnel.

MCCANN-ERICKSON

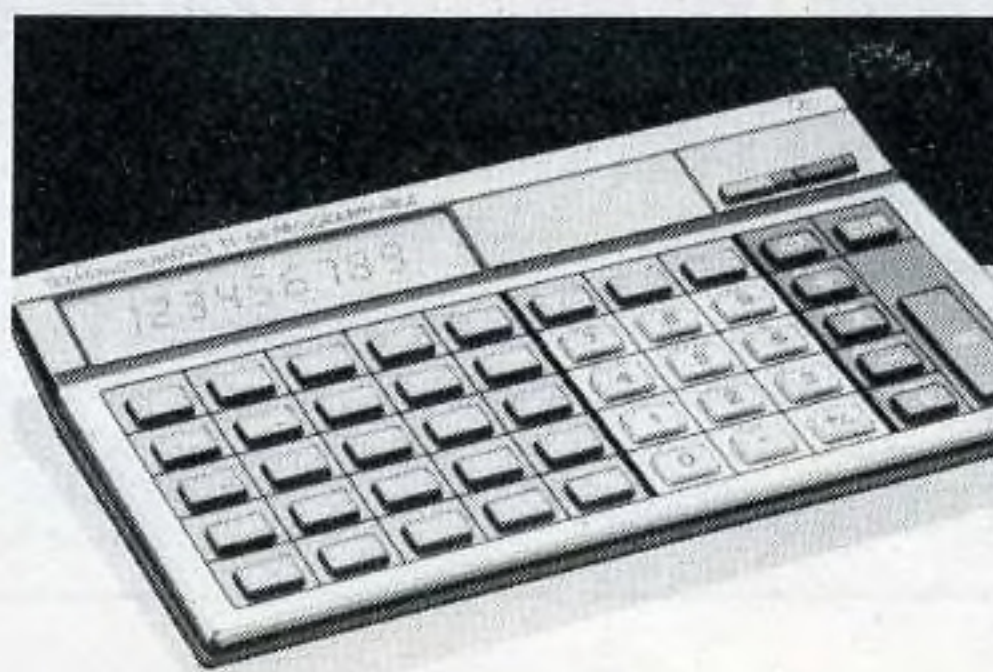
La calculatrice TI-66 programmable de Texas Instruments fait partie d'une classe à part. Comparée à toutes les autres, elle vous offre des performances exceptionnelles à un prix qui l'est aussi.

Elle a tout ce dont vous avez besoin pour vous donner un réel avantage et faire face aux problèmes professionnels les plus épineux ou aux études supérieures les plus poussées.

Jugez vous-même ses performances : la TI-66 programmable a toutes les fonctions nécessaires pour venir à bout des calculs répétitifs et séquentiels. Elle peut comprendre jusqu'à 512 niveaux de programme, et possède

64 mémoires de données. De plus, la TI-66 de Texas Instruments peut être connectée à l'imprimante PC 200.

Performance supplémentaire :



vous pouvez y entrer les programmes de la TI-58C (qui couvrent une très large gamme de fonctions indispensables à certaines tâches professionnelles). Vous pouvez également développer ou concevoir vos propres programmes.

Essayez la TI-66 programmable de Texas Instruments. Vous découvrirez une calculatrice remarquable qui, à prix égal, possède des qualités supérieures.



## TEXAS INSTRUMENTS



# COURS DE PROGRAMMATION

## OU EN SOMMES-NOUS ? OU ALLONS-NOUS ?

Nous avons étudié successivement :

1) Les notions générales sur l'Informatique et les Ordinateurs (LED MICRO n° 1, 2, 3, 4 et 5)

2) Les notions fondamentales du BASIC (LED MICRO n° 6, 7 et 8)

3) Quelques notions pratiques sur les systèmes d'Exploitation (LED MICRO n° 9, 10, 11 et 1ère partie du n° 12).

Depuis la fin du numéro 12, nous avons repris l'étude systématique du BASIC.

Dans le chapitre 3.15 (en cours) nous étudions **l'affectation**.

Dans le (futur) chapitre 3.16 nous examinerons la **sélection**

Dans le chapitre 3.17 les **boucles**.

Dans le chapitre 3.18 les **sous-programmes**.

Puis nous aborderons la quatrième (et dernière) partie du cours d'initiation (cette « première couche ») en parlant « méthode » : organigramme, programmation, structure, mise au point.

Comme vous pouvez le constater, nous progressons avec une sage lenteur, en approfondissant chaque notion nouvelle : nous nous sommes engagés à vous fournir une base solide.

Au fur et à mesure que notre capital de connaissances s'accroîtra, nous pourrons vous proposer des exercices plus complexes et plus distrayants. Ceux de nos lecteurs qui nous réclament des problèmes plus difficiles seront bientôt satisfaits.

## EXERCICES D'APPLICATION ET EXERCICES DE RECAPITULATION

Nous vous rappelons la distinction que nous faisons entre nos « exercices d'application » et nos « exercices de récapitulation » :

Les **exercices d'application** sont des exercices très courts et très faciles, qui forment une illustration immédiate du cours. Dans les cours oraux, ces exercices sont faits immédiatement par les élèves dès que le professeur a fourni les définitions et explications nécessaires. Dans le présent cours de LED MICRO, ces exercices doivent, de même, être effectués par les élèves au fur et à mesure de leur lecture : ils font partie intégrante du cours.

Les **exercices de récapitulation** sont des exercices un peu plus longs, pouvant faire appel à des notions apprises il y a quelque temps (et peut-être oubliées), pouvant comporter des pièges, etc.

Dans les cours oraux, ces exercices sont le plus souvent effectués et entrés sur leur micro-ordinateur par les élèves en séance de Travaux Pratiques. Dans le présent cours de LED MICRO, ces exercices sont proposés en fin de leçon. Nous remercions ceux de nos lecteurs qui veulent bien nous envoyer leurs réponses : cela nous permet de vérifier que notre cours est assimilé par la majorité d'entre eux (et de corriger le tir par des notes complémentaires si cela s'avérait nécessaire).

Le présent numéro de LED MICRO ne comporte pas d'exercices de récapitulation. Vous en aurez bientôt.

## UN ENSEIGNEMENT A PLUSIEURS COUCHES

Comme nous l'avons déjà dit, l'actuelle 3<sup>e</sup> partie de notre cours constitue une « première couche » d'enseignement. Par exemple dans les chapitres 3.15.6 et 3.15.7 de ce numéro, nous ne faisons que citer l'existence de quelques fonctions et opérateurs sur les chaînes de caractères (MID\$, CHR\$, LEN, STR\$...) : nous reviendrons en détail sur toutes ces fonctions lors de la « deuxième couche », c'est-à-dire lorsque vous disposerez de suffisamment de connaissances pour pouvoir utiliser ces fonctions et/ou opérateurs dans des exercices : apprendre à un débutant une foule d'instructions qu'il ne pratiquera pas immédiatement, c'est se moquer de lui.

De la même façon, nous ne faisons que rappeler comment BASIC « déclare » la précision d'un nombre (caractères #, ! et % à la fin du nom de la variable) pour que l'élève ne soit pas surpris de rencontrer ces signes bizarres (en lisant des programmes de jeu prêts à être entrés tels quels par exemple), mais nous n'étudierons « vraiment » ce problème que lorsque, lors de la « deuxième couche », nous nous entraînerons à « optimiser » nos programmes et à éviter différents pièges de « conversion » ou « d'arrondi ».

Par contre, les (peu nombreuses) notions nouvelles que nous aborderons ici seront étudiées avec une surabondance d'explications et d'exercices : rien ne doit être laissé dans l'ombre si on veut s'assurer au départ cette fameuse « base solide ».

## NOUS NE VOUS APPRENONS PAS QUE LE BASIC

« Le BASIC s'apprend en 2 heures ».

Faut-il traiter de menteurs ceux qui l'affirment ? Pas forcément. J'ai vu mon ami M. de la B. assis à son bureau devant moi feuilleter le manuel BASIC du Personal Computer d'IBM. Il n'avait jamais écrit une seule ligne en BASIC et pourtant deux heures plus tard il avait déjà rédigé et fait tourner un petit programme de tri en BASIC.

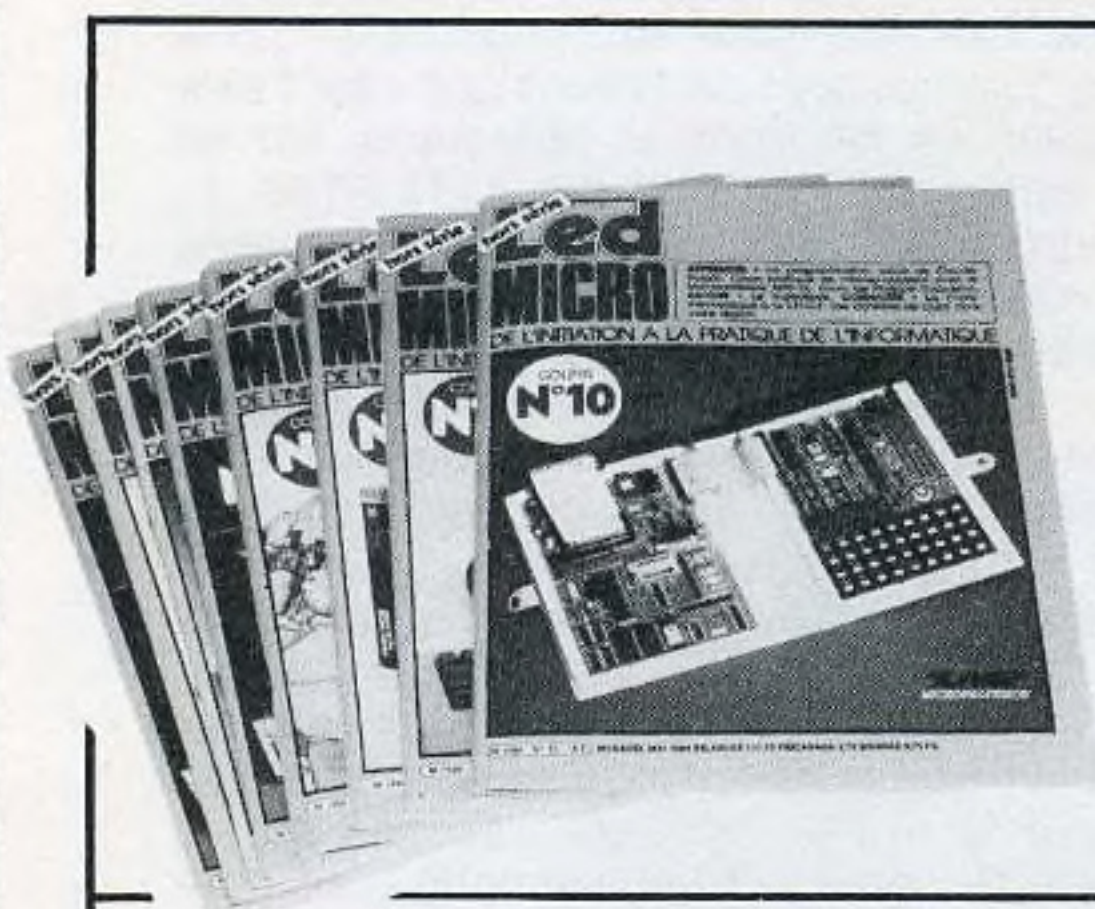
Seulement, M. de la B. avait derrière lui 10 ans de programmation en FORTRAN, COBOL et divers assembleurs. Il savait programmer.

Nous supposons que nos lecteurs n'ont pas l'expérience de Michel. Il nous faut leur apprendre à la fois le BASIC et la programmation.

Si nous ne voulions que vous distraire, nous ne vous ferions pas « approfondir » la notion de variable pendant plus de 2 numéros (1/2 LED MICRO n° 12 + LED MICRO n° 13 et n° 14). En travaillant comme nous vous le proposons, vous assimilerez toute une collection de notions telles que : type de variable, déclaration, affectation, opérateur, de sorte que, sans que vous vous en rendiez compte, vous êtes (presque) prêts à aborder le PASCAL rapidement et sans douleur.



■ Vous découvrez Led-Micro avec ce n°13  
 La partie cours vous intéresse et vous désirez  
 l'ensemble des numéros parus (depuis le n°1)  
*Voici ce que nous vous proposons :*



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

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



les 10 premiers  
 numéros agrafés  
 dans leur reliure\*  
**180 F PORT COMPRIS**

\* Pour obtenir la reliure seule, voir  
 notre publicité en page 58

**SOMMAIRE DES COURS**

**N°1** Introduction générale - Vocabulaire et notions de base - L'emploi des ordinateurs • Fonctions de base

**N°2** Configuration d'un système - L'unité centrale et ses interfaces - Ecran, clavier, imprimante • Opérateurs de base

**N°3** Disquettes et cassettes - Machine à dessiner, numériseur, photostyle, souris • Opérateurs de base

**N°4** Langages compilés et interprétés - Les systèmes d'exploitation - Les progiciels - Classification et choix d'un micro • Opérateurs de base

**N°5** Choisir, installer, brancher - La pratique du clavier - Mise en route • Arithmétique binaire

**N°6** Premier programme en Basic - Ponctuation dans le Print - Exercices sur le Print • Arithmétique binaire

**N°7** Déroulement d'un programme - Représentation des nombres - Corrigé d'exercices • Les bascules

**N°8** Calculs en BASIC - Corrigé d'exercices - Les registres - Les compteurs

**N°9** Notion de format - Le NEWDOS - Corrigé d'exercices • Architecture d'un système à microprocesseur

**N°10** Le NEWDOS (fin) - Le CP/M80 - Les registres du Z80 - Déroulement d'un programme - L'U.A.L.

**N°11** Utilisation d'un fichier enregistré en MBASIC (sous CP/M) - Le formatage • Le hardware du MPF-1B

**N°12** L'affectation - Variables chaînes booléennes - Le langage du Z80<sup>R</sup>

■ Vous désirez un ou plusieurs numéros qui vous  
 manquent (de 1 à 12): **17 F par numéro** PORT COMPRIS

**BON DE COMMANDE**

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

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

Je désire : 1 collection complète des 10 premiers numéros **reliés**

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

1 2 3 4 5 6 7 8 9 10 11 12



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

Mon nom : ..... prénom : .....

Mon adresse : ..... Code postal .....



# COURS DE PROGRAMMATION(13)

## TROISIEME PARTIE (SUITE)

### Premiers travaux sur ordinateur

<b>3. 1.</b> But et contenu de cette 3 <sup>e</sup> partie <b>3. 2.</b> Les systèmes types <b>3. 3.</b> Choisir, installer, brancher <b>3. 4.</b> La pratique du clavier <b>3. 5.</b> De la mise en route au caractère d'attente	LED-MICRO n° 5
<b>3. 6.</b> Un premier programme en Basic <b>3. 7.</b> Modifications et complétons ce programme <b>3. 8.</b> La ponctuation dans le PRINT <b>3. 9.</b> Exercices sur le PRINT	LED-MICRO n° 6
<b>3.10.</b> Le déroulement d'un programme <b>3.11.</b> Nombres et calculs (1 <sup>re</sup> partie : les nombres)	LED-MICRO n° 7
<b>3.11.</b> Nombres et calculs (2 <sup>e</sup> partie : les calculs)	LED-MICRO n° 8
<b>3.12.</b> Conventions et notations <b>3.13.</b> Notions élémentaires sur les fichiers et les SED (1 <sup>re</sup> partie)	LED-MICRO n° 9
<b>3.13.</b> Notions élémentaires sur les fichiers et les SED (2 <sup>e</sup> partie)	LED-MICRO n° 10
<b>3.13.</b> Notions élémentaires sur les fichiers et les SED (3 <sup>e</sup> partie)	LED-MICRO n° 11
<b>3.14.</b> Complément sur le listage <b>3.15.</b> L'affectation. Variables numériques	LED-MICRO n° 12
<b>3.15.</b> L'affectation (suite). Variables chaînes et booléennes.	LED-MICRO n° 13



### 3.15.4. Variables chaînes de caractères

#### A. Rafraichissons nos connaissances

Si vous ne vous souvenez pas :	Relisez :
— de ce qu'est une chaîne de caractères	LM n° 6 - chapitre 3.8 (pages 24 à 27)
— de ce qu'est une variable numérique — de l'affectation d'une valeur à une variable numérique	LM n° 7 - § 3.11.3 (pages 24 et 25)
	LM n° 12 - § 3.15.1 à 3.15.3 (pages 22 à 26)
— de ce que l'on appelle «chaîne vide» — de la façon de réaliser des dessins avec des chaînes de caractères	LM n° 6 - §3.8.1.C (page 25)

#### B. Variables «chaînes de caractères»

Considérons le programme ci-dessous :

```

10 REM PROGRAMME AFFE04
20 HOME
30 LET M$ = "MONSIEUR"
40 LET D$ = "MADAME"
50 LET B$ = "BONJOUR"
60 PRINT B$ ; D$
70 PRINT B$ , M$
    
```

et suivons son exécution pas à pas :

30 LET M\$ = "MONSIEUR"	L'ordinateur prend la chaîne de caractères "MONSIEUR" et la met dans une petite boîte qu'il appelle M\$ [prononcez : ème-dollar] Voir dessin page ci-contre.
40 LET D\$ = "MADAME"	Procédé identique. Disons la même chose avec d'autres mots : «L'ordinateur affecte la chaîne "MADAME" à la variable D\$».
50 LET B\$ = "BONJOUR"	Procédé identique : B\$ ← "BONJOUR"
60 PRINT B\$ ; D\$	L'ordinateur affiche sur l'écran le contenu des cases B\$ et D\$. Comme ces deux arguments sont séparés par un point-virgule, il affiche ces deux mots «collés» l'un à l'autre, à savoir : BONJOURMADAME
70 PRINT B\$ , M\$	L'ordinateur affiche sur l'écran le contenu des cases B\$ et M\$. Comme ces deux arguments sont séparés par une virgule, il affiche ces deux mots dans des colonnes-mots différentes, à savoir : BONJOUR MONSIEUR



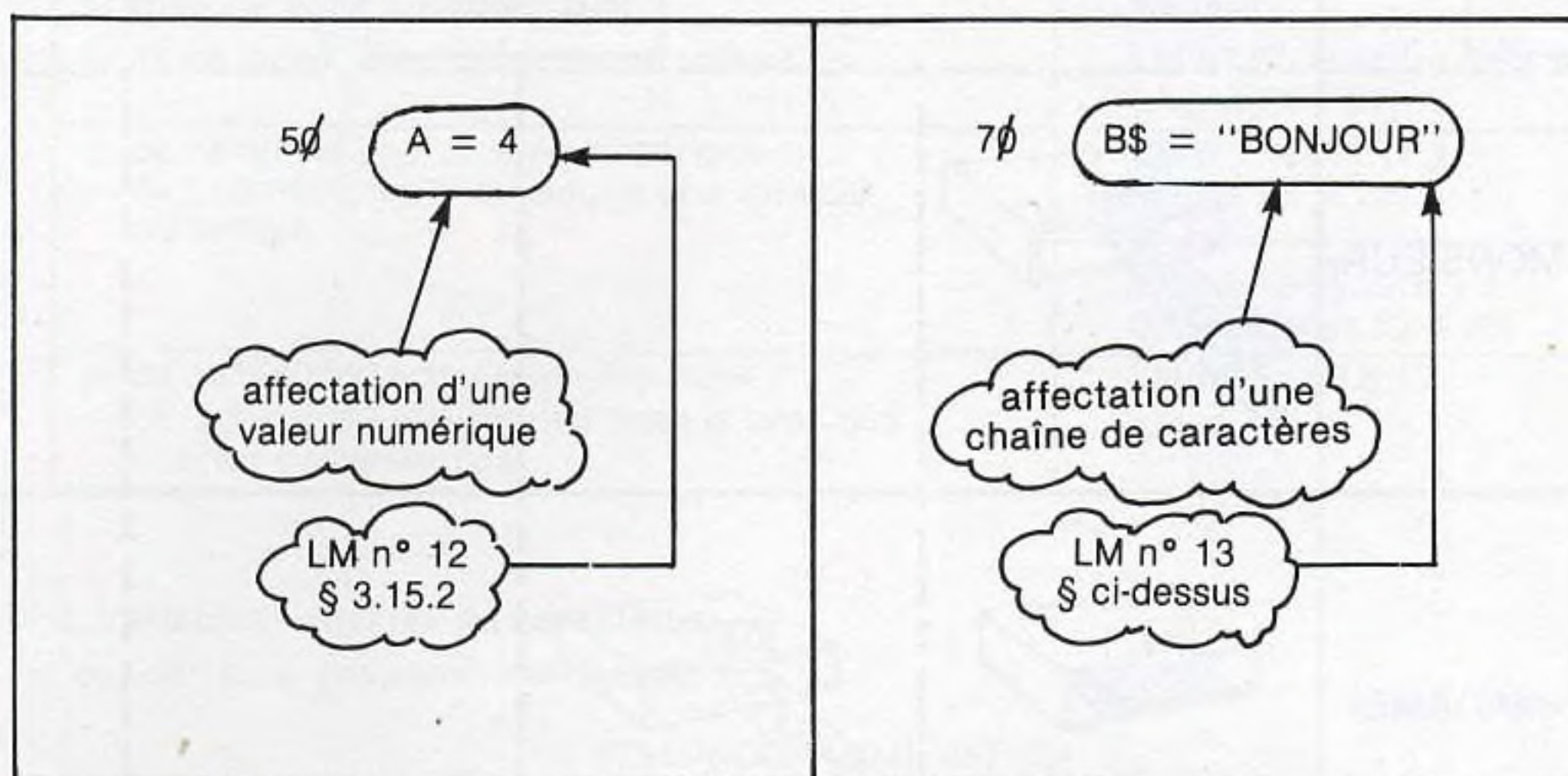
C. Bonjour Madame, bonjour Monsieur

Programme Basic	Illustrations				
30 LET M\$ = «MONSIEUR»					
40 LET D\$ = «MADAME»					
50 LET B\$ = «BONJOUR»					
60 PRINT B\$ ; D\$					
70 PRINT B\$, M\$					



## D. Déclaration du type «variables chaîne»

Si on compare :



On remarque que, en BASIC :

- les cases destinées à contenir une chaîne de caractères portent un nom terminé par le signe \$ (dollar) ;
- les cases destinées à contenir une valeur numérique portent un nom... qui n'est pas terminé par le signe \$.

On peut dire en parlant «moins bébé» : «les noms des variables chaînes de caractères se terminent par le signe \$».

**Note aux puristes :** Il existe dans les Basics évolués des instructions qui permettent d'affecter une chaîne de caractères à une variable dont le nom ne se termine pas par un \$... Ne nous compliquons pas la vie pour le moment !

## E. Longueur d'une chaîne

Vous ne pouvez pas ranger dans une chaîne un nombre quelconque de caractères. Il y a une limite à la capacité des variables chaînes. Dans le cas du BASIC de l'IBM PC cette capacité limite est de 255 caractères : il y a de quoi faire (plus de 3 lignes de 80 caractères !). Nous avons vu (LED MICRO n° 6 § 3.8.1.C bas de la page 25) que le BASIC considère comme «valides» des chaînes vides, c'est-à-dire des chaînes de longueur nulle.

## F. Valeur initiale d'une chaîne de caractères

- Tant que l'on ne leur a pas affecté de valeurs numériques particulières, le BASIC considère que les variables numériques ont des valeurs nulles (voir LM n° 7 §3.11.3).
- De façon analogue, tant que l'on n'a pas affecté une chaîne de caractères particulière à une variable chaîne, le BASIC considère que cette chaîne est «vide».
- Ne confondez pas «chaîne vide» et «valeur numérique égale à 0».

## G. L'affectation est plus forte que la pile Wonder

Le programme AFFE04 ci-dessus affiche le contenu de B\$ trois fois de suite : à la ligne 80, à la ligne 90 et à la ligne 100.

Il pourrait l'utiliser 1 000 fois sans l'user.

L'explication que nous avons donnée dans LED MICRO n° 12 paragraphe 3.15.4 en assimilant l'affectation au remplissage d'un récipient n'est pas tout à fait exacte : on remplit C d'huile... mais il reste toujours de l'huile dans A... tant que l'on n'y a pas mis autre chose.

On ne «vide» pas la valeur affectée à une variable : on «l'écrase» en la remplaçant par une autre valeur.



## H. Affichage

Considérons le programme:

```
10 CLS
20 A = 5
30 A$ = "5"
40 PRINT A
50 PRINT A$
```

Voici ce qui apparaît à l'écran lorsqu'on en commande l'exécution.

L'affichage de la valeur numérique 5 est précédé d'un espace. Cet espace apparaît chaque fois qu'on fait afficher un nombre positif. Avec une valeur négative, cet espace est rempli avec le signe -

Ici 5 n'est pas un nombre, mais une chaîne de caractères, pour laquelle les notions de «positif» ou «négatif» n'ont pas de sens.

	1	2		5	6	7		10						15	
1				5											
2				5											
3															
4															
5															
6															

Nous avons déjà remarqué cette différence (LED MICRO n° 6 § G3.8.1.A page 24.)

## I. Ne confondez pas les types !

Nous avons défini deux «types» de variables :

- le type numérique
  - le type chaîne de caractères.
- Nous en verrons d'autres très bientôt :
- le type nombre entier
  - les types simple précision et double précision
  - le type booléen.

Lorsque le moment sera venu, nous définirons :

- le type «tableau numérique»
- le type «tableau de chaînes».

Le PASCAL impose que le programmeur commence par «déclarer» le type de chacune des variables qu'il utilisera. Le BASIC est plus expéditif : il suffit de placer des guillemets ou un signe \$ au moment où l'on crée une chaîne de caractères pour qu'il comprenne... mais il ne faut pas l'oublier.

## J. Un peu d'anglais Informatique

Type mismatch	Type de valeur non concordant
Out of string space	Dépassement d'espace chaîne
String too long	Chaîne trop longue
Overflow	Dépassement
Integer	Entier (nombre)



### K. Rappel sur «l'écran fictif standardisé»

Nous avons défini dans LED MICRO n° 8 paragraphe G3.11.5.F (page 22) un «écran fictif standardisé».

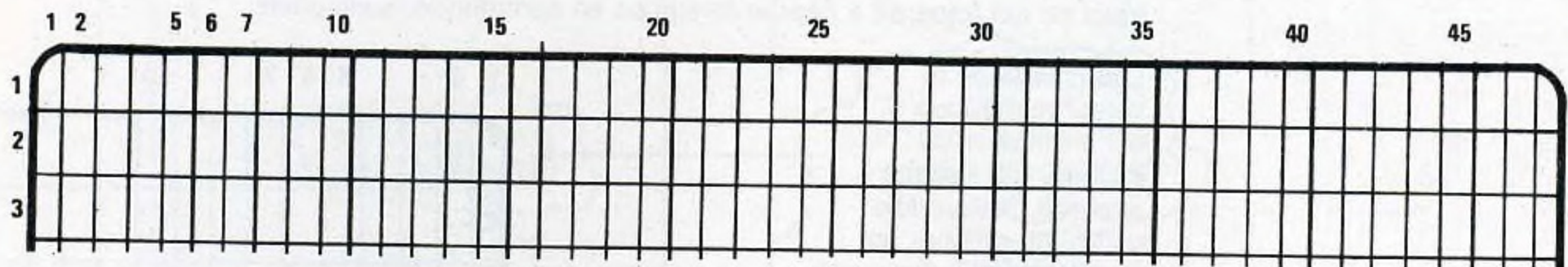
Cet écran comporte 16 lignes de 48 colonnes.

Lorsque l'on utilise des virgules dans le PRINT, ces virgules divisent l'écran en 3 tranches verticales de 16 colonnes chacune.

Le mot «colonne» peut prêter à confusion. Lorsque c'est nécessaire nous distinguerons :

— les «colonnes-caractères», qui sont les 48 colonnes de cet écran, chaque case de ces colonnes peut contenir un caractère et un seul.

— les «colonnes-mots», qui sont les 3 tranches regroupant 16 colonnes-caractères.



### L. Exercice d'application A17

Essayez de prévoir ce que donne sur notre «écran fictif standardisé» l'exécution du programme ci-dessous. La solution se trouve page ci-contre. Ne trichez pas. Prenez un crayon et écrivez sur du papier quadrillé.

```
1 0 P $ = " P A P A "
2 0 M $ = " M A M A N "
3 0 B $ = " B E B E "
4 0 C L S
5 0 P R I N T P $ ; M $ ; B $
6 0 P R I N T P $ ; " " ; M $ ; " E T " ; B $
7 0 P $ = M $
8 0 P R I N T P $ , M $ , B $
```

Solution page ci-contre.

### M. Exercice d'application A18

Trouvez les erreurs contenues dans le programme ci-dessous.

```
1 0 R E M P A S B O S U R R I E N D U T O U T
2 0 N E W
3 0 B = " B O N S O I R "
4 0 C $ = L E S C A M A R A D E S
5 0 P R I N T B : C $
6 0 P R I N T " C $ " ; " " ; B
7 0 E N D
```

Solution page ci-contre.

### N. Exercice d'application A19

Trouvez les erreurs contenues dans le programme ci-dessous.

```
1 0 R E M P A S M I E U X
2 0 6 = A
3 0 P R I N T A
4 0 A = B = C = 3
```

Solution page ci-contre.



### O. Solution de l'exercice A17

	1	2	5	6	7	10	15	20	25	30	35	40	45			
1	P	A	P	A	M	A	M	A	N	B	E	B	E			
2	P	A	P	A	M	A	M	A	N	E	T	B	E	B	E	
3	M	A	M	A	N			M	A	M	A	N	B	E	B	E
4																

### P. Solution de l'exercice A18

- Ligne 10 Pas d'erreur : on peut mettre n'importe quoi après un REM.  
 Ligne 20 Etourderie qui a des conséquences graves : il faut mettre CLS. Voir LED-MICRO n° 7, paragraphe G3,10.4.E page 16 et paragraphe 3.10.9 page 21.  
 Ligne 30 "BONSOIR" est une chaîne de caractères. Il faut donc l'affecter à une variable chaîne, se terminant par \$. D'autre part, il vaut mieux écrire ce mot avec les lettres O plutôt qu'avec les chiffres 0. Il aurait donc fallu écrire :

30 B\$ = "BONSOIR"

- Ligne 40 Une chaîne de caractères doit être entourée de guillemets.

40 C\$ = "LES CAMARADES"

- Ligne 50 Supposons que B ait été remplacé par B\$.  
 L'ordinateur refusera de comprendre cette ligne parce que le programmeur a confondu : (deux points) avec ; (point virgule).

Le symbole : permet d'écrire plusieurs instructions sur une même ligne de programme (voir LED-MICRO n° 7 - § 3.10.5 page 17) et un C\$ tout seul ne saurait constituer une instruction, il faudrait (par exemple) le faire précéder d'un PRINT.

Supposons que l'on remplace donc le : par un ; l'ordinateur comprendra — mais affichera les deux mots collés l'un à l'autre. Ce n'est pas très joli. Il aurait été préférable d'insérer un espace entre ces deux arguments, comme on a essayé de le faire à la ligne 60.

- Ligne 60 Du fait que l'on a mis C\$ entre guillemets, l'ordinateur va afficher sur l'écran C\$ et non LES CAMARADES. Du fait que l'on a oublié le \$ après B, l'ordinateur va afficher le chiffre 0 pour traduire la valeur de B. Vous vous en souvenez ? Sinon, relisez LED-MICRO n° 7 § 3.11.3.D page 25. La ligne 60 correcte sera :

60 PRINT C\$ ; " " ; B\$

- Ligne 70 Pas d'erreur !

### Q. Solution de l'exercice A19

- Ligne 10 Pas d'erreur

- Ligne 20 Il faut écrire l'inverse, à savoir :

$$A = 6$$

(Voir LED-MICRO n° 12 § 3.15.1)

- Ligne 30 Pas d'erreur

- Ligne 40 Certains BASICs interprètent cette ligne en la traduisant

$$40 A = 3 : B = 3 : C = 3$$

Mais les BASICs que nous vous avons définis **ne** comprennent **pas** cette syntaxe... (sinon nous vous l'aurions dit !)



### 3.15.5. Nom des variables

#### A. Nombre de caractères

Jusqu'à présent, nous avons utilisé des noms de variables ne comprenant :

- qu'une seule lettre pour les variables numériques (A, B,... X, Z) ;
- qu'une seule lettre suivie de \$ pour les variables chaînes (A\$, B\$,...)

Certains BASICs anciens et rudimentaires, ne peuvent effectivement utiliser qu'une seule lettre comme nom de variables. Mais la plupart des BASICs récents sont beaucoup plus riches en ce domaine.

L'APPLESOFT (le BASIC standard de notre système de référence APPLE IIe) peut utiliser des noms de variables numériques composés :

- soit d'une seule lettre (exemples : A, B, C, F...) ;
  - soit d'une lettre majuscule suivie d'une deuxième lettre ou d'un chiffre (exemples : A1, A5, B4, AG, AZ, BF, K7...).
- Pour les variables chaînes de caractères, les mêmes combinaisons sont suivies du \$ (exemples : A\$, A3\$, B6\$, X5\$...).

Le BASIC du PC IBM admet des noms de variables comportant jusqu'à 40 caractères. (On peut même dépasser 40 caractères mais au-delà de 40 ils sont dits «non significatifs», c'est-à-dire que deux noms dont les 40 premiers caractères sont identiques seront considérés comme définissant la même variable.)

Ce nombre maximum de caractères ainsi que le nombre de caractères significatifs varient considérablement d'un système à un autre : consultez la notice de votre BASIC.

#### B. Définition de la précision dans les variables numériques

Nous avons indiqué (LED-MICRO n° 7 - § G3.11.4.B page 26) que l'on peut définir la précision avec laquelle on veut que les nombres soient définis en terminant le nom de la variable avec l'un des signes %, ! ou #.

Bien sûr, tout ceci se généralise avec des identificateurs comportant plusieurs caractères.

Identificateur terminé par :	Type	A titre d'exemple dans le cas de l'IBM PC
%	Nombre entier (en anglais : integer)	Ne permet de mémoriser que des nombres compris entre - 32 768 et + 32 767).
!	Simple précision	6 chiffres significatifs
#	Double précision	16 chiffres significatifs

#### C. «Identificateurs»

Au lieu de dire «nom» d'une variable, les informaticiens disent «identificateur».

#### D. Caractères autorisés par les identificateurs

Les règles varient un peu avec les systèmes. D'une façon générale :

- + le premier caractère doit être une lettre ;
- + les caractères suivants peuvent être soit une lettre, soit un chiffre, mais ne peuvent généralement pas être un signe ou un espace [Exceptions nombreuses : par exemple, le BASIC du PC IBM accepte des identificateurs comportant un point] ;
- + il ne faut pas utiliser certains «mots réservés» (voir §F page ci-contre).

#### E. Comment choisir un nom de variables

Lorsqu'on a la chance d'utiliser un BASIC admettant des identificateurs comportant plusieurs caractères, il faut profiter de cette opportunité.

Au lieu de donner des noms de variables tout «bêtes» tels que

- A = prix de revient
- B = marge bénéficiaire
- C = prix de vente

Il sera souvent plus commode de les appeler

- PRIX REV
- MARGE
- PRIX VEN

La lecture des programmes devient plus aisée.

Inversement, prendre des identificateurs trop longs aboutit à des expressions algébriques peu claires. Personnellement, je choisis généralement des identificateurs comprenant entre 3 et 6 caractères... Faites selon votre goût !



## F. Mots réservés

Certains mots ont une signification précise en BASIC. C'est ainsi que nous avons rencontré des mots tels que : RUN, LOAD, LIST, EDIT, END, PRINT, NEW, SAVE, EXP. Et il y en a beaucoup d'autres : la liste varie avec le BASIC que vous utilisez.

Ces mots sont dits «mots réservés» : on ne peut pas les employer comme identificateurs.

Dans certains systèmes, il ne faut pas utiliser de mots comportant une suite de caractères formant un mot réservé. Avec eux, vous ne pourrez pas utiliser LIST\$ (qui contient LIST) ni PRUNE (qui contient RUN).

## G. Exemple : les mots réservés du BASIC de l'IBM PC

ABS	AND	ASC	ATN	AUTO
BEEP	BLOAD	BSAVE	CALL	CDBL
CHAIN	CHR\$	CINT	CIRCLE	CLEAR
CLOSE	CLS	COLOR	COM	COMMON
CONT	COS	CSNG	CSRLIN	CVD
CVI	CVS	DATA	DATE\$	DEF
DEFDBL	DEFINT	DEFSNG	DEFSTR	DELETE
DIM	DRAW	EDIT	ELSE	END
EOF	EQV	ERASE	ERL	ERR
ERROR	EXP	FIELD	FILES	FIX
FNxxxx	FOR	FRE	GET	GOSUB
GOTO	HEX\$	IF	IMP	INKEY\$
INP	INPUT	INPUT#	INPUT\$	INSTR
INT	KEY	KILL	LEFT\$	LEN
LET	LINE	LIST	LLIST	LOAD
LOC	LOCATE	LOF	LOG	LPOS
LPRINT	LSET	MERGE	MID\$	MKD\$
MKI\$	MKS\$	MOD	MOTOR	NAME
NEW	NEXT	NOT	OCT\$	OFF
ON	OPEN	OPTION	OR	OUT
PAINT	PEEK	PEN	PLAY	POINT
POKE	POS	PRESET	PRINT	PRINT#
PSET	PUT	RANDOMIZE	READ	REM
REM	RESET	RESTORE	RESUME	RETURN
RIGHT\$	RND	RSET	RUN	SAVE
SCREEN	SGN	SIN	SOUND	SPACE\$
SPC	SQR	STEP	STICK	STOP
STR\$	STRIG	STRING\$	SWAP	SYSTEM
TABC	TAN	THEN	TIMES\$	TO
TROFF	TRON	USING	USR	VAL
VARPTR	VARPTR\$	WAIT	WEND	WHILE
WIDTH	WRITE	WRITE#	XOR	

Avec ce BASIC on peut utiliser des identificateurs incluant des mots réservés (exemples : PRUNE, LISTE, EXPERT sont «valides»).

## H. Un Basic vicieux

L'APPLESOFT prend la liberté de supprimer ou de rajouter des espaces dans votre texte pour le «corriger».

Si, par exemple : vous écrivez :

```
IF B = A THEN PRINT "EGALITE"
```

il supprimera l'espace entre le A et le THEN,

d'où

```
IF B = AT HEN PRINT "EGALITE"
```




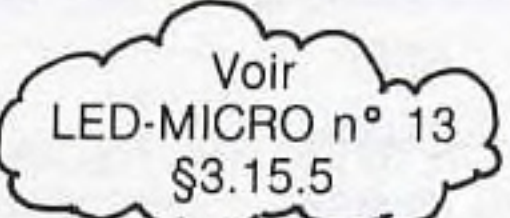
et comme AT est un mot réservé, et que HEN ne signifie rien, il vous insultera en vous disant «Syntax error». Il faut savoir !

Une fois qu'on s'est fait «piégé», on écrit :

```
IF A = B THEN PRINT "EGALITE"
```



## I. Résumé

	Numériques :	Chaînes de caractères :
Constantes :	5 0.23 3.14 5E7  Voir LED MICRO n° 7 §3.11.4	"BONJOUR" " " "VIVE MOI !"  Voir LED-MICRO n° 6 §3.8.1
Variables :	A Z A3 PRIX  Voir LED-MICRO n° 7 §3.11.3	Z\$ PRIX\$ POPAUL\$ A3\$  Voir LED-MICRO n° 13 §3.15.5

Il ne faut jamais «mélanger» une valeur numérique et une chaîne de caractères. Si l'ordinateur s'attend à ce que vous lui donniez une valeur chaîne (parce que vous avez utilisé un identificateur se terminant par \$) et que vous lui fournissiez une valeur numérique — ou vice versa —, il ne vous obéira pas. S'il est compréhensif, il vous expliquera pourquoi en vous disant (dans son langage) : «type de valeur non concordant».

## J. Un Basic de référence

### 1. Une exception à nos «principes»

Nous vous avons maintes fois exposé l'un de nos principes d'enseignement de la programmation : «Pas de système fictif simplifié, qui élude les difficultés et laisse l'étudiant désarmé devant les problèmes pratiques qu'il rencontrera sur son ordinateur.»

Mais nous voici (provisoirement !) obligés de faire le contraire : il y a tant de variantes dans les désignations des variables et la présentation de l'écran qu'il nous a paru préférable de définir (provisoirement, j'insiste) un système fictif dont nous vous demanderons de respecter les «règles» pour effectuer les exercices qui suivent.

Voilà d'ailleurs un excellent réflexe à acquérir : lorsque, dans la pratique sur votre système, vous recevrez un message d'erreur, celui-ci portera souvent sur le non-respect des particularités de votre système. Ne lisez pas «en diagonale» les notices du BASIC d'un nouvel appareil, même si vous connaissez bien «le» BASIC !

### 2. Instructions Fonctions-Commandes

Nous utiliserons le BASIC de Microsoft (celui du TRS 80, de PROF 301, de l'IBM-PC, etc.). C'est-à-dire que nous emploierons les mots-clés suivants :

PRINT - GOTO - END - LPRINT - LIST - LLIST - REM - RUN - CLS - SQR

et **non** les mots-clés particuliers à l'APPLESOFT :

HOME - PR # 1

### 3. Règles concernant les identificateurs

(Règles concernant les «noms» des variables)

1. Un identificateur doit commencer par une lettre majuscule.
2. Les noms des variables chaînes se terminent par \$.
3. Les seuls caractères autorisés sont les lettres majuscules et les chiffres (plus le \$ à la fin des caractères chaîne).
4. L'identificateur doit comporter un maximum de 12 caractères (plus l'éventuel \$) dont seuls les 8 premiers sont significatifs.
5. Les mots réservés de notre système sont ceux du PC d'IBM (voir §3.15.5.E ci-dessus).
6. Un nom de variable ne peut ni être un mot réservé ni contenir un mot réservé. Il ne peut pas non plus commencer par FN.



### K. Des noms pour les constantes

Supposons qu'on a à faire des calculs utilisant plusieurs fois la valeur  $\pi$ .  
On a intérêt à écrire une fois pour toutes :

$$PI = 3.141592$$

au lieu de :

$$\begin{aligned} 10\emptyset \text{ CIRCON} &= 2 * 3.141592 * R \\ 11\emptyset \text{ SURFAC} &= 3.141592 * R * R \\ 12\emptyset \text{ VOLUME} &= (2/3) * 3.141592 * R * R * R \end{aligned}$$

on écrira :

$$\begin{aligned} 9\emptyset \text{ PI} &= 3.141592 \\ 10\emptyset \text{ CIRCON} &= 2 * PI * R \\ 11\emptyset \text{ SURFAC} &= PI * R * R \\ 12\emptyset \text{ VOLUME} &= (2/3) * PI * R * R * R \end{aligned}$$

Donner un nom à une constante est non seulement commode, mais accélère considérablement les temps de calculs : l'ordinateur, au lieu de décoder la chaîne de chiffres, n'a qu'à aller chercher la valeur qui attend, bien au chaud, dans sa case. Nous développerons ce thème lorsque nous serons des fortiches et que nous essaierons d'«optimiser» nos programmes.

Comme le BASIC ne donne pas explicitement des noms aux constantes (contrairement au PASCAL), cette pratique (très recommandable) est appelée souvent «traiter les constantes comme des variables».

### L. Exercice d'application n° A20

Parmi les mots suivants, certains peuvent être utilisés comme identificateurs (noms «valides») et d'autres pas. Indiquez les noms valides, en supposant que vous avez affaire à un système respectant les règles du «BASIC de référence» que nous venons de définir.

P R I X   N E T	P R I X - B R U T	A 2 $\emptyset$ 4
2 $\emptyset$ 4 A	5 E 7	F r a n c e
P O R T I O N 3	F N L O G	A \$ 5
" N O M B R E "	V E R S . $\emptyset$ 3	B A S S I N E

### M. Corrigé de l'exercice A20

PRIX NET	non	Pas d'espace à l'intérieur d'un identificateur
PRIX-BRUT	non	Pas de petit tiret à l'intérieur d'un identificateur
A2 $\emptyset$ 4	O.K.	Il faut bien un identificateur acceptable !
2 $\emptyset$ 4A	non	Un identificateur doit commencer par une lettre
5E7	non	
France	non	Un identificateur ne doit pas contenir de lettres minuscules
PORTION3	non	Ce mot contient le mot réservé OR
FNLOG	non	Commence par FN
A\$5	non	Pas de \$ à l'intérieur d'un identificateur
"NOMBRE"	non	Pas de "dans un identificateur
VERS. $\emptyset$ 3	non	Pas de . dans un identificateur
BASSINE	non	Contient le mot réservé SIN

Rappelons que ces «interdictions» correspondent aux règles que nous venons de vous donner (page ci-contre). Beaucoup de ces mots seraient admissibles sur beaucoup de BASICs. Vérifiez ce que permet «votre» BASIC : parfois il ne veut pas de lettres minuscules, parfois il les transforme automatiquement en majuscules.



### 3.15.6. Les opérations sur les chaînes de caractères (Notions élémentaires)

#### A. Comparaison

Dans LED MICRO n° 8, paragraphe G3.11.5.E.4 (page 20) nous avons simplement indiqué l'existence en BASIC des opérateurs de comparaison utilisables avec des valeurs numériques

= (égal) > (plus grand que) < (plus petit que)  
<> (différent de) <= (plus petit ou égal) >= (plus grand ou égal)

Ces mêmes symboles sont utilisés par le BASIC avec les chaînes de caractères. Mais alors leur signification est différente.

Par exemple : A1\$ < B2\$ signifie

«Si on classe les textes des chaînes de caractères par ordre alphabétique A1\$ précède B2\$».

Nous étudierons plus tard les opérations de comparaison (lorsque nous aurons le moyen de les utiliser dans des exemples, c'est-à-dire avec l'instruction IF... THEN). Contentons-nous pour le moment de noter que :

- 1) les symboles de comparaison ont des significations différentes selon qu'on les utilise pour comparer des valeurs numériques ou pour comparer des valeurs chaînes de caractères.
- 2) en assimilant «l'ordre alphabétique» avec «l'ordre des caractères dans le code ASCII», on fait une simplification abusive : dans l'ordre ASCII toutes les majuscules sont avant toutes les minuscules, les chiffres et les signes ont chacun une place bien définie (en particulier l'espace est un caractère ASCII comme un autre) dans cet ordre et qui n'est peut-être pas celle que l'on s'attend à voir.

#### B. Concaténation

Employé avec des **valeurs numériques**, le signe + représente l'opération «addition». C'est ainsi qu'en mode commande on obtiendra :

```
P R I N T   5 + 7  
1 2
```

Employé avec des **chaînes de caractères**, le signe + jouera un autre rôle. Essayons :

```
P R I N T   " J O L I E " + " M A D A M E "  
J O L I E M A D A M E
```

Nous voyons que le + a pour effet de «réunir» les deux chaînes de caractères en les plaçant l'une à la suite de l'autre. On dit qu'ici le + est un opérateur de «concaténation».

L'exemple ci-dessous montre encore mieux, que le signe + joue des rôles différents :

```
P R I N T   1 0 + 2 0  
2 0  
P R I N T   - " 1 0 " + " 1 0 "  
1 0 1 0
```

Pour montrer cette différence, certains BASICs (celui de Tektronix, par exemple) utilisent comme opérateur de concaténation non pas le symbole + mais le symbole & (le «et commercial»).



### C. Instructions portant sur des chaînes de caractères

Le BASIC utilise un certain nombre d'instructions et de fonctions portant sur des chaînes de caractères :  
MID\$ (K\$, 5, 2) : analyse la chaîne de caractères K\$ et en extrait 2 caractères à partir du caractère de position 5  
CHR\$ (72) : fournit le caractère dont le code ASCII est le nombre 72 (c'est-à-dire la lettre H majuscule)  
LEN (Z\$) : fournit la longueur de la chaîne de caractères Z\$.  
Elles permettent de résoudre élégamment différents problèmes de manipulation des textes : classement par ordre alphabétique, vérification de vraisemblance d'une réponse de l'opérateur, présentation agréable des tableaux, etc...  
Nous les examinerons plus tard.

### D. Ne parlons pas «bébé»

Au lieu de dire :  
«CHR\$ (72) fournit le caractère dont le code ASCII est le nombre 72, c'est-à-dire la lettre H».  
Les informaticiens écriront :  
«La fonction CHR\$ (n) renvoie le caractère dont le code est n en ASCII».  
Les informaticiens parlant français préféreront même dire «retourne» au lieu de «renvoie».  
Petit à petit nous emploierons ce vocabulaire «comparatiste» : il faut le connaître pour lire facilement les articles, livres et notices traitant d'informatique.  
Mais, pour être clair, nous illustrerons ces définitions d'exemples souvent traduites en «mode commande».

```
P R I N T   C H R $   ( 7 2 )
H
N = 3 7
P R I N T   C H R $   ( N )
%
```

### E. Exercice d'application A21

Que donnera l'exécution du programme ci-dessous ?

```
10 B$ = "BLA"
20 B$ = B$ + B$
30 PRINT B$
40 GOTO 20
```

### F. Solution de A21

	1	2	5	6	7	10	15	20	25	30	35											
1	B	L	A	B	L	A																
2	B	L	A	B	L	A	B	L	A	B	L											
3	B	L	A	B	L	A	B	L	A	B	L	A	B	L	A	B	L	A	B	L	A	
4	O	U	T	O	F	S	T	R	I	N	G	R	A	N	G	E	I	N	8	0		

Effectivement, à chaque fois que l'on passe dans l'instruction 20, la taille du mot contenu dans la chaîne B\$ double. Il arrive un moment où l'on dépasse la longueur admissible. Voir ci-dessus §3.15.4.E.

### G. Exercices d'application A22 et A23

Que donne l'exécution des programmes ci-dessous ?

**A22 :**  
10 A\$ = "PIF"  
20 B\$ = "paf" + A\$  
30 PRINT B\$ ,  
40 GOTO 20

**A23 :**  
10 A\$ = "PIF"  
20 B\$ = A\$ + "paf"  
30 PRINT B\$ ,  
40 GOTO 20

### H. Solutions des exercices A22 et A23

Dans les deux cas ci-dessus, la chaîne B\$ (définie à la ligne 20) ne grandit pas indéfiniment (contrairement au cas de l'exercice précédent).

Dans l'exercice A22, cette chaîne vaut successivement pafPIF puis pafPIF puis pafPIF puis pafPIF, etc.

Dans l'exercice A23, la chaîne vaut successivement PIFpaf, puis PIFpaf, puis PIFpaf, puis pifPAF, etc.



### 3.15.7. Changement de type de variables VAL et STR\$

#### A. Passage d'un type à un autre

Nous pensons avoir suffisamment insisté pour que le lecteur n'oublie pas qu'il ne faut jamais mélanger les «chaînes de caractères» et les «valeurs numériques» : ce sont des «types» de constantes ou de variables tout à fait différents et utilisant des opérateurs différents.

Mais nous verrons (très bientôt) qu'il est parfois utile d'effectuer certains «changements de type» :

#### Exemple 1

Un programme demande à l'utilisateur d'indiquer sa date de naissance. L'opérateur répond :

25 OCTOBRE 1926

A partir de cette réponse, l'ordinateur veut calculer l'âge de son client. Il lui faudra :

- d'abord extraire 1926 de cette suite de caractères (ce qui se fera grâce à des fonctions que nous étudierons plus tard). Il obtiendra alors «1926» sous forme de chaînes de caractères ;
- puis, il lui faudra transformer cette chaîne de caractères «1926» dans le nombre 1926 de façon à pouvoir effectuer la soustraction

$$1984 - 1926 = 58 \text{ ans}$$

En effet, on ne peut pas effectuer de soustraction sur des chaînes de caractères.

#### Exemple 2

Une espèce de loterie respecte la règle du jeu suivante :

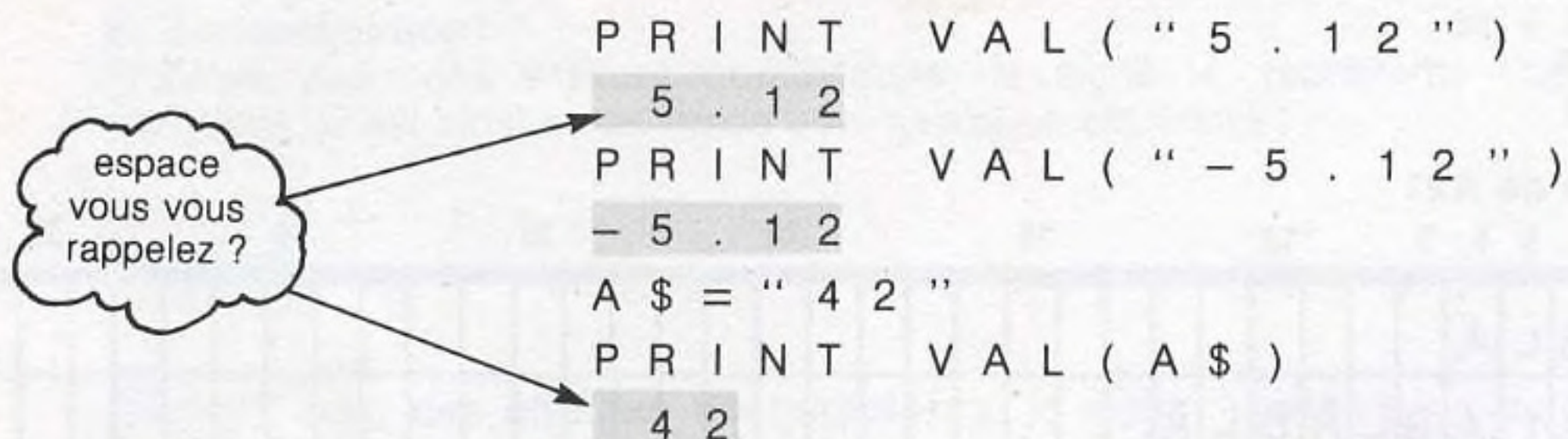
Chaque joueur tire un nombre composé de 5 chiffres (nombre fabriqué «au hasard» par l'ordinateur, par exemple). Le joueur gagne si ce nombre comporte 2 chiffres placés n'importe où.

Nous verrons que la programmation de ce jeu est très facile si l'on commence par transformer le nombre (par exemple 30172) en une chaîne de caractères (qui sera alors "30172"). On pourra alors utiliser une des fonctions chaînes-de-caractères pour analyser l'un après l'autre les 5 caractères de cette chaîne.

#### B. La fonction VAL - Définition

Employons le «vocabulaire des informaticiens» pour définir VAL : «La fonction VAL(X\$) renvoie la valeur numérique de la chaîne X\$».

Exemple :



#### C. La fonction VAL - Compléments

Exposé par des exemples

VAL (" 53") = 53	Les caractères espaces précédant les chiffres sont éliminés.
VAL (" - 32") = - 32 VAL (" - 4.5") = - 4.5 10 A\$ = " - 18" 20 PRINT VAL (A\$) RUN - 18	Le signe - est pris en compte de façon «normale»
VAL ("JULIE") = 0 10 X\$ = "GLOP" 20 PRINT VAL (X\$) RUN 0	Lorsque le premier caractère de la chaîne est une lettre, le VAL «retourne» la valeur 0.
10 V\$ = "" 20 PRINT VAL (V\$) RUN 0	Le VAL d'une chaîne vide est 0. ATTENTION ! pas avec tous les BASICs. Certains refusent cette expression
PRINT VAL (15 RUE DARU) 15	Lorsque les premiers caractères sont des chiffres, le VAL permet de les extraire (ce qui est souvent très commode en gestion).



#### D. La fonction STR\$

C'est la fonction inverse : elle convertit une valeur numérique en chaîne de caractères.

Nous en verrons très bientôt l'intérêt dans un cas très fréquent (avec l'instruction INPUT).

#### E. Exercice d'application A24

Les cinq lignes suivantes sont correctes :

```
10 A = 20
20 B = 30
30 C$ = "30"
40 D$ = "A60"
50 E$ = "20"
```

Indiquez ce que donne chacune des lignes suivantes (certaines de ces dernières lignes peuvent comporter des erreurs).

```
100 E = A + B
110 PRINT E + C$
120 X1 = VAL(A)
130 PRINT X1 + B
140 PRINT A + VAL(C$)
150 PRINT C$ + STR$(A)
```

#### F. Solution de l'exercice A24

ligne 110	On n'a pas le droit d'additionner une valeur numérique (E) et une chaîne (C\$). Résultat : Erreur.
ligne 120	On ne peut pas prendre la valeur numérique d'une variable numérique : Erreur.
ligne 130	Puisque X1 n'est pas défini, la ligne 130 ne signifie rien.
ligne 140	VAL(C\$) = 30 (valeur numérique) donc 140 affiche 50.
ligne 150	Affiche la concaténation des deux chaînes. Résultat affiché : 3020.



### 3.15.8. Variables booléennes.

#### A. Rappel sur la notion de variable binaire

Rappelons ce que nous avons dit à ce sujet dans LED-MICRO n° 1 chapitre 1.3.1.

Il existe quantité d'éléments qui peuvent prendre deux états et deux états seulement :

- une porte P peut être ouverte ou fermée ( $P = 0$  ou  $P = 1$ )
- un relais R peut être alimenté ou non-alimenté en courant ( $R = 0$  ou  $R = 1$ )
- une lampe L peut être allumée ou éteinte ( $L = 0$  ou  $L = 1$ )
- une proposition L peut être vraie ou fausse ( $K = 0$  ou  $K = 1$ )
- sur certaines prises de sortie d'un ordinateur on ne peut rencontrer que deux tensions : 0 volt et 5 volts ( $S = 0$  ou  $S = 1$ )
- etc...

Tous ces éléments sont des «variables binaires».

On convient de représenter par 0 et par 1 chacun des deux états que peut prendre une variable binaire.

Après le type «numérique» et le type «chaîne», nous voici en présence d'un autre type de variable : le type «binaire», plus communément appelé «type booléen» (prononcez : boulé-in).

Dans le BASIC APPLESOFT, les deux «états» que peut prendre une variable booléenne sont représentés par 0 et 1. Dans les BASICs d'origine MICROSOFT (TRS 80, PROF 301, IBM PC et compatibles), ces deux états sont représentés par 0 et -1.

Lorsqu'il sera nécessaire de préciser avec quel BASIC on travaille, nous dirons : soit «convention 0,1», soit «convention 0, -1».

#### B. La «condition» considérée comme une variable booléenne.

Si  $A = 8$  et  $B = 5$  la condition

$$A > B$$

est vraie.

Si  $A = 3$  et  $B = 5$ , la condition

$$A < B$$

est fausse.

Si  $A = 3$  et  $B = 5$  la condition

$$A = B$$

est fausse.

On peut donc considérer l'ensemble des deux termes d'une comparaison comme une variable booléenne puisqu'elle peut présenter deux états (et deux états seulement) :

- être vraie (symbole 1, ou -1 suivant «le» Basic utilisé)
- être fausse (symbole 0).

Dialogue en mode commande - Basic Microsoft -

```
A = 5
PRINT A > 3
- 1
PRINT A = 3
0
```

#### C. Nom des variables booléennes

Si on distingue les variables chaînes par leur suffixe \$, il n'en est pas de même pour les variables de type booléen : le nom d'une variable booléenne a la même forme que le nom d'une variable numérique. Nous verrons d'ailleurs bientôt que l'on peut mélanger variable numérique et variable booléenne dans une même expression : le Basic (généalement simple) ne se complique pas la vie avec des distinctions pratiquement inutiles.

#### Exemples

```
10 A = 5
20 B = A + 3
30 C = A > B
40 PRINT C
RUN
0
```

La ligne 30 indique que C est une valeur booléenne qui serait vraie ( $C = 1$  ou  $C = -1$ ) lorsque  $A > B$  et qui sera fausse ( $C = 0$ ) lorsqu'on aura  $A \leq B$ .

Dans le cas présent, C est faux. De sorte que lorsqu'on lance l'exécution de ce programme, la ligne 40 affiche 0.



### 3.15.9. L'instruction IF... THEN... - Première approche

#### A. Rafrichissons nos connaissances

Si vous ne vous souvenez pas de :	Relisez :
Ce qu'est un format	LM 9 - §3.12.3 pages 14 et 15
Ce qu'est une expression	LM 8 - §3.11.5 page 13

#### B. Format de IF... THEN...

IF signifie SI ; THEN signifie ALORS ; PRINT signifie AFFICHER (SUR L'ECRAN) [en général] l'instruction :

50 IF A > 5 THEN PRINT "A EST GRAND"

signifie :

«Monsieur l'ordinateur, si vous trouvez que A est plus grand que 5, alors affichez "A est grand". Sinon, ne faites rien et contentez-vous de passer à l'instruction suivante».

Autre exemple :

```

1 0  A  =  5
2 0  B  =  7
3 0  IF  A  =  B  THEN  PRINT  "  C  '  E  S  T  P  A  R  E  I  L  "
4 0  IF  A  <  B  THEN  PRINT  "  A  E  S  T  P  E  T  I  T  "
R U N
A  E  S  T  P  E  T  I  T
    
```

On peut dire :

Le format de IF... THEN... est :

IF <condition> THEN <instruction>

Si la condition est vraie (= 1) alors l'instruction est exécutée.

Si la condition n'est pas vraie (= 0) alors l'instruction n'est pas exécutée.

#### C. IF <condition> THEN <Instruction d'affectation>

Il est possible de mettre après THEN n'importe quelle instruction,

à savoir :	par exemple :
une instruction d'affichage	PRINT "C'EST PAREIL"
une instruction d'affectation	LET C\$ = "ATCHOUM"
une instruction de branchement	GOTO 50 (en fait on verra que l'on peut supprimer le GOTO)

Pour le moment, contentons-nous de considérer le cas où l'instruction après le THEN est une instruction d'affectation, telle que :

IF A=5 THEN LET X=3

Or, comme on peut supprimer le LET

IF A=5 THEN X=3

Dans cette écriture :

— le premier signe = représente un opérateur de comparaison (au même titre que les opérateurs >, <, <>, <=>, =>);

— le deuxième signe = est le symbole de l'affectation (que nous avons décortiqué §3.15.1).

Donc, au lieu d'écrire :

IF A=5 THEN X=3

on peut inverser les deux termes suivant le IF

IF 5=A THEN X=3

mais on **ne** peut **pas** inverser les deux termes suivant le THEN :

IF A=5 THEN 3=X

C'est la troisième fois que nous disons la même chose : il est indispensable que la distinction entre «condition» et «instruction» soit parfaitement assimilée.



### 3.15.10. Les opérateurs logiques

#### A. Qu'est-ce que l'algèbre de Boole (ou algèbre logique)

Nous avons vu que :

- + l'on peut combiner des valeurs numériques avec :
  - des opérateurs arithmétiques : + - /\* MODULO
  - des opérateurs relationnels (ou de comparaison)
- + l'on peut combiner des valeurs chaîne avec :
  - des opérateurs relationnels : = < > <= >= )
  - un opérateur spécifique : concaténation

De même, les variables binaires (de «type» booléen) peuvent être combinées à l'aide d'opérateurs dits «opérateurs logiques». Les opérateurs logiques du BASIC sont :

- NOT (ou «négation» ou «complément logique»)
- AND (ou «ET» ou «conjonction»)
- OR (ou «OU inclusif» ou «disjonction»)
- XOR (ou «OU exclusif»)
- IMP (ou «implication»)
- EQV (ou «équivalence»)

L'art de combiner ces différents opérateurs entre eux s'appelle «l'algèbre de Boole». Dans cette «première couche», nous n'étudierons cet algèbre de Boole que de façon extrêmement sommaire, et en nous contentant d'examiner les 3 fonctions NOT, AND et OR.

#### B. L'opérateur NOT ou «négation» ou «complément logique»

Exposé fait en «convention 0, 1».

Si A est une variable booléenne.

La variable X définie par

$$X = \text{NOT } A$$

est une variable binaire, qui prend les valeurs «complémentaires» à celle de A. C'est-à-dire que :

$$\begin{aligned} \text{si } A = 0 \text{ on aura : } X &= 1 \\ \text{si } A = 1 \text{ on aura : } X &= 0 \end{aligned}$$

Ceci peut se traduire par le tableau suivant :

A	X = NOT A
0	1
1	0

#### Exemple

```
1 0 A $ = " S O L E I L "
2 0 X = ( A $ = " L U N E " )
3 0 P R I N T ( N O T X )
R U N
- 1
```

X est une variable booléenne qui est fautive X = 0 donc NOT X est vrai (= - 1 en BASIC Microsoft).



**C. L'opérateur AND («ET» ou «conjonction»)**

Exposé fait en convention (0, 1).

Soient 3 variables booléennes A, B et C.

L'expression

$$X = A \text{ AND } B$$

définit une variable X qui

1) est booléenne

2) prend la valeur 1 lorsque à la fois  $A = 1$  ET  $B = 1$ .

Dans les autres cas X vaut 0

Ceci peut se traduire par le tableau suivant :

B	A	X
0	0	0
0	1	0
1	0	0
1	1	1

L'expression :

$$Y = A \text{ AND } B \text{ AND } C$$

définit une variable Y qui

1) est booléenne

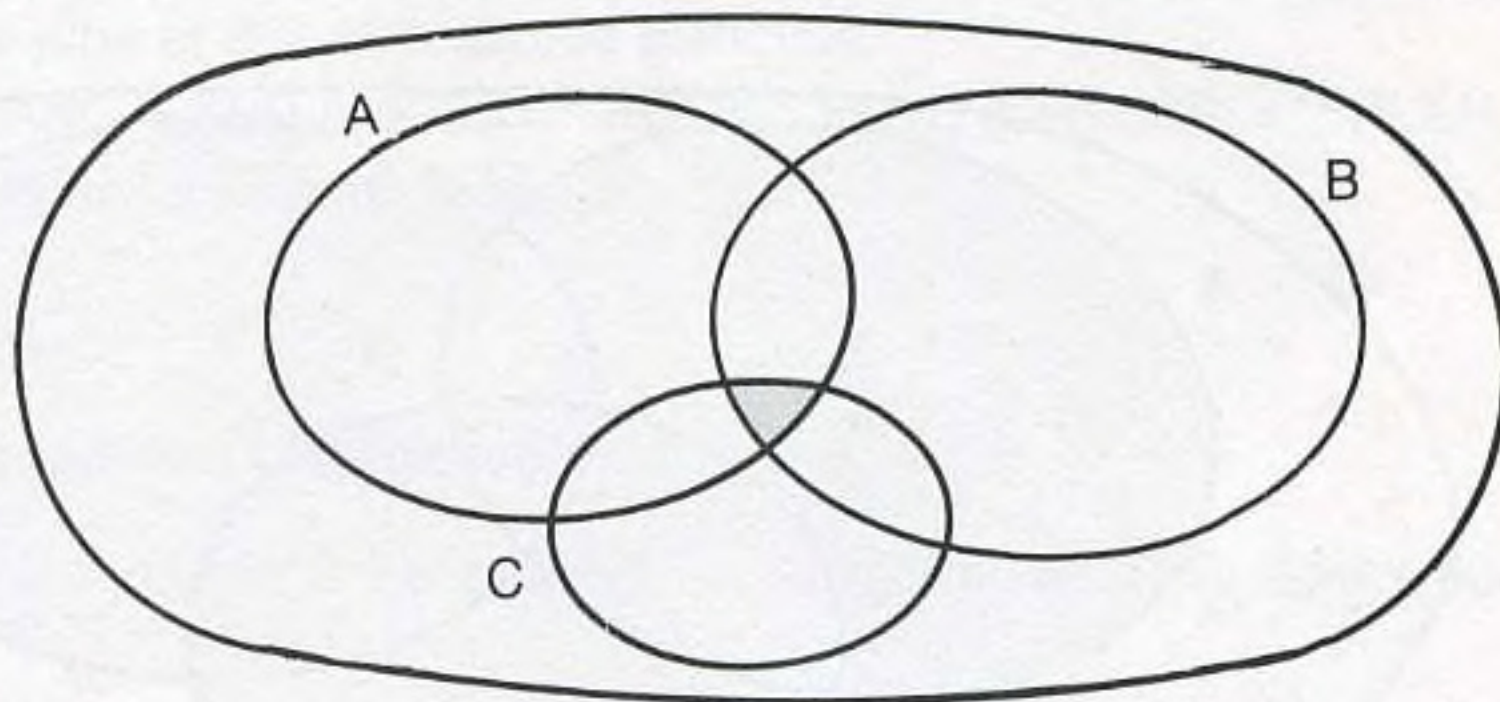
2) prend la valeur 1 lorsque à la fois  $A = 1$  ET  $B = 1$  ET  $C = 1$ .

Dans les autres cas Y = 0

Ceci peut se traduire par le tableau suivant :

C	B	A	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

On peut représenter cette fonction ET de 3 variables A, B et C par les «cercles d'Euler» :



Seule est égale à 1 la surface commune aux surfaces A, B et C.

Exemple :

```

10 A = 5
20 B = 3
30 C = 3
40 X = (A > B)
50 Y = (C > A)
60 PRINT (X = Y / AND Y)
RUN
-1
    
```

X est vrai X = 1

Y est faux Y = 0



#### D. L'opérateur OR («OU inclusif» ou «disjonction»)

Exposé fait en convention (0, 1).

Soient trois variables booléennes, A, B et C.

L'expression

$$X = A \text{ OR } B$$

définit une variable X qui :

1°) est booléenne ;

2°) prend la valeur 1 lorsque ou bien  $A = 1$ , ou bien  $B = 1$ , ou bien à la fois  $A = 1$  et  $B = 1$  (c'est l'un ou l'autre ou les deux, par opposition à l'opérateur XOR avec lequel on a ou  $A = 1$  ou  $B = 1$  mais pas à la fois  $A = 1$  et  $B = 1$ ).

Ceci peut se traduire par le tableau suivant :

B	A	X
0	0	0
0	1	1
1	0	1
1	1	1

l'expression

$$Y = A \text{ OU } B \text{ OR } C$$

définit une variable Y qui

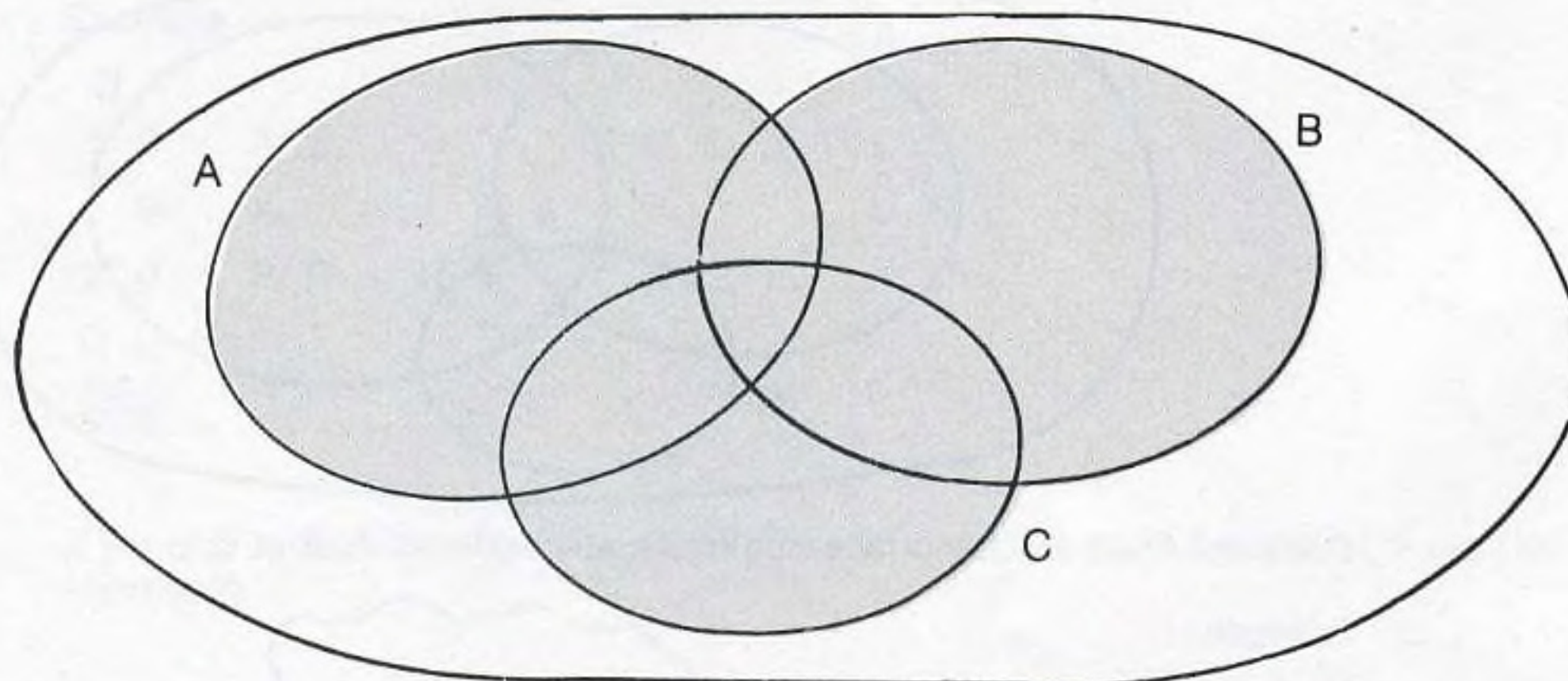
1°) est booléenne ;

2°) prend la valeur 1 lorsqu'on a OU bien  $A = 1$ , OU bien  $B = 1$ , OU bien  $C = 1$ , OU bien plusieurs de A, B et C sont égaux à 1.

Ceci peut se traduire par le tableau suivant :

C	B	A	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

On peut imaginer l'opérateur OU à l'aide des cercles d'Euler :



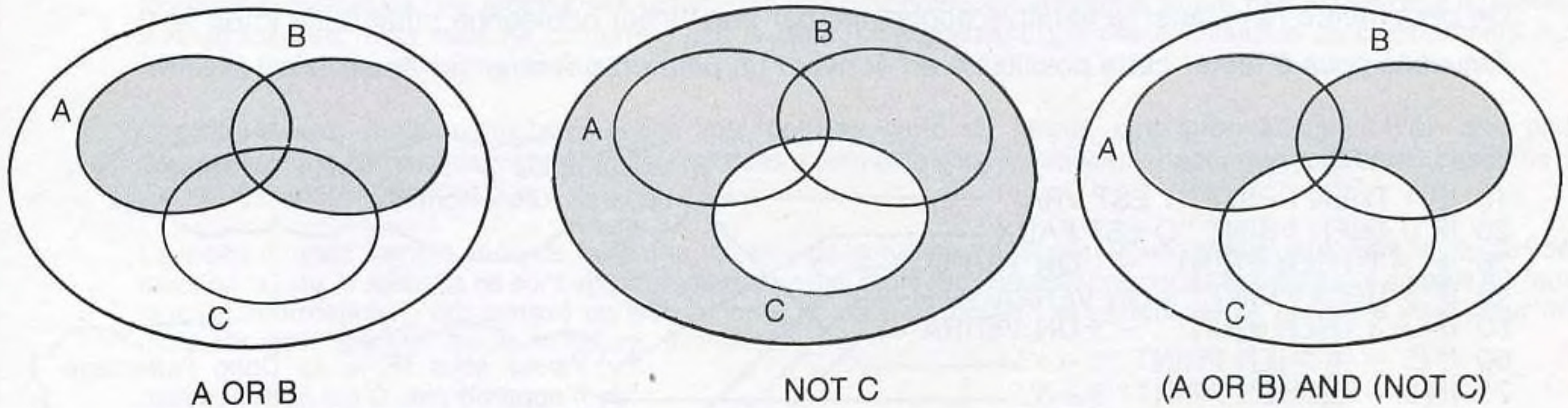
La surface égale à 1 représente l'ensemble des surfaces de A, de B et de C : un point est noir s'il est OU BIEN à l'intérieur de A OU BIEN à l'intérieur de B OU BIEN à l'intérieur de C.



## E. Combinaison d'opérateurs logiques

On peut combiner les trois opérateurs NOT, AND et OR dans des expressions complexes — que l'on simplifie d'ailleurs très facilement grâce aux règles de l'algèbre de Boole — ou simplement en considérant les cercles d'Euler (lorsqu'on ne combine pas plus de trois variables).

Exemple :



## F. Exemple

### Enoncé

Mademoiselle Cauchois peut porter une veste (VEST\$) qui peut être de quatre couleurs différentes

soit VEST\$ = "ROUGE"

soit VEST\$ = "BLEU"

soit VEST\$ = "VERT"

soit VEST\$ = "BLANC"

Elle peut porter une jupe (JUP\$) qui peut avoir cinq couleurs "ROUGE", "BLEU", "VERT" et "JAUNE". Elle peut porter des chaussures (CHAUS\$) qui ne peuvent avoir que deux couleurs "MARRON" et "BLANC".

Mademoiselle Cauchois se trouve élégante dans deux cas :

**1<sup>er</sup> cas** : si elle a une veste et une jupe de même couleur avec des chaussures marron.

Ce qui peut s'écrire

$$\text{CAS1} = \underbrace{(\text{VEST\$} = \text{JUP\$})}_{= 1 \text{ si la veste et la jupe ont la même couleur.}} \text{ AND } \underbrace{(\text{CHAUS\$} = \text{"MARRON"})}_{= 1 \text{ si les chaussures sont marrons.}}$$

**2<sup>e</sup> cas** : si elle a une veste, une jupe et des chaussures blanches.

$$\text{CAS2} = (\text{VEST\$} = \text{"BLANC"}) \text{ AND } (\text{JUP\$} = \text{"BLANC"}) \text{ AND } (\text{CHAUS\$} = \text{"BLANC"})$$

Pour que Mademoiselle Cauchois soit contente, il faut donc

soit que CAS1 = 1

soit que CAS2 = 1

### Programme exprimant cette condition d'élégance

On indiquera cet ensemble de conditions à l'ordinateur en rédigeant les trois lignes suivantes de BASIC :

```
100 CAS1 = (VEST$=JUP$) AND (CHAUS$ = "MARRON")
200 CAS2 = (VEST$="BLANC") AND (JUP$="BLANC") AND (CHAUS$ = "BLANC")
300 IF CAS1 OR CAS2 THEN PRINT "CA VA"
```



### G. Confusion de variables booléennes et de variables numériques

Il est possible de généraliser l'emploi de l'instruction

IF <condition> THEN <instruction>

en écrivant son format ainsi :

IF <variable booléenne> THEN <instruction>

On peut même remplacer la variable booléenne par une valeur booléenne numérique (0 ou -1).

Amusons-nous à tester cette possibilité en écrivant un petit programme sur Apple II :

```
10 IF 1 THEN PRINT "1 EST VRAI"
20 IF 0 THEN PRINT "0 EST FAUX"
30 IF -1 THEN PRINT "-1 ON VERRA"
40 IF 3 THEN PRINT "3 ON VERRA"
50 IF -3 THEN PRINT "-3 ON VERRA"
60 IF(5 - 4) THEN PRINT "5 - 4"
70 IF(5 - 5) THEN PRINT "5 - 5"
RUN
1 EST VRAI
-1 ON VERRA
3 ON VERRA
```

Valeur sous IF = 1. Donc l'affichage apparaît. C'est normal.

Valeur sous IF = 0. Donc l'affichage n'apparaît pas. C'est normal aussi.

Mais en plaçant sous le IF des valeurs différentes de 0 on peut avoir des surprises. C'est bien fait : IF... THEN... n'est pas fait pour ça !

### H. Expressions mixtes (algébrique et booléenne)

Il est possible de mélanger dans une même expression des variables numériques et booléennes.

Exemple :

$$Y = \underbrace{(A + B)}_{\text{variable numérique}} * \underbrace{(A > B)}_{\text{variable booléenne}}$$

**SI** A n'est pas plus grand que B, la condition  $(A > B)$  sera fausse donc cette variable booléenne sera égale à 0 et on aura

$$Y = 0$$

**SI** A est plus grand que B, la condition  $(A > B)$  sera vraie et, selon le BASIC que l'on utilisera, on aura

$$\text{soit } Y = (A + B) \\ \text{soit } Y = - (A + B)$$

Il est possible également de mélanger des opérateurs algébriques et des opérateurs booléens.

Exemple :

$$Y = ((A > B) \text{ OR } (C = D)) * (A + B + C + D)$$

Nous avons vu (LED-MICRO n° 8 §3.11.5.E.4) les règles de priorité que doivent respecter ces opérateurs (lorsqu'on veut obtenir des expressions plus courtes économisant des parenthèses).

Ces mélanges d'opérateurs permettent de résoudre de façon simple divers problèmes (expression algébrique contenant des discontinuités... tables de décision...). Nous en verrons des exemples lors de notre deuxième vague !



### 3.15.11. Révision de jargon informatique

L'étude de l'informatique et la lecture des notices techniques sont souvent difficiles pour les débutants simplement parce qu'ils ne possèdent pas bien le sens de différents mots que les informaticiens utilisent soit par nécessité soit par snobisme.

Les informaticiens ne sont d'ailleurs pas les seuls à se créer un vocabulaire hermétique : les philosophes, les journalistes politiques, les juristes, les publicistes, les psychologues arrivent à écrire des textes dont le profane connaît tous les mots mais ne comprend pas le sens (en supposant que ces spécialistes se comprennent eux-mêmes ; mais cela est une autre histoire).

Chemin faisant, nous avons appris pas mal de ces mots du jargon des informaticiens. Bien que nous désapprouvions l'emploi de mots «franglais», il nous a bien fallu vous les enseigner pour que vous soyez capables de lire la «littérature» informatique.

Le moment nous semble venu de faire une révision générale de ce jargon. Vous remarquerez que la plupart des mots de la liste ci-dessous ne sont apparemment pas des mots techniques : ce sont de braves mots de tous les jours que les informaticiens ont annexé en leur donnant un sens particulier, et parfois même plusieurs sens (pour des mots tels que : attribut, mode, initialiser, format...).

Mots	Voir :		
affectation	LM 7 §G3.11.3.A page 24 LM 12 §3.15.1 page 22	initialiser	une disquette : LM 11 §3.13.5.D page 23 une variable : LM 7 §3.11.3.D page 25 LM 7 §G3.11.3.E page 24
ambigu	LM 10 §3.13.4.F.3 page 27	inversion vidéo	LM 2 §2.6.6 page 33
argument	LM 6 §3.6.1 page 11 LM 6 §3.8.2 page 27	maître (master)	LM 11 §3.13.5.A page 19
attribut	LM 2 §2.6.6 page 23 LM 10 §G3.13.3.C.3 page 14	mnémorique	LM 4 §G2.11.5.E page 22
autochargeable	LM 9 §3.13.2 page 19	mode	LM 2 §2.6.6 page 33 LM 7 §3.11.2 page 23
backup	LM 11 §G3.14.4.G.3 page 10	mot-clé	LM 6 §3.6.1 page 11 LM 8 §3.11.6.D.1 page 29
buffer	LM 2 §2.7.5 page 37	off line	LM 2 §2.8.2.D page 39
certifier	LM 3 §2.9.2.C page 15	on line	LM 2 §2.8.2.D page 39
charger	LM 9 §G3.13.1.B page 16	opérateur	LM 8 §3.11.5.A.1 page 13
commande	LM 7 §3.11.2 page 23	option	LM 9 §G3.12.1.B page 12
configurer	LM 2 §2.8.2.B page 39	planté	LM 11 §G3.13.4.I.1
conversationnel	LM 11 §3.13.4.I.3 page 13	priorité	LM 8 §3.11.5.E page 21
dédié (ou dédié)	LM 2 §2.8.2.C page 39	protection	LM 10 §3.13.3.G page 15
défaut (par défaut)	LM 9 §G3.12.1.B page 12	release	j'ai oublié de vous signaler ce mot (très employé en franglais informatique). Il signifie «version». Voir LM 4 §G2.11.9.C
directory	LM 9 §3.13.3.B page 23 LM 9 §G3.13.1.B page 16	renvoyer	LM 8 §G3.11.16.B.1 page 24 LM 13 §3.15.6.D
écraser	LM 10 §G3.13.3.G.2 page 14	réserve	LM 13 §3.15.5.E
éditable (caractère)	LM 1 §1.3.3.C page 23	résident	LM 9 §3.13.2.C page 21
encodage	LM 3 §2.9.2.B page 13	retourner	LM 8 §G3.11.16.B.1 page 24 LM 13 §3.15.6.D
expression	LM 9 §3.12.3 page 15 LM 8 §3.11.5 page 13	scrolling	LM 6 §3.6.6 page 17
extension	LM 9 §3.13.3.C page 25	station	LM 10 §3.13.4.A page 23
fonction	LM 8 §3.11.6 page 25	syntaxe	LM 9 §3.13.3.C page 23 LM 10 §G3.13.4.A page 22 LM 9 §G3.12.3.A page 14
format	LM 7 §G3.11.4.E page 28 LM 10 §G3.13.3.F page 13 LM 10 §3.12.3 page 15	transitoire	LM 9 §3.12.2.C page 21
identificateur	LM 13 §3.15.5.C		
implicite	LM 9 §G3.12.1.B page 12 LM 10 §3.13.4.D page 25		





## NOS LECTEURS NOUS ECRIVENT

Vos lettres nous font toujours très plaisir. Elles traduisent votre intérêt pour Led-Micro. Et nous vous en remercions. Mais comment répondre efficacement à toutes ces lettres qui arrivent chaque jour ? Bon nombre de lecteurs soulèvent des problèmes très proches les uns des autres. Y répondre individuellement est impossible. De plus, ces problèmes intéressent certainement d'autres lecteurs. Aussi avons-nous choisi de retenir les lettres qui posent les questions les plus pointues et qui, néanmoins, traduisent des préoccupations communes et d'y apporter une réponse globale. Nous publierons également des lettres auxquelles nous pensons que des lecteurs pourraient apporter une réponse et ainsi créer une sorte de bourse d'échanges, d'idées et de réflexions. Nombreux sont également les lecteurs qui nous demandent des conseils quant à l'achat d'un micro. Le meilleur micro du monde est celui qui correspond à ce que l'on veut en faire. Notre avis ne pourrait être suffisamment motivé, faute de connaître parfaitement les souhaits de chacun. Aussi le meilleur conseil que nous puissions vous donner est de bien déterminer besoins et budget.

■ *... je trouve vos explications très claires, mais je rencontre encore des problèmes quand je veux transposer sur mon TI99 des programmes qui m'intéressent et qui sont destinés au ZX81 ou au TRS80. Pourriez-vous me dire comment transposer les instructions suivantes ...*

*G.E. Paris (75011)*

Je vais vous faire un aveu : je ne connais pas le Basic du TI99, et je n'ai pas la possibilité de me procurer les micros et de les essayer tous. Donc, je ne peux répondre aujourd'hui à votre question. Toutefois, j'envisage de faire paraître à la fin du cours de Basic un dictionnaire destiné à aider nos lecteurs à transcrire des programmes sur les micros les plus répandus dans le commerce... Pour le moment nous n'avons abordé que le Basic «initial» dont toutes les instructions sont communes à tous les systèmes.

Mais la transposition d'un programme d'un Basic dans un autre pose beaucoup plus que des problèmes de vocabulaire et de traduction ligne par ligne des instructions. Regardez les pièges dans lesquels nous sommes tombés en vous proposant de traduire «le plongeur» sur vos micros : Michel Lopez a été tout de suite conduit (et il a raison !) à restructurer ce programme.

Notre (savant) lecteur M. M.P. ne s'est pas cantonné lui non plus dans une traduction mot à mot du programme de Harumi Takahashi ; il a pris le programme initial simplement comme base de départ.

J'ai donc l'intention de reprendre dans «le Coin des Fortiches» le problème général de la transcription des programmes, mais :

1°) Seulement quand nos lecteurs (même les débutants) seront initiés à la programmation structurée (en prin-

cipe : en janvier), aux caractères de commandes, etc.

2°) Ces transcriptions ne seront pas du «vous n'avez qu'à recopier sans comprendre» (d'ailleurs ce n'est pas ce que vous souhaitez, je suppose).

■ *... j'ai pensé acquérir un micro-ordinateur, mais je suis rebuté par le fait que ces appareils ne «dialoguent» qu'en anglais, langue que je ne connais pas et que je ne désire pas apprendre... Existe-t-il un (ou plusieurs) micro travaillant entièrement en français ?...*

*C.D. à Toulouse (31100)*

Le L.S.E. (qui est «le Basic» de l'Education Nationale) parle français... mais (à ma connaissance) aucun micro de prix abordable n'utilise le L.S.E. Il y a eu des tentatives pour «franciser» le Basic : le BASICOIS et le BASI-



QUE... Ces tentatives sont restées sur le papier.

Bien que personnellement, je n'aie pas étudié l'anglais à l'école et que je ne sois ni doué ni intéressé par l'étude des langues, il m'est tout de suite apparu indispensable d'acquérir un minimum de vocabulaire anglais pour pouvoir faire autre chose que d'utiliser des programmes de jeux. C'est en suivant cette idée que je parsème mon cours de Basic de notion d'anglais technique. Si vous ne voulez pas apprendre un minimum d'anglais, vous n'irez pas bien loin. Je le regrette autant que vous.

■ ... Je travaille actuellement dans une Direction départementale de... Je suis néophyte mais ai étudié les dix premiers numéros de LED-MICRO. J'envisage de me présenter à l'examen dont le programme est le suivant... Je souhaiterais avoir votre opinion sur les chances de réussite en suivant votre cours de LED-MICRO, le concours ayant lieu en septembre 1985... Je vous précise que personne n'a encore pris l'option informatique à ce concours, je n'ai donc pu avoir les sujets précédemment posés...

J.P.C. à Mende (48000)

■ ... Je suis étudiant en deuxième année de DEUG... Ayant trouvé votre cours de Basic très intéressant et d'une approche facile par rapport au cours que j'avais suivi, je me permets de vous demander de m'indiquer (ou de m'expédier) des cours correspondant plus spécialement à l'Informatique générale... Voici le genre de question qui risquent de tomber : «Capacité approximative d'une bande magnétique»,...

P.B. à Marseille (13006)

J'ai déjà répondu partiellement à ces questions (en particulier dans l'édito-

rial de LED-MICRO n° 9). LED-MICRO n'a pas la prétention de former des informaticiens professionnels, mais seulement d'aider les débutants à acquérir une base solide.

D'autre part, nous ne pouvons pas connaître les niveaux et les programmes de tous les examens d'informatique : chaque école fait ce qu'elle veut... ou ce qu'elle peut.

■ ... Possesseur d'un MODEM acceptant les normes V 21 et V 23, j'ai donc accès à de nombreux réseaux, et en particulier le réseau TELETEL...

Après avoir écrit les routines de décodage en assembleur 650Z, je me suis aperçu que les transmissions issues du réseau TELETEL étaient truffées de caractères de contrôle et de caractères VIDEOTEX.

Le but de ma lettre est de vous demander s'il existe un ouvrage traitant des spécifications techniques de TELETEL, de manière à améliorer mon logiciel de décodage, et à obtenir des écrans à peu près parfaits.

P.B. à Paris (75020)

Je profite de votre question pour faire une expérience : «Lecteurs spécialistes d'un problème, aidez les lecteurs dans l'embarras». Si je ne reçois pas de réponse, je vous répondrai par courrier personnel, ou par la voie de cette rubrique si votre problème intéresse d'autres lecteurs.

■ ... Dans votre numéro 5 de LED-MICRO vous proposez le jeu du plongeur et sollicitez d'autres versions de ce programme. Je me suis mis au travail sur un Oric 1. Lorsque je m'apprêtais à vous l'envoyer, une version sur Oric apparaissait dans le numéro 10.

De plus, mes enfants (10 et 12 ans) jouant avec ce jeu et le trouvant

trop lent, j'ai poursuivi mon effort et ai écrit une version en assembleur 6502 sur Oric-Atmos.

Je n'ai modifié que peu de choses à l'esprit du jeu (seulement deux éléments : le jeu s'arrête sur gain ou perte, le plongeur se déplace toujours à la même vitesse. Je vous adresse ci-joint ce jeu (cassette, listing et organigrammes de principe)...

J'apprécie votre revue qui m'a surtout intéressée par ses articles «d'électronique digitale». Je souhaiterais des articles sur les interfaces ; peut-être que LED-ROBOT ira dans ce sens ? La partie Basic et programmation est trop importante à mon goût.

M.P. à Manziat (01920)

Le travail que nous a remis M. M.P. m'a étonné par son ampleur et sa qualité. Son éventuelle publication dans LED-MICRO pose «le problème de base de notre revue».

Il nous apparaît que LED-MICRO comporte deux catégories de lecteurs : — d'une par les débutants, qui sont tous intéressés par le cours de programmation, et dont beaucoup apprécient le cours d'électronique digitale ; — d'autre part les fortiches qui ne sont pas intéressés par le cours de programmation.

De toute évidence, M.P. fait partie des fortiches.

Il fut une période où, dans le cadre des rubriques pédagogiques, je pensai faire une part égale aux débutants et aux fortiches... mais les fortiches ont à leur disposition une multitude de revues professionnelles, et semblent très minoritaires parmi vous. Le directeur de la publication vous a expliqué dans l'éditorial du numéro 12 les conclusions qu'il en a tirées.

En ce qui concerne les interfaces, notre ami P. Faugeras (spécialiste de ce problème) est en train de finir la série d'articles que vous souhaitez.



# COURS PRATIQUE DE MICROPROCESSEUR AVEC LE MICROPROFESSOR MPF-IB

## CINQUIEME PARTIE

### Le langage du Z80<sup>R</sup> (1)

#### SOMMAIRE

##### I. INTRODUCTION

##### II. TRANSFERT DE DONNEES (1 OCTET)

- II. 1. Définition d'un transfert
- II. 2. Transfert entre registres
- II. 3. Transfert entre registre A et case mémoire
- II. 4. Exemples
- II. 5. Chargement immédiat
- II. 6. Transfert avec adressage par (BC) ou (DE)
- II. 7. Transfert avec adressage par (HL)
- II. 8. Exemples
- II. 9. Transfert avec les registres «R» et «I»
- II.10. Transfert avec registre d'index
- II.11. Conclusions sur les transferts 1 octet

##### REPertoire INSTRUCTIONS DU Z-80<sup>R</sup>

##### EXERCICES

#### I. INTRODUCTION

Les quatre premières parties de notre cours de microprocesseur avaient pour but de présenter au lecteur les différents éléments internes qui constituent un microprocesseur, ainsi que quelques composants externes (mémoires et circuits périphériques) indispensables.

Comme nous l'avons déjà dit, la connaissance des aspects matériels n'est pas réellement indispensable pour une bonne compréhension du logiciel, mais elle y contribue cependant beaucoup. Aussi n'hésitez pas à effectuer de temps à autre un petit retour en arrière et notamment à la deuxième partie (LED-MICRO n° 10) intitulée «Hardware du Z80», de manière à bien «mémoriser» (pour rafraîchir...) l'architecture interne du micro.

Avec cette cinquième partie, nous abordons la seconde phase consacrée à l'aspect logiciel (software en anglais). De nombreux exemples et exercices, exécutables avec notre Microprofessor MPF-IB, vous permettront d'acquérir une bonne maîtrise de la programmation en langage machine.

Nous présentons à la fin de cette partie, le répertoire complet du jeu d'instructions du Z80<sup>R</sup>. Pour chacune d'elles, nous indiquons le code mnémorique et un bref descriptif.

Les instructions sont rangées dans un ordre alphabétique, qui n'est pas celui de notre étude pour laquelle

nous avons préféré un ordre par complexité croissante, mieux adapté.

#### II. TRANSFERT DE DONNEES (1 OCTET)

##### II.1. Définition d'un TRANSFERT

Historiquement, on distinguait deux types de mouvement de données. Le premier permettait de charger (LOAD) un registre du CPU avec le contenu d'une case mémoire. Le second, l'opération inverse, consistait à enregistrer une donnée dans la mémoire, elle portait le nom de stockage (store).

Actuellement, l'instruction «transfert» désigne n'importe quel mouvement de données. La source et la destination sont définies dans la partie «opérande» de l'instruction. Le mnémorique du code de transfert (comme d'ailleurs la plupart des mnémoriques) varie d'un microprocesseur à un autre. Il en est ainsi dans le cas du Z80<sup>R</sup>. Ce code est LD (abréviation de LOAD), tandis que pour le 8080, le mnémorique est MOV (abréviation de MOVE).

Toute instruction commençant par «LD» correspond à un mouvement de données : transfert intégral du contenu d'un registre dans un emplacement mémoire spécifié, ou copie conforme du contenu d'un emplacement mémoire dans un registre donné. Dans tous les cas, le registre ou emplacement mémoire «origine» constitue la «source» et reste inchangé après l'exécution de l'instruction.

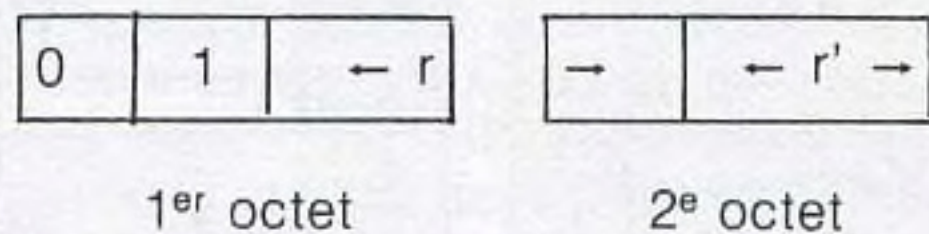


L'emplacement qui reçoit la donnée est le «destinataire». Avant l'exécution de l'opération de transfert, le registre destinataire contient une certaine donnée qui sera détruite par l'opération «LOAD», ou la nouvelle donnée, objet du transfert, se substitue au contenu présent avant l'exécution de l'opération.

Les tailles (1 ou 2 octets) du destinataire et de la source doivent être identiques. Nous reviendrons sur ce point très important dans les transferts de 2 octets.

Le schéma symbolique d'une opération de transfert est le schéma 1.

Et le code binaire s'obtient de la manière suivante :



avec r, r'

1 1 1 = A  
 0 0 0 = B  
 0 0 1 = C  
 0 1 0 = D  
 0 1 1 = E  
 1 0 0 = H  
 1 0 1 = L

c) Lire les contenus des registres A et B.

Vous devez constater que le contenu du registre source B contient toujours 24 (inchangé) tandis que le registre destinataire A contient désormais lui aussi 24 au lieu de 00.

#### Exercice 2

Pour vous entraîner, déterminez les codes hexadécimaux pour les instructions suivantes :

LD C, A  
 LD E, C  
 LD D, E  
 LD L, D  
 LD H, L

Constituez un second programme (à partir de 1802) avec les cinq instructions de chargement, dans l'ordre indiqué, sans oublier de le terminer par F7. Introduisez-le et exécutez-le. Examinez tous les registres, ils doivent tous contenir 24.

Réponse :

1802	4F	LD C, A
1803	59	LD E, C
1804	53	LD D, E
1805	6A	LD L, D
1806	65	LD H, L
1807	F7	FIN. Retour au moniteur

### II.3. Transferts entre registre A et case mémoire.

Dans le paragraphe précédent, l'opération consistait à recopier le contenu d'un registre dans un autre registre : l'opération était interne au CPU. Nous étudierons maintenant les opérations de **transferts entre l'accumulateur et la mémoire externe**.

Dans ce type d'instruction, seul l'accumulateur peut intervenir comme registre «Destinataire» ou «Source».

Le schéma de l'instruction du chargement de A avec le contenu (1 octet) d'un emplacement mémoire quelconque désigné par «nn» (2 octets pour l'adresse)(voir schéma 2)

Exemple :

Soit à transférer dans A le contenu de l'emplacement d'adresse 04 B1. Qui, dans le cas du MPF-1B, appartient à la zone «MONITEUR» et contient 85 (voir page 29 du listing source).

Le programme correspondant à cette opération est indiqué au schéma 3.

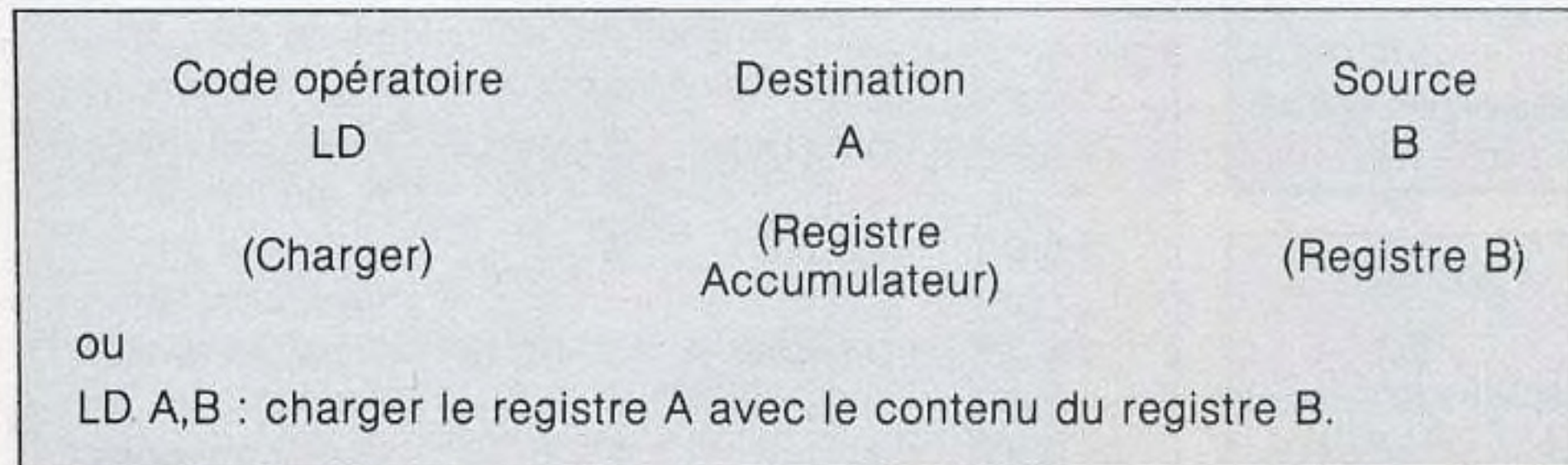


Schéma 1

Il est important de noter que la «**DESTINATION**» précède la «**SOURCE**» dans l'écriture symbolique.

Nous étudierons tout d'abord les «transferts de données» qui se présentent sous la forme d'un OCTET (8 bits).

### II.2. Transfert entre registres

Nous avons vu dans ce qui précède (LED-MICRO n° 10) que le CPU dispose, en plus du registre Accumulateur A, de six autres registres auxiliaires. Dans les opérations de transfert entre registres, chacun d'eux peut être soit la «source» soit la «DESTINATION». Le nombre d'instructions de transfert entre registres est de 7x7, soit 49.

**Nota :** Nous rappelons qu'à un instant donné, un seul jeu de registres A, B, C, D, E, H ou A', B', C', D', E', H' et L' est **en service** tandis que l'autre est en «**stand by**» : **ce qui exclut les transferts de registre à registre prime et vice versa.**

Le code symbolique des opérations de transfert entre registres est :

LD r', r

avec :

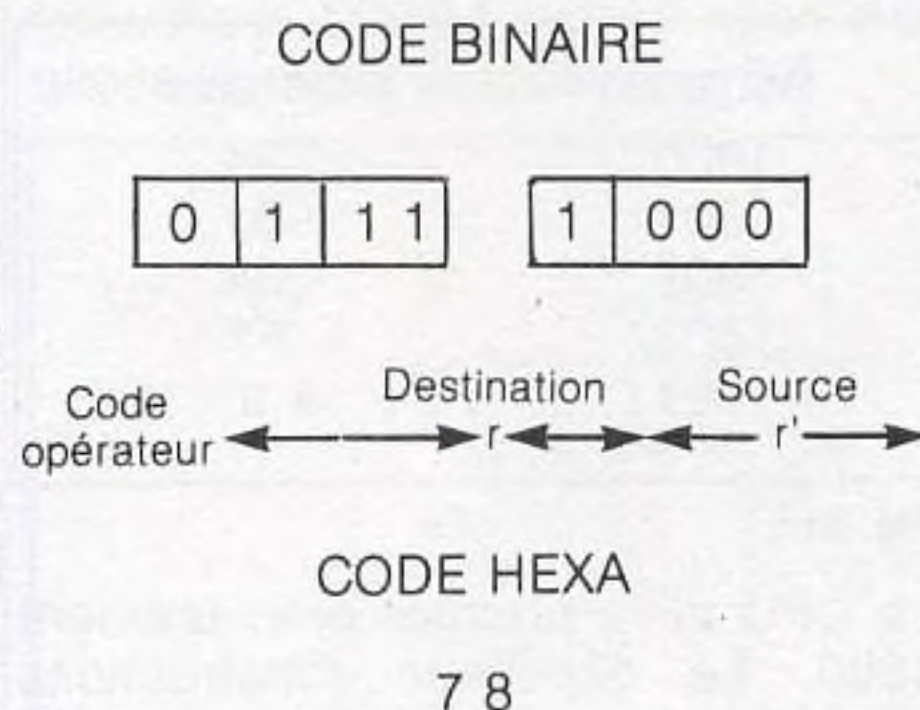
r : registre destinataire

r' : registre source.

(Les registres r et r' appartiennent **au même jeu de registres**, voir la note qui précède).

Exemple :

Etablissons le code de LD A,B. Pour cela, il suffit de remplacer r par 111 qui représente le registre A et r' par 000 qui représente le registre B, ce qui donne :



Exercice 1

Avec le MPF-1B, charger respectivement sous le contrôle du MONITEUR, les registres A avec 00 et B avec 24H (REG, AF, 00, etc.)

Introduire le programme suivant à partir de 1800H :

1800	78	LD A, B
1801	F7	FIN (retour au moniteur)

a) Lire le contenu des registres A et B et vérifier qu'ils contiennent bien respectivement 00 et 24.

b) Exécuter le programme qui commence en 1800.



Code opératoire LD	Destination A	Source (nn)
Charger	Registre A	Contenu de l'emplacement d'adresse «nn»

ou  
LD A,(nn) : charger A avec le contenu de l'emplacement d'adresse nn.

Schéma 2

Code opératoire LD	Destination (nn)	Source A
Charger	(emplacement d'adresse nn)	Registre A

ou  
LD (nn), A : charger l'emplacement d'adresse nn avec l'octet contenu dans le registre A.

Schéma 4

Code opératoire LD	Destination r	Donnée n
(charger)	(registre r)	(l'octet «donnée»)

ou LD r,n : charger le registre r avec «n», où r qui désigne l'un des 7 registres comme dans l'instruction LD r,r' (figure 98).

Schéma 5

1 8 0 0	3A B1 04	LD A,(04 B1)
1 8 0 3	F7	FIN (retour au moniteur)

Schéma 3

1 8 0 0	32 00 19	LD (1900), A
1 8 0 3	F7	FIN (retour au moniteur)

Tableau 1

3 A	0011 1010	LD A,(nn)
3 2	0011 0010	LD (nn), A

Tableau 2

Adresse	Code opératoire	Code machine	Commentaires
1 8 0 0	LD A,(1A00)	3A 00 1A	Charge du 1 <sup>er</sup> opérande dans A
1 8 0 3	LD C ← A	4F	Transfert A dans C
1 8 0 4	LD A,(1A01)	3A 01 1A	Charge du 2 <sup>e</sup> opérande dans A
1 8 0 7	ADD A,C	81	Add de A à C, résultat dans A
1 8 0 8	LD (1A02), A	3E 02 1A	Transfert de A dans (nn)
1 8 0 B	FIN	F7	Retour au moniteur

Tableau 3

Après l'exécution de ce programme, le contenu du registre A est 85 : copie de l'emplacement 04 B1.

Quelques remarques s'imposent à propos de cette instruction LD A,(nn).  
1. Seul le registre A peut être chargé à partir d'un emplacement mémoire quelconque. Le registre Accumulateur A est contenu **implicitement** dans le code opérateur 3A de l'instruction ; ce qui n'était pas le cas dans l'instruction LD r,r'.

2. L'adresse nn de la **source est placée entre parenthèses**. La présence des parenthèses ( ) signifie que le code qu'elles renferment ne doit pas être considéré comme une quantité mais comme l'**ADRESSE de l'EMPLACEMENT** qui contient la donnée sur laquelle porte l'opération (ici le transfert).

3. Le code objet correspondant à l'instruction LD A, (04 B1) est 3A B1 04. **L'octet de poids faible B1, partie inférieure de l'adresse précède l'octet de poids fort 04.** Cette inversion devra toujours avoir lieu pour les codes de deux octets qui représentent une adresse ou une quantité hexadécimale sur 16 bits. Jusqu'à présent, nous n'avons rencontré que des instructions d'un seul octet, tandis que celle-ci en comporte 3.

Etudions ce qui se passe.

Le programme, défini plus haut, est chargé dans la mémoire RAM à partir de l'adresse 1800 H comme l'indique la figure 94.

Adresse	Donnée
1800	3A
1801	B1
1802	04
1803	F7
1804	..

Fig. 94

Le CPU lit le premier emplacement 1800. Le décodeur d'instructions identifie le premier octet «3A» comme étant le premier byte d'une instruction qui en comporte trois. **Le CPU procède alors, avant toute exécution, à la recherche des deux autres octets**, qu'il sait contigus au premier. Comme ceux-ci sont respectivement les 8 bits de poids faible et les 8 bits de poids forts de l'adresse, ils sont déposés sur le bus d'adresses pour sélectionner l'emplacement «Source».

Puis l'exécution de transfert est exécutée.

L'opération inverse qui consiste à transférer le contenu du registre A dans un emplacement mémoire, pré-

sente une structure similaire. Dans ce cas, le registre A constitue la «source» tandis que l'emplacement d'adresses «nn» devient la «destination». A noter toutefois que l'emplacement doit appartenir à la zone mémoire vive, sinon l'opération serait inefficace. Dans l'instruction précédente, l'emplacement mémoire, qui constituait la source, pouvait être soit une zone mémoire morte, comme l'exemple l'indiquait, soit une zone mémoire vive, comme nous le montrerons dans un prochain exemple (voir schéma 4).

Le schéma est :

Exemple

Supposons que le registre A contienne l'octet 85 et que l'on souhaite le transférer dans la case d'adresse 1900. Le programme correspondant est :

Les trois remarques que nous avons faites pour l'instruction LD A,(nn) restent valables ; seul le contenu du registre A peut être transféré en mémoire, l'adresse de l'emplacement destinataire (exclusivement RAM) est placée entre parenthèses et comme précédemment le code opérateur 32 est suivi immédiatement de l'octet de poids faible (00) puis de l'octet de poids fort 19 de l'adresse. L'examen **des codes binaires** des



deux instructions montre que seule la valeur du bit 3 du code opération diffère d'une instruction à l'autre (voir tableau 2).

Ces deux instructions nous permettent d'utiliser la mémoire vive comme une zone de stockage «longue durée», très étendue, et ainsi de libérer les registres auxiliaires.

#### II.4. Exemple

Nous en savons maintenant assez pour écrire un programme plus conséquent.

Objet : Réaliser l'addition du contenu des cases mémoires 1A00H et 1A01H, puis placer le résultat somme dans l'emplacement 1A02H. (Il s'agit de quantités décimales).

Ecrivez ce programme, introduisez-le dans votre MPF-1B à partir de l'adresse 1800H. Vérifiez.

**Nota :** Les quantités placées dans 1900 et 1901 sont des quantités hexadécimales, la somme sera donc effectuée en base 16. Pour obtenir une addition décimale, il suffit de rajouter DAA (code 27) après l'addition et de choisir des quantités initiales telles que leur somme soit inférieure à 99.

Essayez d'écrire le programme avant de lire la solution ci-après.

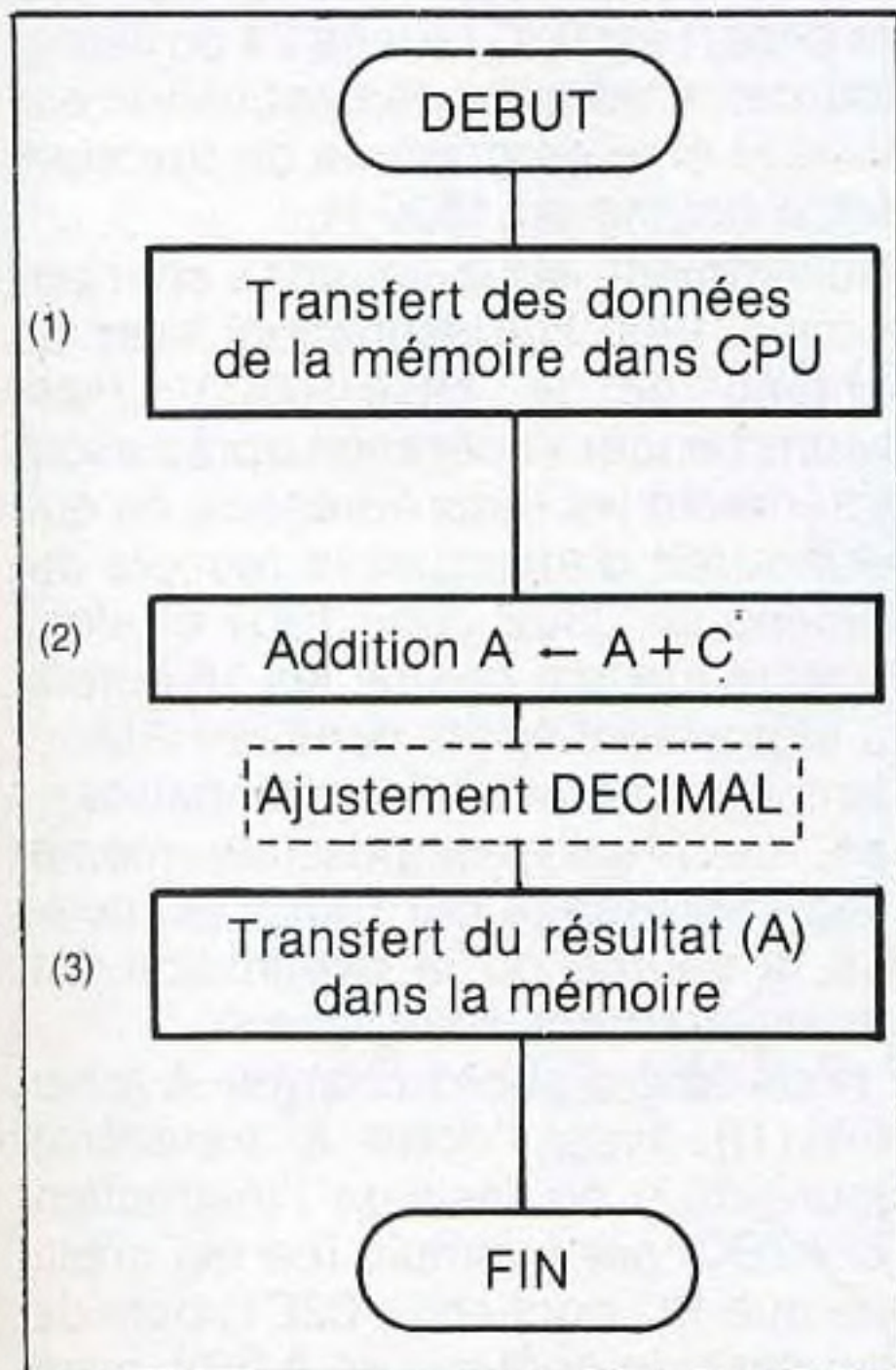


Fig. 95

#### SOLUTION

La figure 95 décrit d'une manière succincte les trois phases de notre

programme. C'est un organigramme (ou Flow Chart).

La phase (1) est la préparation des données dans le CPU.

La phase (2) l'addition elle-même avec le résultat dans l'accumulateur. La phase (3) le transfert de la somme dans la mémoire (voir tableau 3).

Le commentaire est facultatif. Il facilite toutefois la compréhension du programme. Chaque instruction occupe une ligne. Chaque octet occupe un emplacement «mémoire», ce qui conduit à un empilement continu des bytes dans la mémoire.

La figure 96 indique la disposition du programme dans la mémoire.

1800	3A
1801	00
1802	1A
1803	4F
1804	3A
1805	01
1806	1A
1807	81
1808	3E
1809	02
180A	1A
180B	F7

Fig. 96

**Nota :** Pour ceux qui n'auraient pas trouvé que LD C,A est 4F nous indiquons le détail ci-après (fig. 97) :

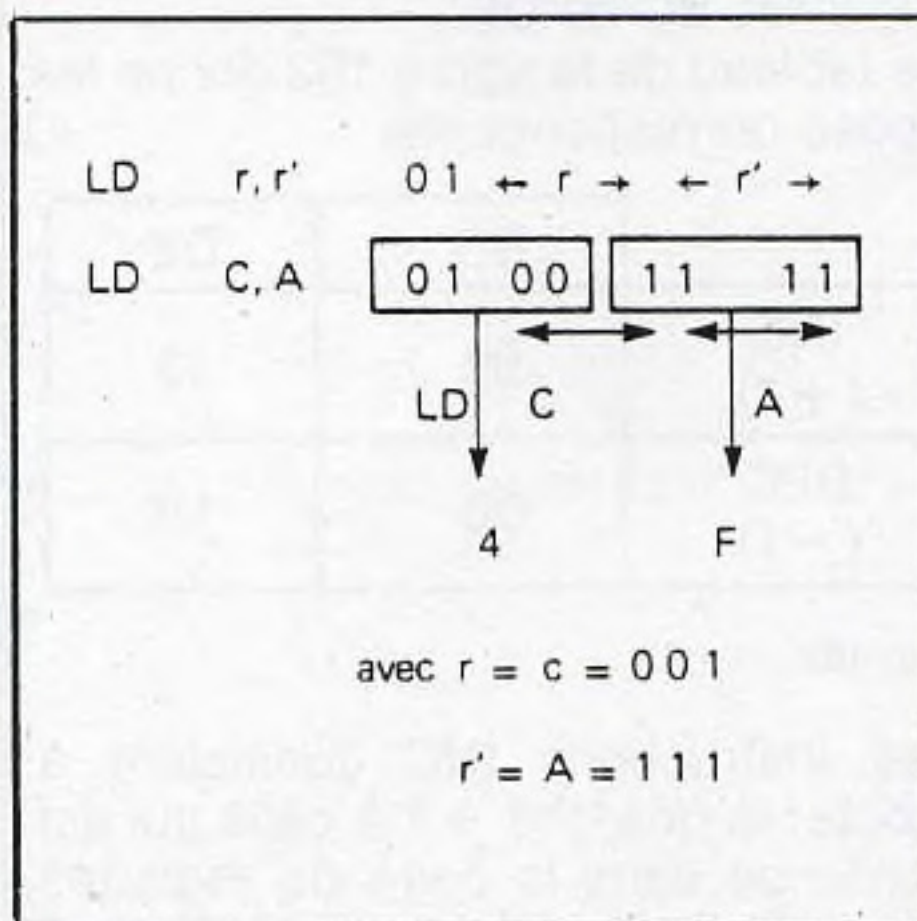


Fig. 97

#### II.5. Chargement immédiat

Une autre manière de charger un registre est d'utiliser les instructions de chargement immédiat. Le registre destinataire est identifié dans le premier octet et la donnée est l'octet immédiatement suivant.

Le schéma de l'instruction est donné au schéma 5.

Le code binaire est :

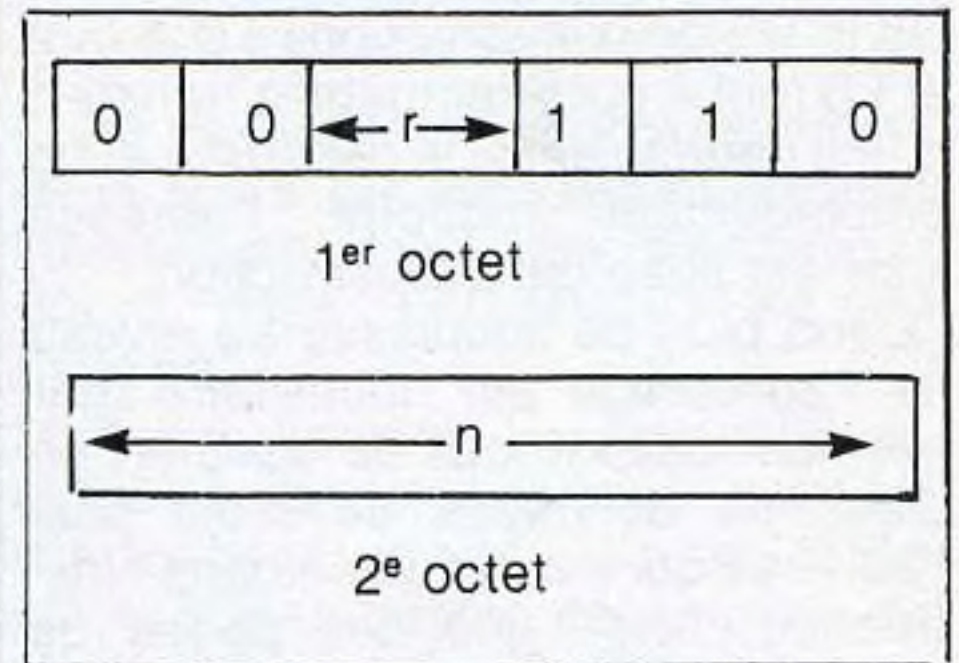


Fig. 98

Exemples

n représente une quantité hexadécimale comprise entre 00H et FFH, c'est-à-dire une valeur décimale arithmétique comprise entre 0 et 255.

1) Soit à charger le registre A avec 3F. Le code symbolique est LD A,3F ce qui donne pour le code binaire :

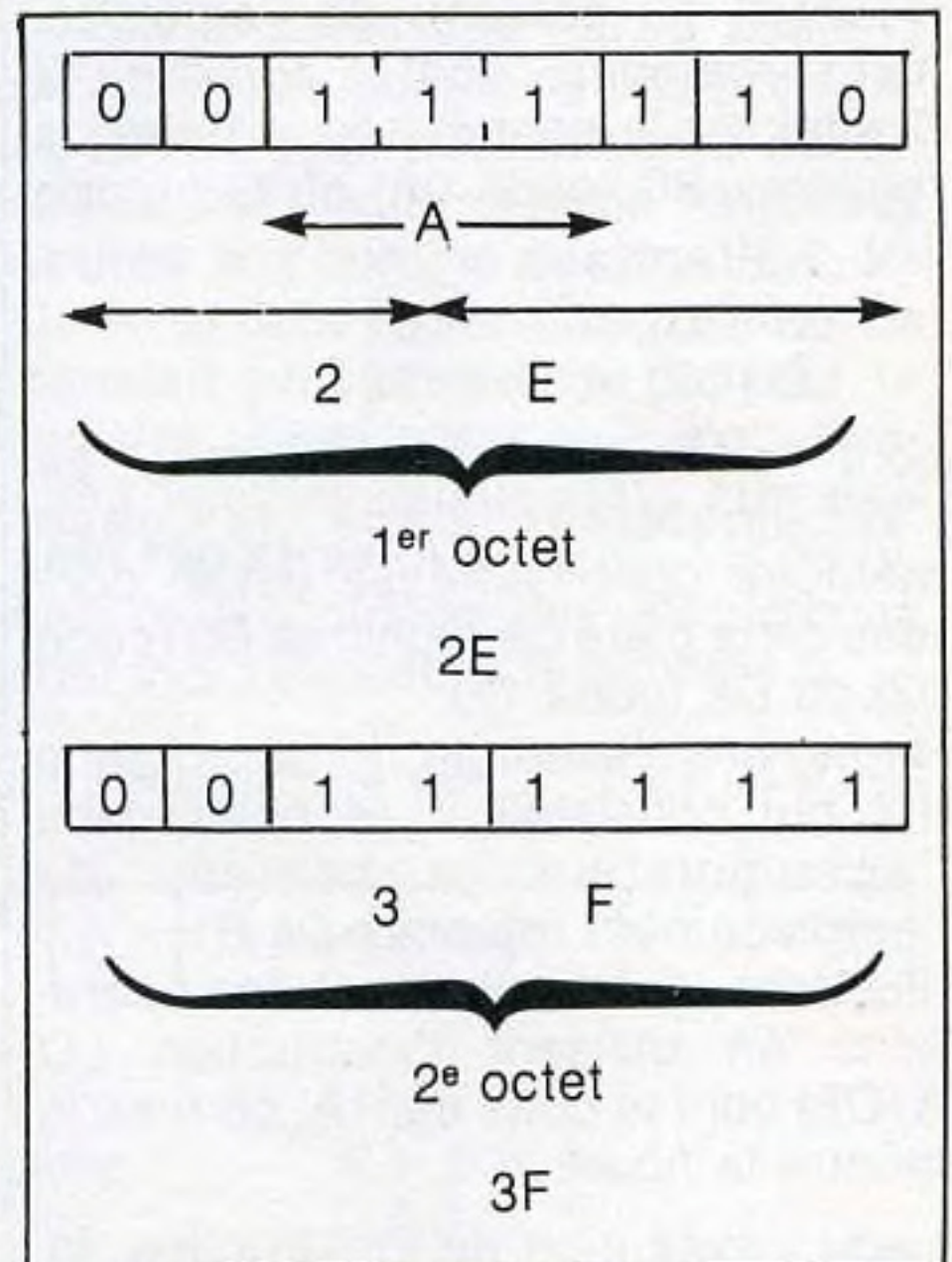


Fig. 99

et pour le code hexadécimal :

LD A,3F : 2E 3F

2) Soit à charger le registre B avec 76.

Le code binaire du premier octet est :

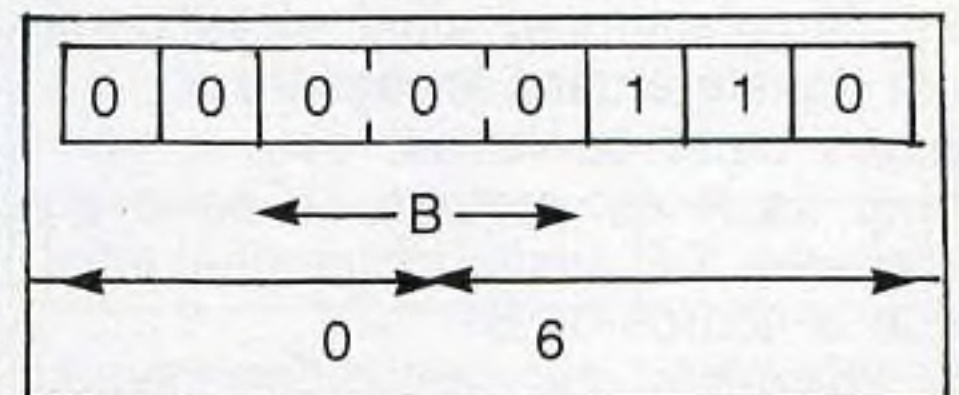


Fig. 100

ce qui donne pour le code hexadécimal : LD B,76 : 06 76



## II.6. Transfert avec adressage par (BC) ou (DE)

Dans les deux instructions LD A,(nn) et LD (nn),A qui effectuent le transfert d'une donnée entre le registre A et un emplacement mémoire, l'adresse «nn» est figée dans l'instruction.

Quand plus de souplesse au niveau de l'adressage est nécessaire (par exemple dans le cas de transfert de blocs de données), ce n'est plus l'adresse qui est spécifiée dans l'instruction mais l'une des paires de registres doubles BC ou DE. Dans ce cas, **c'est le contenu de la paire de registres désignée qui constitue l'adresse «Source» ou «Destination».**

Les quatre codes mnémoniques et hexadécimaux sont :

- a) LD A,(BC)      0A  
LD A,(DE)      1A

pour le transfert dans l'accumulateur (1 octet) du contenu de l'emplacement mémoire dont l'adresse (2 octets) est le contenu de la paire de registres BC (code 0A) ou DE (code 1A).

- b) LD (BC), A      02  
LD (DE), A      12

pour le transfert du contenu de l'accumulateur dans la case mémoire dont l'adresse est le contenu de la paire de registres BC (code 02) ou DE (code 12).

Reprenons l'exemple du paragraphe II.3 qui consistait à recopier dans l'accumulateur le contenu de l'emplacement mémoire 04 B1.

Illustrons le cheminement des opérations en utilisant l'instruction LD A,(DE) dont le code est 1A, comme le montre la figure 101.

Avant l'exécution de l'instruction, la paire de registres DE contient la quantité 04 B1 (l'octet de poids faible dans E, l'octet de poids fort dans D). Au moment de l'exécution de l'instruction 1A, le CPU utilise le contenu de DE comme pointeur, dépose celui-ci sur le bus d'adresses et le contenu de l'emplacement ainsi sélectionné est transféré dans le registre A.

Après cette opération, seul le contenu de A est modifié, la paire de registres DE reste inchangée ainsi que la source 04 B1.

Si l'exemple qui vient d'être décrit, détaille le fonctionnement des opérations de transfert avec adressage par registre, il n'en montre que partiellement l'intérêt.

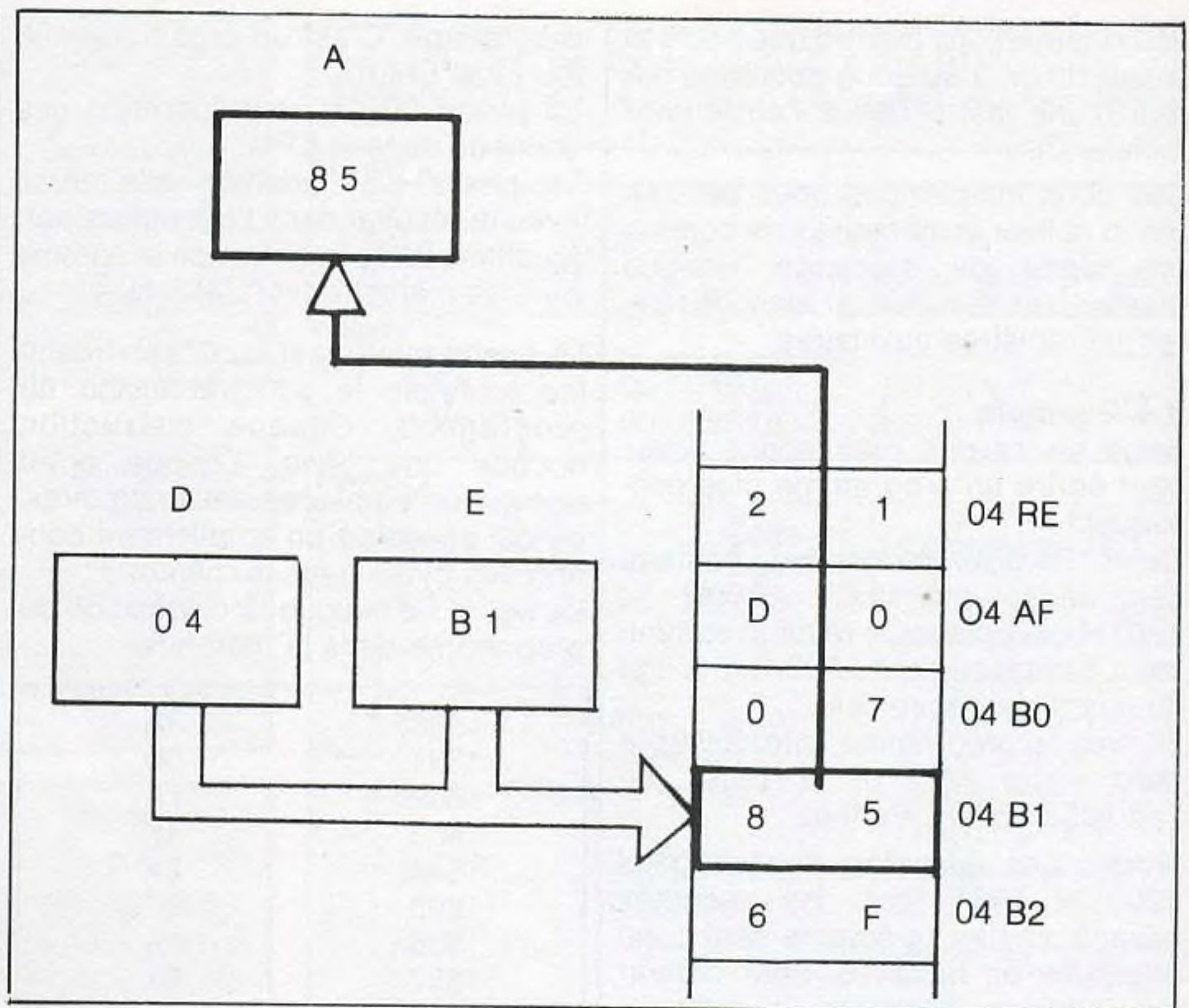


Fig. 101

Pour que le lecteur comprenne bien, nous allons donner un second exemple.

Auparavant, il nous faut anticiper sur le cours et introduire les instructions INC (incrément) et DEC (décrément) du contenu de la paire de registres BC ou DE.

Le tableau de la figure 102 donne les codes correspondants.

	BC	DE
INC (+1)	03	13
DEC (-1)	0B	1B

Fig. 102

Les instructions INC consistent à ajouter la quantité +1 à celle qui est contenue dans la paire de registres désignée.

Ainsi, si le contenu de DE est 2050 après l'exécution de l'instruction 13, le contenu est 2051.

De même, si BC contient 1EFF après l'exécution de l'instruction 03, le contenu est 1F00.

Les instructions DEC opèrent d'une manière identique, mais la quantité 1 est retirée du contenu de la paire de registres au lieu d'y être ajoutée.

Le problème à traiter consiste à transférer un bloc de 18 (décimal) octets de la zone mémoire ROM pour le placer dans la mémoire vive RAM : dans cet exemple, il s'agit de la sous-routine d'initialisation dont l'adresse de départ est 03C1 (page 24 du listing source). L'adresse de destination est 1900 H et le programme de transfert a son origine en 1800 H.

Intuitivement, vous voudriez charger la case 1900 H (destination) avec le contenu de la case 03C1. Puis recommencer l'opération après avoir incrémenté les deux adresses, ce qui permettrait d'effectuer la recopie du contenu de 03C2 dans 1901 et ainsi de suite jusqu'à ce que les 18 octets du bloc soient écrits dans la RAM.

Dans l'état actuel de nos connaissances, ce n'est pas possible même avec l'adressage par registres, puisque la source ou la destination est nécessairement le registre A.

Il nous faut d'abord charger A [chemin (1)] avec l'octet à transférer (figure 103) en utilisant l'instruction LD A,(BC) par exemple (ce qui implique que BC contienne 03C1) puis de transférer le contenu de A [(2)], avec l'instruction LD (DE), A (ce qui implique que DE contienne 1900).

Ceci étant fait, on incrémente BC qui pointe sur l'octet suivant (03C2) ainsi



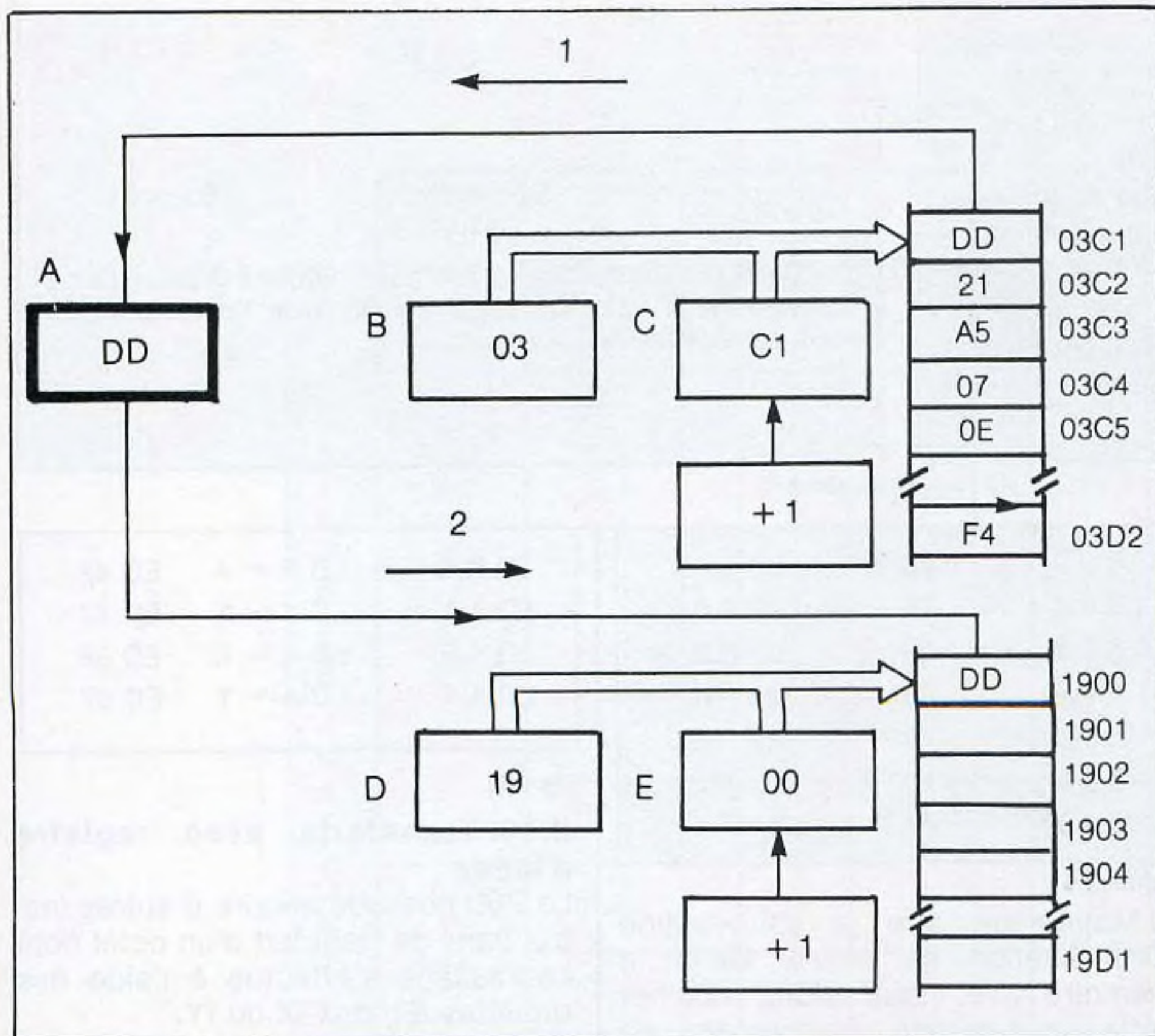


Fig. 103

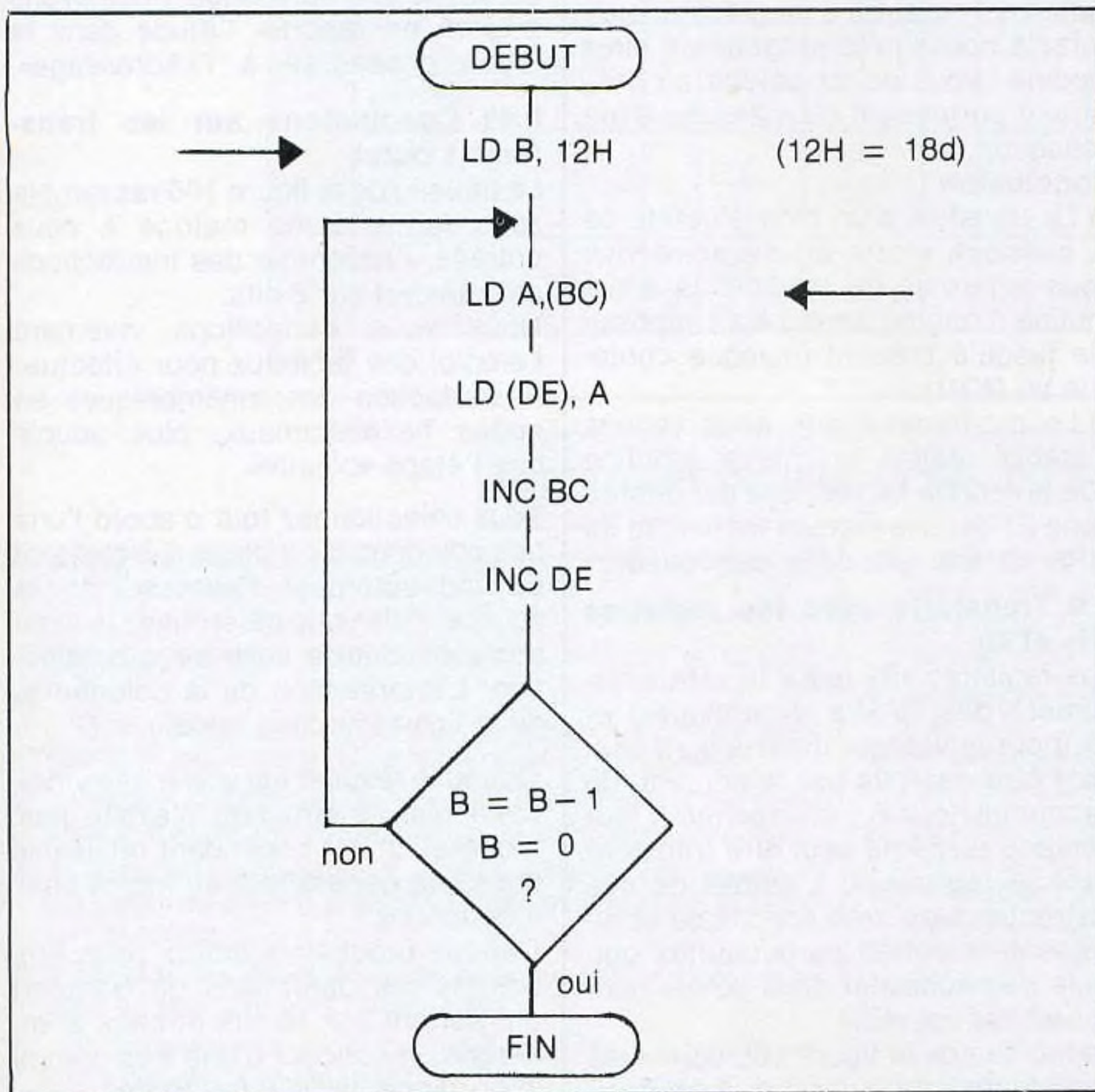


Fig. 104

que DE qui adresse l'emplacement 1901. En répétant 18 fois l'opération, le bloc sera recopié.

C'est ce que traduit l'organigramme de la figure 104 dans lequel apparaît l'instruction DJNZ que nous avons décrit dans LED-MICRO n° 10 page 62.

Malheureusement, le registre B est utilisé à la fois comme pointeur d'adresses dans l'instruction LD A,(BC) et comme décompteur de boucles (instruction DJNZ).

Sommes-nous dans une impasse ? Non, tant s'en faut. Une première solution (il y en a d'autres) est d'utiliser l'instruction de transfert avec adressage par la paire de registres HL.

### II.7. Transfert avec adressage par (HL)

Dans les quatre instructions de transfert avec adressage par le contenu de l'une des paires de registres BC ou DE, le registre A intervenait IMPLICITEMENT soit comme **élément source** soit comme **destination**.

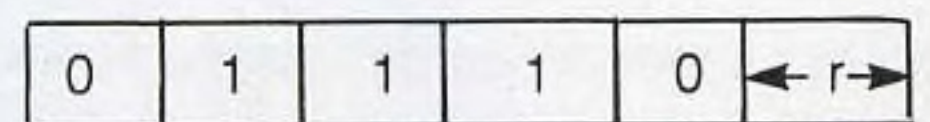
Dans les deux types d'instructions de transfert avec adressage par (HL), le registre utilisé n'est plus exclusivement l'accumulateur **mais l'un des sept registres** (A ou B, C, ... H ou L). Les mnémoniques des instructions sont indiquées dans le schéma 6.

Le code binaire est :



avec : r :	1 1 1	A
	0 0 0	B
	0 0 1	C
	0 1 0	D
	0 1 1	E
	1 0 0	H
	1 0 1	L

Le code binaire est :



même signification que r précédemment.

### II.8. Exemples

1) Soit à charger le registre D avec



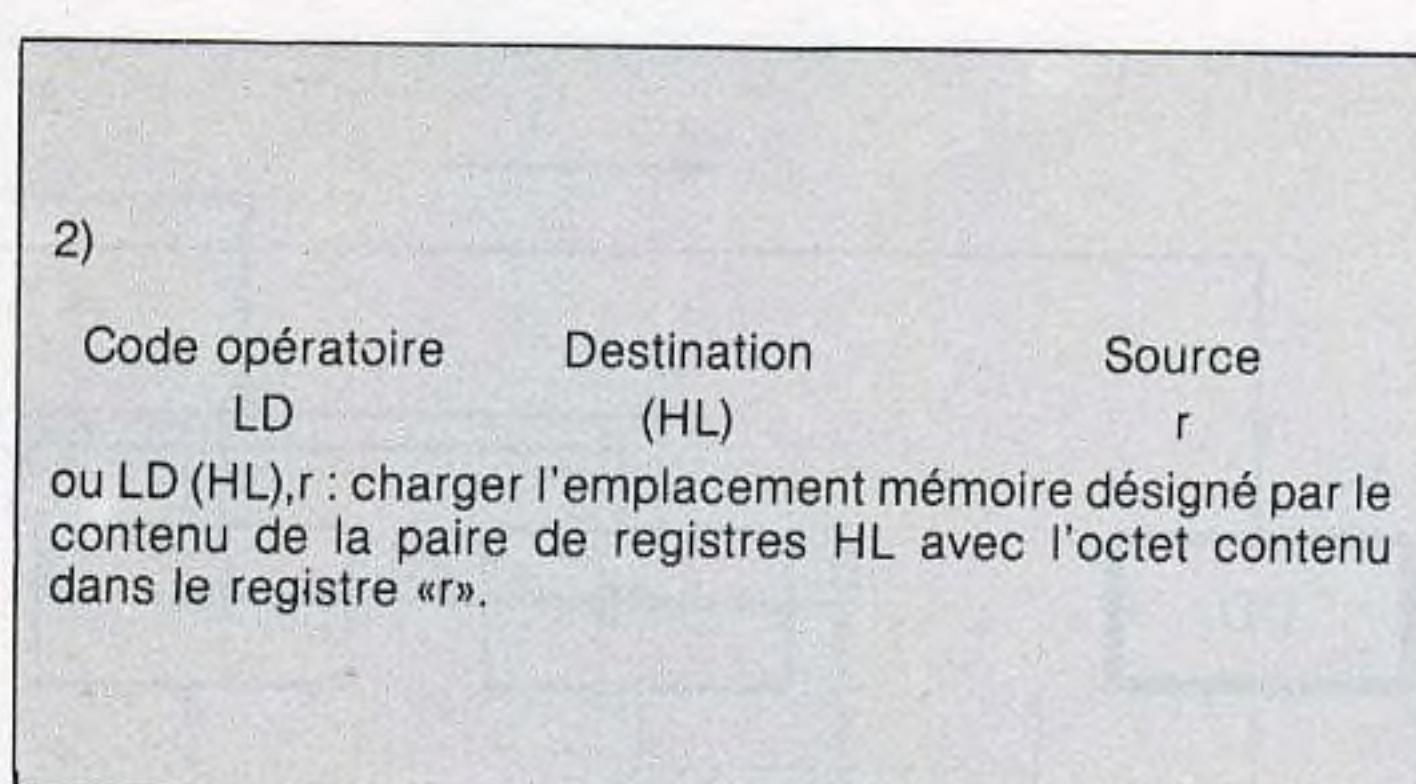
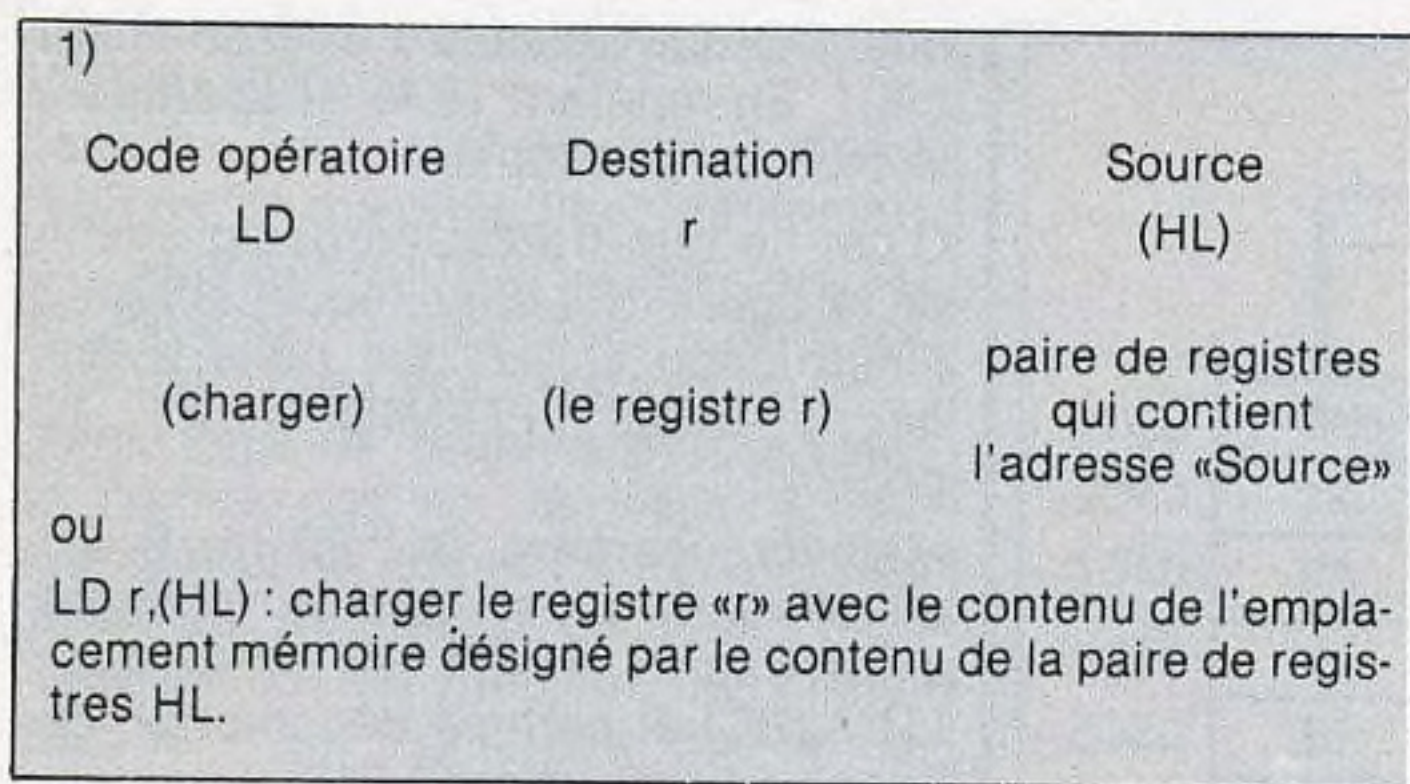


Schéma 6

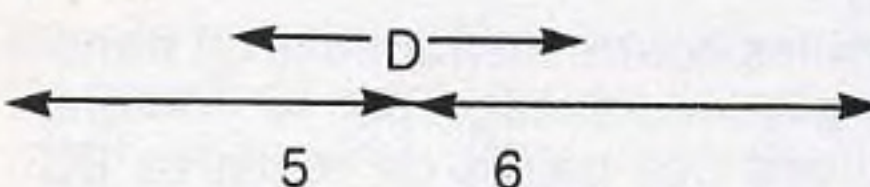
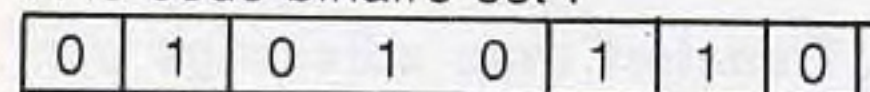
Schéma 7

l'octet de l'emplacement adressé par la paire de registres HL.

Le code mnémotique est :

LD D,(HL)

et le code binaire est :

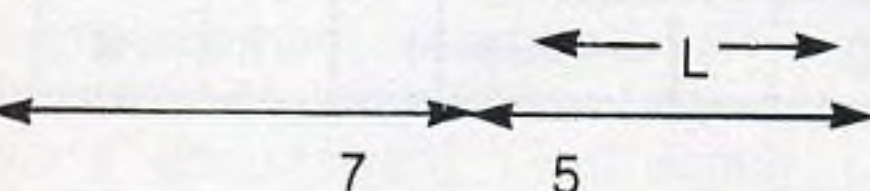
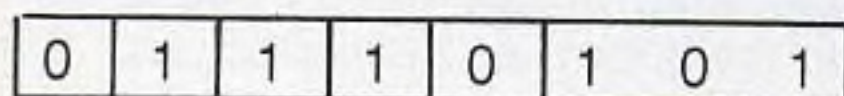


Ce qui donne pour le code hexadécimal : 56.

2) Soit à charger l'emplacement dont l'adresse est le contenu de la paire HL avec l'octet contenu dans le registre L, le code mnémotique est :

LD (HL), L

et le code binaire est :



ce qui donne pour le code hexadécimal : 75. (voir schéma 7).

3) En remplaçant, dans l'exemple du transfert d'un bloc (fig. 104) Ld A,(BC) par Ld A,(HL) et en chargeant HL avec 03C1 d'une part et en incrémentant HL (code 23) au lieu de BC, nous obtenons le programme complet pour le transfert d'un bloc de données.

Etablissez le programme, introduisez-le et exécutez-le.

**Nota :** L'instruction de test est 10 et la valeur du déplacement FA ce qui donne 10 FA.

Si c'est correct, vous devez voir apparaître le message UPF-1 comme lors d'une mise sous tension. Voici la solution (voir schéma 8).

1 8 0 0	06 12	LD B,12
1 8 0 2	7E	LD A,(HL)
1 8 0 3	12	LD (DE),A
1 8 0 4	23	INC HL
1 8 0 5	13	INC DE
1 8 0 6	10 FA	DJNZ, -5
1 8 0 8	F7	FIN

Schéma 8

4) Maintenant que la sous-routine d'initialisation se trouve dans la mémoire vive, nous allons modifier cette sous-routine. Le contenu du dernier octet (adresse : 19D1) est F4, mettez-y l'octet EE à sa place et exécutez à nouveau le programme ainsi modifié : vous devez obtenir un défilement permanent du message d'initialisation.

#### Conclusion :

1) Le transfert d'un bloc d'octets de la mémoire morte en mémoire vive nous a permis de modifier la sous-routine d'origine, ce qui était impossible jusqu'à présent (puisque contenue en ROM).

2) Le programme que nous venons d'établir réalise la même fonction que la touche MOVE. Elle est décrite page 27 dans le manuel technique du MPF-1B. Essayez cette commande.

#### II.9. Transferts avec les registres «R» et «I»

Les registres «R» (pour le rafraichissement des RAM's dynamiques) et «I» (pour le vecteur Interruption) peuvent être chargés par le contenu de l'accumulateur A ; inversement, leur contenu respectif peut être transféré dans le registre A. L'emploi de ces instructions est très spécifique et ne présente d'autres particularités que celle de nécessiter deux octets dont le premier est «ED».

Le tableau de la figure 105 donne les mnémotiques et codes hexadécimaux pour ces quatre instructions.

LD R,A	LD R ← A	ED 4F
LD 1,A	LD 1 ← A	ED 47
LD A,R	LD A ← R	ED 5F
LD A,1	LD A ← 1	ED 57

Fig. 105

#### II.10. Transferts avec registre d'Index

Le Z80 possède encore 6 autres instructions de transfert d'un octet dont l'adressage s'effectue à l'aide des registres d'Index IX ou IY.

Compte tenu de la légère complexité de ce mode d'adressage, nous avons préféré en reporter l'étude dans la leçon consacrée à l'«Adressage».

#### II.11. Conclusions sur les transferts 1 octet

Le tableau de la figure 106 rassemble sous forme d'une matrice à deux entrées, l'ensemble des instructions de transfert sur 8 bits.

Nous vous conseillons vivement l'emploi des tableaux pour effectuer la traduction des mnémotiques en codes hexadécimaux, plus souple que l'étape «binaire».

Vous sélectionnez tout d'abord l'une des colonnes qui indique directement ou indirectement l'adresse de la source. Puis vous déterminez la ligne correspondant à celle de la destination. L'intersection de la colonne et de la ligne fournit le code.

Quand le résultat est une «case vide», c'est que l'instruction n'existe pas. L'opération est cependant réalisable mais elle nécessitera au moins deux instructions.

Dans le prochain numéro, nous étudierons les opérations de transfert qui portent sur 16 bits et nous aborderons un concept d'une très grande importance, la PILE (ou Stack).

**Philippe Duquesne**



		SOURCE														
		Imp.		Registre								Reg. Ind.			Adr. éten	
Mode d'adress.		I	R	A	B	C	D	E	H	L	(BC)	(DE)	(HL)	(nn)	n	
D E S T I N A T I O N	Registre	A	ED 57	ED 5F	7F	78	79	7A	7B	7C	7D	0A	1A	7E	3A n n	3E n
		B			47	40	41	42	43	44	45			46		06 n
		C			4F	48	49	4A	4B	4C	4D			4E		0E n
		D			57	50	51	52	53	54	55			56		16 n
		E			5F	58	59	5A	5B	5C	5D			5E		1E n
		H			67	60	61	62	63	64	65			66		26 n
		L			6F	68	69	6A	6B	6C	6D			6E		26 n
	Reg. indir.	(BC)			02											
		(DE)			12											
		(HL)			77	70	71	72	73	74	75					36 n
	Adresse étendue	(nn)			32 n n											
	Impl.	I			ED 47											
		R			ED 4F											

Fig. 106

## REPERTOIRE INSTRUCTIONS DU Z-80<sup>R</sup>

Notations utilisées dans la description en mnémoniques du répertoire d'instructions du Z80 :

- 1) r désigne n'importe lequel des registres suivants : A, B, C, D, E, H, L.
- 2) (HL) désigne le contenu d'un emplacement mémoire dont l'adresse est le contenu de la paire de registres HL.
- 3) n désigne une donnée de un octet, dont la valeur est comprise entre 0 et 255 en décimal ou 00 et FF en hexadécimal.
- 4) nn désigne une quantité (donnée ou adresse) dont la valeur est comprise entre 0 et 65535 en décimal ou 0000 et FFFF en hexadécimal.
- 5) d désigne une quantité sur un octet en valeur algébrique, donc compris entre -128 et +127.
- 6) (nn) désigne le contenu d'un emplacement mémoire dont l'adresse est nn (2 octets).
- 7) b désigne une quantité de 0 à 7.
- 8) e désigne un déplacement sur un octet en valeur algébrique donc compris entre -126 et +129.
- 9) cc indique l'état des indicateurs pour les instructions conditionnelles JR, JP; CALL et RET.
- 10) qq désigne n'importe quelle paire de registres BC, DE, HL ou AF.
- 11) ss désigne l'une quelconque des paires de registres suivantes : BC, DE, HL et SP.
- 12) pp désigne l'une quelconque des paires de registres suivantes : BC, DE, IX et SP.
- 13) rr désigne l'une quelconque des paires de registres suivantes : BC, DE, IY et SP.
- 14) s désigne l'un des opérandes suivants : r, n, (HL), (IX + d), (IY + d).
- 15) dd désigne l'une des paires de registres suivants : BC, DE, HL et SP.
- 16) m désigne l'un des opérandes suivants : r, (HL), (IX + d), (IY + d).



Code mnémorique	Opération	Code mnémorique	Opération
ADC HL, ss	Additionner HL et le registre double ss avec le report. Résultat dans HL.	CPIR	Comparaison du contenu d'emplacement mémoire pointé par HL avec l'accumulateur. HL est incrémenté et BC décrétementé. L'opération est répétée jusqu'à ce que BC = 0. Résultat positionnement des indicateurs.
ADC A, s	Additionner l'accumulateur et l'opérande s avec le report. Résultat dans A.	CPL	Complémentation à 1 de l'accumulateur. Résultat dans l'accumulateur.
ADD A, n	Additionner l'accumulateur et la valeur n. Résultat dans A.	DAA	Ajustement décimal de l'accumulateur.
ADD A, r	Additionner l'accumulateur et le registre r. Résultat dans A.	DEC m	Décrémentation de l'opérande m.
ADD A, (HL)	Additionner l'accumulateur et l'emplacement mémoire dont l'adresse est le contenu de HL. Résultat dans A.	DEC IX	Décrémentation du registre IX.
ADD A, (IX + d)	Additionner l'accumulateur et l'emplacement mémoire dont l'adresse est le contenu de IX + d. Résultat dans A.	DEC IY	Décrémentation du registre IY.
ADD A, (IY + d)	Additionner l'accumulateur et l'emplacement mémoire dont l'adresse est le contenu de IY + d. Résultat dans A.	DEC ss	Décrémentation du registre double ss.
ADD HL, ss	Additionner HL et le registre double ss. Résultat dans HL.	DI	Interdiction des interruptions.
ADD IX, pp	Additionner IX et le registre double pp. Résultat dans IX.	DJNZ e	Décrémentation du registre B et saut relatif de e si B est égal à 0.
ADD IY, rr	Additionner IY et le registre double rr. Résultat dans IY.	EI	Autorisation des interruptions.
AND s	ET logique entre l'accumulateur et l'opérande s. Résultat dans A.	EX (SP), HL	Echange du registre HL avec le sommet de la pile.
BIT b, (HL)	Test du bit b de l'emplacement mémoire dont l'adresse est le contenu de HL. Résultat dans Z (indicateur).	EX (SP), IX	Echange du registre IX avec le sommet de la pile.
BIT b, (IX + d)	Test du bit b de l'emplacement mémoire dont l'adresse est le contenu de IX + d. Résultat dans Z (indicateur).	EX (SP), IY	Echange du registre IY avec le sommet de la pile.
BIT b, (IY + d)	Test du bit b de l'emplacement mémoire dont l'adresse est le contenu de IY + d. Résultat dans Z (indicateur).	EX AF, A'F'	Echange entre les registres AF et A'F'.
BIT b, r	Test du bit b du registre r. Résultat dans Z (indicateur).	EX DE, HL	Echange entre les registres DE et HL.
CALL cc, nn	Saut à la sous-routine d'adresse nn si la condition cc est vraie.	EXX	Echange entre les registres BC, DE et HL avec le contenu respectivement des registres B'C', D'E' et H'L'.
CALL nn	Saut incondionnel à la sous-routine d'adresse Nn.	HALT	Arrêt du CPU (attente d'une interruption ou RAZ).
CCF	Complémentation de l'indicateur C. Résultat dans C.	IMO	Sélection du mode d'interruption 0.
CP s	Comparaison de l'opérande s et de l'accumulateur. Résultat positionnement des indicateurs.	IM1	Sélection du mode d'interruption 1.
CPD	Comparaison du contenu d'emplacement mémoire pointé par HL avec l'accumulateur. HL et BC sont décrétementés. Résultat positionnement des indicateurs.	IM2	Sélection du mode d'interruption 2.
CPDR	Comparaison du contenu d'emplacement mémoire pointé par HL avec l'accumulateur. HL et BC sont décrétementés. L'opération est répétée jusqu'à ce que BC = 0. Résultat positionnement des indicateurs.	IN A, (n)	Chargement de l'accumulateur à partir du contenu du port d'adresse n.
CPI	Comparaison du contenu d'emplacement mémoire pointé par HL avec l'accumulateur. HL est incrémenté et BC décrétementé. Résultat positionnement des indicateurs.	IN r, (C)	Chargement du registre r à partir du contenu du port dont l'adresse est le contenu de C.
		INC (HL)	Incrémentation de l'emplacement mémoire pointé par HL.
		INC IX	Incrémentation de IX.
		INC (IX + d)	Incrémentation de l'emplacement mémoire pointé par IX + d.
		INC IY	Incrémentation de IY.
		INC (IY + d)	Incrémentation de l'emplacement mémoire pointé par IY + d.
		INC r	Incrémentation du registre r.
		INC ss	Incrémentation de la paire de registres ss.
		IND	Chargement de l'emplacement pointé par HL avec le contenu du port dont l'adresse est le contenu de C, décrémentation de HL et de B.
		INDR	Chargement de l'emplacement pointé par HL avec le contenu du port dont l'adresse est le contenu de C, décré-



Code mnémorique	Opération	Code mnémorique	Opération
	mentation de HL et de B. L'opération est répétée jusqu'à ce que $B = 0$ .		dont l'adresse est le contenu de HL avec celui du registre r.
INI	Chargement de l'emplacement pointé par HL avec le contenu du port dont l'adresse est le contenu de C, incrémentation de HL et décrémentation de B.	LD I, A	Chargement du registre I avec le contenu de A.
INIR	Chargement de l'emplacement pointé par HL avec le contenu du port dont l'adresse est le contenu de C, incrémentation de HL et décrémentation de B. L'opération est répétée jusqu'à ce que $B = 0$ .	LD IX, nn LD IX, (nn)	Chargement du registre IX avec nn. Chargement du registre IX avec la donnée contenue aux adresses nn et nn + 1.
JP (HL)	Saut inconditionnel à l'adresse contenue dans HL.	LD (IX + d), n LD (IX + d), r	Chargement de l'emplacement mémoire d'adresse IX + d avec la donnée n. Chargement de l'emplacement mémoire d'adresse IX + d avec le contenu du registre r.
JP (IX)	Saut inconditionnel à l'adresse contenue dans IX.	LD IY, nn LD IY, (nn)	Chargement du registre IY avec nn. Chargement du registre IY avec la donnée contenue aux adresses nn et nn + 1.
JP (IY)	Saut inconditionnel à l'adresse contenue dans IY.	LD (IY + d), n LD (IY + d), r	Chargement de l'emplacement mémoire d'adresse IY + d avec la donnée n. Chargement de l'emplacement mémoire d'adresse IY + d avec le contenu du registre r.
JP cc, nn	Saut à l'adresse nn, si la condition cc est vraie.		
JP nn	Saut inconditionnel à l'adresse nn.		
JR C, e	Saut à l'adresse PC + e si l'indicateur C = 1.	LD (nn), A	Chargement de l'emplacement mémoire d'adresse nn avec le contenu de A.
JR e	Saut inconditionnel à l'adresse PC + e.	LD (nn), dd	Chargement des emplacements mémoires nn et nn + 1 avec le contenu de la paire de registres dd.
JR NC, e	Saut à l'adresse PC + e si l'indicateur C = 0.	LD (nn), HL	Chargement des emplacements mémoires nn et nn + 1 avec le contenu de HL.
JR NZ, e	Saut à l'adresse PC + e si l'indicateur Z = 0	LD (nn), IX	Chargement des emplacements mémoires nn et nn + 1 avec le contenu de IX.
JR Z, e	Saut à l'adresse PC + e si l'indicateur Z = 1.	LD (nn), IY	Chargement des emplacements mémoires nn et nn + 1 avec le contenu de IY.
LD A, (BC)	Chargement de l'accumulateur avec le contenu de l'emplacement mémoire dont l'adresse est le contenu de BC.	LD R, A	Chargement du registre R avec le contenu de A.
LD A, (DE)	Chargement de l'accumulateur avec le contenu de l'emplacement mémoire dont l'adresse est le contenu de DE.	LD r, (HL)	Chargement du registre r avec le contenu de l'emplacement mémoire dont l'adresse est le contenu de HL.
LD A, I	Chargement de l'accumulateur avec le contenu du registre I.	LD r, (IX + d)	Chargement du registre r avec le contenu de l'emplacement mémoire dont l'adresse est le contenu de IX + d.
LD A, (nn)	Chargement de l'accumulateur avec le contenu de l'adresse nn.	LD r, (IY + d)	Chargement du registre r, avec le contenu de l'emplacement mémoire dont l'adresse est le contenu de IY + d.
LD A, R	Chargement de l'accumulateur avec le contenu du registre R.	LD r, n	Chargement du registre r avec la donnée n.
LD (BC), A	Chargement de l'emplacement mémoire dont l'adresse est le contenu de BC avec celui de A.	LD r, r'	Chargement du registre r avec le contenu de r'.
LD (DE), A	Chargement de l'emplacement mémoire dont l'adresse est le contenu de DE avec celui de A.	LD SP, HL	Chargement du registre SP avec le contenu de HL.
LD (HL), n	Chargement de l'emplacement mémoire dont l'adresse est le contenu de HL avec la donnée n.	LD SP, IX	Chargement du registre SP avec le contenu de IX.
LD dd, nn	Chargement de la paire de registres dd avec la donnée nn.	LD SP, IY	Chargement du registre SP avec le contenu de IY.
LD dd, (nn)	Chargement de la paire de registres dd avec la donnée contenue aux adresses nn et nn + 1.	LDD	Chargement de l'emplacement d'adresse DE avec le contenu de
LD HL, (nn)	Chargement de HL avec la donnée contenue aux adresses nn et nn + 1.		
LD (HL), r	Chargement de l'emplacement mémoire		



Code mnémorique	Opération	Code mnémorique	Opération
LDDR	l'emplacement d'adresse HL. Décrémenter de DE, HL et BC. Chargement de l'emplacement d'adresse DE avec le contenu de l'emplacement d'adresse HL. Décrémenter de DE, HL et BC. L'opération est répétée jusqu'à ce que BC = 0.	OTDR	Chargement du port dont l'adresse est le contenu de C avec le contenu de l'emplacement d'adresse HL. Décrémenter de HL et de BC. L'opération est répétée jusqu'à ce que BC = 0.
LDI	Chargement de l'emplacement d'adresse DE avec le contenu de l'emplacement d'adresse HL. Incrémenter de DE et HL. Décrémenter de BC.	OTIR	Chargement du port dont l'adresse est le contenu de C avec le contenu de l'emplacement d'adresse HL. Incrémenter de HL et décrémenter de BC. L'opération est répétée jusqu'à ce que BC = 0.
LDIR	Chargement de l'emplacement d'adresse DE avec le contenu de l'emplacement d'adresse HL. Incrémenter de DE et HL. Décrémenter de BC. L'opération est répétée jusqu'à ce que BC = 0.	OUT (C), r	Chargement du port dont l'adresse est le contenu de C avec le contenu du registre r.
POP IX	Chargement de IX avec le sommet de la pile.	OUT (n), A	Chargement du port d'adresse n avec le contenu de A.
POP IY	Chargement de IY avec le sommet de la pile.	OUT D	Chargement du port dont l'adresse est le contenu de C avec le contenu de l'emplacement d'adresse HL. Décrémenter de HL et BC.
POP qq	Chargement de la paire de registres qq avec le sommet de la pile.	OUT I	Chargement du port dont l'adresse est le contenu de C avec le contenu de l'emplacement d'adresse HL. Incrémenter de HL et décrémenter de BC.
PUSH IX	Chargement du sommet de la pile avec IX.	RLC r	Rotation à gauche du contenu du registre r. Recopie du bit de poids fort dans l'indicateur C.
PUSH IY	Chargement du sommet de la pile avec IY.	RLCA	Rotation à gauche du contenu de l'accumulateur. Recopie du bit de poids fort dans l'indicateur C.
PUSH qq	Chargement du sommet de la pile avec le contenu de la paire de registres qq.	RLD	Rotation décimale à gauche.
RES b, m	Remise à zéro du bit b de l'opérande m.	RR m	Rotation à droite de l'opérande m au travers de l'indicateur C.
RET	Retour d'un sous-programme.	RRA	Rotation à droite du contenu de l'accumulateur au travers de l'indicateur C.
RET cc	Retour d'un sous-programme si la condition cc est vraie.	RRC m	Rotation à droite de l'opérande m. Recopie du bit de poids faible dans l'indicateur C.
RETI	Retour d'un sous-programme d'interruption.	RRCA	Rotation à droite du contenu de l'accumulateur. Recopie du bit de poids faible dans l'indicateur C.
RETN	Retour d'un sous-programme d'interruption non masquable.	RRD	Rotation décimale à droite.
RL m	Rotation à gauche de l'opérande m au travers de l'indicateur C.	RST p	Saut à l'adresse p.
RLA	Rotation à gauche de l'accumulateur au travers de l'indicateur C.	SBC A, s	Soustraire de l'accumulateur l'opérande s ainsi que le report. Résultat dans A.
RLC (HL)	Rotation à gauche du contenu de l'emplacement mémoire d'adresse HL. Recopie du bit de poids fort dans l'indicateur C.	SBC HL, ss	Soustraire de HL le registre double ss avec le report. Résultat dans HL.
RLC (IX + d)	Rotation à gauche du contenu de l'emplacement mémoire d'adresse (IX + d). Recopie du bit de poids fort dans l'indicateur C.	SCF	Mise à 1 de l'indicateur C.
RLC (IY + d)	Rotation à gauche du contenu de l'emplacement mémoire d'adresse (IY + d). Recopie du bit de poids fort dans l'indicateur C.	SET b, (HL)	Mise à 1 du bit b de l'emplacement mémoire dont l'adresse est le contenu du HL.
NEG	Complément à 2 du contenu de l'accumulateur. Résultat dans A.	SET b, (IX + d)	Mise à 1 du bit b de l'emplacement mémoire dont l'adresse est le contenu de IX + d.
NOP	Pas d'opération.	SET b, (IY + d)	Mise à 1 du bit b de l'emplacement mémoire dont l'adresse est le contenu de IY + d.
OR s	Opération «OU logique» entre l'opérande s et le contenu de A. Résultat dans A.		



Code mnémorique	Opération	Code mnémorique	Opération
SET b, r	Mise à 1 du bit b du registre r.		
SLA m	Décalage arithmétique à gauche de l'opérande m. Le bit de poids fort est dans l'indicateur C.	SUB s	m. Le bit de poids faible est dans l'indicateur C. Soustraire l'opérande s de l'accumulateur. Résultat dans l'accumulateur.
SRA m	Décalage arithmétique à droite de l'opérande m. Le bit de poids faible est dans l'indicateur C.	XOR s	OU exclusif logique entre l'accumulateur et l'opérande s. Résultat dans l'accumulateur.
SRL m	Décalage logique à droite de l'opérande		

## EXERCICES

### Nota :

1. Dans les exercices qui suivent, vous pouvez être amenés à utiliser l'instruction DJNZ et donc de devoir effectuer le calcul d'une adresse relative. Votre travail peut être grandement facilité en utilisant la commande RELA décrite dans le Manuel Technique du MPF-I, pages 31 et 32.  
2. Terminez vos programmes avec une instruction FIN (F7).

### Exercice 1

Etablir le code binaire et en déduire le code hexadécimal pour les opérations de transferts suivantes. Vérifiez avec le tableau de la figure 106.

- LD A,C
- LD A,H
- LD D,A
- LD B,E
- LD L,B

### Exercice 2

Expliciter les codes hexadécimaux suivants :

- 3A 0F 18
- 32 0F 18
- 3É FE
- ED 5F
- 2E 4F

### Exercice 3

Quelle différence existe-t-il (s'il en existe une) entre les deux instructions suivantes :

LD A, BC  
LD A, (BC)

Indiquez les codes hexadécimaux correspondants.

### Exercice 4

Même question entre les instructions suivantes :

LD DE, A  
LD (DE), A

### Exercice 5

Déterminez le code hexadécimal des instructions suivantes :

- LD C, (HL)
- LD A, BE
- LD A, (BE)
- LD A, 1900
- LD I, (HL)
- LD A, (1900)
- LD HL, L
- LD C, (DE)
- LD BC, A
- LD R, A

Remarques.

### Exercice 6

Ecrire un programme à l'aide des instructions de transfert sur 8 bits qui permet de charger les paires de registres HL et DE avec les quantités respectives 03C1 et 1900. Reprendre l'ensemble de l'exercice II.8.3.

### Exercice 7

A l'aide des instructions de transfert sur 8 bits, établissez le programme qui permet d'inverser le contenu de deux cases mémoires consécutives. Exemple :

AVANT Exécution

1 9 5 0 1 1  
1 9 5 1 2 2

APRES Exécution

1 9 5 0 2 2  
1 9 5 1 1 1

### Exercice 8

Même question dans le cas de six cases consécutives.

AVANT Exécution

1 9 5 0 0 5  
1 9 5 1 0 4  
1 9 5 2 0 3  
1 9 5 3 0 2  
1 9 5 4 0 1  
1 9 5 5 0 0

APRES Exécution

1 9 5 0 0 0  
1 9 5 1 0 1  
1 9 5 2 0 2  
1 9 5 3 0 3  
1 9 5 4 0 4  
1 9 5 5 0 5

### Exercice 9

a) Réaliser l'opération

LD L, (HL + 1)

b) Ecrire le programme qui réalise la séquence suivante :

LD L, (HL)  
LD H, (HL + 1)

par exemple HL contient 1A00 et les cases mémoires 1A00 et 1A01 contiennent respectivement FF et 20.





# UN EDITEUR DE TABLEAU

## 1. BUT DE L'UTILITAIRE

On veut afficher un tableau constitué d'un certain nombre de colonnes et de lignes, pouvoir remplir dans un ordre quelconque les différentes zones ainsi définies et éventuellement corriger les erreurs de frappe. Si le tableau à remplir contient plus de lignes que l'écran, on veut pouvoir le dérouler — comme un papyrus égyptien — («scrolling» pour parler informaticien), en laissant sur l'écran les titres des colonnes.

## 2. CAHIER DES CHARGES

### Vu par l'opérateur

L'opérateur verra apparaître le tableau avec le nom de chacune de ses colonnes et le numéro de ses lignes. Il remplira chacune des zones en frappant les caractères au clavier. Il corrigera ses erreurs en déplaçant le curseur sur tout le tableau grâce aux quatre flèches du clavier. Il indiquera qu'il a terminé le remplissage du tableau en appuyant sur la touche [ENTER].

### Vu par le programmeur

Le programmeur définira dans son programme principal :

- + le nom de chaque colonne (ou le texte qu'il veut faire apparaître en haut de l'écran) ;
- + la position de chaque colonne ;
- + le nombre de colonnes et le nombre de lignes ;
- + éventuellement, il peut indiquer la largeur de chacune d'elles (par défaut, on prendra la largeur maximale).

L'éditeur de tableau sera considéré comme un sous-programme (accès par un GOSUB).

## 3. INSERTION DE L'UTILITAIRE DANS UN PROGRAMME

Dès les premières lignes de programme, il faut initialiser trois variables de manière à réserver la place nécessaire en mémoire :

- $TB\$(I,J)$  où  $I$  est le nombre de lignes du tableau le plus long alors que  $J$  est le nombre de colonnes du tableau le plus large.
- $P(J)$  et  $L(J)$  sont initialisés comme  $TB\%$  par le nombre maximum de colonnes.

C'est ce qui est fait à la ligne 20, cette partie est commune à tous les tableaux et ne doit être faite qu'une seule fois. Par exemple, si le programmeur fait appel à trois tableaux comportant respectivement 50 lignes et 7 colonnes, 90 lignes et 4 colonnes et 60 lignes et 5 colonnes, il devra initialiser  $TB\%$ ,  $L$  et  $P$  par 90 suivant l'indice  $I$  et par 7 suivant  $J$  car on a au maximum 90 lignes ou 7 colonnes.

Quelques lignes avant le Gosub, le programmeur doit préciser tous les paramètres concernant le tableau qu'il veut faire apparaître c'est-à-dire le nombre de lignes dans  $NL$ , le nombre de colonnes dans  $NC$ , la position du premier caractère de chaque colonne dans  $P(J)$  (lignes 35 et 45), ainsi  $P(1)$  contiendra la position du premier caractère de la première colonne,  $P(2)$  celle du premier caractère de la deuxième, etc.

On doit toujours avoir  $P(1)$  supérieur ou égal à 4, de manière à conserver les trois premiers caractères de chaque ligne pour indiquer le numéro de celle-ci.

Eventuellement, le programmeur peut indiquer le nombre de caractères de chaque colonne dans  $L(J)$ . Par

défaut, le sous-programme calculera la longueur maximum de chaque colonne et prendra cette valeur.

Puis, il doit afficher sur la première ligne de l'écran, le nom de chaque colonne (ligne 55). La ligne suivante du programme peut alors être le «Gosub» vers l'utilitaire éditeur de tableau.

Il existe trois points d'entrée dans ce sous-programme qui correspondent à des niveaux d'initialisation différents.

— Si la largeur de chaque colonne n'a pas été spécifiée : c'est le cas le plus général, on doit alors avoir un Gosub 9941.

— Si la largeur de chaque colonne a été spécifiée dans  $L(J)$ , mais que les chaînes de caractères du tableau  $TB\$(I,J)$  n'ont pas toutes la longueur correspondante aux colonnes. On doit alors avoir un Gosub 9943.

— Si  $L(J)$  a été spécifié et que les chaînes  $TB\$(I,J)$  ont la bonne longueur, on doit avoir Gosub 9951, cette dernière méthode est la plus rapide mais n'est pas toujours utilisable.

## 4. LES VARIABLES INTERNES ET EXTERNES

Les variables internes n'apparaissent que dans le sous-programme et les variables externes permettent de faire passer des informations du programme principal vers l'utilitaire et vice-versa.

### • Les variables internes

Elles sont initialisées par le sous-programme (lignes 9952 et 9953) à chaque fois que l'on fait appel à l'édi-



teur de tableau. Elles n'apparaissent pas dans le programme principal, et permettent de connaître la position du curseur sur l'écran ainsi que la position du tableau par rapport à l'écran.

YR : indique la position relative du tableau par rapport à l'écran. Sa valeur est le numéro de la première ligne affichée moins un. Elle varie de 0 (première ligne affichée) à NL-S (lorsque la dernière ligne du tableau est affichée)

XT : on y trouve le numéro de la colonne dans laquelle se trouve le curseur. XT varie de 1 lorsque le curseur est dans la 1ère colonne à NC (nombre de colonne) lorsqu'il est dans la dernière.

YT : indique le numéro de la ligne dans laquelle se trouve le curseur. Elle varie de 1 lorsque le curseur écran est sur la 1ère ligne du tableau, à NL lorsqu'il est sur la dernière.

C : contient la position du curseur dans une colonne. Il varie de 1 lorsque le curseur est sur le 1er caractère d'une colonne, à L (J) lorsqu'il est sur le dernier caractère.

#### • Les variables externes

Contrairement aux précédentes, elles sont initialisées par le programmeur dans le programme principal et servent à faire passer des informations du programme principal à l'utilisateur et vice versa.

NC et NL : qui contiennent le nombre de colonnes et le nombre de lignes du tableau affiché.

TB\$ (I, J) dans lequel on trouve ce qui a été inscrit dans le tableau à la Ième ligne et à la Jème colonne.

P (J) : on doit y avoir mis la position du premier caractère de la Jème colonne.

L (J) : on y trouve le nombre de caractères que contient la Jème colonne (largeur de la colonne).

## 5. L'ORGANIGRAMME DE L'UTILITAIRE

— l'initialisation

On y trouve les 3 niveaux d'accès aux lignes 9941, 9943 et 9951.

— Impression du tableau

Celle-ci n'a lieu qu'au début et lorsqu'on déplace le tableau par rapport à l'écran c'est-à-dire à la suite d'une action sur ↓ ou ↑.

— Saisie d'un caractère

Rien de particulier.

— Interprétation d'un caractère

Dans cette partie, on commence par examiner la nature du dernier caractère frappé au clavier. Si celui-ci n'est pas un caractère de contrôle ([CF] ; ↑ ; ↓ ; → ; ←), il est introduit dans la chaîne TB\$ (ligne 9974 à 9976).

— Les déplacements du curseur

• Droite (→) et gauche (←) : rien de particulier

• haut (↑) et bas (↓) : pour ne pas avoir à réimprimer le tableau à chaque fois qu'on change de ligne, on déplace ce dernier lorsque le curseur écran se trouve suffisamment excentré.

## 6. ETUDE DE QUELQUES LIGNES PARTICULIÈRES

Les lignes 9974 à 9976 permettent de remplacer le Cème caractère de TB\$ par le dernier caractère frappé au clavier. Pour ceci, on utilise 3 types d'instructions de découpage de chaîne, MID\$, LEFT\$, RIGHT\$.

### MID\$

Elle permet d'extraire le milieu d'une chaîne de caractères. La syntaxe est la suivante :

MID\$ (chaîne de caractères, expression numérique 1, expression numérique 2). La première expression numérique indique la position du premier caractère et la seconde, la longueur de la chaîne que l'on veut.

Exemple :

MID\$ («ABCDEFGH», 5, 2) = «EF»

de la même manière si

A\$ = «ABCDEFGH»

B = 5

C = 3

MID\$ (A\$, B, C - 1) = «EF»

### LEFT\$

Elle permet d'extraire la partie gauche d'une chaîne. La syntaxe est la suivante :

LEFT\$ (chaîne de caractères, expression numérique) où l'expression numérique indique le nombre de caractères que l'on prend à partir de la gauche (Left = gauche en anglais).

Exemple :

LEFT\$ («ABCDEFGH», 4) = «ABCD»

### RIGHT\$

Elle permet d'extraire la partie droite d'une chaîne. La syntaxe est la suivante :

RIGHT\$ (chaîne de caractères,

expression numérique) où l'expression numérique indique le nombre de caractères que l'on prend à partir de la droite (Right = droite en anglais). Exemple :

RIGHT («ABCDEFGH», 3) = «FGH»

A l'aide de ces instructions et de la concaténation (+) on remplace 1 seul caractère de TB\$ (I, J) par celui contenu dans S\$.

## 7. ADAPTATION DE L'UTILITAIRE A D'AUTRE TAILLE D'ECRAN

Nous allons voir quelles sont les lignes à modifier pour une autre taille d'écran.

Sur le listing, on trouvera les valeurs pour un écran de 24 lignes de 40 caractères.

### • ligne 9941

Dans la variable, S doit se trouver le nombre de caractères par ligne plus un. Ainsi avec un écran de 80 caractères par ligne on aura : 9941 S = 81

### • ligne 9951

Dans S, nous devons mettre le nombre de lignes de l'écran moins un. Ainsi si l'écran fait 40 lignes, on aura :  
9951 S = 39 : IF NL 39 then S = NL

### • ligne 9992

Dans la 2ème partie de cette ligne, on calcule la position du curseur sur l'écran (YT - YR) et on regarde si le curseur est dans le quart supérieur et si la première ligne du tableau est affichée (YR = 0)

(YT - YR nombre de lignes/4)

Ainsi, avec un écran de 40 lignes.

9992 : YT = YT - 1 : IF YT - YR 10 OR YR = 0 then 9964

### • ligne 9993

Dans la première partie de cette ligne, on fait redescendre le tableau d'à peu près un quart de la hauteur de l'écran : avec 40 lignes on aura  
9993 : YR = YR - 10 IF YR 0 then YR = 0

### • ligne 9997

On calcule la position du curseur par rapport à l'écran et si celui-ci est dans le quart inférieur YT - YR 17 ou que la dernière ligne est affichée (YR : NL - S) on ne déplace pas le tableau par rapport à l'écran

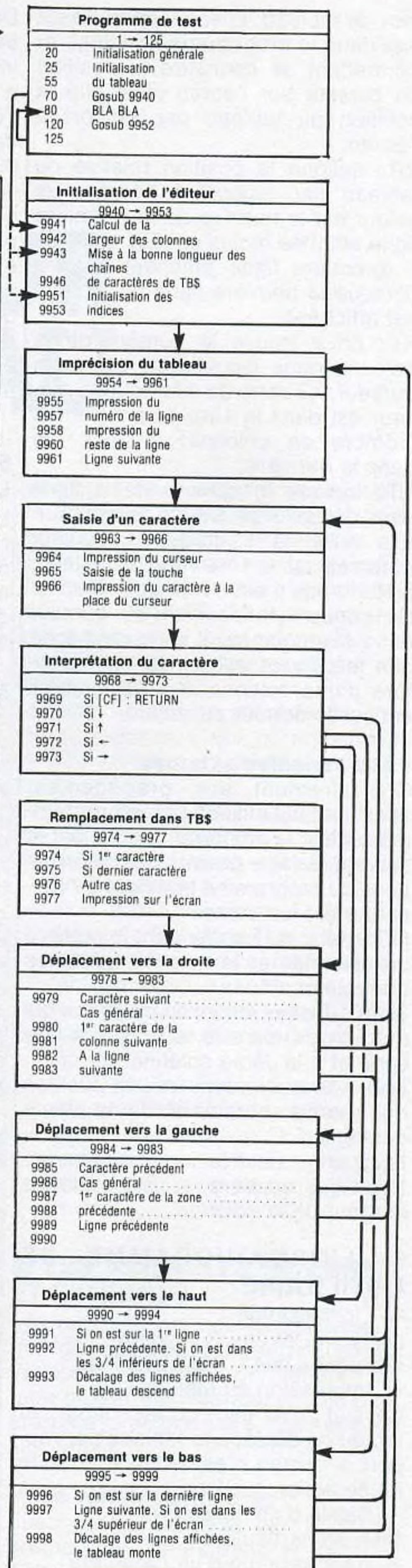


## Listing du programme

```

1 REM EDITEUR DE TABLEAU
2 REM ECRIT PAR BRUNO LILAMAND
3 REM ENREGISTRE SUR EDITEUR DE
  TABLEAU
4 REM COPYRIGHT LED MICRO/B.LIL
  AMAND
5 REM *****
10 REM PETIT PROGRAMME POUR TES
  TER L'UTILITAIRE
15 REM DIMENSIONNEMENT DE TB#,P
  ,L
20 DIM TB$(50,4): DIM P(4): DIM
  L(4)
25 HOME
30 REM INITIALISATION DES VARIA
  BLES EXTERNES
35 NL = 40:NC = 4
45 P(1) = 4:P(2) = 12:P(3) = 25:P
  (4) = 34
50 REM NOM DES COLONNES
55 PRINT "  PRODUIT DISTRIBUTEU
  R QUANTITE PRIX"
70 GOSUB 9940
80 HOME
85 PRINT "VOULEZ VOUS FAIRE DES
  MODIFICATIONS(O/N)?";
90 GET S#
100 IF S# = "N" THEN STOP
110 PRINT "  PRODUIT DISTRIBUTEU
  R QUANTITE"
120 GOSUB 9952
125 GOTO 80
9940 REM *****INITIALISATION****
9941 S = 41
9942 FOR J = NC TO 1 STEP - 1:L
  (J) = S - P(J) - 1:S = P(J):
  NEXT J
9943 FOR I = 1 TO NL
9944 FOR J = 1 TO NC
9945 TB$(I,J) = LEFT$(TB$(I,J) +
  ".....",L(J))
9946 NEXT J: NEXT I
9951 S = 23: IF NL < 23 THEN S =
  NL
9952 YR = 0:YT = 1
9953 XT = 1:C = 1
9954 REM IMPRETIION DU TABLEAU
9955 FOR I = 1 TO S
9956 HTAB (1): VTAB (I + 1): IF
  I + YR < 10 THEN PRINT " ";
9957 PRINT I + YR:
9958 L = I + YR
9959 FOR J = 1 TO NC
9960 HTAB (P(J) - 1): PRINT "!";
  TB$(L,J);
9961 NEXT J: NEXT I
9963 REM SAISIE D'UN CARACTERE
9964 HTAB (P(XT) + C - 1): VTAB
  (YT - YR + 1)
9965 GET S#
9966 PRINT MID$(TB$(YT,XT),C,1
  );
9968 REM INTERPRETATION DE LA T
  OUCHE ACTIONNEE
9969 IF ASC (S#) = 13 THEN RETURN
9970 IF ASC (S#) = 21 THEN 9978
9971 IF ASC (S#) = 8 THEN 9985
9972 IF ASC (S#) = 11 THEN 9991
9973 IF ASC (S#) = 10 THEN 9996
9974 IF C = 1 THEN TB$(YT,XT) =
  S# + RIGHT$(TB$(YT,XT),L(X
  T) - 1): GOTO 9977
9975 IF C = L(XT) THEN TB$(YT,XT
  ) = LEFT$(TB$(YT,XT),C - 1
  ) + S#: GOTO 9977
9976 TB$(YT,XT) = LEFT$(TB$(YT,
  XT),C - 1) + S# + RIGHT$(T
  B$(YT,XT),L(XT) - C)
9977 HTAB (P(XT) + C - 1): PRINT
  S#;
9978 REM A DROITE *****
9979 C = C + 1: IF C < > L(XT) +
  1 THEN 9964
9980 C = 1:XT = XT + 1
9981 IF XT < > NC + 1 THEN 9964
9982 XT = 1
9983 GOTO 9996
9984 REM A GAUCHE *****
9985 C = C - 1
9986 IF C < > 0 THEN 9964
9987 C = 1:XT = XT - 1
9988 IF XT < > 0 THEN 9964
9989 XT = 1
9990 REM VERS LE HAUT *****
9991 IF YT = 1 THEN 9964
9992 YT = YT - 1: IF YT - YR > 6 OR
  YR = 0 THEN 9964
9993 YR = YR - 5: IF YR < 0 THEN
  YR = 0
9994 GOTO 9954
9995 REM VERS LE BAS *****
9996 IF NL = YT THEN 9964
9997 YT = YT + 1: IF YT - YR < 17
  OR YR = NL - S THEN 9964
9998 YR = YR + 5: IF YR > NL - S THEN
  YR = NL - S
9999 GOTO 9954
  
```

## Organigramme



donc, avec 40 lignes, on aura:  
9997: YT = YT + 1: IF YT - YR  
29 OR YR = NL - S then 9964

### • ligne 9998

Avec cette ligne, on fait remonter le tableau d'à peu près un quart de la hauteur de l'écran (YR = YR + (24/4 - 1) et on vérifie que la dernière ligne du tableau n'est pas au-dessus de la dernière ligne de l'écran: YR > NL - S).

Avec 40 lignes on aura:  
9998: YR = YR + 9: IF YR  
NL - S then YR = NL - S

## 8. AUGMENTATION DU NOMBRE DE LIGNES DU TABLEAU

Nous allons voir comment porter le nombre maximal de lignes du tableau de 99 à 999.

Il suffit de prévoir une colonne de 4 caractères et non de 3 pour numéroter les lignes du tableau. Seule la ligne 9956 est à modifier.

9956: HTAB (1): VTAB (I + 1):  
IF I + YR > 10 then PRINT " " ;  
IF I + YR > 10 then PRINT " " ;  
Remarque: P(1) devra toujours être supérieur ou égal à 5. **B. Lilamand**



# ALLEZ PLUS LOIN AVEC VOTRE COMMODORE



## LISEZ COMMODORE MAGAZINE

Oui, explorez toutes les possibilités de votre ordinateur Commodore avec **COMMODORE MAGAZINE**, le nouveau magazine destiné aux utilisateurs du VIC 20, du COMMODORE 64 et des CBM 8000.

Découvrez chez vous, tranquillement et en profondeur, tout l'intérêt d'un magazine consacré exclusivement à votre ordinateur.

Au sommaire de chaque numéro :

- des informations,
- des reportages,
- du vécu,
- des astuces,
- des programmes.

Le numéro 2 vient de paraître, commandez-le aujourd'hui-même. Vous pouvez aussi obtenir le numéro 1, encore disponible, au prix promotionnel de 15 F.

**OUI**  Je désire recevoir le numéro 2 de **COMMODORE MAGAZINE** au prix de 30 F.  
 Envoyez-moi aussi le numéro 1 à moitié prix soit 15 F.  
Ci-joint un chèque de F. \_\_\_\_\_ à l'ordre de **COMMODORE MAGAZINE**,  
19, rue des Parisiens - 92600 Asnières

Nom \_\_\_\_\_ Prénom \_\_\_\_\_

Adresse \_\_\_\_\_

Tél. \_\_\_\_\_



# LIBRE PROPOS

**L**a formation en informatique est-elle bonne ? Est-elle bon marché ? L'informatique, phénomène miracle de la fin du 20<sup>e</sup> siècle, a effectué une ascension vertigineuse ces dernières années. Il existe aujourd'hui toutes sortes de formations en la matière, cela va de l'Ecole Polytechnique au stage d'initiation en deux jours. Toutes les astuces sont bonnes pour attirer le client (ex. : don d'un micro-ordinateur). En fait, tout cela dépend de ce que l'on recherche, ainsi que du niveau requis. Les prix sont, en général, assez variables ; toutefois dès qu'on s'adresse aux professionnels : organismes de renom, grands constructeurs, etc., les prix deviennent comparables. Le budget nécessaire s'échelonne de 1 000 F la journée (6 heures d'initiation), à 1 500 F (perfectionnement). Mais dans certains domaines très spécialisés, il faudra prévoir 3 000 F par jour pour les hyper-spécialistes, pour certains mini-stages de 2 à 5 jours.

Il ne faut pas être dupe, une bonne formation ne peut pas être réalisée en peu de temps, de nombreux cours sont nécessaires, ainsi que beaucoup de pratique (analyse, programmation, etc.). Si vous êtes salarié, en utilisant le 1 % formation de votre entreprise, il faudra compter l'équivalent d'un mois de stage réparti en quatre semaines sur une année, soit un budget de 20 à 30 000 francs. Grâce à cela, vous pourrez avoir des bases pour utiliser correctement un micro. Si vous voulez être un véritable «pro», ce sera de un à trois ans à plein temps.

Mais ne nous trompons point, il n'est pas nécessaire d'avoir un sérieux bagage pour se servir d'un clavier. La majorité des micro-ordinateurs vendus aux sociétés aujourd'hui s'accompagne de logiciels. Quelques heures suffisent à l'utilisateur potentiel pour comprendre le mode d'emploi. Lorsque vous passez votre permis de conduire, vous n'avez pas à être mécanicien, encore moins ingénieur. Alors, avant d'aborder toute formation, analysez bien vos besoins. Est-il nécessaire de suivre telle filière ? Ce stage est-il suffisant ? Dans 99 % des cas, quel que soit le niveau des intéressés, la même remarque revient toujours : pas assez de travaux pratiques. L'astuce est donc de disposer d'une machine pour s'entraîner en dehors des cours.

Dans tous les cas, essayez de rencontrer une personne ayant déjà effectué la formation que vous désirez suivre. S'il s'agit d'une reconversion, demandez ce que sont devenus les anciens «élèves». Quels postes occupent-ils aujourd'hui ? Quels sont leurs appointements ? Ne vous trompez pas de spécialisation, demandez une documentation détaillée sur le contenu de la formation, vérifiez que la finalité de vos besoins est bien respectée.

Charles Henry Delaleu



# SOMMAIRE

POUR BIEN CHOISIR L'IMPRIMANTE, IL FAUT CONNAITRE LES DIFFERENTS TYPES EXISTANTS SUR LE MARCHE ET QUI CORRESPONDENT CHACUN A DES UTILISATIONS PRECISES ET DETERMINENT LE PRIX **P. 50**. AVEC LE LASER 3000, NOUS OUVRONS VRAIMENT NOTRE RUBRIQUE CONTRE-MESURES **P. 54**. DANS CETTE GRANDE KERMESSE AUX TECHNOLOGIES NOUVELLES QU'EST LE SICOB, VOICI LA MICRO ET LES MICROS **P. 59**. POUR PASSER UN MOMENT AGREABLE, VOUS TROUVEREZ **P. 64** NOTRE SELECTION DES LIVRES DU MOIS.



## imprimante: le choix dépend de l'usage

De tous les périphériques du micro-ordinateur, l'imprimante est celui qui est le plus nécessaire. Si dans le domaine professionnel, cela ne fait aucun doute, pour un usage familial, cela ne semble pas indispensable au départ. Puis très rapidement, le besoin d'avoir une trace écrite se fait sentir. Le choix de l'imprimante, comme celui du micro d'ailleurs, est fonction des besoins.



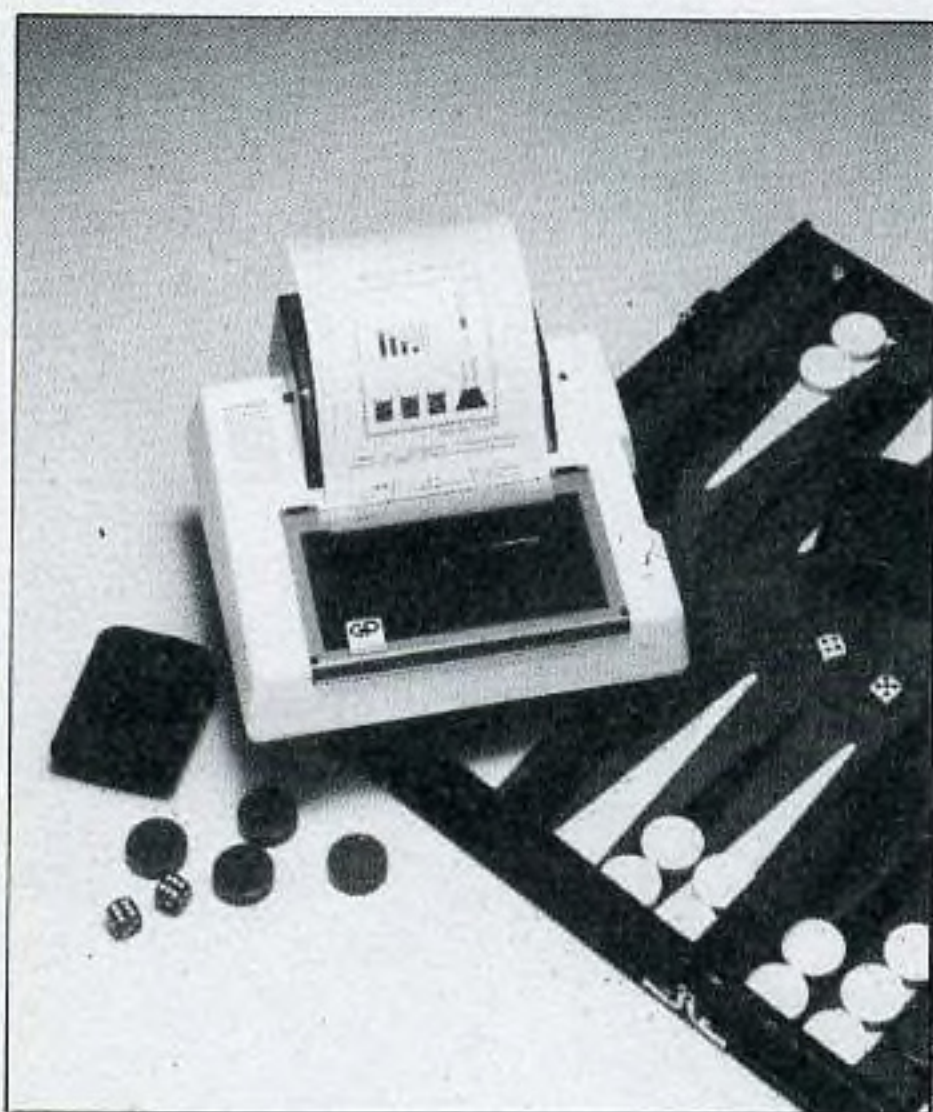
Il existe sur le marché des dizaines et des

dizaines d'imprimantes dont les prix varient d'un millier à plusieurs milliers de francs. Elles sont regroupées en cinq grandes catégories : imprimantes thermiques, imprimantes matricielles, imprimantes à marguerite, imprimantes à jet d'encre et imprimantes à laser. A chaque type correspondent des performances différentes et par conséquent des usages différents et bien évidemment des prix.

### LES IMPRIMANTES THERMIQUES

Ces imprimantes font appel à un procédé d'impression thermique. De petites électrodes brûlent en surface un papier spécial. Le papier est revêtu

d'une fine couche conductrice et présente un aspect argenté, ou bien il est traité chimiquement. Les électrodes qui constituent la pointe de lecture détruisent la pellicule argentée ou modifient la substance chimique et créent des points qui forment la lettre. Ce procédé requiert une technique de fabrication simplifiée, d'où un coût relativement bas de ces imprimantes. Les imprimantes thermiques sont silencieuses, mais le dessin des caractères n'est pas toujours très fin ce qui parfois ne facilite pas la lecture. La vitesse d'impression varie de 15 à 40 caractères par secondes ; il faut environ une à deux minutes pour imprimer une page. L'imprimante thermique de par ses performances modestes, n'est pas destinée à un usage professionnel. En revanche, elle convient bien à un usage



L'imprimante thermique Seikosha GP 50.





*L'imprimante thermique Tandy TP 10.*

familial, ou éventuellement professionnel dans la mesure où elle n'est pas encombrante, et facilement transportable. Les fabricants de micro-ordinateurs familiaux proposent tous un modèle adapté à leur machine. Canon, Epson, Seiksha pour ne citer que quelques uns ont à leur catalogue des imprimantes thermiques.

## **LES IMPRIMANTES MATRICIELLES**

Ce sont de loin les imprimantes les plus répandues sur le marché. Ce type d'imprimante a été spécialement développé pour les micro-ordinateurs. La



*La matricielle multi-vitesse SPG 8010.*

tête d'impression est constituée d'une matrice d'aiguilles qui vient frapper un ruban de type machine à écrire. Chaque caractère est reconstitué point par point. La lisibilité dépend du nombre de points, donc du nombre d'aiguilles. Les textes qui sortent d'une imprimante matricielle conservent un aspect «listing», et n'ont pas la qualité courrier offerte par les machines à écrire électroniques. En revanche, ces imprimantes ont une vitesse d'impression plus élevée que celle des imprimantes thermiques : de 40 à 300 caractères par secondes.

Pour améliorer la qualité de l'impression, certains constructeurs utilisent une technique dite «le double passage». Chaque caractère est frappé deux fois, la densité est donc doublée et la lisibilité améliorée, mais la vitesse d'impression est diminuée. Certains



*Itoch CI 3500 compatible IBM.*

modèles peuvent reproduire fidèlement sur papier un graphisme haute résolution réalisé à l'écran, d'autres au moyen de rubans de couleurs peuvent produire une impression couleur. Le principal inconvénient de ces imprimantes est le bruit, mais il est désormais possible de le réduire en utilisant un kit d'insonorisation proposé par certains constructeurs (Mannesman Tally). L'imprimante matricielle vient de plus en plus concurrencer l'imprimante thermique pour le prix. La Seiksha GP 200 par exemple vaut environ 2 000 F. De par la diversité des performances, les imprimantes matricielles peuvent couvrir la quasi totalité des applications. A preuve, aujourd'hui elles constituent 70 % du marché.

Les prix des imprimantes matricielles se situent entre 5 000 et 55 000 F TTC



*La qualité graphique de l'Epson FX 80.*



# MATERIEL

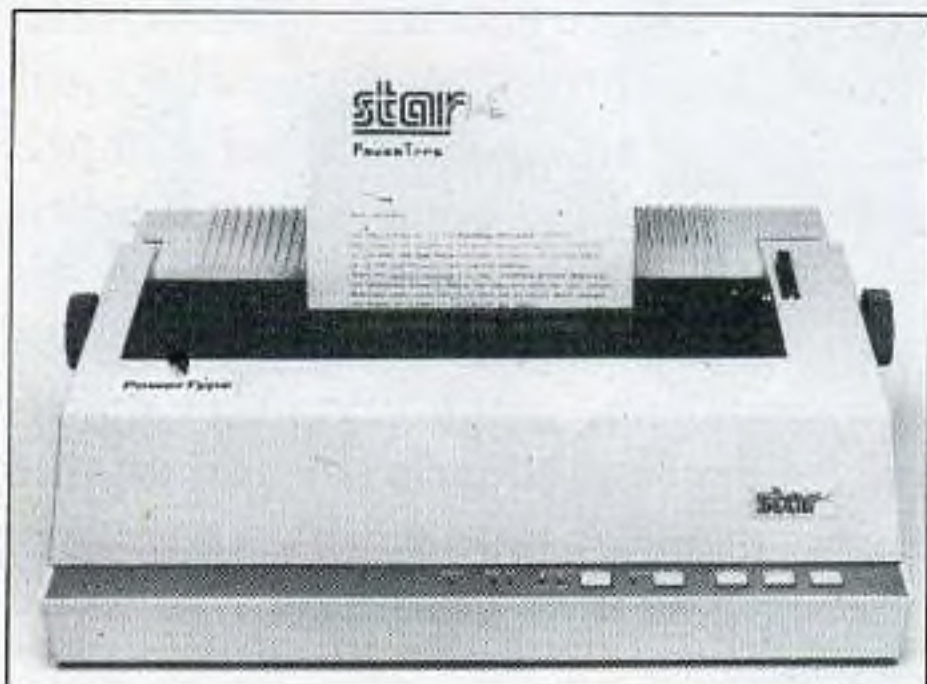


CBS : l'Adam avec son imprimante.

environ. Parmi les moins onéreuses, les Epson RX80 et FX80 et chez Data-products la Microprism et la SPG 8010.

## LES IMPRIMANTES A MARGUERITE

Le principal inconvénient, on l'a vu, des imprimantes matricielles, est qu'elles n'offrent pas une qualité courrier nécessaire pour les lettres, les rapports. L'imprimante à marguerite, elle, apporte cette qualité. Avec ce procédé, le caractère est préformé sur un support constitué par une roue contenant la police de caractères complète. Il suffit de changer de roue, comme l'on change de boule sur machine à



Star, l'imprimante à marguerite.

écrire, pour changer de police de caractères. L'inconvénient majeur de ces imprimantes est leur lenteur d'impression. D'autre part, on ne peut pas changer de police en cours d'impression.

## BIEN CHOISIR, C'EST BIEN CONNAITRE SES BESOINS

*Le choix d'une imprimante dépend de ce que l'on veut faire. Une imprimante à usage familial n'a pas besoin d'être très rapide. Si l'on a besoin de faire du courrier, il faut une imprimante à marguerite qualité courrier ou une jet d'encre qui a l'avantage d'être tous usages. Il est parfois préférable de dépenser un peu plus afin de n'être pas limité par la suite par les possibilités de la machine. Pour un usage professionnel, les normes courantes sont de 40, 80 et 132 caractères par ligne. Si l'on souhaite diversifier la présentation des documents par des caractères différents, il est préférable de choisir des imprimantes matricielles ou à jet d'encre qui permettent la sélection de la police par logiciel.*

*Avant tout achat, il est bon de déterminer ses besoins, texte, courrier, graphique, volume des documents à traiter.*

sion. Les pannes mécaniques sont fréquentes. Etant donné que ces imprimantes incluent de la mécanique de haute précision, le prix est élevé. Il se situe entre 6 000 F et 50 000 F. Ce type d'imprimante est destiné aux utilisations professionnelles pour lesquelles la qualité d'impression est primordiale et le volume de documents à traiter relativement réduit. A signaler que CBS propose avec son micro-ordinateur familial une imprimante à marguerite.

Les prix des imprimantes à marguerite évoluent de 4 500 à 35 000 F TTC, pour les modèles de haut de gamme comme le HP 2601A de Hewlett-Packard (132 colonnes, 96 caractères par marguerite, entraînement par friction). Le prix moyen se situe autour de 12 000 F (Silver Reed EXP 770).



Thinkjet d'Hewlett-Packard.

## TECHNOLOGIES DE POINTE : JET D'ENCRE ET LASER

Les imprimantes à jet d'encre sont apparues récemment sur le marché. Avec ce procédé le point d'impression (pixel) est obtenu par la projection d'un très fin jet d'encre sur le papier, préalablement électrisé au point d'impact pour permettre la fixation de l'encre. La densité obtenue est comparable à celle que l'on a avec les imprimantes à marguerite, avec en plus le graphisme, la couleur et la vitesse. Les prix sont de plus très compétitifs : à partir de 8 000 F. A cette catégorie





**Imprimante à laser.**

appartiennent l'A1210 de Canon et la toute récente «Thinkjet» de Hewlett-Packard qui est très compacte et prend peu de place sur un bureau. L'un des principaux avantages de la technologie à jet d'encre est de pouvoir réaliser aisément le hard copy d'écran graphique, souvent en couleur.

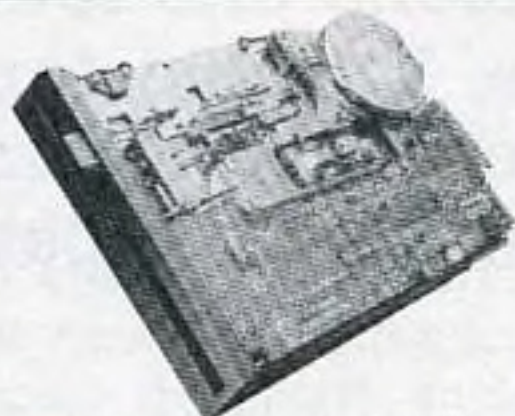
L'imprimante à laser, aux dires de certains, représente la solution à tous les problèmes d'impression. En effet, elle a tous les avantages désirés : vitesse d'impression, qualité de production, et elle est silencieuse. Le rayon laser frappe un tampon recouvert d'une pellicule photosensible sur laquelle la page de texte est reproduite de façon quasi instantanée. L'impression se fait comme un tirage photographique dans le nombre d'exemplaires souhaité. A l'occasion du Sicob, Hewlett-Packard a présenté un modèle qui en raison de son prix (35 000 F HT) met pour la première fois cette technologie à la portée des utilisateurs de micro-informatique, professionnelle cela s'entend.

**C.R.**

**Quelques marques parmi les plus représentatives du marché**

- Epson (Technologie Resources). Tél : 757.31.33.
- Dataproducts. Tél : 910.77.91.
- Centronics (Data Computer). Tél : 828.40.51.
- Seikosha (Tekelec Airtronic). Tél : 534.75.35.
- Oki (Tekelec Airtronic). Tél : 534.75.35.
- Texas Instruments. Tél : 946.97.12.
- Hewlett Packard. Tél : 077.83.83.
- Star (Hengstler). Tél : 866.22.90.
- Diablo Geneke Electronic. Tél : 654.15.82.
- Brother (Brother France). Tél : 859.96.16.
- Facit. Tél : 780.71.17.
- Qume (Métrologie). Tél : 790.62.40.
- Silver Reed (Ern). Tél : 956.00.11.
- Olympia. Tél : 630.21.42.
- Triumph-Adler. Tél : 732.92.45.
- Olivetti. Tél : 266.91.44.

# electro-puce



**Lecteurs de disquettes 3.5"**

- YE DATA** prix T.T.C.
- YD 620:67,5 TPI  
Slim Line DF/DD  
500 Ko 2350,00
  - YD 640:135 TPI  
Slim Line DF/DD  
1 Mo 2850,00

**Lecteurs de disquettes 5.25"**

- BASF (CANON)** prix T.T.C.
- 6128:48 TPI  
Slim Line DF/DD  
500 Ko 2150,00
  - 6138:96 TPI  
Slim Line DF/DD  
1 Mo 2550,00

Les lecteurs de disquettes 3.5" sont compatibles 5.25"

- Supports, Connecteurs : 3M, TB & OEC, AUGAT, EMC...
- Coffrets et Cartes Format Europe : EUROBOX.
- Systèmes d'essai : OK, 3M-PROTOKIT...
- Transferts : MECANORMA Electronic.

## OFFRE SPÉCIALE IMPRIMANTE GEMINI 10X



**3500 F T.T.C.**

- 120 CPS bidirectionnel optimisé
- graphique quadruple densité
- caractères redéfinissables
- friction et traction

**VENTE PAR CORRESPONDANCE**

(frais d'envoi : - 1000 FTTC : 20 F / + 1000 FTTC : 2 % du Total TTC)



**Claviers Capacitifs**

- ALPHAMERIC** prix T.T.C.
- 63 touches 963,00
  - 83 touches 1323,50  
(pavé numérique)
  - 117 touches 1838,50  
(touches fonctions)



- Moniteur vert  
HITACHI 1500,00
- Moniteurs couleurs  
MICROVITEC

4, rue de Trétaigne 75018 PARIS Métro Jules Joffrin Tél : (1) 254.24.00

(Heures d'ouverture : 9 h 30-12 h - 14 h-18 h 30 du Lundi au Samedi)

Egalement disponible chez : CPPM 11, rue Alexandre Dumas 75011 PARIS Métro Mouton Duvernet Tél : (1)371.51.54

(Heures d'ouverture : 9 h - 18 h du Lundi au Vendredi)



# le laser 3000



Le mois dernier, nous vous annonçons une toute nouvelle rubrique «contre-mesures». En voici donc le premier épisode, consacré au Laser 3000, remarquable par ses multiples applications et sa compatibilité avec Apple II.

**L**

e Laser 3000 est un appareil qui se veut complet dans le format 8 bits. En effet, architecturé autour d'un 6502 A, il est possible de lui adjoindre une carte Z80, qui permet alors l'utilisation de nombreux logiciels. Un émulateur permet en option de travailler sous CP/M-86 (format compatible 8088). D'après le constructeur, le «plus» du Laser 3000 concerne sa compatibilité avec l'Apple II, pour un prix plus attractif, car en version de base, il possède l'option 80 colonnes, la carte sonore, le graphisme couleur et les touches de fonction. Il peut travailler en BASIC, PASCAL, FORTRAN et LOGO.

L'appareil est accompagné d'un manuel Basic, d'un manuel d'exploitation, d'un câble Péritel, de deux cor-

dons pour raccorder un magnétophone à cassette (deux options suivant la norme Jack ou Cinch) et d'une unité centrale.

Des possibilités d'adressage en mémoire vive s'étendent de 64 K à 92 Koctets.

L'appareil est d'une esthétique correcte, et sa finition est bonne. Le clavier mécanique, par contre, est nettement mieux réalisé que pour ses concurrents directs. Grande taille, pavé numérique séparé, touches de fonction bien localisées. Son utilisation est agréable et la frappe devient rapidement aisée. Si, actuellement, ce clavier est configuré en QWERTY, une version AZERTY est imminente et fera le plaisir des utilisateurs de logiciels de traitement de texte.



# 5" 1/4

Le Laser 3000 et son lecteur de disquette 5" 1/4.



## L'ENVIRONNEMENT LASER

Le Laser 3000 se raccorde à un téléviseur externe par l'intermédiaire d'une prise Péritel. Notons que, pour les possesseurs de postes anciens, il existe désormais des adaptateurs spécialisés chez certains revendeurs de micro.

Comme interfaçage, il est possible d'avoir accès au bus système et à une interface RS 232. Il est regrettable que cette dernière ne soit pas équipée d'un connecteur standardisé permettant de bons contacts et une utilisation aisée.

Une prise DIN cinq broches permet de brancher un lecteur de cassettes. De même, un modem téléphonique existe en option mais nous ne savons pas s'il est importé en France. Les différentes extensions du Laser 3000 (contrôleur de disquette, extension mémoire) se connectent à l'extérieur du boîtier principal. De ce fait, dans une configuration maximum, on se retrouve avec quelque chose d'inesthétique, mais surtout on perd le côté pratique de la

compacité de la machine en version de base s'assimilant à un clavier séparé. En effet, le Laser 3000 possède des excroissances sur le côté droit, et sur la face arrière, il n'est alors pas souhaitable de le bouger sous peine, à terme, d'avoir du jeu dans les logements incriminés.

## MEMOIRE DE MASSE

Deux types de mémoire de masse peuvent être connectés au boîtier soit, comme nous venons de le voir un lecteur de cassettes, soit un lecteur de disquette 5" 1/4. Il existe un lecteur simple disquette, ou double disquette. Chacun pouvant contenir 164 Koctets.

## VIDEO

La partie vidéo du Laser 3000 est équipée d'une définition six couleurs (texte et graphique). Un light-pen (crayon lumineux) peut être branché. Le choix d'utilisation permet, soit le format Péritel, soit le format RGB classique. La définition graphique est plutôt bonne. Elle possède une précision de  $560 \times 192$  pixels. Notons que pour ce genre d'appareil, on gagne en netteté avec de petits postes téléviseurs.

## LA DOCUMENTATION

Video Technologie (l'importateur) fournit une documentation copieuse sur le Laser 3000 :

- 1 USER'S MANUAL (manuel de l'utilisateur)
- 1 BASIC REFERENCE (manuel de Basic)
- 1 DOS MANUAL (manuel du DOS)

- 1 Technical reference MANUAL (manuel des références techniques).

### USER'S MANUAL (ou manuel d'exploitation)

Ce document permet de se familiariser avec les différentes options langage de la machine ainsi que l'éditeur, les clés de fonctions, le raccordement des périphériques, etc. On regrettera la page 39 tirée à l'envers.

### DOS MANUAL (manuel d'utilisation des disquettes)

Il concerne le système d'exploitation et l'utilisation des lecteurs de disquettes. Il est très aéré et facile à lire.

### BASIC REFERENCE MANUAL (manuel de référence en Basic)

Après une présentation du BASIC, ce manuel reprend chaque ordre et le développe clairement, appuyé par des exemples de petits programmes qui facilitent la compréhension. Très clair, bien réalisé, il permettra au débutant d'apprendre le BASIC très rapidement.

### TECHNICAL REFERENCE MANUAL (manuel des références techniques)

Ce manuel, qui n'est pas livré avec la machine, peut être fourni par l'importateur sur demande. Il concerne la partie hard du Laser 3000 et reprend chaque élément, sous-ensemble et ensemble du micro-ordinateur et fera donc la joie des rois du fer à souder qui désirent raccorder telle ou telle carte.

NOTA : Le dépliant publicitaire que l'on peut trouver chez le revendeur n'est pas conforme à la distribution du Laser 3000 en France. Il fait référence à une bibliothèque de logiciels non disponible chez l'importateur du Laser 3000.



# CONTRE-MESURES

## UTILITAIRES

Trois utilitaires sont disponibles sur Laser 3000 :

- The FILER PROGRAM
- Fast COPY
- DISK SPEED AND DISK TEST.

Le premier permet de travailler sur le contenu des disquettes, copies, catalogues, etc. Le second réalise des copies rapides de programme et le dernier autorise des tests sur les disques afin

d'optimiser la vitesse de rotation du lecteur et de vérifier les différents circuits.

## COMPATIBILITE APPLE

Un des grands arguments publicitaires du Laser 3000 concerne sa compatibilité avec Apple. Voyons de plus près ce que cela signifie :

### Compatibilité HARD

Elle est quasi-inexistante. En effet ici,

le standard Apple est oublié et ne peut donc être pris en compte.

### Compatibilité langage

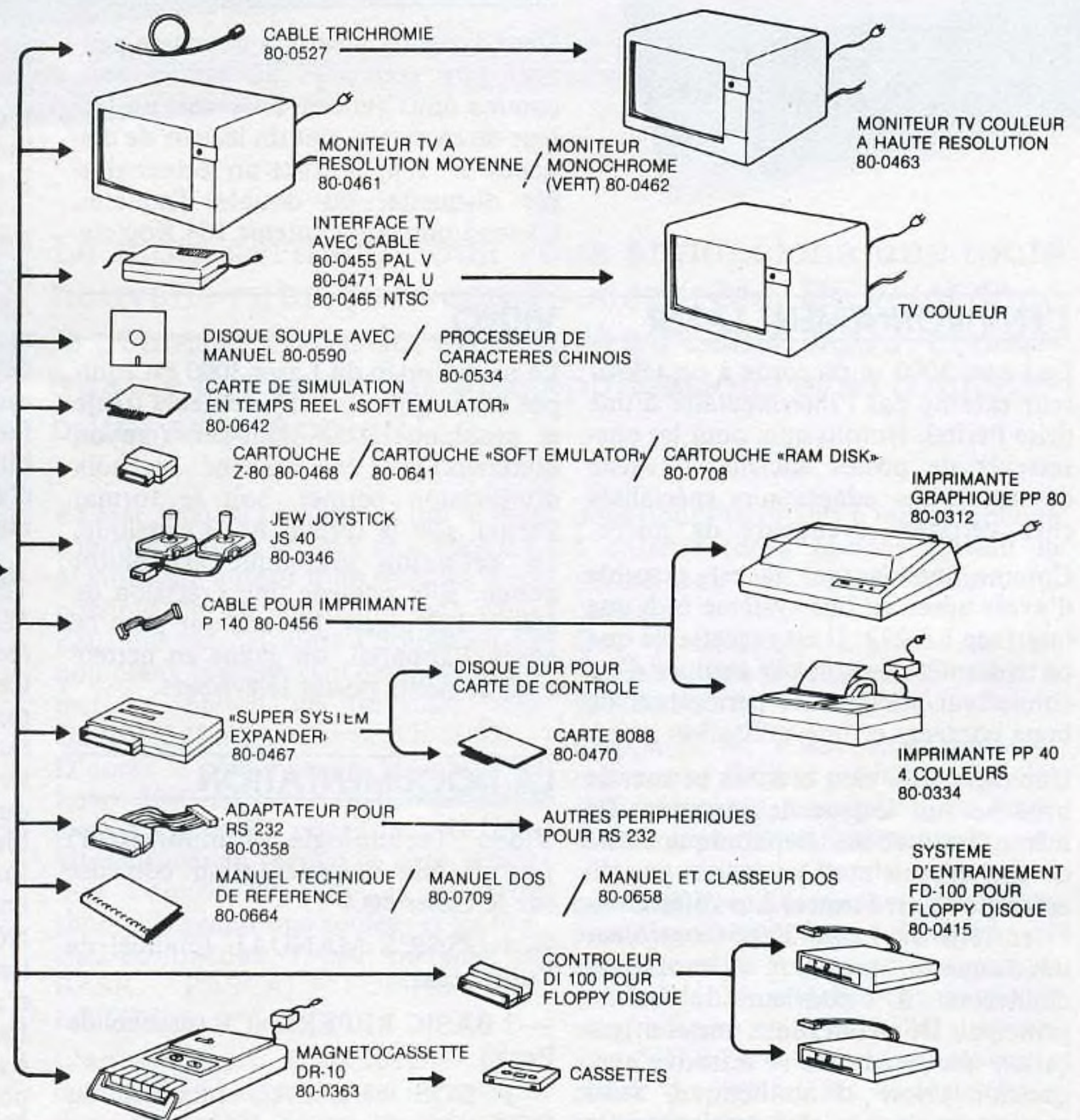
Le Laser 3000 reprend dans son intégralité l'Apple Soft et ajoute quelques ordres concernant le graphisme et les sons. De ce fait, l'utilisateur se retrouve avec un langage assez complet.

### Compatibilité logiciels

La compatibilité logiciel du Laser 3000 avec Apple II n'est pas évidente. Si cer-



Les différentes configurations du Laser 3000.





tains programmes tournent comme sur Apple II, cela n'est pas une règle générale. En effet, des logiciels deviennent capricieux quand ils sont branchés sur le Laser 3000 et ont un comportement incompatible. Après avoir questionné l'importateur, celui-ci nous a affirmé que 80 % des logiciels étaient compatibles. Mais il n'y a là rien de très grave pour les bons programmeurs qui, avec un peu d'astuce, arrivent à faire tourner l'ensemble des logiciels.

## FINALITE DU LASER 3000

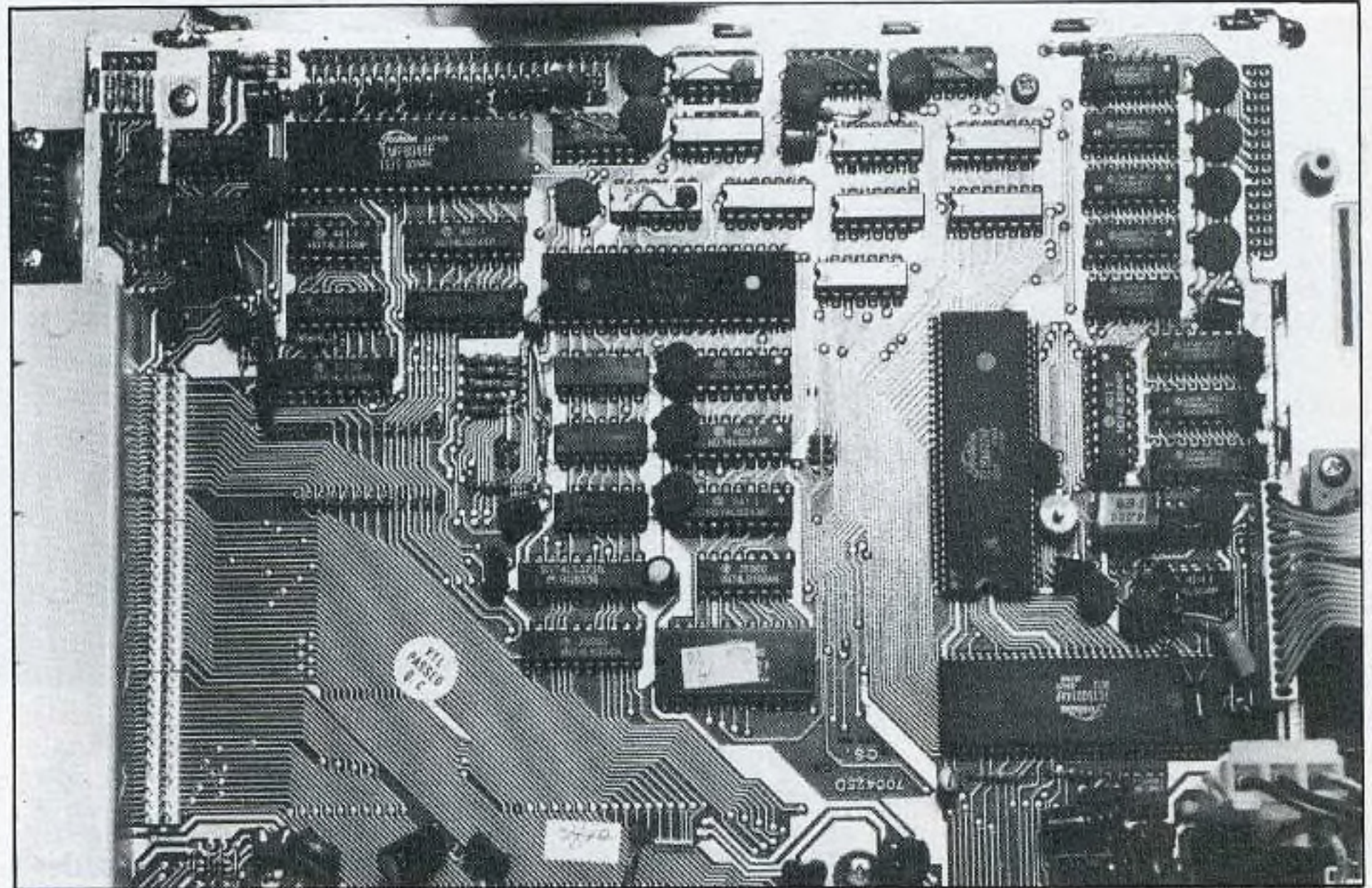
Quelle est la finalité du Laser 3000 ? A qui s'adresse-t-il ? En fait, il s'agit là d'un appareil plutôt complet et agréable à utiliser. Ses deux grandes orientations sont l'apprentissage de langages de programmation, donc l'initiation à la programmation, d'autre part les jeux et les applications domestiques. La capacité de stockage des disquettes ne permet pas d'application professionnelle au sens large du terme. Au bureau avec un fichier carnet d'adresses et quelques applications simples et la machine sera déjà saturée.

## CONCLUSION

Le Laser 3000 a quelques défauts, mais beaucoup de qualités qui en font un appareil d'un rapport qualité-prix très intéressant. En effet, les deux principaux défauts concernent une capacité relative des disquettes et la prise RS 232. Les qualités, elles, sont nombreuses : documentation très bien faite (en anglais), la couleur, les colonnes, les sons, le clavier, les touches de fonctions, etc.

Le Laser 3000 sera la machine idéale à placer entre le petit micro à 600 F (suez mon regard !) et le micro professionnel à plus de 30 000 F. Un bon compromis.

C.-H. Delaleu



La carte CPU du Laser 3000. Les composants sont câblés sur circuit en verre époxy.

## LE COUT DU LASER 3000

<b>Ordinateur ± Périphériques + Interfaces</b>	<b>T.T.C.</b>
LASER 3000 Secam Péritel complet comprenant :	
Unité centrale + Prise Péritel + Alimentation	
+ K7 démonstration + Câble lecteur K7	
+ Livres techniques + Garantie .....	<b>5 980 F</b>
Lecteur disquette FD 100/250 K : 160 K Formaté .....	<b>1 980 F</b>
Interface disquette + disquette de démonstration .....	<b>790 F</b>

## FICHE SIGNALÉTIQUE DU LASER 3000

<b>Référence</b>	: LASER 3000
<b>Importateur</b>	: VIDEO TECHNOLOGIE FRANCE
<b>Processeur</b>	: Microprocesseur 6502 A
<b>Langage</b>	: BASIC version de base
<b>Définition graphique</b>	: 560 × 192 pixels
<b>Editeur</b>	: 80 ou 40 colonnes
<b>Interface</b>	: RS 232
<b>Couleur</b>	: Rouge, vert, bleu, cyan, magenta, jaune
<b>Mémoire de masse</b>	: Disquettes 5" 1/4 - Lecteur K7
<b>Vidéo</b>	: Interface Péritel



# CONTRE-MESURES

## le départ est pris...

### Contre-mesures

Dans Led-Micro n° 12, vous avez pris connaissance de la philosophie des contre-mesures. Actuellement, pour des raisons techniques, les lecteurs ne choisissent pas encore les appareils testés dans cette rubrique. En effet, ce choix doit se faire en deux temps :

- Le choix
- La critique

### Le choix

Par courrier, les lecteurs peuvent demander que Led-Micro passe à la contre-mesure tel ou tel micro-ordinateur. Ce micro-ordinateur sera, soit la dernière machine à la mode, soit un grand classique. La gamme des produits testés peut être vaste, cela va du petit micro bas de gamme au micro de bureau. Si possible, lorsque vous nous adressez une demande de contre-mesures, indiquez-nous les raisons de votre choix.

### La critique

Les appareils sélectionnés pour une contre-mesure doivent être jugés par Led-Micro en toute objectivité, mais rien n'interdit aux lecteurs de donner leurs sentiments. Une critique peut être sévère, élogieuse, dans tous les cas de figures elle doit reposer sur des faits, sur des observations. De ce fait, Led-Micro indiquera à ses lecteurs, deux à trois mois à l'avance, les appareils testés afin que vous puissiez nous donner votre sentiment. Bien entendu, une critique ne

peut être gratuite. Pour cela ne pourront être publiées les remarques transmises sans justificatif. Prenons par exemple le cas de Monsieur Dupont qui a acheté un micro-ordinateur TOTO. Depuis son achat, Monsieur Dupont n'a quasiment pas utilisé son TOTO car il ne cesse de faire des allers-retours entre son domicile et le SAV-TOTO. Les remarques sur la fiabilité de TOTO, transmises à Led-Micro par Monsieur Dupont, devront être accompagnées de justificatifs (photocopie des bordereaux, etc). Les critiques peuvent être très variées :

- Publicité non conforme à la réalité
- Prix non justifié
- Logiciel non performant
- Utilisation difficile
- Documentation
- Assistance technique
- Délais de livraison
- Ventes illicites de copies de logiciels
- Compatibilité inexistante
- Connecteurs non standardisés, etc.

De toute façon, nous sommes persuadés que les critiques positives ou négatives émanant de nos lecteurs seront fondées, soit sur le plan de l'utilisation, soit sur le plan commercial. Dans tous les cas, ces critiques seront publiées avec l'autorisation de ces lecteurs et sous leur responsabilité.

Nous sommes dès à présent en mesure de vous indiquer le programme des prochains contre-mesures.

- Novembre : Commodore SX 64
- Décembre : Yashica MSX
- Janvier : Victor S1

— Février : Sinclair QL

Dès novembre (et pourquoi pas avant), écrivez-nous pour formuler vos remarques si vous êtes possesseur d'un Victor S1 ou en attente d'un QL Sinclair. Le choix des lecteurs devenant effectif dès mars, écrivez-nous dès aujourd'hui pour nous faire part de vos désirs.

### Nouveautés 85

Afin de bien commencer l'année 85, Led-Micro éditera chaque mois un second contre-mesures. Ce sera la contre-mesure Logiciel. En effet, les logiciels, et surtout les progiciels, représentent un chiffre d'affaires grandissant chaque jour. Ils sont désormais déterminants dans le choix d'une machine. Il y aura donc, dans chaque numéro de Led-Micro à partir de 85, une contre-mesure Hard et une contre-mesure Soft. La philosophie reste la même que pour les machines, à savoir :

— Choix du logiciel ou du progiciel à expertiser par le lecteur

— Critique de Led-Micro et des lecteurs en conclusion de chaque contre-mesures logiciel ou progiciel.

Comme pour les machines, le choix effectif démarre en mars, donc écrivez-nous dès aujourd'hui pour nous transmettre vos impressions.

Le choix pour commencer l'année en beauté s'est orienté vers deux grands classiques du progiciel :

- Janvier : Visicalc (tableau)
- Février : Delta (gestion de fichier).



Prix : l'unité 35 F prise à nos bureaux.  
Envoi par poste recommandé + 14,70 F  
soit 49,70 F

Venez chercher votre (vos) exemplaires, ou envoyez ce bon de commande, accompagné de votre règlement à :  
EDITIONS FREQUENCES  
1, boulevard Ney, 75018 Paris

Nom .....

Adresse .....

Ci-joint le montant de .....

CCP  Chèque bancaire  Mandat



## habilitez votre collection

# LedMICRO

avec une superbe  
reliure toilée jaune





La micro-informatique envahit le Sicob. Elle est partout. Plus personne aujourd'hui ne semble contester l'ampleur du phénomène micro. Mais plus que jamais ce terme recouvre des machines différentes dans leur conception et leur utilisation, quels points communs entre un micro d'initiation et un super-micro multi-utilisateur ?

L'important n'est pas là, mais dans la passion, le goût de savoir des enfants qui, au carrefour de la communication, manipulaient des T07 ou des Oric.

# AU 35<sup>e</sup> SICOB la micro et les micros

**P**our la deuxième fois cette année, tous les moyens d'informations se sont braqués sur la voûte de béton et de verre du Cnit où s'est tenu le 35<sup>e</sup> Sicob. La télévision, les journaux, émissions ou éditions spéciales ont tout dit ou presque sur l'informatique, la bureautique, la télématique et autres merveilles en «tique». Quelques 500 000 visiteurs se sont bousculés dans les allées cherchant parfois vainement à apercevoir les dernières prouesses technologiques de l'industrie mon-

diale. Pas facile de s'y retrouver. Imaginez plus de 870 stands et 4 000 produits différents.

Cette manifestation venant quatre mois après le Spécial Sicob consacré uniquement à la micro-informatique, il ne fallait pas s'attendre à trouver beaucoup de nouveautés, d'autant que sous l'effet de son développement considérable dans le monde, la micro-informatique se standardise de plus en plus. Standardisation qui conditionne l'accélération de son





développement et l'abaissement des coûts.

Le Sicob est un formidable show capable de séduire tous les publics tant les scènes et les tableaux sont variés. La révélation du spectacle a, sans aucun doute, été le «DG One». Un micro portable, réellement portable, présenté par Data General. Il pèse moins de 5 kg et est le premier système à disposer d'un écran LCD de 25 lignes de 80 caractères, capable de visualiser des données graphiques. Il est bien entendu compatible IBM-PC. Construit autour d'un microprocesseur Intel 8086 à technologie CMO, il dispose d'une mémoire de 128 K extensible à 312 K. Deux micro-disquettes 3,5" de 737 K de capacité formatée peuvent lui être raccordées ainsi qu'un châssis d'extension susceptible de recevoir cinq cartes (mémoire additionnelle, disquette 5,25", disque dur 5,25"...), ainsi qu'une mini-imprimante. Doté d'une remarquable panoplie de logiciels (Lotus 1-2-3, dBase II, Wordstar, Multiplan, Supercalc, Pfs, Easywriter, etc., sans oublier Cobol et Fortran. De plus CP/M 86 et Venix sont disponibles en option), DG One est l'outil par excellence du manager. Grâce à un logiciel intégré de bureautique, il peut se raccorder à un réseau bureautique et ainsi expédier des documents et en recevoir d'autres. Cette machine a sans nul doute quelques longueurs d'avance sur ses concurrentes et préfigure ce que deviendront les portables à échéance de dix-huit mois à deux ans.

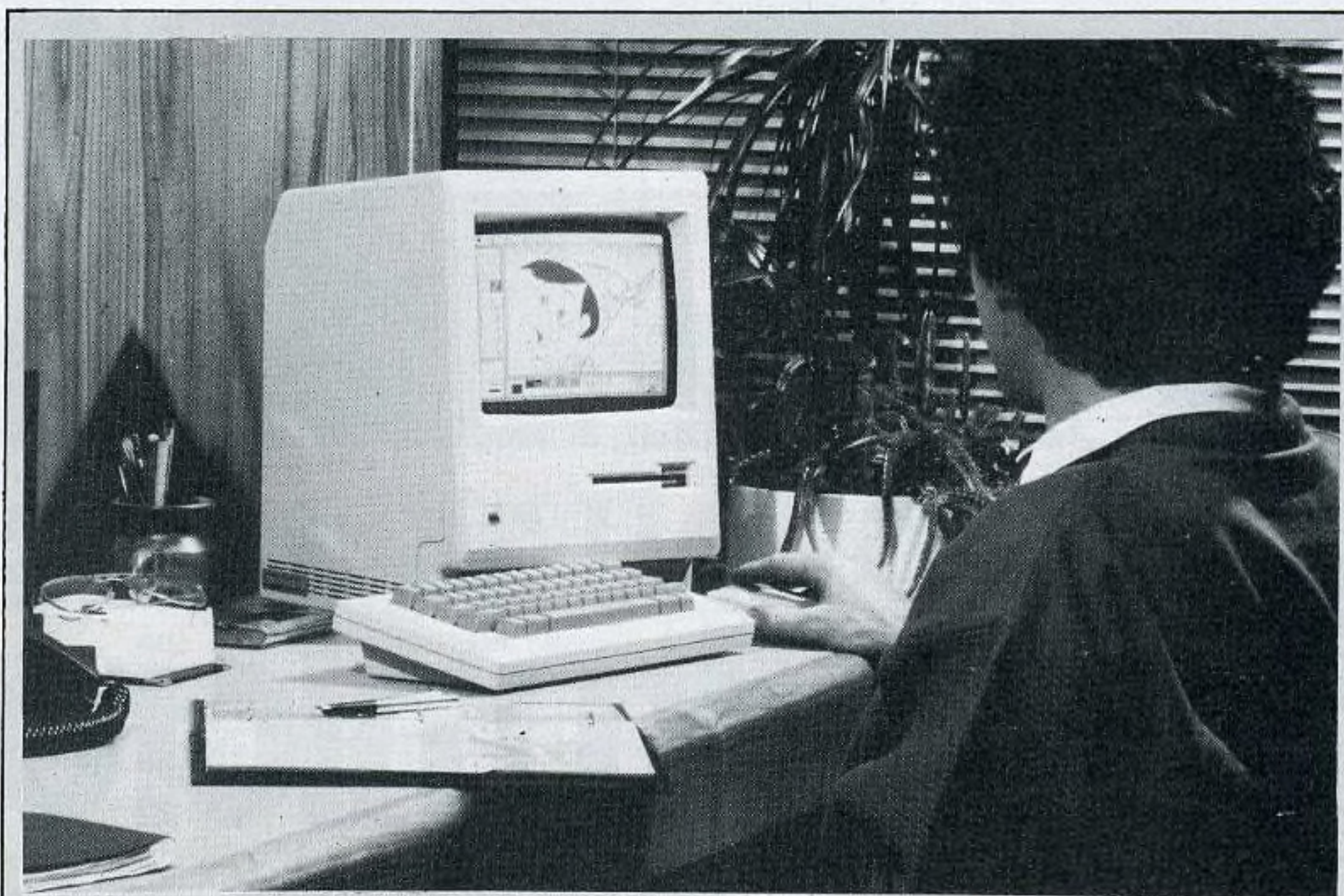
Vedettes déjà connues, les portables ont réalisé de belles pres-

tations. Le portable type pèse moins de 15 kg, il est équipé d'un écran le plus souvent de 9", de deux disquettes 5,25". Il est compatible IBM-PC (il faut exécuter des logiciels conçus pour le PC). Un nouveau venu particulièrement intéressant est

le Compaq Plus, doté d'un disque dur Winchester de 10 Mo et d'une unité de sauvegarde sur bande. Cette société américaine encore inconnue en France est actuellement première au hit-parade US des vendeurs de portables. Parmi les portables,

quelques noms à retenir : M21, la version portable du M24 d'Olivetti, Z160, frère jumeau du Z150 de Zenith, le portable de Texas Instruments, le Vicky de Victor, l'Apricot portable d'ACT.

Après les transportables, les



L'ordinateur Macintosh d'Apple.

## REGARDS SUR LE SICOB BOUTIQUE

*La boutique est plus humaine que le Sicob. L'atmosphère y est plus agréable même si ce bâtiment provisoire coupé en deux n'est pas très fonctionnel. Cette année il y avait cent vingt exposants, dont un grand nombre de boutiques (des chaînes comme Siréa, Computerland, des boutiques comme Illel, KA, La Règle à calcul, Espace 01...), ainsi que des distributeurs de logiciels. A la boutique, on pouvait voir un bon nombre de micro-ordinateurs domestiques, tels*

*l'Atmos, le QL de Sinclair dont la commercialisation en France n'interviendra pas avant plusieurs mois, le micro au standard MSX de Yamaha et celui de Yashica. La boutique, c'était aussi Apple et IBM ainsi que de nombreux compatibles comme le Pap de Toshiba, le Z160 de Zenith, l'Hitachi 16000, des portables comme le nouveau biprocesseur 8116 koctets de SKS, le SKS 3500 distribué par Sodiep qui va bientôt commercialiser également un compatible*

*IBM d'origine allemande, le Computer Molecular.*

*Côté logiciel, la boutique était fort bien pourvue, notamment des logiciels verticaux mais aussi des logiciels pour micro domestique. A signaler, Virgule, un traitement de texte destiné au Commodore 64, Microscript, un traitement de texte en français pour l'IBM-PC, Papyrus un autre traitement de texte, mais destiné à l'Apple IIe et Apple IIc, distribué par Ediciel.*



portables, les portatifs, voici les super-micros. Le développement de microprocesseurs 16 bits de la seconde génération a donné naissance à une nouvelle famille de micros adaptée à un environnement multi-utilisateur. Ces super-micros

*Data General One, premier ordinateur portable à écran cristaux liquides.*



requièrent des ressources de stockage importantes (bandes magnétiques, disques durs intégrés) et un système d'exploitation multi-utilisateur : Unix ou ses dérivés. A cette catégorie appartiennent l'IBM-AT annoncé dans le courant de l'été, la série Professional de Dec, les systèmes Mapper 5 et 6 de Sperry, le B25 de Burroughs, ou vedette américaine, le Macintosh...

Apple présente et commercialise dès maintenant une version 512 K de Macintosh ainsi que l'extension mémoire permettant de passer du 128 K au 512 K. Progressivement Macintosh s'est doté de logiciels : une soixantaine distribuée en France. Pour moitié ces logiciels sont de conception française. Le Macintosh 512 K est vendu avec un traitement de texte Mac Write, un logiciel de dessin Mac Point et «une visite guidée», 29 995 F soit 10 000 F de plus que le 12 K ; la diffé-

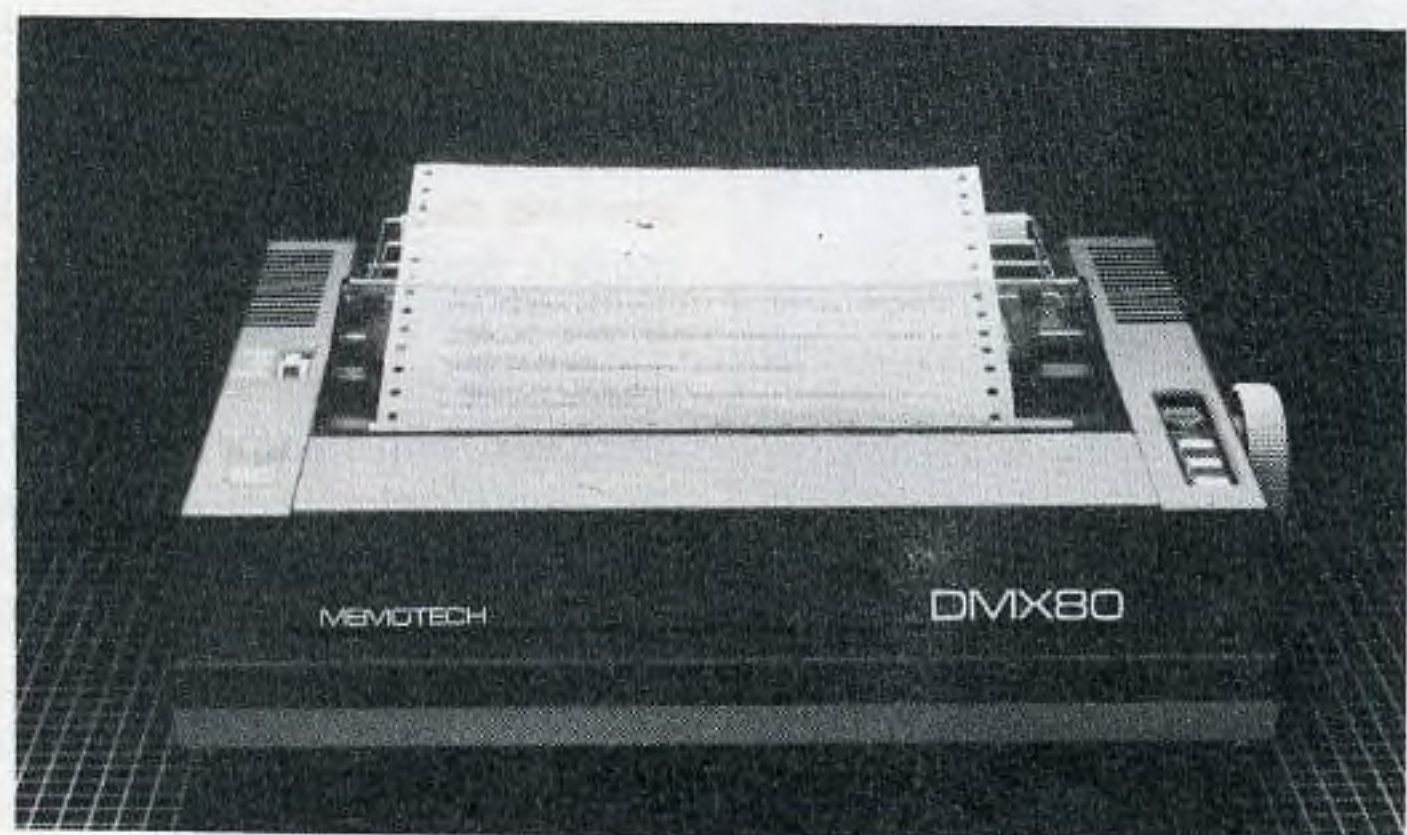
rence représente le prix du kit d'extension.

A l'autre bout de la chaîne se trouvent toutes les petites machines, moins spectaculaires, elles n'en sont pas moins intéressantes car c'est grâce à elles que la micro-informatique commence à faire partie de notre quotidien. Elles valent entre 1 000 et 5 000 F en moyenne. Aux premiers micros au standard MSX déjà connus — Sanyo, Canon, Yamaha — est venu s'ajouter au Sicob, Yashica. Ce constructeur d'appareils de photos se lance à son tour dans la micro avec un modèle équipé d'une mémoire de 64 K et doté d'une gamme d'accessoires (magnétophones, moniteur, imprimante). Habillé de rouge comme l'Alice de Matra, le Yashica sera commer-



*Une nouvelle version sortie au Sicob : biprocesseur 8 116 bits de SKS.*





*MTX 512 de Memotech distribué par Beauvais Matic...  
... et son imprimante.*



*Une des gammes complètes de Matra : Alice 90.*

cialisé environ 3 900 F pour les fêtes de fin d'année. Matra propose désormais une gamme de quatre modèles de 4, 12, 32 et 56 K de mémoire. Les modèles 32 K et 56 K (baptisé Alice 90) existent en version coffret. Dans une superbe

valise rouge sont rassemblés un micro, un lecteur-enregistreur, quatre à cinq logiciels, un guide Basic, un guide d'instructions de l'éditeur-assembleur, des câbles (Péritel, raccordement au lecteurs-enregistreur, alimentation secteur).

Dans ce domaine, les Français tiennent un rôle non négligeable. A côté d'Alice et du M05 et du T07-70 de Thomson, voici Exelvision, fabriqué par la CGCT. Ce micro développé par des anciens de Texas-Instruments, a été conçu pour l'utilisateur. Et tout a été pensé pour son confort et son agrément : clavier détachable communiquant avec l'unité centrale par infra-rouges, manettes de

jeu à infrarouges, synthèse vocale incorporée. Venu de Grande-Bretagne, le MTX de Memotech joue double jeu si l'on peut dire. En effet, il peut être utilisé à des fins personnelles ou semi-professionnelles. Memotech a



*Le QL de Sinclair.*



*Le CPC 464 d'Amstrad.*





L'ordinateur personnel M21 d'Olivetti.

en effet opté pour un système à cartes enfichables dans le châssis, ce qui permet de faire évoluer la machine en fonction des besoins. Le MTX existe en deux versions (32 ou 64 K de mémoire vive). Il peut recevoir une unité de disquettes FDX sous CP/M 2.2 et un disque dur de 20 Mo. Atari a annoncé une baisse substantielle du prix du 800XL qui passe de 3 500 F à 2 500 F. La société ayant été rachetée par un patron de choc, Jack Tramiel (ex-patron de Commodore), on peut s'atten-

dre à un renouveau d'Atari. En face du Sicob, dans le centre commercial des «4 Temps», la société Amstrad présentait son micro CPC464, qui a la particularité d'offrir intégré au clavier, un lecteur de cassettes. Ce micro de 64 K de Rom est proposé avec un moniteur vert ou couleur. Il est doté de trois modes écran, d'une palette de 27 couleurs, d'un clavier de 74 touches mais hélas de type Qwerty, d'un générateur de son à trois voies de 7 octaves, d'une sortie imprimante. C.R.

## DEUXIEME FESTIVAL DU LOGICIEL FRANÇAIS

Invité par le Carrefour International de la Communication, le Festival du Logiciel a présenté quarante programmes français parmi les plus significatifs de la vie quotidienne. Trente sont des créations originales. Choisis parmi trois cents, par huit mille personnes venues au Festival de juillet à la Chartreuse de Villeneuve-lez-Avignon, ces logiciels témoignent de la créativité existante. A noter que huit des auteurs ont moins de vingt ans. Le grand prix a été décerné à «Transat», simulation

d'une course transatlantique à la voile, sur Apple, réalisé par Jean-Marc Sornin. Le deuxième grand prix est allé à un jeu d'adresse «Billard». Le prix de la Fondation de France à «Stroumpfe-moi un mot !» un programme destiné aux enfants de quatre à sept ans. Un objet ou un animal apparaît sur l'écran. Et l'enfant écrit le nom dans une case. Pour ceux qui ne savent pas lire, le mot apparaît et l'enfant cherche les lettres sur le clavier. Ce logiciel a été créé par Roger Weinachter sur Apple.

# FORMATION

## département informatique d'Educatel CHOISISSEZ UNE CARRIERE D'AVENIR

Devenir informaticien en 1984, c'est choisir une carrière d'avenir, avec l'assurance de trouver immédiatement de nombreux débouchés, et des perspectives d'autant plus intéressantes que la place de l'ordinateur ne cesse de s'accroître dans tous les domaines: économique, social, administratif, etc.

Quel que soit votre niveau de formation, Educatel se charge de vous apprendre en quelques mois par les moyens les plus modernes, et avec un enseignement personnalisé à votre cas, le métier informatique qui vous convient le mieux.

Demandez, sans aucun engagement de votre part, notre documentation gratuite en nous renvoyant le bon ci-dessous ou en nous téléphonant au (1) 208.50.02.

METIERS PREPARES	DUREE DE LA FORMATION	NIVEAU D'ACCES
OPERATEUR SUR ORDINATEUR	8 mois	3 <sup>e</sup> B.E.P.C.
PUPITREUR	13 mois	3 <sup>e</sup> B.E.P.C.
PROGRAMMEUR D'APPLICATION	17 mois	3 <sup>e</sup> B.E.P.C.
PROGRAMMEUR SUR MICRO-ORDINATEUR	9 mois	3 <sup>e</sup> B.E.P.C.
ANALYSTE PROGRAMMEUR	30 mois	Baccalauréat
B.T.S. INFORMATIQUE	32 mois	Baccalauréat
ANALYSTE	15 mois	Bac + 2 ans

Si vous êtes salarié, votre étude peut être prise en charge par votre employeur (loi du 16-7-1971 sur la formation continue).  
EDUCATEL - 1083, route de Neufchâtel  
3000 X - 76025 ROUEN Cédex

**Educatel**

G.I.E. Unieco Formation  
Groupement d'écoles spécialisées.  
Etablissement privé d'enseignement  
par correspondance soumis au contrôle  
pédagogique de l'Etat.

POSSIBILITE  
DE COMMENCER  
VOS ETUDES  
A TOUT MOMENT  
DE L'ANNEE

### BON pour recevoir GRATUITEMENT

et sans aucun engagement une documentation complète sur le secteur ou le métier qui vous intéresse, sur les programmes d'études, les durées et les tarifs.

M.  Mme  Mlle

NOM .....

Prénom .....

Adresse: N° ..... Rue .....

Code postal [ ] [ ] [ ] [ ] Ville .....

(Facultatifs)

Tél. .... Age ..... Niveau d'études .....

Profession exercée .....

Précisez le ou les métiers qui vous intéressent:

EDUCATEL G.I.E. Unieco Formation  
3000 X - 76025 ROUEN CEDEX

Pour Canada, Suisse, Belgique: 49, rue des Augustins - 4000 Liège  
Pour TOM-DOM et Afrique: documentation spéciale par avion.

ou téléphonez à Paris  
(1) 208.50.02

SOGEX

LEM007



# A LIRE

## M05 votre micro-ordinateur

Serge Poutslajus. Collection micro-onde. Cedic Nathan.

C'est avec plaisir que nous voyons Thomson persévérer sur le marché de la micro-informatique (l'exemple anglais ferait-il des envies : Sinclair, Oric, BBC...). Après le T07, voici le M05, micro-ordinateur de moyenne gamme vendu aux alentours de 2 500 F. Comme toujours le lancement d'un nouveau micro-ordinateur est accompagné de nombreux livres de présentation, c'est le cas avec cet ouvrage des éditions Nathan (collection micro-onde) qui est un «voyage au pays de la micro-informatique et du M05» indique l'auteur.

Ce livre débute par une description externe et interne du M05. Vous apprendrez ainsi comment mettre en route le M05 et quels sont les principaux composants (microprocesseur, mémoire, E/S...) constituant ce micro-ordinateur. Il faut noter que le M05 est équipé du microprocesseur 6809 (Motorola) qui est un microprocesseur 8 bits très performant. La seconde partie de cet ouvrage aborde l'aspect logiciel du M05. Comme la plupart des micro-ordinateurs, le M05 dispose du Basic Microsoft, de ce côté donc pas de surprise. Au niveau graphisme, le M05 offre 16 couleurs (au lieu de 8 pour le T07). Les principales instructions graphiques (SCREEN, PSET, LINE, BOX) sont développées à l'aide de nombreux exercices (réalisation des drapeaux des USA et de l'URSS par exemple).

Dans la troisième partie l'auteur approfondit le Basic du M05 et donne quelques exemples de sous-programmes utiles. Enfin la dernière partie de ce livre est consacrée aux différents périphériques que l'on peut relier au M05 et à leur gestion (enregistreur de cassettes, lecteur de disquettes, crayon optique, imprimante...).

En conclusion un livre très pédagogique à un prix d'étudiant (35 F !).



## 50 jeux avec du papier et des crayons

François Pingaud, Jean François Gerne. Editions du Rocher.

La clientèle des jeux de réflexion est de plus en plus importante. Cet engouement a été accentué par l'introduction dans les familles de la micro-informatique. En effet, peut-on rêver d'un meilleur adversaire que le micro-ordinateur ? Ce nouveau livre (collection jeux et stratégie) propose 50 jeux qui nécessitent comme seuls accessoires une gomme, un crayon et... occasionnellement un micro-ordinateur.

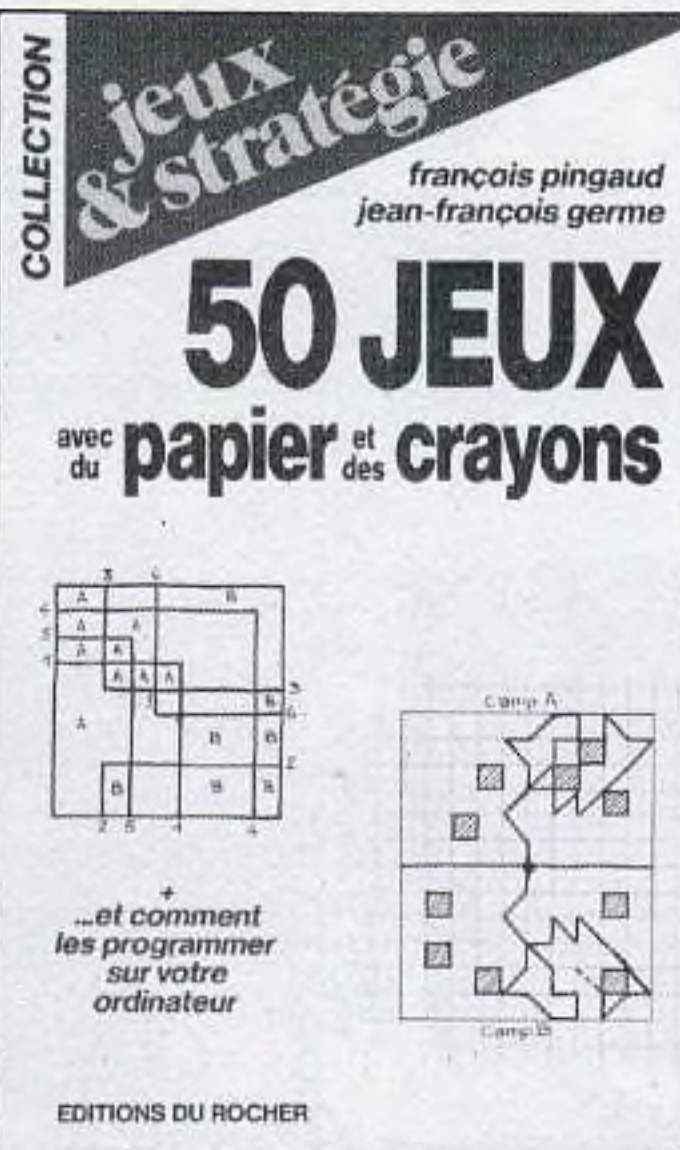
Dans une première partie, une analyse et le mode d'emploi de 50 jeux sont donnés.

Dans une seconde partie les auteurs proposent un traitement informatique de 10 des 50 jeux. Pour chaque jeu, un listing écrit en Basic (PET COM-MODORE) est développé, accompagné de quelques explications concernant la structure du programme et le mode d'emploi du jeu.

## Minitel votre guide pratique

Collection Micromonde. Cedic Nathan.

La télématique à domicile est devenue une réalité grâce au Minitel qui permet à de nombreux abonnés d'être reliés à des banques de données.



Ce nouveau livre de J. David est un guide pratique pour les utilisateurs qui désirent exploiter toutes les possibilités de ce périphérique. Personnellement je trouve cette initiative très judicieuse car les notices d'emploi des PTT sont bien... maigres.

Dans une première partie l'auteur donne quelques renseignements pratiques et techniques sur le mode d'emploi du terminal. Les principales touches du clavier sont examinées ainsi que la liaison entre Minitel et le téléphone. Une fois la mise en route effectuée l'utilisateur peut rentrer en conversation avec les banques de données. C'est tout d'abord l'annuaire électronique qui permet de trouver un abonné très facilement. Les différentes opérations à effectuer sont explicitées sous la forme de petits organigrammes.

Outre l'annuaire, Minitel permet d'accéder à de nombreux autres services (journaux, Air



Inter, banques). Un guide des principaux services est donné dans ce livre ainsi que le moyen de s'y connecter.

Dans une dernière partie plus technique l'auteur décrit l'architecture du réseau Minitel et les différents composants qui permettent la liaison du terminal Minitel avec l'ordinateur d'un autre serveur.

Ce petit livre se lit très facilement et est accessible à tous les «non informaticiens».

## Oric et son microprocesseur

François Normand, Frédéric Blanc. Editions Micro programmes 5.

L'Oric est devenu en peu de temps un des piliers de la micro-informatique. Ce succès n'est pas démenti par la quantité impressionnante de livres qui paraît tous les mois sur ce micro-ordinateur. Ce nouveau livre traite du microprocesseur 6502 de l'Oric. Dans une première partie, toutes les instructions de base du 6502 sont étudiées. Il est bien évident que cette partie très générale peut être appliquée à tous les micro-ordinateurs disposant d'un 6502 (APPLE, COMMO-DORE...). Cette initiation au



langage machine est suivie de quelques exemples de programmes écrits en langage assembleur comme les routines ANTINEW, RESTORE ou SCROLLING.

La dernière partie de ce livre est consacrée au désassemblage complet de la ROM basic de l'Oric sous la forme d'un listing. Ce listing est une mine de renseignements pour tous les utilisateurs avides de sous-programmes écrits en assembleur.

Les curieux pourront ainsi apprendre comment écrire une fonction sinus en assembleur ou encore comment gérer un écran de visualisation. Point important, aucun commentaire n'accompagne ce listing, ce qui implique une bonne connaissance de l'assembleur et beaucoup de patience pour reconstituer l'interpréteur Basic et le moniteur.



### L'informatique professionnelle.

AOÛT-SEPTEMBRE 84, Editions d'informatique

L'implantation des micros dans les entreprises est vigoureuse, elle pose de sérieux problèmes

d'harmonisation et d'organisation.

«Le PC noyauté l'entreprise» ! Ce titre humoristique de l'éditorial du numéro de septembre de l'Informatique Professionnelle donne le ton.

De plus en plus les micros envahissent l'entreprise. Cela nécessite une maîtrise de la situation si on ne veut pas s'installer dans l'anarchie. Des interviews d'utilisateurs et de directeurs informatiques, des analyses d'experts, des comparaisons avec ce qui se passe ailleurs (aux Etats-Unis entre autres) constituent le sommaire de l'étude que cette revue consacre ce mois à la micro-informatique. On y trouve notamment une interview très piquante d'un directeur informatique «anti-micro». Il explique pourquoi avec des arguments préoccupants et analyse son action pour contenir la vague micro qui a

envahi son entreprise. Une réflexion positive et qui fait clairement le point.

Cet autre qui, connaissant bien la situation de la micro-informatique aux U.S.A. annonce, après lecture de la presse américaine, que les budgets informatiques sont en passe d'être considérablement augmentés. Autre phénomène : la création d'une nouvelle grande fonction : «le micro-manager» dont la principale qualité devra être la lucidité. Enfin, un effort sérieux devra être fourni quant à la formation du personnel

C'est avec humour que Laurel et Hardy terminent ce numéro. Dans un article intitulé : «Apocalypse now !», nos deux compères font un tour d'horizon des côtés positifs et des côtés négatifs du phénomène micro-informatique, dans lequel personne n'est épargné ! P. F

## Des bons métiers où les jeunes sont bien payés



### INFORMATIQUE

B.P. Informatique diplôme d'Etat.

Pour obtenir un poste de cadre dans un secteur créateur d'emplois. Se prépare tranquillement chez soi avec ou sans Bac en 15 mois environ.

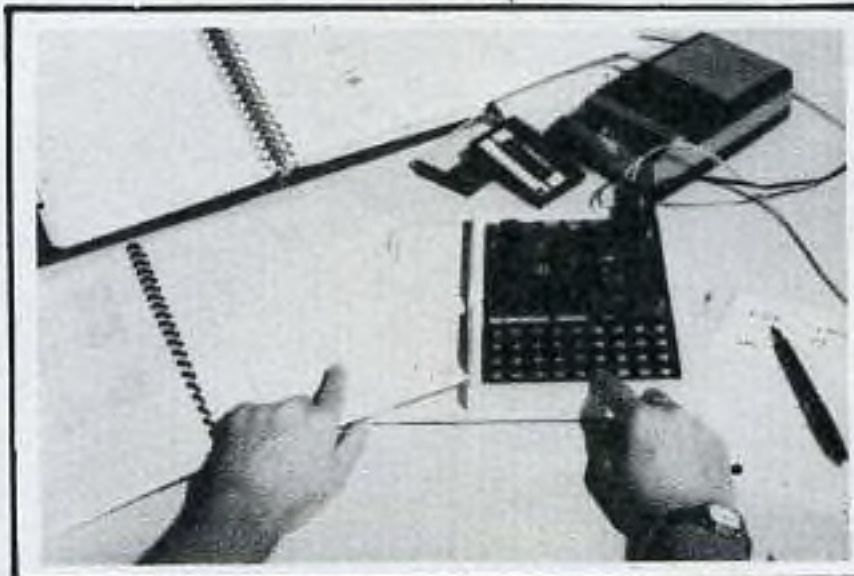
**Cours de Programmeur, avec stages pratiques sur ordinateur.**

Pour apprendre à programmer et acquérir les bases indispensables de l'informatique. Stage d'une semaine dans un centre informatique régional sur du matériel professionnel. Durée 6 à 8 mois, niveau fin de 3<sup>e</sup>.

### MICRO-INFORMATIQUE

**Cours de BASIC et de Micro-Informatique.**

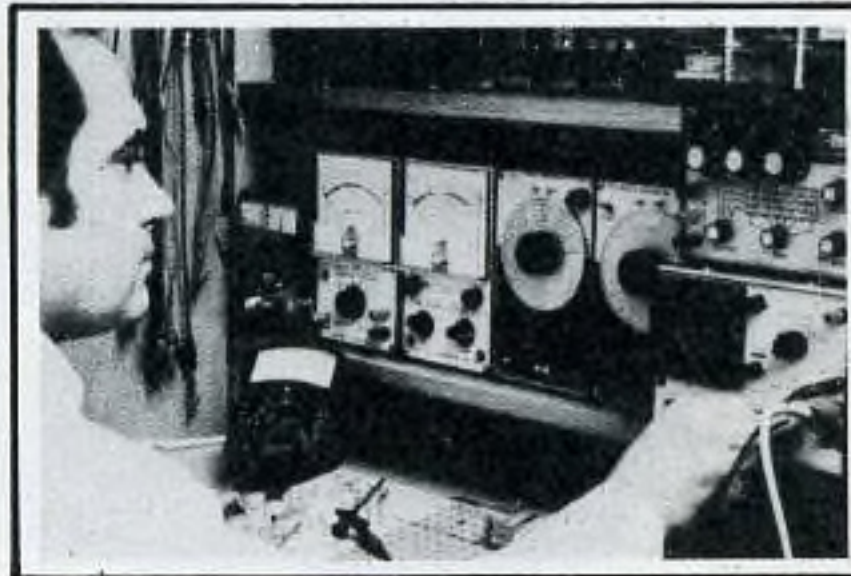
En 4 mois environ, vous pourrez dialoguer avec n'importe quel "micro". Vous serez capable d'écrire seul vos propres programmes en BASIC (jeux, gestion...). Niveau fin de 3<sup>e</sup>.



### MICROPROCESSEURS

- Cours général microprocesseurs/micro-ordinateurs.

Un cours par correspondance pour acquérir toutes les connaissances nécessaires à la compréhension du fonctionnement interne d'un micro-ordinateur et à son utilisation. Vous serez capable de rédiger des programmes en langage machine, de concevoir une structure complète de micro-ordinateur autour d'un microprocesseur (8080-Z80). Un micro-ordinateur MPF 1B est fourni en option avec le cours. Durée moyenne des études : 6 à 8 mois. Niveau conseillé : 1<sup>er</sup> ou Bac.



### ELECTRONIQUE "84"

- Cours de technicien en Electronique/micro-électronique. Ce nouveau cours par correspondance avec matériel d'expériences vous formera aux dernières techniques de l'électronique et de la micro-électronique. Présenté en deux modules, ce cours qui comprend plus de 100 expériences pratiques, deviendra vite une étude captivante. Il représente un excellent investissement pour votre avenir et vous aurez les meilleures chances pour trouver un emploi dans ce secteur favorisé par le gouvernement. Durée : 10 à 12 mois par module. Niveau fin de 3<sup>e</sup>.

INSTITUT PRIVÉ  
D'INFORMATIQUE  
ET DE GESTION  
92270 BOIS-COLOMBES  
(FRANCE)  
Tél.: (1) 242.59.27



IPIG

Pour la Suisse : JAFOR  
16, avenue Wendt - 1203 Genève

Envoyez-moi gratuitement et sans engagement votre documentation N° X 3651 sur : L'INFORMATIQUE  LA MICRO-INFORMATIQUE  LES MICROPROCESSEURS  L'ELECTRONIQUE

Nom \_\_\_\_\_ Prénom \_\_\_\_\_  
Adresse \_\_\_\_\_  
Ville \_\_\_\_\_  
Code postal \_\_\_\_\_ Tél \_\_\_\_\_



## P.A. GRATUITES

Vends ZX81 16 K 3 K7 alimentation 6 manuels. Prix : 700 F. Tél. : 985.22.39 après 18 h.

Le micro à l'école, ce n'est pas pour demain ? J'aimerais initier les jeunes et monter un mini-Microclub. Cherche donateurs tout matériel micro. Merci. Tél. H.R. : (56) 82.44.53

Vds kit 6800 D2 Monté en coffret : 650 F. Carte Micro Texas 990/189 : 650 F. Imprimante MX80 III + RS 232 : 2 500 F. M. Rouer Alain, 20 bd St Georges, 06400 Cannes.

A vendre Sharp M2 720 + cassettes + livres. A vendre jeu électronique Sector. Tél. H.R. : (56) 02.35.80 ou (56) 83.49.74.

Vends cause départ les onze premiers numéros de Led-Micro pour 125 F (franco de port), le lot est indivisible. Tél. av. : (16-1) 842.09.78 le soir ou écrire à : F. Palaci 1, rue Léon Delagrangé, 75015 Paris.

Vends ordinateur TRS-80 modèle 4. Etat neuf. Prix à débattre. Tél. : (88) 69.92.12 après 19 heures. Christ. Rémy, 67370 Woelkenheim.

Vends Dragon 32 Péritel + câbles + nombreux logiciels (Forth, Donkey-Kong, Invasion, Voice, Phantom-Slayer, etc., valeur 3 000 F). Le tout : 2 800 F. Lot de logiciels pour ZX81 (ZXAS, ZXDB, ZXTK, ZXM, Chess, VU-Calc, etc.) : 300 F. S. Llorente 13, rue de Montchauvet, 78200 Mantes-la-Ville. Tél. : (3) 092.75.64.

Vends PC 1500 A + CE 150 + manuels (PSI, Eyrolles), matériels sous garantie (achats juin 84) à Fnac. Valeur 3 880 F (facture), vendu 3 300 F. Région Versailles. Tél. : 908.46.47 h. bur.

Vends Thomson T07 + magnéto + Basic. 6 mois de garantie : 3 000 F. Tél. : 981.56.88 ou 989.67.69.

Recherchons programmeurs Z80 6502 6809 pour jeux sur ordinateurs. Rémunération + matériel. A.R.G. 5, av. Monnot 71100 Chalon s/Saône. Tél. : (85) 93.34.82.

Etudiant en 3<sup>e</sup> année d'agronomie, recherche pour travaux de fin d'études en génétique un généreux donateur d'Apple II ou d'équivalent. Merci. Rannou Thierry, E.S.A. 24, rue Auguste Fonteneau, 49044 Angers.

Cède Triumph-Adler Alpha-PC + 1 drive 320 K, CP/M 2.2 (12/83) : 9 500 F à débattre. Fis, Ecole Maternelle 45240 Menestreau-en-Villette. Tél. : (38) 76.94.20.

STOP-AFFAIRE : Sega SC 3000 sous garantie (valeur 2 300 F), sacrifié 1 800F avec 2 joystick + manuel + K7 Basic. Nbrs K7 jeux disponibles à prix intéressant. Sur R.D.V. au 909.74.76 (après 19 h).

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



### Bulletin d'Abonnement

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

Nom ..... Prénom .....

N° ..... Rue .....

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

Envoyez ce bon accompagné du règlement à l'ordre des Editions Fréquences à :  
EDITIONS FREQUENCES, 1 boulevard Ney, 75018 PARIS

MODE DE PAIEMENT : CCP  - Chèque bancaire  - Mandat



# LASER 3000

pour 5.980,00 F\*,  
jouez la double  
compatibilité !



\* Prix T.T.C. de la console

## LASER 3000 : Compatible à votre budget

Vidéo Technologie fait partie des grandes entreprises qui cherchent constamment à améliorer l'informatique de demain !

Le LASER 3000 est de la race des grands microordinateurs accessibles à tous pour toutes les applications.

C'est un ordinateur universel !

## LASER 3000 : Compatible aux plus grandes bibliothèques mondiales de logiciels

Polyvalent grâce à ses différents microprocesseurs 6502A, Z 80, 8088, le LASER 3000 offre aux bouts des doigts le plaisir d'utiliser les meilleurs logiciels de langages, de gestion, d'organisation, de calculs scientifiques, de musique, soit des milliers de programmes.

PROCESSEUR	: 6502 A
HORLOGE	: 2 mhz
ROM	: 24 K - 32 K
RAM	: 64 K extensible à 192 K
RÉSOLUTION	: 256 x 192 8 couleurs
HAUTE RÉSOLUTION	: 560 x 192 6 couleurs
AFFICHAGE	: 40/80 colonnes 24 lignes
CLAVIER	: 81 touches + pavé numérique + 24 touches de fonction
SON	: 4 canaux / 6 octaves
IMPRIMANTE	: Interface Centronics parallèle incorporée
SORTIE	: Secam Péritel et vidéo composite
OPTIONS	: Emulateur Soft Carte Z80A (système CPM 80) Carte Intell 8088 Interface RS 232 C



**VIDEO TECHNOLOGIE FRANCE**

19, rue Luisant - 91310 Monthléry  
Tél. (6)901.93.40 - Télex : SIGMA 180 114



DISTRIBUTEUR EXCLUSIF

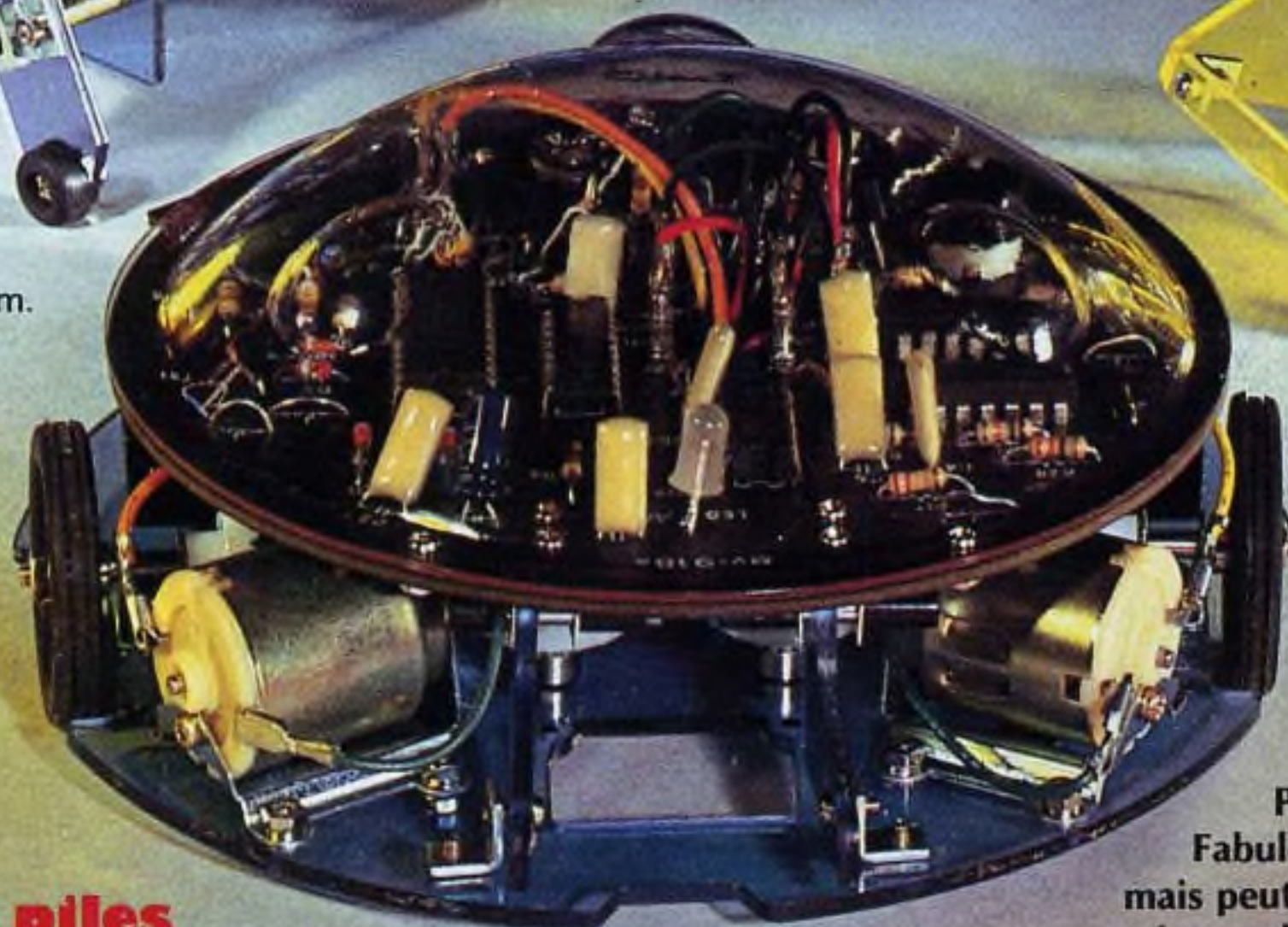
## Assemblez vous-même votre premier robot\*



**AVOIDER**  
14 cm × 14 cm × H. 12 cm.



**SOUND SKIPPER**  
7 cm × 7 cm × H. 10 cm.



**MEMOCRAWLER** Ø 14 cm, H. 6 cm.  
**PROGRAMMABLE** (clavier fourni).  
Fabuleux, se programme à partir d'un clavier  
mais peut aussi être piloté à partir d'un  
micro-ordinateur.

**NOUVEAU :**  
**kits livrés avec piles**

### \* IMPORTANT :

Les montages électroniques  
sont déjà effectués et testés,  
l'assemblage ne nécessite  
aucune expérience ni aucun  
outillage particulier

Revendeurs, contactez-nous  
pour distribuer ces fabuleux produits  
révolutionnaires. Pas de risque de stock,  
nous le maintenons pour vous, pour en  
savoir plus, téléphonez au  
16 (93) 42.49.98 ou écrivez-nous.

### Découvrez les multiples fonctions de la robotique

**SKIPPER MECHA** - 129 F. Le plus simple, se  
déplace rapidement en claudiquant sur ses deux  
jambes.

**SOUND SKIPPER** - 199 F. Le modèle précédent  
équipé d'un micro ampli qui le fait réagir à  
chaque bruit assez fort.

**TURN BACKER** - 299 F. Se déplace sur ses 6 jam-  
bes et effectue un quart de tour à chaque fois  
qu'il perçoit un bruit assez fort par son micro très  
doué pour les slaloms.

**LINE TRACER** - 299 F. Se déplace sur 3 roues et  
suit seul une ligne tracée sur le sol, grâce à une  
cellule photo-sensible.

**PIPER MOUSE** - 329 F. Se déplace sur trois roues  
montées sur amortisseurs et réagit à chaque  
coup de sifflet grâce à son détecteur d'ultrasons.

**AVOIDER** - 329 F. Se déplace sur 6 jambes et  
évite les obstacles placés sur son chemin grâce à

son détecteur à infrarouges, très doué aussi pour  
le slalom.

**CIRCULAR** - 549 F. Il avance, tourne sur lui-  
même en glissant sur deux grands disques caout-  
choutés. Il est livré avec une radiocommande.

**MEMOCRAWLER** - 599 F. Le plus intelligent de la  
famille, il avance, tourne des deux côtés, émet  
un bruit ou s'allume en fonction du programme  
entré en mémoire à partir d'un clavier: (RAM  
256×4 bits). Ne nécessite aucune expérience  
préalable en programmation. Peut aussi être  
commandé à partir d'un micro-ordinateur grâce  
à une interface développée par ROBOTMANIA.  
Indiquez-nous le type de micro que vous pos-  
sédez et nous vous enverrons les détails.

**GUIDE DES ROBOTS FAMILIAUX** 200 pages.  
59 F.

**Votre robot ou le catalogue gratuit chez vous dans 48 h, en téléphonant au 16 (93) 42.57.12.**

Bon de commande ou demande de catalogue gratuit à renvoyer à

ROBOTMANIA, B.P. 3 - 06740 CHATEAUNEUF - FRANCE

SUISSE : ROBOTMANIA-SONAICO  
49, rue du Rhône CH 1204 GENEVE Tél. 022 - 287.866.

BELGIQUE : ROBOTMANIA-LA BONBONNIERE  
B - 6820 FLORENVILLE Tél. 061 - 311.038.

MODÈLES	PRIX
.....	
.....	
.....	
.....	
Participation aux frais de port et d'emballage	25 F.
<b>Total à payer :</b>	

NOM .....

ADRESSE .....

CODE POSTAL .....

VILLE .....

Demande de catalogue gratuit  Règlement : Je joins  un chèque bancaire  CCP 3 volets (ordre ROBOTMANIA) L M 13  
 Je préfère payer au facteur à réception (en ajoutant 20 F pour frais de contre-remboursement).