

hors série

Led

MICRO

APPRENDRE • La programmation, cours de Claude Polgar • Le Microprofessor MPF 1B, cours pratique de microprocesseur par Philippe Duquesne. • Les fortiches : utilitaires, photo d'écran. MAGAZINE • Les boutiques • La contre-mesure du Yashica YC-64 MSX • Nouveautés produits.

DE L'INITIATION A LA PRATIQUE DE L'INFORMATIQUE

COURS
N°15



YASHICA YC64 MSX

ISSN 0757-6889

MICROPROCESSEURS

COMPRENDRE
leur fonctionnement

CONCEVOIR - RÉALISER
vos applications

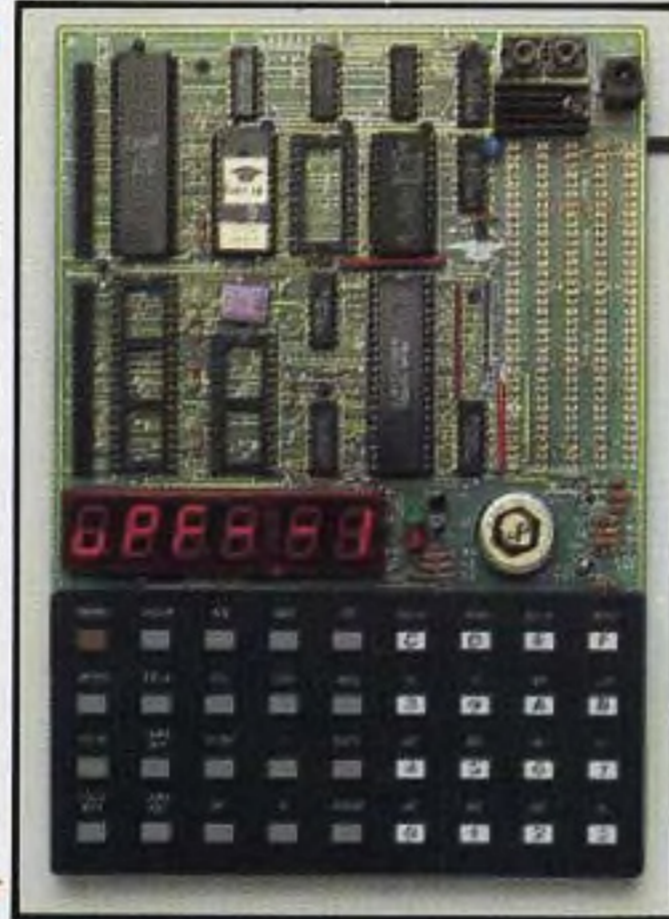


Z 80
R 6502
6809

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



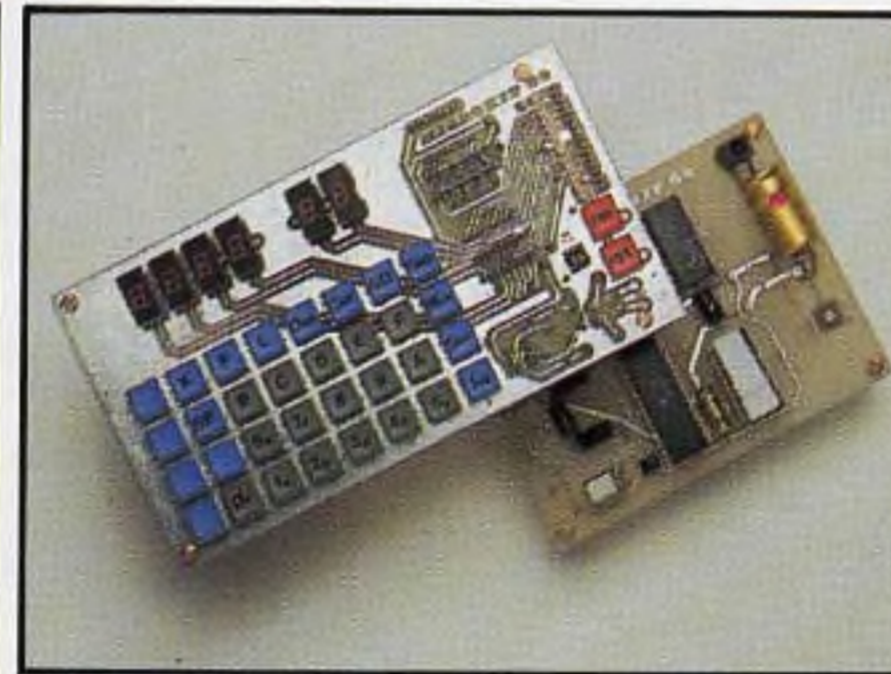
MPF-1 PLUS

- 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 ROM 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 : _____

hors série

LED MICRO

DECEMBRE 84

Société éditrice :
Éditions 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, Jean Yrytov

Rubriques Magazine

Chef de rubrique :

Claude-Hélène Roze
 Ont collaboré à ce numéro :
 Charles-Henry Delaleu, Philippe
 Faugeras

Secrétaire de Rédaction
 Chantal Cauchois

Réalisation graphique
 Serge Fayol

Publicité, à la revue

Tél. : 607.01.97
 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

**COURS PRATIQUE DE MICRO-
 PROCESSEUR**

avec le Microprofessor MPF 1B

par **Philippe Duquesne**

40

LE COIN DES FORTICHES

Utilitaires

par **Jean Yrytov**

44

LE COURRIER DES LECTEURS

Claude Polgar répond

48

LIBRES PROPOS

Réflexions sur la micro-informatique

RUBRIQUES MAGAZINE

50

D'UNE BOUTIQUE A L'AUTRE
 Sourire, compétence et incompé-
 tence

56

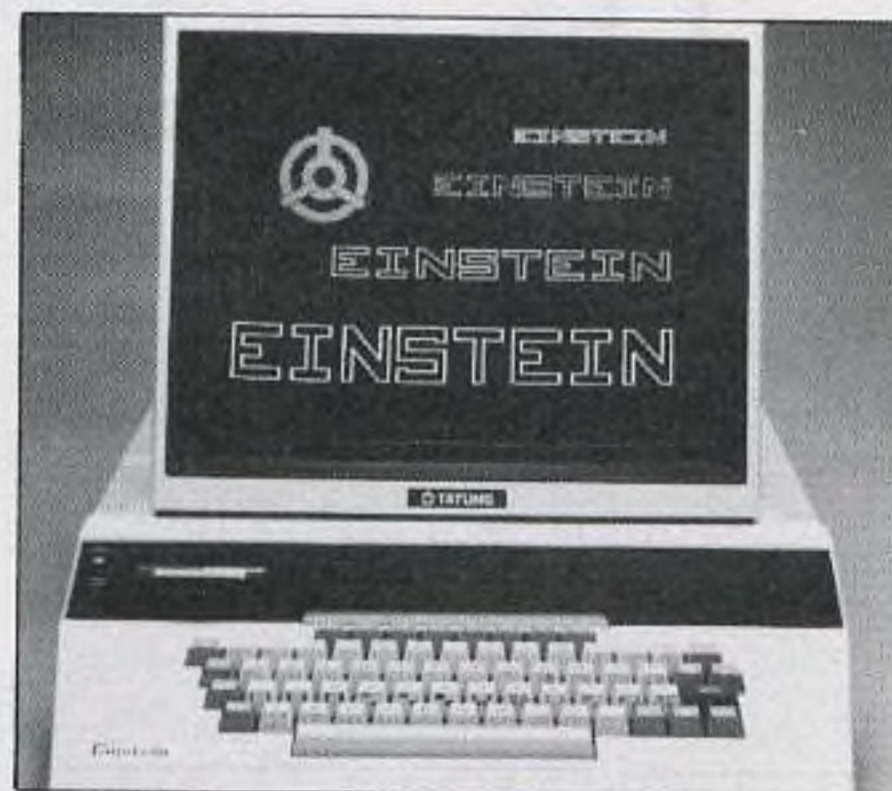
LES CONTRE-MESURES

Le Yashica YC-64



60

LES INFOS ET LES PRODUITS



64

BIBLIOGRAPHIE

A lire

NOTRE COUVERTURE : Le Yashica YC-64 MSX, Importé par Ségimex.



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-58 C (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.

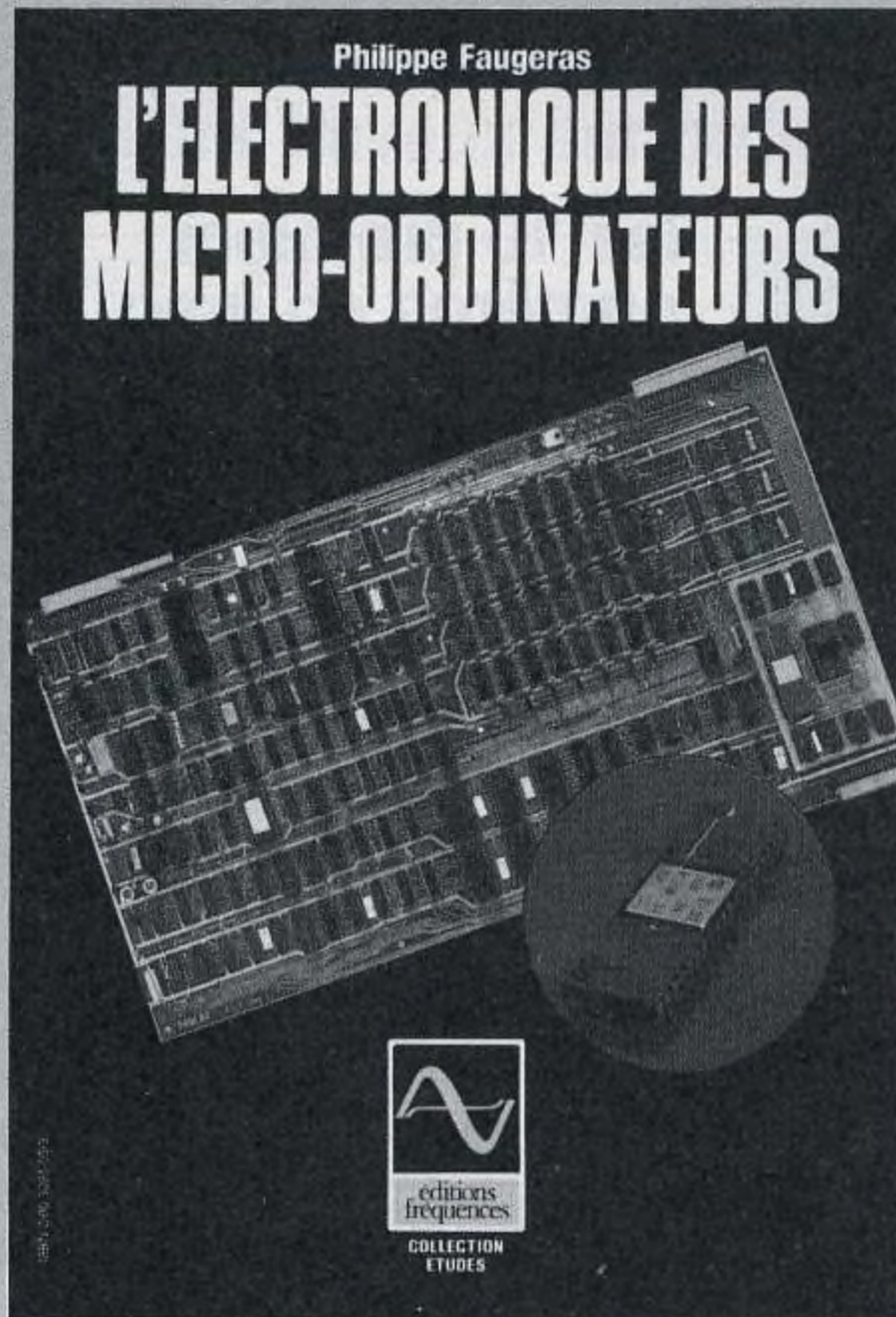


TEXAS INSTRUMENTS

VOYAGE AU CŒUR DES MICRO-ORDINATEURS

dans la
COLLECTION
«ETUDES»
aux
éditions
fréquences

En vente
chez tous les libraires et
aux Editions Fréquences



**une véritable
schémathèque**

- 128 pages
 - 101 schémas
 - 34 tableaux
- Prix : 165 F**
(port compris)

Que ce soit pour concevoir des interfaces ou optimiser un programme (utilisation des périphériques, encombrement mémoire...) «un micro-informaticien performant» doit posséder une bonne connaissance de son matériel.

Ce livre s'adresse donc à tous les électroniciens qui désirent découvrir les différents

composants constituant un micro-ordinateur. Articulé autour du microprocesseur Z80, cet ouvrage contient de nombreux schémas (plan mémoire, interfaces série et parallèle, interface clavier, interface vidéo, CAN, CNA...) qui pourraient être le thème... de nouvelles extensions.

BON DE COMMANDE

Je désire recevoir l'ouvrage «**L'électronique des micro-ordinateurs**»

Nom

Adresse

A adresser aux **EDITIONS FREQUENCES** 1 boulevard Ney, 75018 Paris

Règlement ci-joint :

Par chèque bancaire par chèque postal par mandat

Philippe Faugeras, Docteur-ingénieur en électronique a acquis son expérience dans de grandes entreprises françaises où pendant cinq ans, il a travaillé sur des systèmes d'automatismes à base de microprocesseurs. Philippe Faugeras est responsable de la rubrique «Raconte-moi la micro-informatique» dans la revue LED.

Vous découvrez Led-Micro avec ce n°15
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

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 Z80R
- N°13** L'affectation (fin) : INPUT - La sélection (1^{re} partie : Sélection simple).
- N°14** La sélection - Le langage Z80R (suite).

Vous désirez un ou plusieurs numéros que vous manquent (de 1 à 14) : 18 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)

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(15)

TROISIEME PARTIE (SUITE)

Premiers travaux sur ordinateur

<p>3. 1. But et contenu de cette 3^e partie</p> <p>3. 2. Les systèmes types</p> <p>3. 3. Choisir, installer, brancher</p> <p>3. 4. La pratique du clavier</p> <p>3. 5. De la mise en route au caractère d'attente</p>	LED-MICRO n° 5
<p>3. 6. Un premier programme en Basic</p> <p>3. 7. Modifications et complétons ce programme</p> <p>3. 8. La ponctuation dans le PRINT</p> <p>3. 9. Exercices sur le PRINT</p>	LED-MICRO n° 6
<p>3.10. Le déroulement d'un programme</p> <p>3.11. Nombres et calculs (1^{re} partie : les nombres)</p>	LED-MICRO n° 7
<p>3.11. Nombres et calculs (2^e partie : les calculs)</p>	LED-MICRO n° 8
<p>3.12. Conventions et notations</p> <p>3.13. Notions élémentaires sur les fichiers et les SED (1^{re} partie)</p>	LED-MICRO n° 9
<p>3.13. Notions élémentaires sur les fichiers et les SED (2^e partie)</p>	LED-MICRO n° 10
<p>3.13. Notions élémentaires sur les fichiers et les SED (3^e partie)</p>	LED-MICRO n° 11
<p>3.14. Complément sur le listage</p> <p>3.15. L'affectation. Variables numériques</p>	LED-MICRO n° 12
<p>3.15. L'affectation (suite). Variables chaînes et booléennes.</p>	LED-MICRO n° 13
<p>3.15. L'affectation (fin) : INPUT</p> <p>3.16. La sélection (1^{re} partie : sélection simple)</p>	LED-MICRO n° 14
<p>3.16. La sélection (suite) Compléments sur les chaînes de caractères</p>	LED-MICRO n° 15

Changement de référence

A partir de maintenant, au lieu de prendre comme «BASIC de référence» le BASIC du PROF 80 (c'est-à-dire celui du TRS 80) et comme «BASIC secondaire» le BASIC étendu de l'APPLE II (l'«APPLESOFT»), nous prendrons comme référence le BASIC Microsoft utilisé sur le PC d'IBM et ses compatibles.

Nous vous fournirons, à partir du mois prochain, les «variantes» sur les systèmes APPLE II, COMMODORE 64, ZX Spectrum, TO7, ORIC (dans le cas où il y a des différences bien sûr), et en négligeant la traduction de CLS en HOME (que vous devez commencer à connaître).

Le BASIC du PC d'IBM est un BASIC d'origine Microsoft, pratiquement identique à celui du BASIC (Microsoft également) du TRS 80. Vous ne serez donc pas déroutés. Une petite différence toutefois dans les fonctions aléatoires (vous ne vous rappelez pas ? relisez LED MICRO n° 8 §3.11.6.E pages 30 et 31).

Le BASIC du PC utilise le sigle RND (abréviation de Random) pour «renvoyer» une série de nombres pseudo aléatoires compris entre 0,00000 et 0,99999.

Mais chaque fois que l'on réinitialise le système (en particulier chaque fois que l'on fait «RUN») cette série de nombres recommence au même point de départ. Pour le faire partir à une valeur différente, il faut faire précéder le RND de

RANDOMIZE <expression numérique>

Nous prendrons une expression numérique fournissant un nombre point de départ différent à chaque lancement une expression obtenue en extrayant les secondes de la variable TIME\$: nous verrons comment justement dans le cours de la présente leçon.

Corrigé de l'exercice R7

Un peu de géographie

Si vous voulez rédiger un programme complet, voici la liste des Etats de l'Amérique du Sud avec leurs capitales respectives.

Argentine : **Buenos Aires** - Brésil : **Brasilia** - Pérou : **Lima** - Chili : **Santiago** - Uruguay : **Montevideo** - Bolivie : **La Paz** - Equateur : **Quito** - Colombie : **Bogota** - Venezuela : **Caracas** - Paraguay : **Asuncion** - Surinam : **Paramaribo** - Guyana : **Georgetown** - Guyane Française : **Cayenne**.

Comme notre but n'est pas de vous enseigner la géographid, nous nous contenterons d'utiliser les 5 premiers de ces pays.

Jouons le jeu

Les «fortiches» résoudre l'exercice R7 avec des READ et des DATA. On pourrait aussi utiliser les variables Tableaux. Ce n'est pas du jeu ! Il faut résoudre le problème uniquement avec les notions que nous avons étudiées.

Une première solution

Le programme CAPITA 01 (figure 1 page ci-contre) vous fournit une première solution possible de R7.

L'idée est la suivante :

- A la ligne 70, l'ordinateur choisit au hasard un chiffre compris entre 0 et 4
- Dans les lignes 80 à 170, il fait correspondre à ce chiffre une chaîne question (Q\$) et sa réponse correcte (R\$)
- A la ligne 180, il pose une question à l'utilisateur
- Et teste si cette réponse est correcte à la ligne 200.

Un peu plus condensé

Le programme CAPITA 02 (figure 2 page ci-contre) est un peu plus condensé et un peu plus élégant.

Il a surtout pour objet de montrer l'intérêt qu'il y a parfois à mettre plusieurs instructions sur une ligne commençant par un IF.

Si, par hasard A est égal à 3, l'ordinateur en commençant à lire la ligne 70 découvre que ce n'est pas la peine d'aller plus loin, il ne lira pas R\$ = «BUENOS AIRES».

L'instruction 80 du programme précédent est fortement simplifiée.

Même «astuce» à la ligne 140 : si E\$ = R\$, l'ordinateur ne lira pas le GOTO 120.

Chacun ses petites manies pour s'organiser. Personnellement je conserve un listing de tous mes programmes, et j'indique au début le moyen de retrouver ce programme dans ma collection de disquettes. Ici le nom du programme est CAPITA 01 et je l'ai enregistré sur la deuxième disquette de mon Toshiba PAP.

```

10 REM PROGRAMME CAPITA01
20 REM ENREGRISTRE SUR CP-DK02-PAP
30 CLS
40 N$ = TIME$
50 V$ = RIGHT$(N$,2)
60 RANDOMIZE VAL(V$)
70 A = INT(RND*5)
80 IF A = 0 THEN Q$ = "ARGENTINE"
90 IF A = 0 THEN R$ = "BUENOS AIRES"
100 IF A = 1 THEN Q$ = "BRESIL"
110 IF A = 1 THEN R$ = "BRASILIA"
120 IF A = 2 THEN Q$ = "PEROU"
130 IF A = 2 THEN R$ = "LIMA"
140 IF A = 3 THEN Q$ = "CHILI"
150 IF A = 3 THEN R$ = "SANTIAGO"
160 IF A = 4 THEN Q$ = "URUGUAY"
170 IF A = 4 THEN R$ = "MONTEVIDEO"
180 PRINT "QUELLE EST LA CAPITALE DE "; Q$ ;
190 INPUT E$
200 IF E$ = R$ THEN 230
210 PRINT "NON - RECOMMENCEZ "
220 GOTO 80
230 PRINT "C'EST BIEN"
240 END

```

Pour faire commencer le tirage au hasard à un début différent à chaque RUN. Voir plus loin.

Donne à A un chiffre au hasard compris entre 0 et 4

Q\$ = la question
R\$ = la réponse exacte
E\$ = la réponse de l'élève

Le test

Fig. 1

```

10 REM PROGRAMME CAPITA02
20 REM ENREGRISTRE SUR CP-DK02-PAP
30 CLS
40 V$ = RIGHT$(TIME$,2)
50 RANDOMIZE VAL(V$)
60 A = INT(RND*5)
70 IF A = 0 THEN Q$ = "ARGENTINE" : R$ = "BUENOS AIRES"
80 IF A = 1 THEN Q$ = "BRESIL" : R$ = "BRASILIA"
90 IF A = 2 THEN Q$ = "PEROU" : R$ = "LIMA"
100 IF A = 3 THEN Q$ = "CHILI" : R$ = "SANTIAGO"
110 IF A = 4 THEN Q$ = "URUGUAY" : R$ = "MONTEVIDEO"
120 PRINT "QUELLE EST LA CAPITALE DE "; Q$ ;
130 INPUT E$
140 IF E$ <> R$ THEN PRINT "NON - RECOMMENCEZ " : GOTO 120
150 PRINT "C'EST BIEN"
160 END

```

Presque pareil

L'emploi du signe <> est commode ici

Remarquez ce GOTO comme deuxième instruction sur une ligne : c'est un procédé très employé et très commode.

Fig. 2

Corrigé de R8

Monsieur René SIPRA de 09500 (Mirepoix) m'a envoyé un corrigé de l'exercice R8 bien mieux rédigé que ce que j'avais préparé. Voici sa rédaction «in extenso».
(Dankon ! Samideano)

La Fonction «dent de scie»

Mon but est de montrer qu'un tel problème peut être abordé à différents niveaux, et qu'il ne faut pas nécessairement être un bon «matheux» pour le résoudre.

Mais attention, il y en aura pour tous les goûts !

Avant de donner une méthode de calcul de la valeur de $f(X)$, voici les lignes communes de tous les programmes : ce sont les instructions d'entrée des données et d'impression du résultat :

```
10 INPUT «PERIODE» ; T
20 INPUT «AMPLITUDE» ; A
30 INPUT «ABSCISSE» ; X
100 PRINT «ORDONNEE» = ";Y
110 END
```

Entre les lignes 30 et 100, nous intercalerons différentes instructions correspondant au niveau d'abstraction choisi. Ceci dit, les «forenmaths» peuvent glisser jusqu'au dernier paragraphe (mais ce serait dommage).

Niveau 1 (Pas de niveau 0 pour ménager les susceptibilités)

a. Notion de périodicité (figure 1)

Si nous traçons une droite horizontale d'ordonnée Y_0 (avec $0 < Y_0 < A$), elle coupe le graphique de la fonction (les «dents de la scie») en plusieurs points. Intéressons-nous aux points situés sur la partie «montante» (on dit «croissante») de la courbe, soit M'_1, M_0, M_1, M_2 , etc...

Observant la «régularité» des dents de la scie, nous constatons que les distances séparant deux points consécutifs de la suite M'_1, M_0, M_1, M_2 , etc... sont toutes égales à la «longueur» d'une dent, soit :

$$T = M'_1 M_0 = M_0 M_1 = M_1 M_2 = \dots$$

Nous appelons «abscisse de M_0 » le nombre X_0 qui mesure la distance séparant M_0 de l'axe vertical, distance comptée positivement si le point est à droite de l'axe vertical, négativement si le point est à gauche. Sur la figure 1, X_0, X_1 et X_2 sont positifs tandis que X'_1 est négatif.

Ainsi, nous pouvons écrire $X_1 = X_0 + T$, $X_2 = X_1 + T$ et $X_0 = X'_1 + T$.

Pour connaître la valeur de l'ordonnée du point M_2 , nous pourrions chercher tout aussi bien celle de M_1 ou de M_0 .

Première conclusion : Quelle que soit la valeur de X , il suffira que je trouve l'abscisse correspondante X_0 par additions ou soustractions successives de T (on ramène le calcul à la première dent). Ainsi, si $X < 0$ et $X > T$ il faudra enlever à X la valeur T , autant de fois que cela sera possible, tandis que si $X < 0$, il faudra lui ajouter T autant de fois que cela sera possible.

Le résultat final sera un nombre compris entre 0 et T .

Complétons le programme

```
40 IF X<0 THEN 70 il faut ajouter T
50 IF X<T THEN 80 Si  $0 \leq X < T$ , j'ai gagné !
60 X = X - T : GOTO 50 Si  $X \geq T$  il faut enlever T
70 X = X + T : GOTO 40 Si  $X < 0$  on ajoute T et on recommence !
Patience ! La ligne 80 annoncée arrive !
```

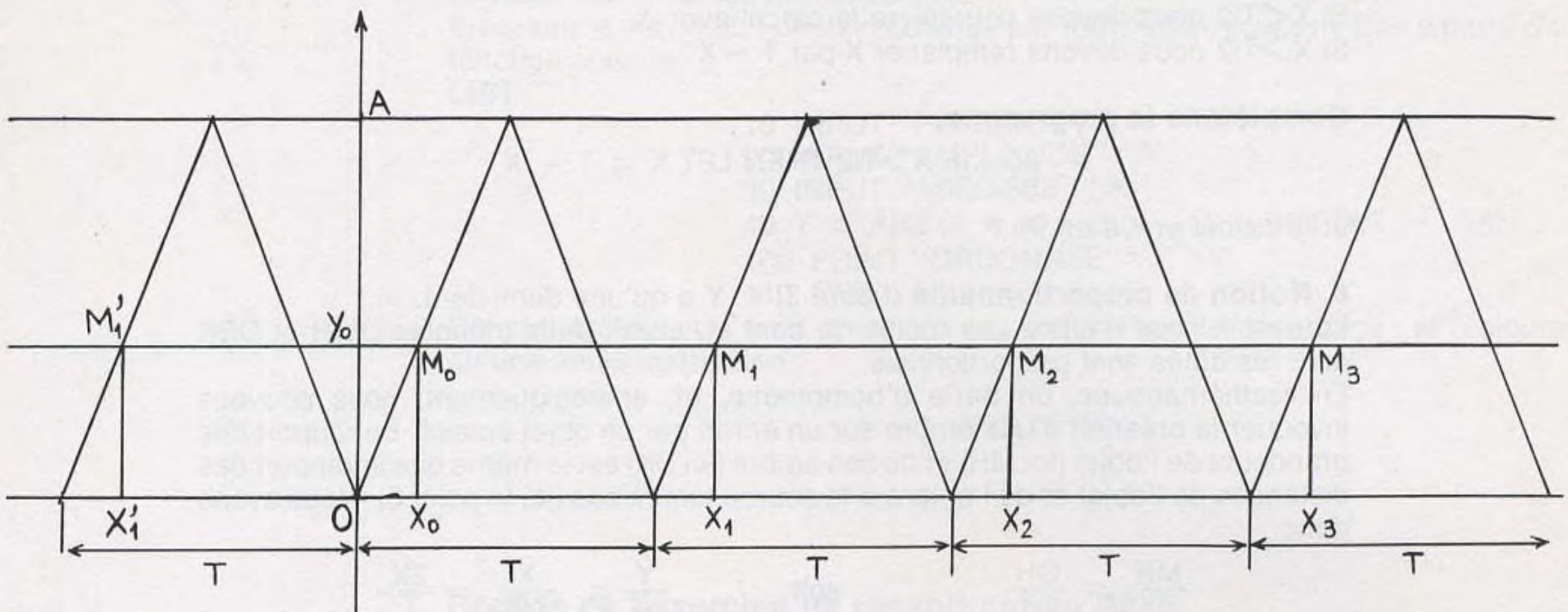



Figure 1

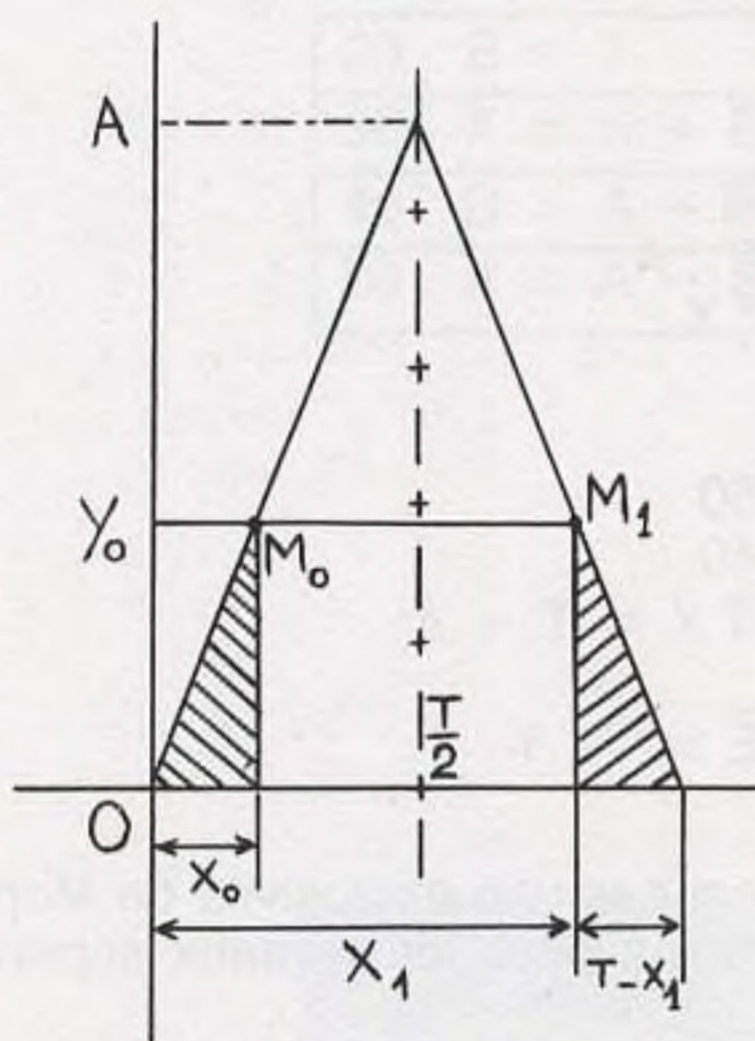


Figure 2

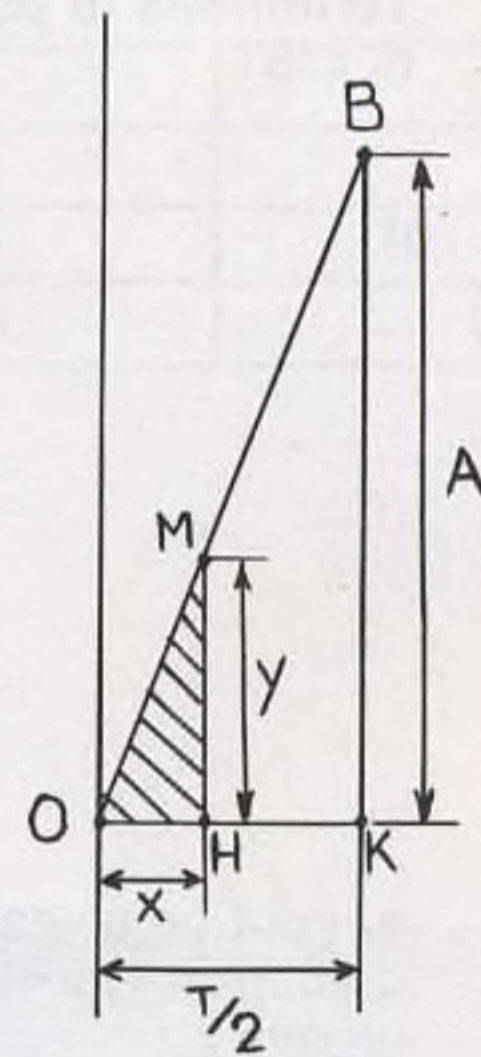


Figure 3

b. Notion de symétrie (figure 2) «...Y a qu'une dent...»

Il y a une partie montante et une partie descendante. Celles-ci sont symétriques par rapport à l'axe vertical d'abscisse $T/2$ (si on plie la dent par rapport à l'axe de symétrie, les deux parties se chevauchent parfaitement).

Prenons une horizontale d'ordonnée Y_0 qui détermine sur la dent deux points M_0 et M_1 . Pour que le calcul de l'ordonnée en fonction de l'abscisse soit le même, nous prendrons les deux triangles hâchurés, qui sont égaux par symétrie.

Nous voyons une condition supplémentaire : M_0 est sur la partie montante ($X_0 < T/2$) et le triangle a pour base X_0 , tandis que M_1 est sur la partie descendante ($X_1 > T/2$) et le triangle a pour base $T - X_1$.

Deuxième conclusion

Si $X \leq T/2$ nous devons poursuivre le calcul avec X
Si $X > T/2$ nous devons remplacer X par $T - X$

Complétons le programme

```
80 IF X > T/2 THEN LET X = T - X
```

et le calcul arrive en 90 !

c. Notion de proportionnalité (figure 3) «...Y a qu'une demi-dent...»

Le dessin nous montre une moitié de dent et, aussi, deux triangles OMH et OBK dont les côtés sont proportionnels.

En mathématiques, on parle d'homothétie, et, analogiquement, nous pouvons invoquer la création d'une ombre sur un écran par un objet éclairé. Le rapport des grandeurs de l'objet (ici MH) et de son ombre (ici BK) est le même que le rapport des distances de l'objet et de l'ombre à la source lumineuse (ici le point O). Nous avons donc

$$\frac{MH}{BK} = \frac{OH}{OK} \quad \text{soit} \quad \frac{Y}{A} = \frac{X}{T/2} = \frac{2X}{T}$$

d'où nous déduisons la formule tant attendue $Y = \frac{2AX}{T}$

Terminons le programme

d. LIST

```
90 Y = 2 * A * X / T  
  
10 INPUT "PERIODE" ; T  
20 INPUT "AMPLITUDE" ; A  
30 INPUT "ABSCISSE" ; X  
40 IF X < 0 THEN 70  
50 IF X < T THEN 80  
60 X = X - T : GOTO 50  
70 X = X + T : GOTO 40  
80 IF X > T/2 THEN LET X = T - X  
90 Y = 2 * X * A / T  
100 PRINT "ORDONNEE = " ; Y  
110 END
```

e. Pour ce niveau 1, j'ai été très bavard (ce n'est pas une exclusivité de Monsieur Claude Polgar !) mais je le serai beaucoup moins pour les niveaux supérieurs, promis !

Niveau 2 Pour les connaisseurs

Les lignes 40 à 70 peuvent être remplacées par une seule ligne. Pour cela, il suffit de demander à l'ordinateur d'effectuer en une seule opération le transfert de X à l'intervalle $[0 ; T]$, donc d'enlever (ou d'ajouter) autant de fois que c'est possible la période. La fonction INT est là pour ça, car ce nombre de fois, c'est $\text{INT}(X/T)$.

D'où $40 \text{ X} = \text{X} - \text{T} * \text{INT}(\text{X}/\text{T})$
et on efface les lignes 50, 60 et 70.

Niveau 3 Pour les «booles»

Et si on n'utilisait pas l'instruction IF... THEN de la ligne 80 ? Les variables booléennes nous aident :

$$80 \text{ X} = \text{X} * (\text{X} \leq \text{T}/2) + (\text{T} - \text{X}) * (\text{X} > \text{T}/2)$$

Ainsi, si $X \leq T/2$ alors $X \leftarrow X$
et si $X > T/2$ alors $X \leftarrow T - X$.

O.K. ?

Niveau «X» Pour les professionnels

Programme succinct. Fonction obtenue par translation, puis symétrie à partir d'une fonction connue.

LIST

```
10 INPUT "PERIODE" ; T
20 INPUT "AMPLITUDE" ; A
30 INPUT "ABSCISSE" ; X
40 Y = ABS (2 * A * (X/T - 1 - INT (X/T - .5)))
100 PRINT "ORDONNEE = " ; Y
110 END
```

Comme promis, il y en a pour tous les goûts. Plus de bavardage ; je n'ajouterai qu'une seule instruction :

GOTO 110

Corrigé de l'exercice de récapitulation R9

Le programme solution proposé	Valeur de la variable A	Valeur de la variable B
10 A = 10	10	
20 B = 7	(10)	(7)
30 A = A + B	17 (= 10 + 7)	7
40 B = A - B	17	10 (= 7 - 7)
50 A = A - B	(7) (= 17 - 10)	(10)

ERRATUM DU COURS N° 14

Page 8, ligne 40 : après 1/A supprimer ;''

Page 14, ligne 70 : avant N\$ ajouter ;

Page 19, 1^{re} bulle à gauche : au lieu d'un ; il faut un :

Page 21, 1^{re} ligne : remplacer ELSE par THEN.

Mes excuses à tous !

A partir du numéro de janvier 1985, je pense que la majeure partie de ces petites erreurs irritantes seront supprimées : au lieu de photocomposer le texte des programmes, nous vous donnerons directement une photo de leurs listings.

3.16.9. Complément sur les organigrammes : notion de structure

A. Structure linéaire

La figure 2 (page ci-contre) représente une structure «toute bête» ne comportant aucune instruction de branchement (ni GOTO, ni IF... THEN...).

B. Structure sélection

C'est la structure que nous sommes en train d'étudier dans le présent chapitre 3.16. Exemple : figure 3 page ci-contre.

C. Structure boucle indéfinie

La figure 4 traduit un programme que nous avons déjà rédigé : un programme qui ne s'arrête jamais de boucler. C'est une mauvaise solution qui peut cependant être utile dans deux cas particuliers :

- Dans des programmes conversationnels où on pose toujours les mêmes demandes de données (exemple : Led-Micro n° 14 page 12 § 3.15.13)
- Lorsqu'on dispose d'un ordinateur ancien (et peu puissant) ne possédant pas l'instruction END (voir Led-Micro n° 7 page 18 § G3.10.7).

D. Structure normale de boucle

Il n'est pas normal de tourner indéfiniment dans une boucle.

En général :

- on entre dans une boucle
- on tourne dans cette boucle un certain nombre de fois
- puis on en sort.

Les figures 5 et 6 montrent comment, à l'aide d'une boucle, on peut afficher des nombres de 1 « 10, et ne pas aller plus loin que 10.

Figure 5 : boucle avec «test en queue» : **on reste** dans la boucle **tant que** la condition $l = 10$ n'a pas été atteinte.

Figure 6 : boucle avec «test en tête» : **on entre** dans la boucle de traitement **jusqu'à** ce que la condition $l = 11$ ait été atteinte.

Nous étudierons la structure de boucle avec beaucoup de détails dans le prochain chapitre 3.17. Les quelques notions que nous venons de vous donner sont suffisantes pour faire comprendre les quelques programmes de contrôle de vraisemblance du présent chapitre 3.16.

E. Autres structures

Pour compléter cette énumération des structures de programmes, citons les «sous-programmes» qui seront étudiés dans le chapitre 3.18.

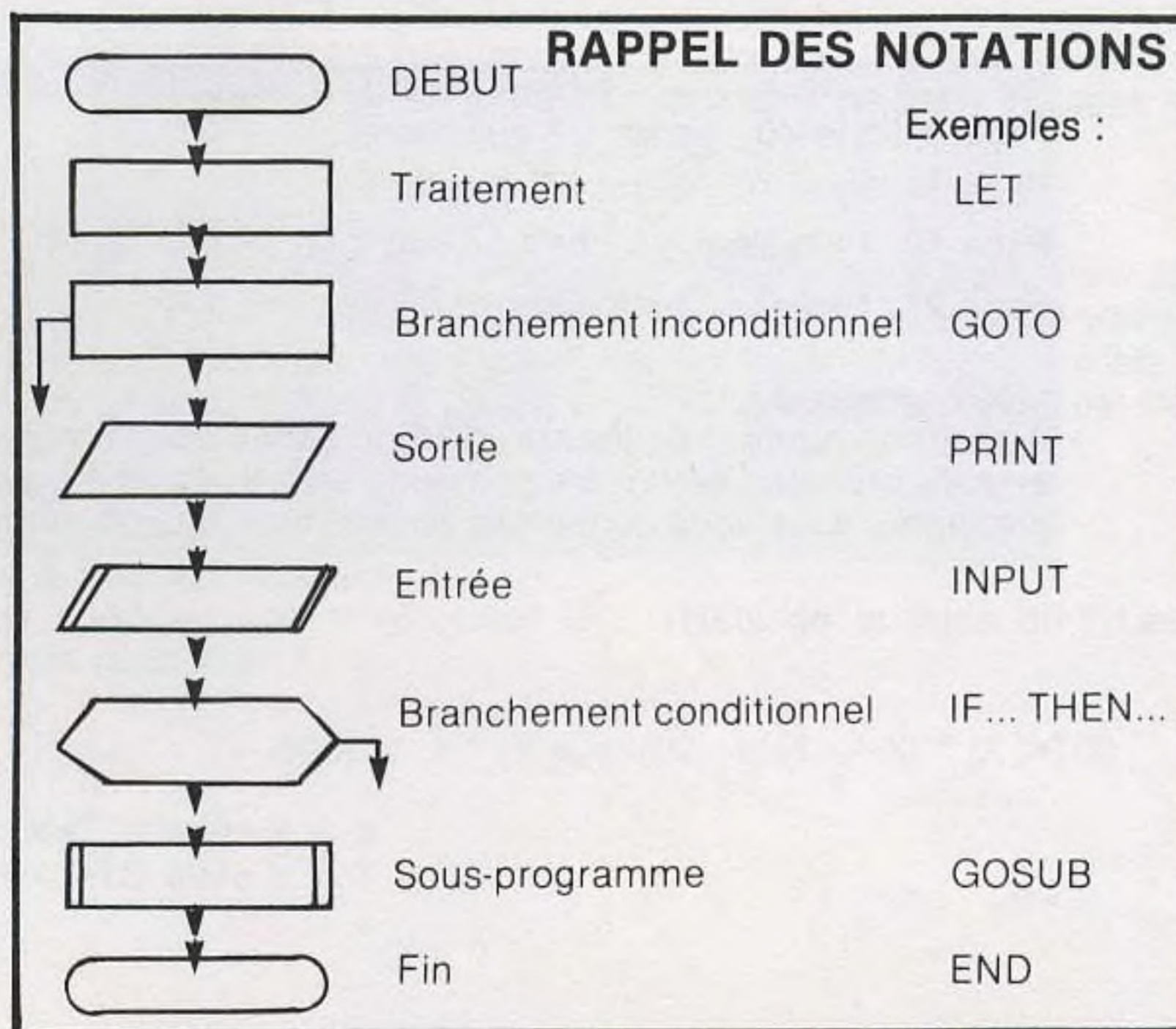
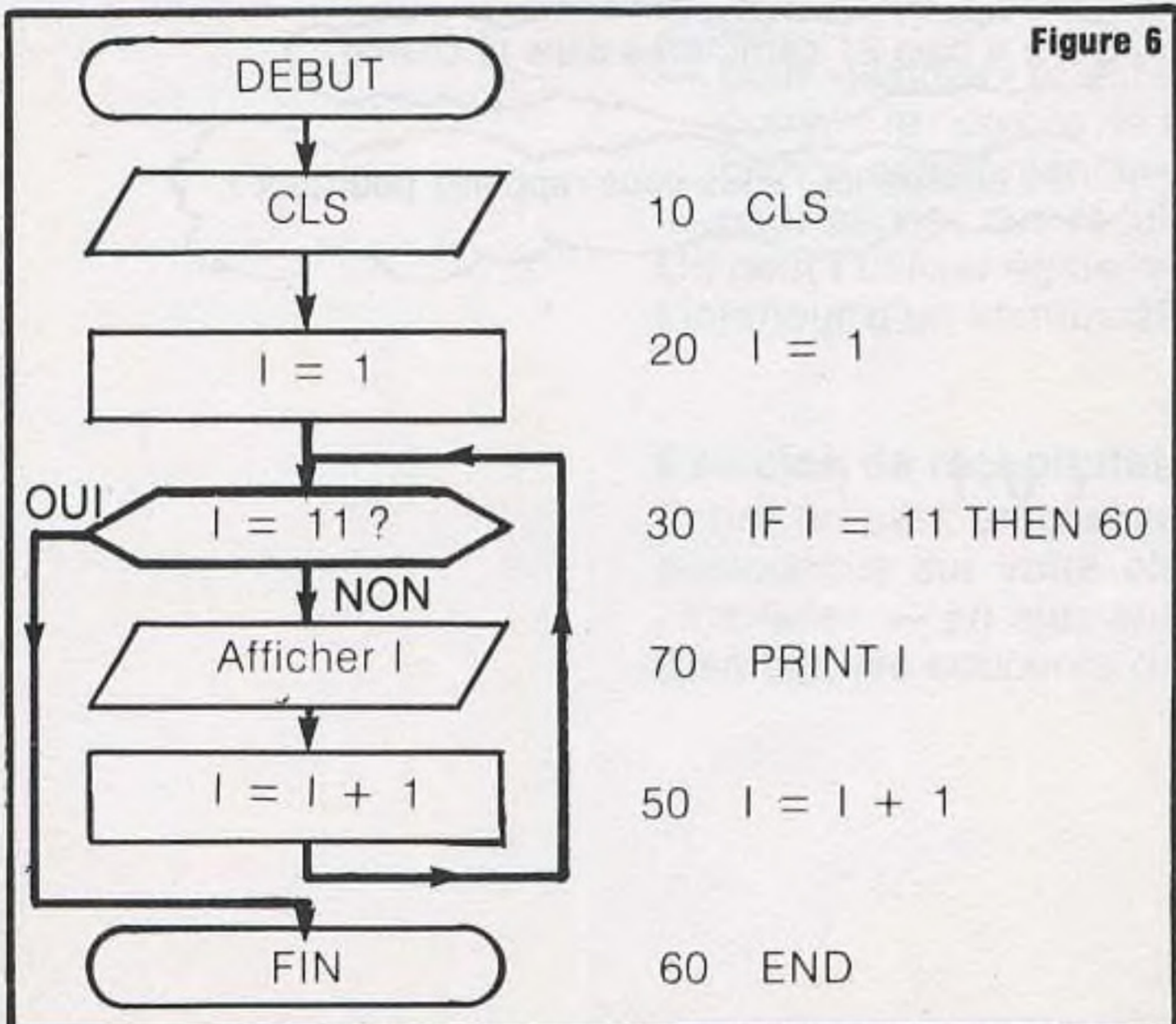
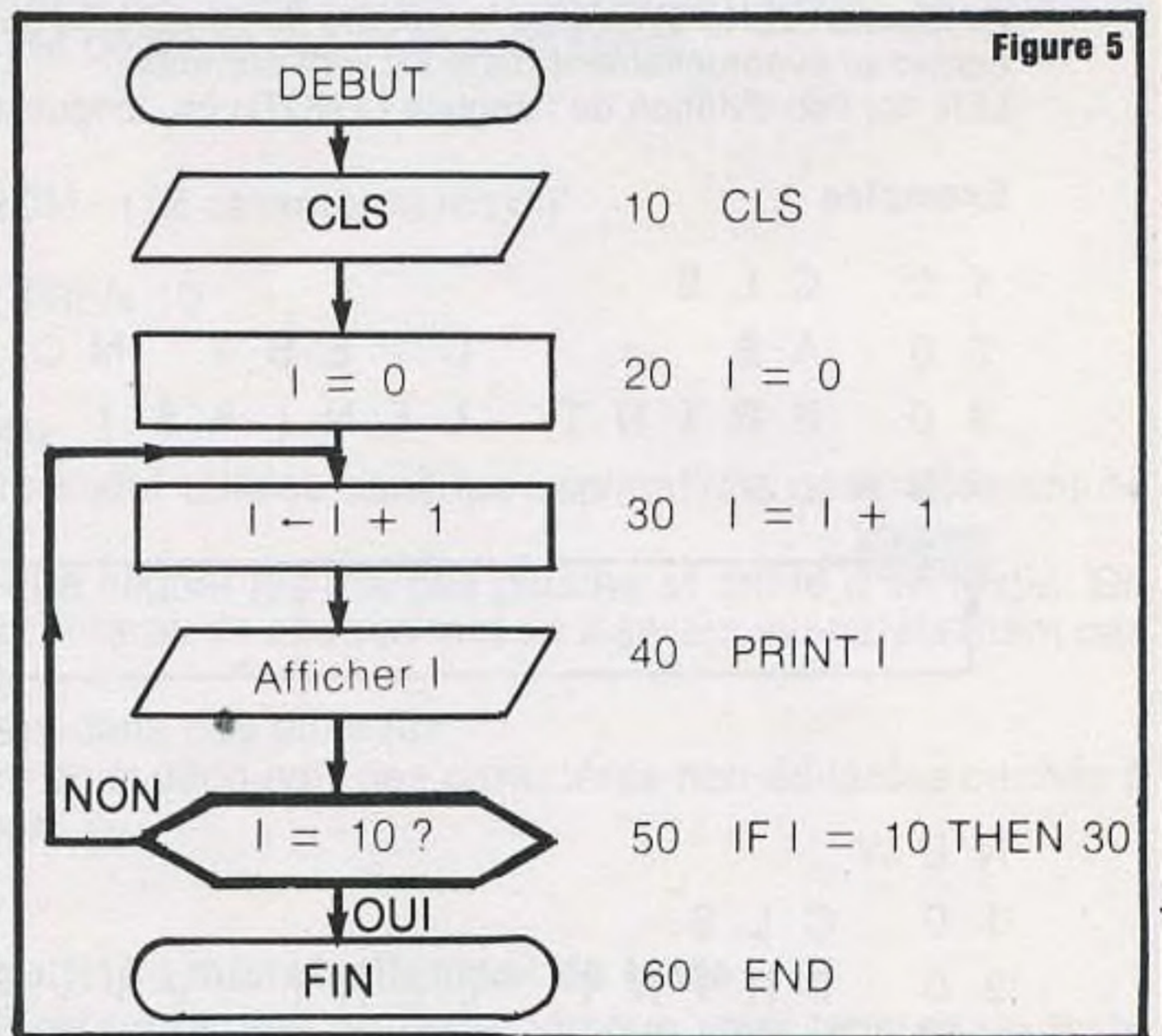
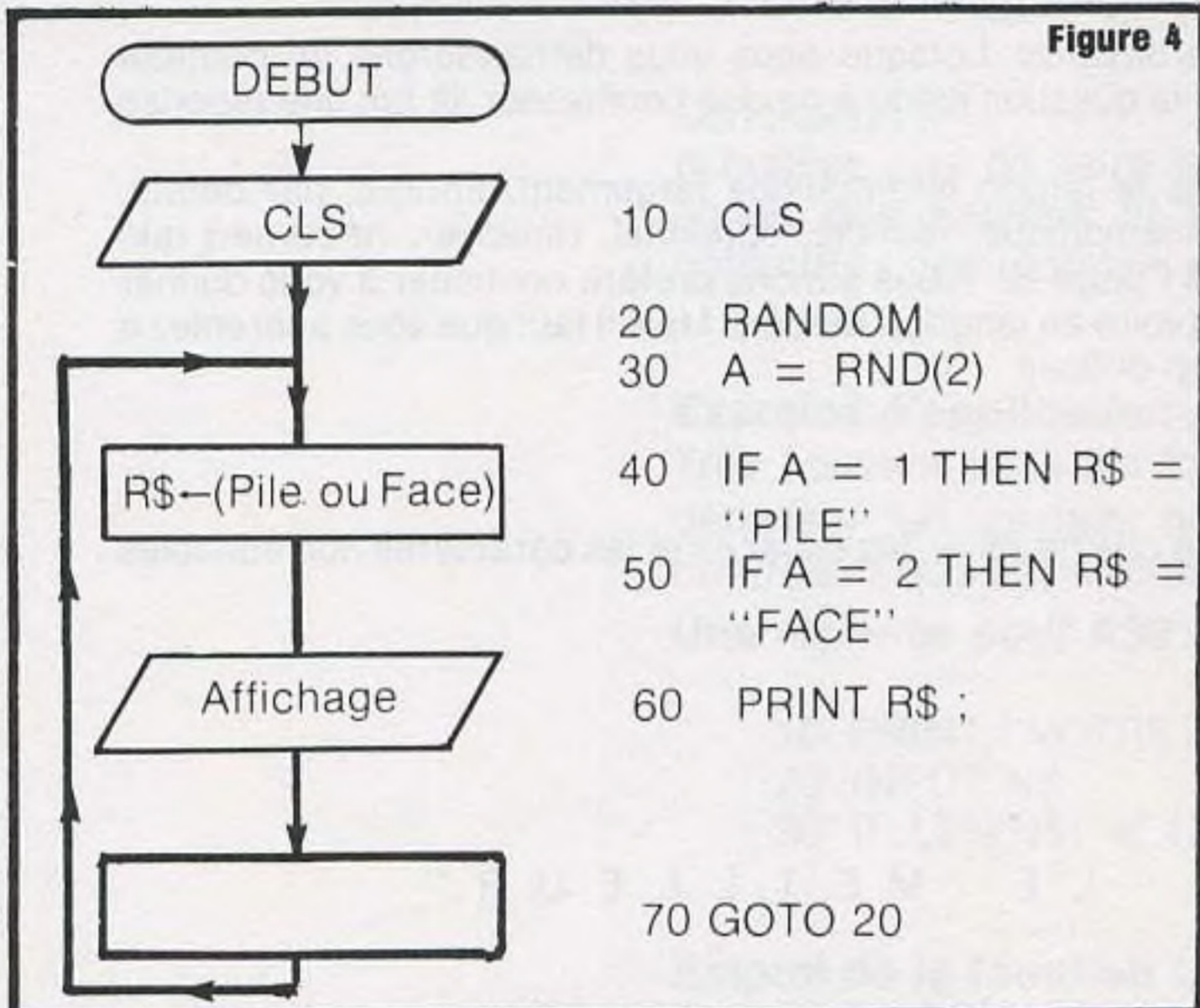
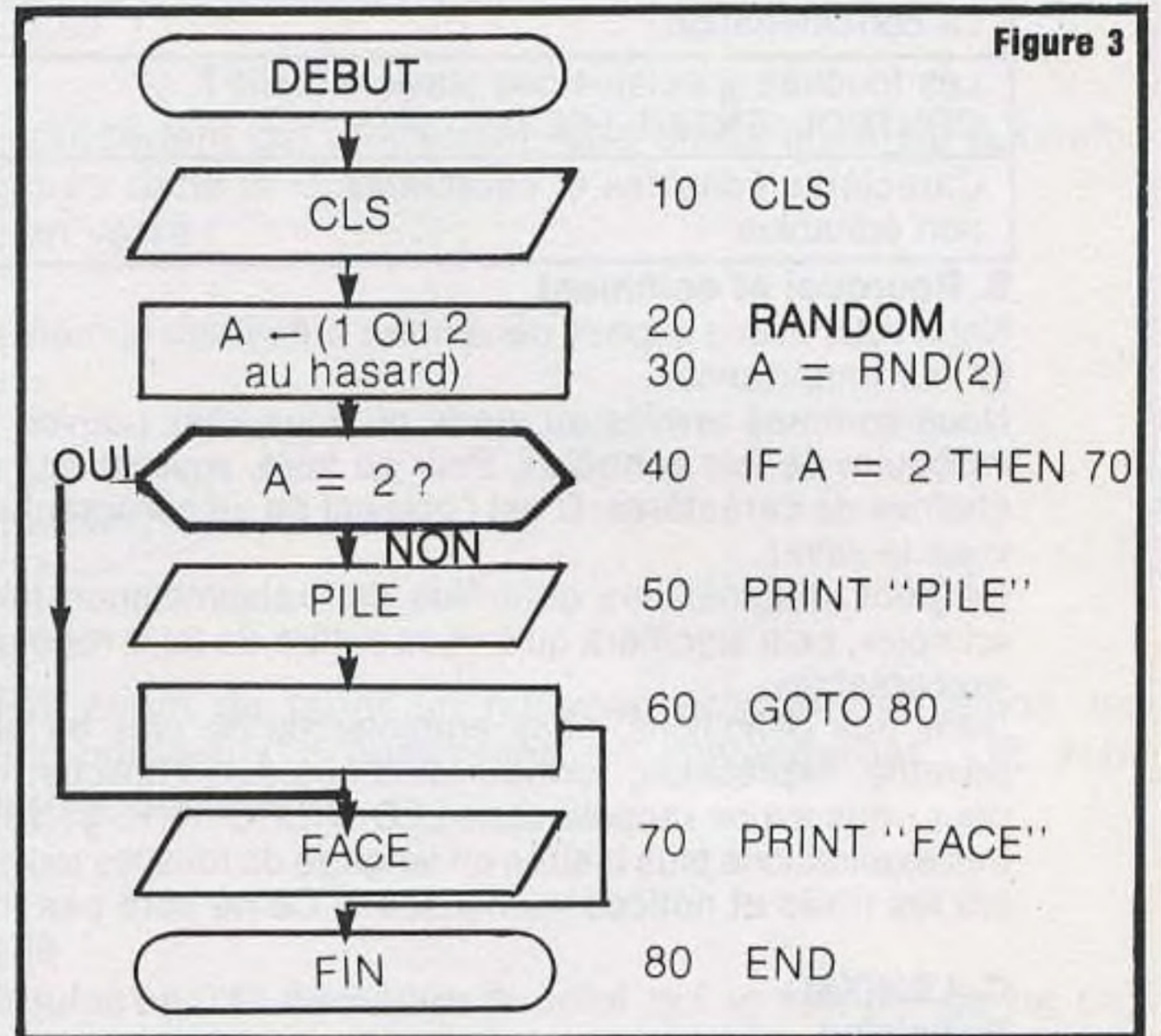
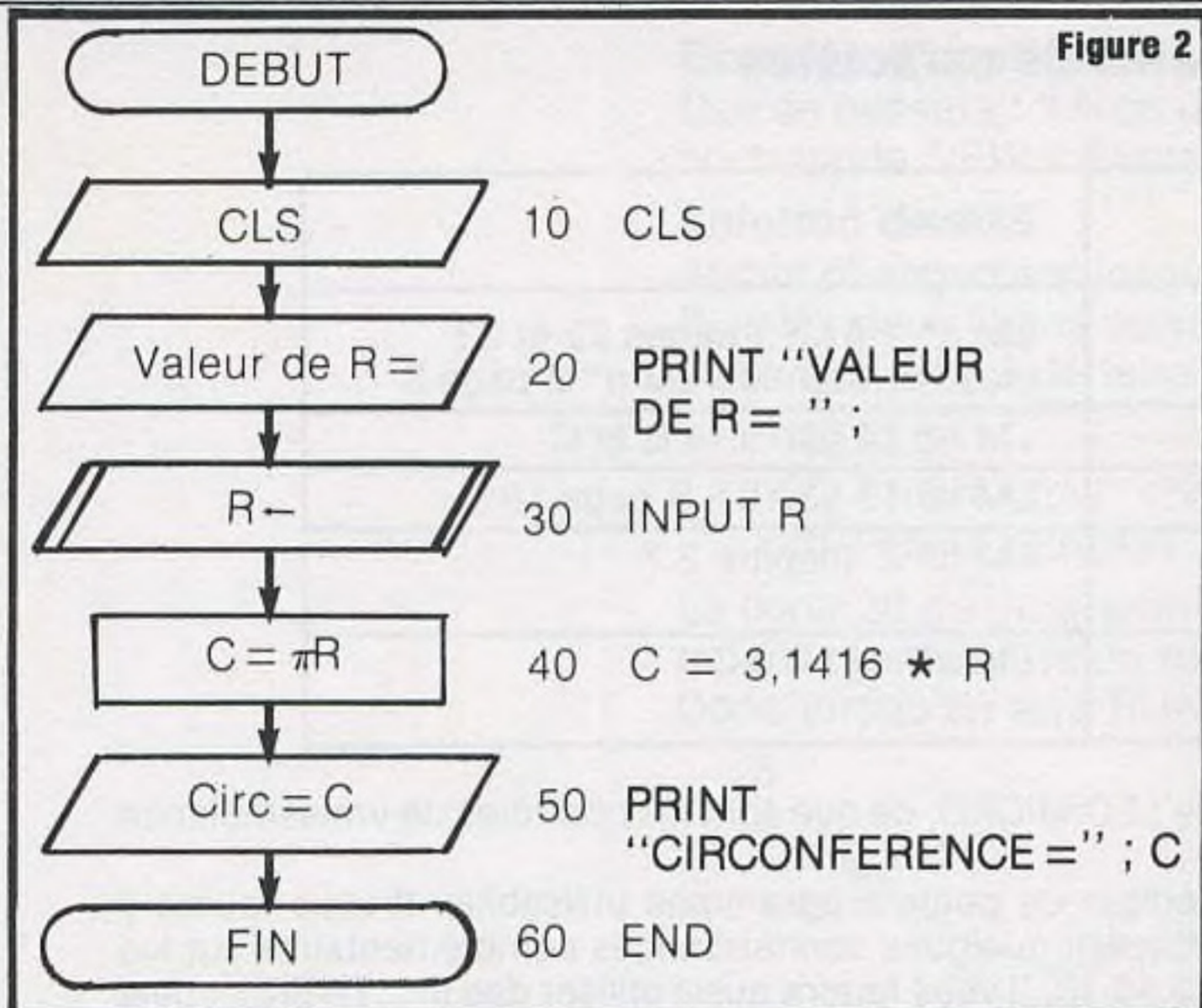


Figure 1



3.16.10. Compléments sur les fonctions chaîne de caractères

A. Rafrichissons nos connaissances

Si vous ne vous souvenez pas de ce qu'est :	Relisez :
Le code ASCII	LM n° 1 §1.3.3 pages 22 et 23 plus l'erratum de LM n° 2 page 8
Le contrôle de vraisemblance	LM n° 14 §3.15.14.B et C
La concaténation	LM n° 13 §3.15.6.B page 18
Les touches spéciales des claviers (SHIFT, CONTROL, ENTER, ESCAPE...)	LM n° 2 chapitre 2.7
Caractères éditables et caractères non éditables	LM n° 1 §1.3.3.C

B. Pourquoi et comment

Nous vous avons exposé dans notre précédent numéro de LED-MICRO, ce que sont les contrôles de vraisemblance et leur importance.

Nous sommes arrivés au stade où vous allez pouvoir rédiger de petits programmes utilisables. Il vous faudra y introduire de tels contrôles. Pour ce faire, vous devrez acquérir quelques connaissances complémentaires sur les chaînes de caractères. C'est l'objectif de ce paragraphe 3.16.10. Il vous faudra aussi utiliser des IF... THEN... : cela vous le savez.

On peut imaginer des contrôles de vraisemblance très bavards. Lorsque nous vous demanderons un contrôle «simple», cela signifiera qu'il vous suffira de faire répéter la question jusqu'à ce que l'opérateur ait fait une réponse «acceptable».

Dans nos définitions, nous emploierons de plus en plus le jargon informatique (argument, attribut, par défaut, éditable, expression, format, identificateur, implicite, mnémonique, mot-clé, optionnel, renvoyer, retourner) que nous vous avons rappelé dans LED-MICRO n° 13 §3.15.11 page 29. Nous aurions préféré continuer à vous donner des explications plus claires en langage de tous les jours (voire en langage «bébé»). Mais il faut que vous appreniez à lire les livres et notices «sérieuses» ! Ce ne sera pas trop difficile.

C. LEN(X\$)

Définition

La fonction LEN(X\$) renvoie le nombre de caractères de la chaîne X\$ — les espaces et les caractères non éditables contenus éventuellement dans X\$ sont comptés.

LEN est l'abréviation de l'anglais LENGTH (= longueur).

Exemples

```
1 0  C L S
2 0  A $ = " C ' E S T   M O I   L E   M E I L L E U R "
3 0  P R I N T   L E N ( A $ )
R U N
```

2 1

Il y a bien 21 caractères dans la chaîne

Un espace ici ; vous vous rappelez pourquoi ?

N E W

```
1 0  C L S
2 0  P R I N T   L E N ( " V I V E   L U I " )
R U N
```

8

Exercice d'application A38

Que se passerait-il si on tapait successivement les lignes rédigées ci-dessus **sauf** la commande NEW ? Essayez de répondre avant de lire la solution ci-dessous.

Solution de A38

Aucun changement jusqu'à la cinquième ligne du texte ci-dessus (affichage de 21). Puis les deux lignes suivantes viendraient écraser les lignes 10 à 20 existantes de sorte que, avant de faire RUN, on aurait en mémoire centrale le programme :

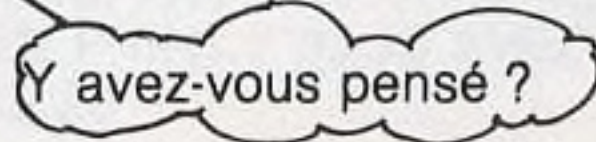
```
10 CLS
20 PRINT LEN("VIVE LUI")
30 PRINT LEN(A$)
```

La ligne 20 du programme précédent qui définissait A\$ a disparu... mais la chaîne qu'elle contenait reste toujours dans la «case» A\$.

Donc lorsqu'on fera RUN, on verra :

8

2 1



Y avez-vous pensé ?

Conclusion :

N'oubliez pas de faire NEW avant de taper un nouveau programme. Sinon, les lignes des anciens et des nouveaux programmes se mélangeront... et vous obtiendrez des résultats parfois curieux.

Exercice d'application A39

Très souvent dans les formulaires, on demande à celui qui le remplit de ne pas dépasser un certain nombre de caractères. Rédigez un (petit) programme contrôlant que l'opérateur ne dépasse pas le code autorisé.

Une réponse pour A39

```
10 PRINT "VOTRE NOM - (12 caractères maxi.)"
20 INPUT N$
30 IF LEN(N$) < 12 THEN 10
```

Emploi de la fonction LEN

La fonction LEN est extrêmement utilisée dans les applications de «traitement de texte» :

- pour «justifier» (c'est-à-dire aligner les marges gauche et droite d'un texte). On compte le nombre de caractères de chaque mot en insérant éventuellement des blancs supplémentaires ;
- pour aligner des résultats dans des tableaux.

On peut l'utiliser également pour découvrir des caractères non éditables cachés à l'intérieur d'un identificateur.

Exercice de récapitulation R10 : microtraitement de texte

Ecrire un petit programme de traitement de texte. lorsque vous taperez un texte quelconque sur votre clavier, l'ordinateur attendra que la ligne soit pleine pour l'imprimer — en ajustant — c'est-à-dire en insérant des espaces entre les mots. Bien sûr, les coupures de mots seront faites à la main.

D. VAL (X\$) et STR \$(X)

Nous avons déjà fait la connaissance de ces deux fonctions (LED-MICRO n° 13 -§3.15.7 - pages 20 et 21).

Exercice d'application n° A 40

Ecrire un programme qui :

1°) demande à l'opérateur quelle est sa date de naissance

2°) lui répond (par exemple) «Alors vous avez 20 ans» (en effectuant la soustraction 1984 - «année de naissance»).

3°) Mais avant d'effectuer ce calcul le programme prend un certain nombre de précautions :

— d'une part la réponse de l'utilisateur doit être reçue sous forme de chaîne de caractères (INPUT A\$) de façon à éviter les «plantages» en cas d'erreur de frappe ;

— d'autre part on suppose que l'utilisateur a entre 3 ans et 120 ans.

Si l'opérateur frappe une chaîne de caractères ou déclare une date de naissance invraisemblable, le système se contentera de lui reposer la même question indéfiniment...jusqu'à l'obtention d'une réponse acceptable.

Cherchez votre solution avant de regarder la réponse (page ci-contre).

Exercice d'application A41

Reprenez le programme que nous vous proposons comme solution de A40 et remplacez-y les quatre lignes 40, 50, 60 et 70 par une seule instruction.

Nous vous proposons une solution (page ci-contre). Cherchez avant de la lire. Remarquez ensuite que la solution précédente est bien plus lisible.

Exercice d'application A42

Que donnera l'exécution du petit programme ci-dessus?

10 P\$ = "132 FRANCS 50 CENTIMES".

20 PRINT VAL (P\$)

Solution page ci-contre.

Exercice d'application A43

Le compte d'un client comporte au début la somme de 150 francs rangée dans la chaîne de caractères F\$ sous la forme :

100 F\$ = "150 FRANCS"

Le programme demande à l'opérateur :

110 PRINT "ENTREZ LE VERSEMENT"

Pour éviter des erreurs, on oblige l'opérateur à entrer ce versement suivant la même syntaxe. Il tapera donc quelque chose comme :

20 FRANCS

Le texte décrivant ce versement sera entré dans une chaîne de caractères que l'on appellera V\$.

On ne cherchera pas à vérifier qu'il en est bien ainsi (pas de contrôle), mais on ajoutera ce montant V\$ au montant F\$.

Completez ce programme :

100 F\$ = "150 FRANCS"

110 PRINT "ENTREZ LE VERSEMENT"

120 INPUT V\$

200 PRINT F\$

de façon à ce que si l'opérateur a entré 20 FRANCS l'exécuteur de la ligne 200 affiche

170 FRANCS

Une solution page ci-contre

Une solution de l'exercice d'application A40

```
10 CLS
20 PRINT "QUELLE ANNEE ETES-VOUS NE ?"
30 INPUT AN$
40 AN = VAL(AN$)
50 AGE = 1984 - AN
60 IF (AGE < 3) OR (AGE > 110) THEN 20
70 PRINT "VOUS AVEZ" ; AGE ; " ANS"
80 END
```

Bien sûr, on pourrait remplacer les deux lignes 40 et 50 par une seule ligne :
AGE = 1984 - VAL(AN\$)

Une solution de l'exercice d'application A42

```
40 IF (VAL(AN$) < 3) OR (VAL(AN$) - 1984) > 110
   THEN PRINT "VOUS AVEZ" ; VAL(AN$) - 1984 ; " ANS"
   ELSE 20
```

Solution de l'exercice d'application A42

On verra apparaître seulement

132

Si vous aviez répondu autre chose, relisez dans LED-MICRO n° 13 §3.15.7.C, les deux dernières lignes de la page 20.

Une solution de l'exercice d'application A43

```
100 F$ = "150 FRANCS"
110 PRINT "ENTREZ LE VERSEMENT"
120 INPUT V$
130 V = VAL(V$)
140 F = VAL(F$)
150 F = V + F
160 F$ = STR$(F) + " FRANCS"
200 PRINT F$
```

Exemple : V\$ = "20 FRANCS"

alors V = 20

et F = 150

d'où nouvelle valeur de F
F = 20 + 150 = 170

Ici le signe + représente
une addition

Ici le signe + représente
une concaténation

Pour concaténer 170 et FRANCS, il faut
transformer 170 en chaîne de caractères,
grâce à STR\$

E. ASC(X\$) et CHR\$(X)

Définitions

- CHR\$(X) renvoie le caractère dont le code ASCII est X.
- ASC(X\$) renvoie le code ASCII (exprimé en décimal) du premier caractère de la chaîne de caractères X\$.

Attention !

Le tableau de la page ci-contre vous fournit la correspondance

(code ASCII exprimé en décimal) ↔ (caractère).

Mais attention ! les constructeurs d'ordinateurs respectent cette correspondance de façon assez fantaisiste. Pourquoi ? Mettez-vous à leur place :

— Je conçois un ordinateur économique n'utilisant pas de minuscules. Je n'ai donc pas besoin des codes ASCII 97 à 121. Quelques mois plus tard, j'ai envie de compléter mon ordinateur en lui faisant faire du graphique basse résolution. Pourquoi ne pas utiliser les codes 97 à 121 qui sont disponibles ? Et voilà, je suis incompatible !

En général, le code ASCII est :

assez bien respecté pour les chiffres et les lettres majuscules

— très incertain pour les autres caractères (le é et le ç par exemple).

Le tableau ci-contre correspond au code utilisé par le PC d'IBM (notre «nouvelle» référence).

Exemples d'emploi en mode programme

```
10 A$ = " B "  
20 PRINT ASC ( A$ )  
RUN  
66
```

```
10 PRINT CHR$ ( 66 )  
RUN  
B
```

Exemples d'emploi en mode commande

```
PRINT ASCI I ( " B O N J O U R " )  
66
```

```
A = 40  
B = 12
```

```
PRINT CHR$ ( A + 2 * B )  
J
```

car 66 est le code ASCII de la première lettre de BONJOUR.

car J est le caractère de code ASCII 74

$20 + 2 * 12 = 74$

Le code ASCII du TRS80 (ou du PROF 301)

Nous avons vu (LED-MICRO n° 1 §1.3.9) que le code ASCII est un code à 7 bits (c'est-à-dire qu'il utilise 7 chiffres binaires). Comme un caractère est généralement rangé dans un octet, le 8^e bit est parfois gaspillé (mis systématiquement égal à 0), parfois utilisé pour effectuer des contrôles de parité (voir LED-MICRO n° 3 §2.9.2.C pages 14 et 15).

Mais il est aussi possible d'utiliser ce troisième bit pour «enrichir» la collection de caractères : au lieu de se limiter aux caractères de code ASCII 0 à 127, on dispose des 256 caractères de code ASCII 0 à 255.

Le TRS80 utilise ainsi les codes 129 à 191 pour définir des «caractères semi-graphiques» obtenus en divisant le «pavé d'un caractère» en six parties (voir figure 2 page ci-contre).

Le code ASCII du Commodore 64

Le Commodore 64 utilise un code ASCII tout à fait particulier : pour les chiffres et les lettres majuscules, il utilise les codes classiques.

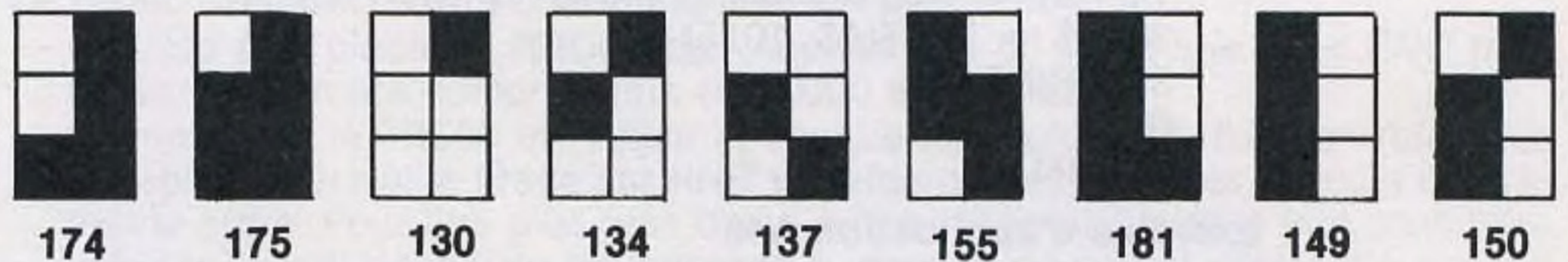
Pour les signes, on trouve des analogies (exemple le code 94 signifie → sur le TRS80 et ↑ sur le Commodore : on remarque un petit air de famille).

Mais pour les caractères, le Commodore utilise des conventions originales : voir des exemples figure 3 page ci-contre.

Le code ASCII de l'IBM-PC

ASCII value	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character	ASCII value	Character	ASCII value	Character	ASCII value	Character		
000	NUL	032	(space)	064	@	096	·	128	Ç	160	á	192	ˆ	224	α
001	SOH	033	!	065	A	097	a	129	ü	161	í	193	±	225	β
002	STX	034	"	066	B	098	b	130	é	162	ó	194	⊕	226	Γ
003	ETX	035	#	067	C	099	c	131	â	163	ú	195	⊖	227	κ
004	EOT	036	\$	068	D	100	d	132	ä	164	ñ	196	—	228	Σ
005	ENQ	037	%	069	E	101	e	133	å	165	ˆ	197	+	229	σ
006	ACK	038	&	070	F	102	f	134	à	166	æ	198	⊕	230	μ
007	BEL	039	'	071	G	103	g	135	ç	167	ø	199	⊖	231	τ
008	BS	040	(072	H	104	h	136	ê	168	ˆ	200	⊕	232	Φ
009	HT	041)	073	I	105	i	137	ë	169	ˆ	201	⊖	233	⊕
010	LF	042	*	074	J	106	j	138	è	170	ˆ	202	⊕	234	Ω
011	VT	043	+	075	K	107	k	139	ı	171	½	203	⊖	235	δ
012	FF	044	,	076	L	108	l	140	ı	172	¾	204	⊕	236	∞
013	CR	045	-	077	M	109	m	141	ı	173	ı	205	⊖	237	∅
014	SO	046	.	078	N	110	n	142	Ä	174	«	206	⊕	238	€
015	SI	047	/	079	O	111	o	143	Å	175	»	207	⊖	239	∩
016	DLE	048	0	080	P	112	p	144	É	176	ˆ	208	⊕	240	≡
017	DC1	049	1	081	Q	113	q	145	Æ	177	ˆ	209	⊖	241	±
018	DC2	050	2	082	R	114	r	146	⊕	178	ˆ	210	⊕	242	≥
019	DC3	051	3	083	S	115	s	147	ö	179	ˆ	211	⊖	243	≤
020	DC4	052	4	084	T	116	t	148	ø	180	ˆ	212	⊕	244	Γ
021	NAK	053	5	085	U	117	u	149	ð	181	ˆ	213	⊖	245	J
022	SYN	054	6	086	V	118	v	150	û	182	ˆ	214	⊕	246	+
023	ETB	055	7	087	W	119	w	151	ü	183	ˆ	215	⊕	247	∞
024	CAN	056	8	088	X	120	x	152	ÿ	184	ˆ	216	⊕	248	•
025	EM	057	9	089	Y	121	y	153	Ö	185	ˆ	217	⊖	249	•
026	SUB	058	:	090	Z	122	z	154	Ü	186	ˆ	218	⊕	250	•
027	ESC	059	;	091	[123	{	155	ç	187	ˆ	219	⊖	251	√
028	FS	060	<	092	\	124		156	É	188	ˆ	220	⊕	252	n
029	GS	061	=	093]	125	}	157	Ë	189	ˆ	221	⊖	253	z
030	RS	062	>	094	^	126	~	158	Ï	190	ˆ	222	⊕	254	■
031	US	063	?	095	_	127	␣	159	ƒ	191	ˆ	223	⊖	255	(blank 'FF')

Les caractères semi-graphiques du TRS80



Code ASCII

Nous avons vu dès le numéro 2 de LED-MICRO que le fait d'appuyer sur une touche du clavier (par exemple sur la touche A) a pour effet d'envoyer dans les circuits de l'ordinateur le nombre binaire 0100 0001, dont l'équivalent décimal est 65.

C'est ainsi que, au lieu d'écrire : PRINT "A", on pourra écrire PRINT CHR\$(65). N'est-ce pas être Shadock que d'utiliser ainsi le CHR\$? (Vous vous rappelez : les Shadocks sont des bêtes qui aiment la complication : pourquoi se casser la tête à faire simple alors qu'il est plus facile de faire compliqué ?).

Certainement ! Mais parfois on ne peut pas faire autrement. Exemple : Pour effacer l'écran :

— beaucoup d'ordinateurs utilisent l'instruction CLS (IBM-PC, TRS80, Sinclair...)

— l'Apple II utilise l'instruction HOME.

Mais certains BASICs ne connaissent pas cette instruction. Que faire !

Consulter les codes de commande de l'écran de votre ordinateur. Si, par exemple, le code ASCII 28 correspond à «Effacement de l'écran», en écrivant :

```
PRINT CHR$(28)
```

vous vous êtes fabriqué votre CLS.

Ce procédé est très commode pour entrer dans un programme des fonctions réalisées par des touches commandant des caractères «non éditables». Par exemple, les codes des touches de mouvement du curseur.

Dans le courrier des lecteurs, vous trouverez une application particulièrement intéressante de ce PRINT CHR\$(X). Merci à son auteur Melle Catherine B de Lyon.

F. LEFT\$, RIGHT\$ et MID\$

Un peu d'anglais

Left = gauche Right = droite Middle = milieu

LEFT\$

Définition :

LEFT\$(X\$, n) renvoie les n premiers caractères de la chaîne X\$.

Exemple :

```
10 A$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
20 B$ = LEFT$(A$, 5)
30 PRINT B$
RUN
ABCDE
```

RIGHT\$

Définition :

RIGHT\$(X\$, n) renvoie les n derniers caractères de la chaîne X\$.

Exemple :

```
10 A$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
20 B$ = RIGHT$(A$, 6)
30 PRINT B$
RUN
UVWXYZ
```

MID\$

Définition

MID\$(X\$, p, n) renvoie les n caractères de X\$ comptés à partir de leur position P dans X\$.

Exemple :

```
10 A$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
20 B$ = MID$(A$, 10, 5)
30 PRINT B$
RUN
JKLMN
```

Exercice d'application A46

Le logiciel ALAMOD (qui permet de dessiner des patrons personnalisés et d'effectuer de la coupe en mesure industrielle) utilise pour définir les mensurations d'une cliente, une collection de sigles standardisés tels que :

TPO = tour de poitrine
TTA = tour de taille
ESE = écartements des seins
etc.

C'est-à-dire des sigles comportant uniquement trois lettres majuscules.

A un moment donné le programme demande à l'opérateur :

```
100 PRINT "ENTREZ LE SIGLE DE LA PREMIERE MENSURATION"
110 INPUT SIG$
```

Rédigez la suite d'instructions permettant de vérifier que le sigle indiqué par la personne chargée de noter cette mensuration contient bien une succession de trois lettres.

N'utilisez que les instructions et notions que vous possédez déjà. Bien sûr, vous pourriez trouver (facilement) une solution plus élégante en testant de la même façon chacun des trois caractères de ce sigle à l'aide d'une boucle : ne trichez pas. N'utilisez que ce que vous connaissez actuellement.

Testez successivement les trois caractères du sigle, mais pour tester le premier sigle, utilisez uniquement ASC comme fonction chaîne de caractères.

Nous vous proposons une solution page ci-contre. Ne trichez pas : cherchez la vôtre avant.

Une solution de l'exercice d'application A46

```
100 PRINT "ENTREZ LE SIGLE DE LA PREMIERE MENSURATION"  
110 INPUT SIG$  
120 IF LEN(SIG$) <> 3 THEN 100  
130 A1$ = LEFT$(SIG$, 1)  
140 IF (A1$ < 125) OR (A1$ > 95) THEN A1$ = A1$ + 32  
150 A2$ = MID$(SIG$, 2, 1)  
160 IF (A2$ < 125) OR (A2$ > 95) THEN A2$ = A2$ + 32  
170 A3$ = RIGHT$(SIG$, 1)  
180 IF (A3$ < 125) OR (A3$ > 95) THEN A3$ = A3$ + 32  
190 SIG$ = A1$ + A2$ + A3$
```

Commentaires

Ligne 130 : on extrait le premier caractère du sigle et on le baptise A1\$

Ligne 140 : si ce premier caractère est une lettre minuscule (c'est-à-dire un caractère dont le code ASCII est compris entre 95 et 125) on le transforme en une lettre majuscule : le tableau de la page 21 montre qu'il y a une différence de 32 entre le code ASCII d'une lettre minuscule et le code ASCII de la même lettre majuscule.

Ligne 150 : on extrait le deuxième caractère du sigle et on le baptise A2\$

Ligne 160 : on transforme ce caractère en majuscule s'il y a lieu

Ligne 170 : on extrait le troisième caractère du sigle et on le baptise A3\$

Ligne 180 : on transforme ce caractère en majuscule s'il y a lieu

Ligne 190 : le sigle SIG\$ ne contient maintenant que des majuscules.

Une application utile

Nous avons vu la façon dont l'Apple II et le TRS80 modèle III traitaient les fonctions aléatoires (LED-MICRO n° 8 § 3.11.6.E pages 30 et 31).

Le BASIC de l'IBM-PC a choisi une solution différente :

— Il utilise à la place du RND(1) de l'Applesoft et du TRS80, le sigle RND pour choisir un nombre compris entre 0,000000 et 0,999999.

— Comme pour le TRS80 et l'Apple II, chaque fois qu'on fait RUN, on remet les variables à zéro et le tirage par RND va refournir des nombres toujours dans le même ordre. Pour être plus près d'une «vraie» série aléatoire, il faut donc faire débiter l'envoi de la liste de nombres à un point de départ différent à chaque lancement.

Pour ce faire, le BASIC Microsoft de l'IBM-PC utilise l'instruction :

RANDOMIZE < expression >

L'expression qui suit le RANDOMIZE a pour effet de choisir le point de départ de la liste de nombres.

Une façon commode de déterminer cette expression consiste à utiliser la variable TIME\$ de l'IBM-PC.

TIME\$ fournit l'heure, la minute et la seconde sous la forme

hh : mm : ss

Il suffit de faire, par exemple :

```
100 A$ = RIGHT$(TIME$)
```

```
110 A = VAL(A$)
```

pour obtenir une expression A de valeur toujours différente (ou presque !) chaque fois qu'on fait RUN.

Vous en avez vu un exemple (à l'avance !) dans notre corrigé de l'exercice R7 (au début de cet article).

G. Exercice de récapitulation R11 : contrôle de validité d'une date

Énoncé initial

L'ordinateur demande à l'opérateur de lui indiquer la date :

```
10 PRINT "DONNEZ LA DATE"  
20 PRINT "sous la forme jj : mm : aa"  
30 INPUT DATE$
```

Puis l'ordinateur :

1°) vérifie que la date entrée est « acceptable »

- la « syntaxe » utilisée par l'opérateur est correcte ;
- la date est « vraisemblable » (mois inférieur à 32 jours, année comprise entre 1984 et 1989, inutile d'aller au-delà) ;

2°) affiche cette date en langage clair

« premier janvier dix neuf cent quatre vingt sept »

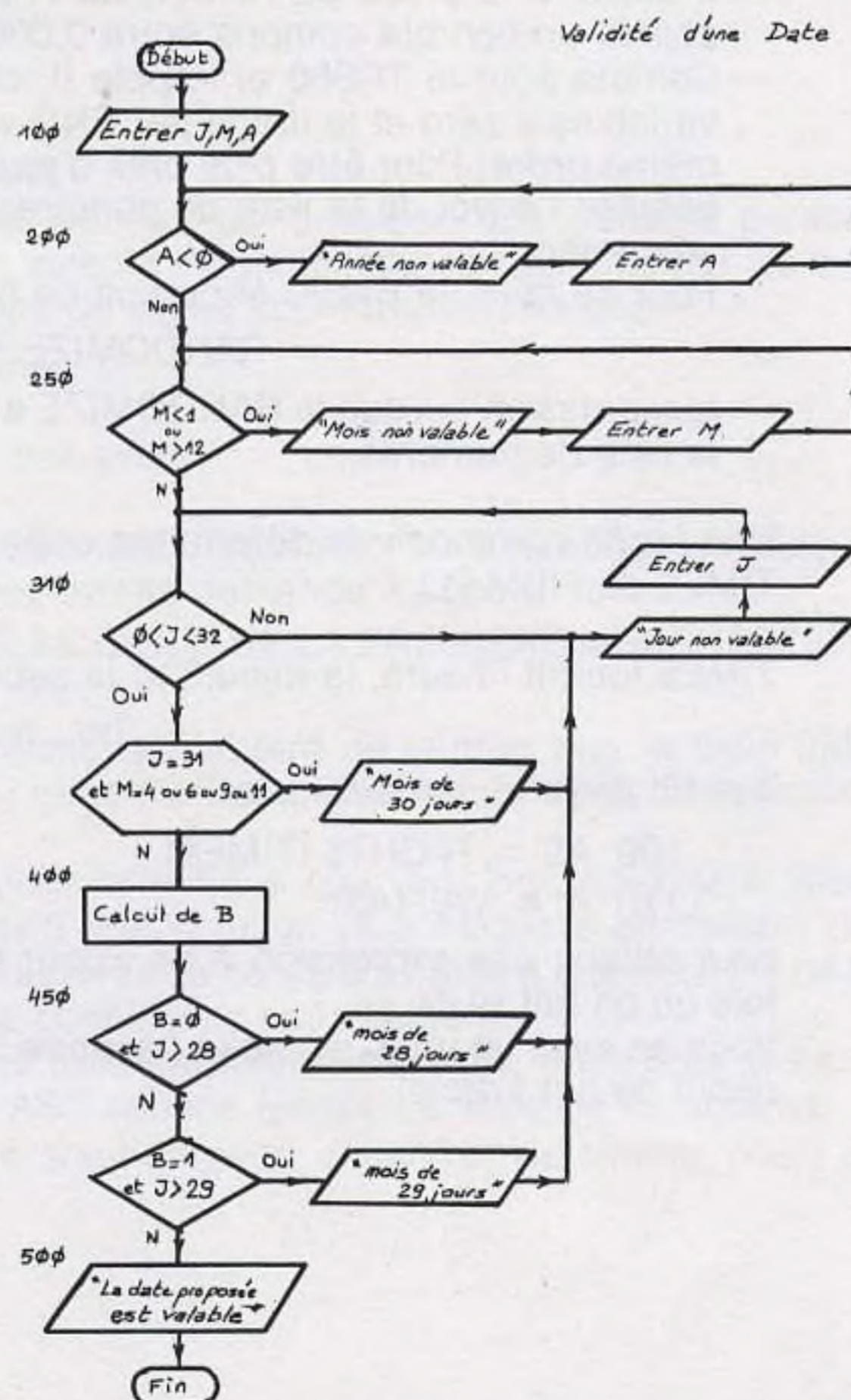
(par exemple).

L'effet des lecteurs

Monsieur G.H. de Palaiseau nous a envoyé une solution à cet exercice avant même que je publie son énoncé (voir le courrier des lecteurs). Le problème qu'il se pose n'est pas exactement le même, mais comme sa solution est intéressante (et, chose rare !), qu'il nous l'a envoyée sous une forme impeccable, nous la publions ci-dessous. Cherchez quand même un peu avant de la lire.

D'autre part, Mademoiselle Catherine B. (de Lyon) nous a appris à jongler avec les TAB(X) et les PRINT CHR\$(30) ; [N'oubliez pas le ;]. Alors utilisez ces connaissances toutes fraîches pour obtenir un effet de surprise : dès que l'opérateur a tapé une date acceptable, l'ordinateur efface cette date exprimée en chiffres et la remplace par la date exprimée en langage clair. Il efface également le ? donnée par l'INPUT.

Ne trichez pas : n'utilisez pas de GET A\$ ni de INKEY\$: vous n'êtes pas censés connaître ces instructions.




```

10 REM      *** VALIDITE D'UNE DATE ***
20 '
30 PRINT CHR$(12)
40 '
50 REM  Variables : J jour en chiffres
60 '                M mois en chiffres
70 '                A annee en chiffres
80 '                B indicateur d'annee bissextile
85 '                B=0 ou B=1
90 '
100 REM      Entree de la Date
110 PRINT "DONNER LA DATE EN CHIFFRES "
120 PRINT "sous la forme JOUR.MOIS,ANNEE"
130 PRINT "par exemple : 15,9,1984 " : PRINT
140 INPUT J.M.A : PRINT
150 '
200 REM      Controle de Validite de l'Annee
210 IF A<0 THEN PRINT"L'annee doit etre posterieure a J.C." : INPUT "ANNEE : ";A: GOTO 210
220 '
250 REM      Controle de Validite du Mois
260 IF M<1 OR M>12 THEN PRINT"Mois non Valable" : INPUT "MOIS :";M : GOTO 260
270 '
300 REM      Controle de Validite du Jour
310 IF J>0 AND J<32 THEN GOTO 350
320 PRINT "Jour non Valable" : INPUT "JOUR EN CHIFFRES :";J : GOTO 310
330 '
350 IF J=31 AND (M=4 OR M=6 OR M=9 OR M=11) THEN PRINT "Mois de 30 jours" : GOTO 320
360 IF M<>2 THEN GOTO 500
370 '
400 REM      Recherche des Annees Bissextils
410 B=0 : A1=INT(A/100) : A2=A MOD 100
420 IF A2<>0 AND A2 MOD 4 = 0 THEN B=1
430 IF A2=0 AND A1 MOD 4 = 0 THEN B=1
440 '
450 REM      Controle du nombre de Jours du Mois de Fevrier
460 IF B=0 AND J>28 THEN PRINT "Mois de 28 jours" : GOTO 320
470 IF B=1 AND J>29 THEN PRINT "Mois de 29 jours" : GOTO 320
480 '
500 REM      Affichage du Resultat
510 PRINT : PRINT
520 PRINT "LA DATE PROPOSEE :";J;",";M;",";A; " EST VALABLE"

```

le programme

... et son execution

DONNER LA DATE EN CHIFFRES
sous la forme JOUR.MOIS,ANNEE
par exemple : 15.9,1984

? 31.22,-1980

L'annee doit etre posterieure a J.C.

ANNEE : ? 1980
Mois non Valable

MOIS :? 2
Mois de 29 jours

Jour non Valable
JOUR EN CHIFFRES :? 3

LA DATE PROPOSEE : 3 , 2 , 1980 EST VALABLE
Ok

3.16.11. Exercice de récapitulation n° R 12 : le cheval errant

A. But du programme

Le petit programme que nous vous demandons d'élaborer nous servira (plus tard !) dans la confection de programmes divers de jeux (d'échecs, de Go, de morpion). Mais, pour le moment, nous limiterons nos ambitions et ne vous proposerons que des problèmes résolubles avec les instructions et fonctions que vous connaissez

B. Conventions de représentation

La figure 1 (page ci-contre) représente le damier d'un jeu d'échecs (si je peux m'exprimer ainsi !). Pour y repérer la position d'une case, le joueur d'échecs définit son abscisse par une lettre (de a à h) et son ordonnée par un chiffre (de 1 à 8). Par exemple, la case entourée est la case C5.

En informatique, il est plus commode de repérer la position de la case uniquement par des nombres.

On pourrait utiliser un tableau à deux dimensions $T(i, j)$

avec i = abscisse, de 1 à 8

et j = ordonnée, également de 1 à 8.

Mais :

- d'une part, ce n'est pas la notation la plus commode,
- et (surtout !) nous n'avons pas encore étudié les tableaux.

Nous nous contenterons donc de numéroter les cases 1, 2, 3... 63, 64.

Il est commode (et classique) d'entourer ce quadrillage de 8×8 par une bordure de cases complémentaires. Ce procédé permettra de représenter facilement les limites du jeu. On pourra ainsi, par exemple

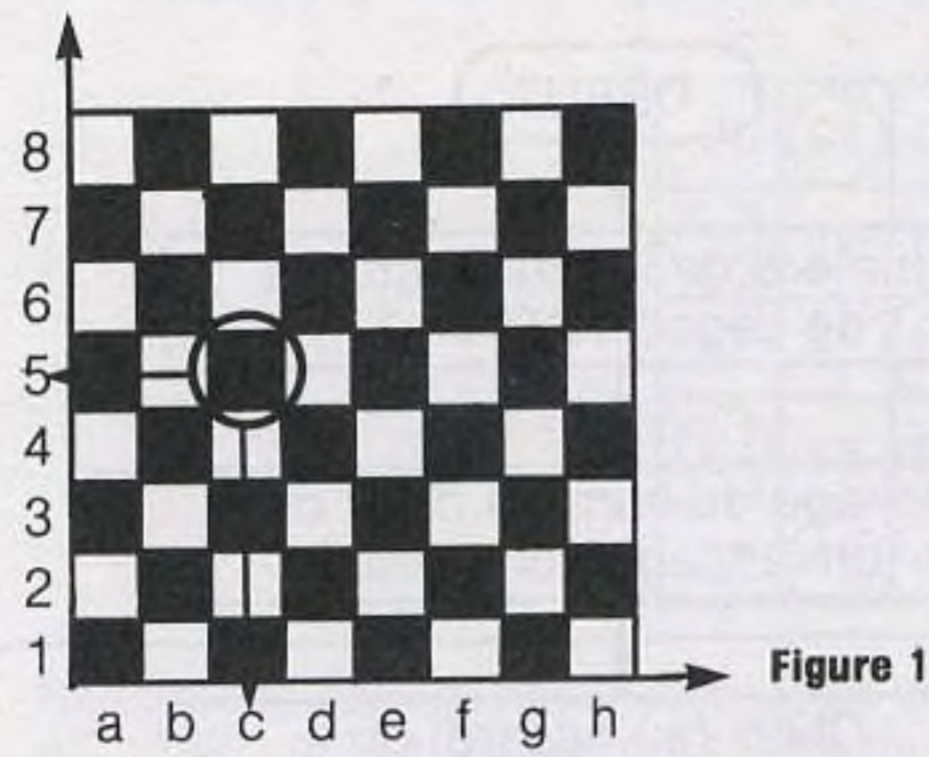
- donner la valeur 0 à une case libre ;
- la valeur +1 à une case occupée par un pion blanc ;
- la valeur +2 à une case occupée par un pion noir ;
- et la valeur -1 à une case interdite (dont les cases de bordures).

Nous n'utiliserons pas cette convention dans le présent exercice, mais nous numéroterons nos cases, comme l'indique la figure 2, pour ne pas avoir à redéfinir de nouvelles conventions à chaque nouvel exercice. On remarque sur cette figure 2 que la bordure est large de deux cases : ceci est commode pour définir les interdictions au mouvement du cheval des échecs, car un cheval sauterait allègrement par dessus une barrière épaisse d'une seule case.

C. Enoncé

Ceci étant posé, nous vous proposerons de résoudre le problème suivant :

- 1°) Position de départ : le cheval est sur la case A1 (c'est-à-dire la case $N = 27$).
 - 2°) L'ordinateur affiche ce numéro de case dans le langage de joueur d'échecs (c'est-à-dire : affiche A1), et s'arrête.
 - 3°) Pour continuer le jeu, l'opérateur appuie sur la touche ENTER.
 - 4°) L'ordinateur va alors déplacer son cheval au hasard mais (bien sûr !) dans une case autorisée. (Ici, ce ne peut être que la case 52 ou la case 41).
Il affiche le numéro de cette case, puis s'arrête.
 - 5°) Pour continuer la partie, l'opérateur appuie sur la touche ENTER.
 - 6°) L'ordinateur va alors déplacer son cheval, etc.
- Le jeu se continue ainsi jusqu'à ce que le cheval se trouve dans la case opposée (case H8) et, à ce moment, il affiche : «C'EST FIN!».



		133	134	135	136	137	138	139	140	141	142	143	144
		121	122	123	124	125	126	127	128	129	130	31	132
8		109	110	111	112	113	114	115	116	117	118	119	120
7		97	98	99	100	101	102	103	104	105	106	107	108
6		85	86	87	88	89	90	91	92	93	94	95	96
5		73	74	75	76	77	78	79	80	81	82	83	84
4		61	62	63	64	65	66	67	68	69	70	71	72
3		49	50	51	52	53	54	55	56	57	58	59	60
2		37	38	39	40	41	42	43	44	45	46	47	48
1		25	26	27	28	29	30	31	32	33	34	35	36
		13	14	15	16	17	18	19	20	21	22	23	24
		1	2	3	4	5	6	7	8	9	10	11	12

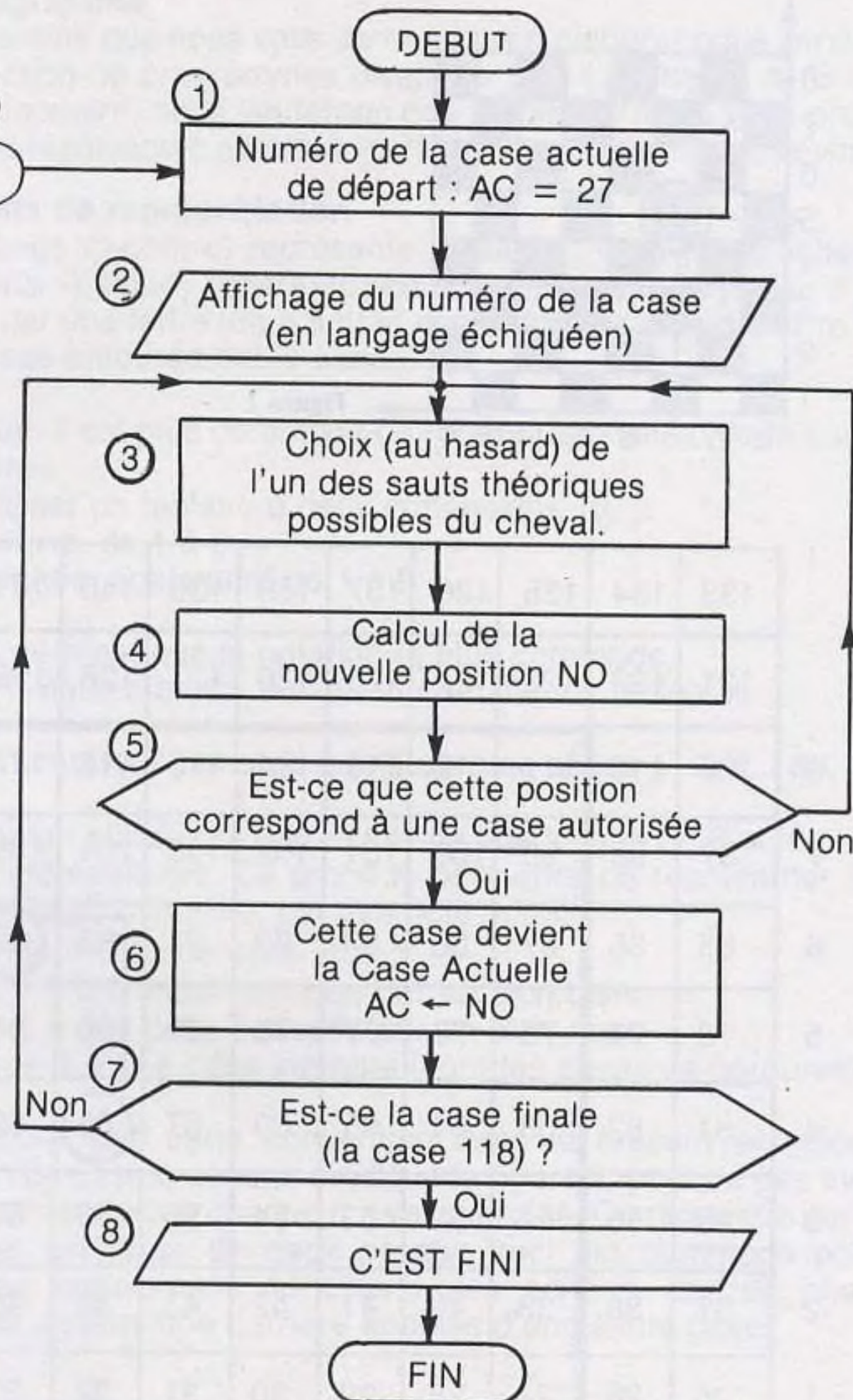
Figure 2

D. Pour commencer à vous mettre sur la voie

Essayez de rédiger l'organigramme traduisant ce problème et comparez avec la solution que nous vous proposons page suivante.

E. Organigramme proposé

Vocabulaire abrégé :
«Case Actuelle» signifie
«case sur laquelle se trouve
actuellement le cheval».



F. Premier problème : traduction des positions en langage échiquéen

Pour résoudre le problème posé par la case 2 de l'organigramme 1, à savoir : «Etant donné un nombre N (par exemple N = 77), trouvez le caractère N\$ traduisant la même position en langage échiquéen (ici N\$ = "C4")».

Bien sûr, vous pourriez écrire toute une collection de IF... THEN..., tels que :

```

100 IF N = 111 THEN C$ = "A1"
110 IF N = 112 THEN C$ = "A2"
120 IF N = 113 THEN C$ = "A3"
... etc...
  
```

Mais ce ne serait ni élégant ni instructif.

Essayez plutôt de «décoder» le numéro N par l'opération «MODULO» (voir LED-MICRO n° 8, §G.3.11.6.C page 26). Mais comme nous supposons que votre système n'a pas l'opérateur MODULO, vous devrez vous débrouiller avec INT, +, * et /.

F. Deuxième problème : choix au hasard de l'un des sauts possibles

90	91	92	93	94
78	79	80	81	82
66	67	68	69	70
54	55	56	57	58
42	43	44	45	46

Raisonnons sur l'exemple représenté sur la figure ci-dessus : le cheval placé en position AC = 68 peut se déplacer sur l'une des huit positions : 78, 91, 93, 82, 58, 45, 43 et 54.

Considérons, par exemple, le saut du cavalier qui consiste à avancer de deux pas vers le nord puis de un pas vers l'est, c'est-à-dire, dans le cas de notre exemple de passer de la case 68 à la case 93 :

$$93 - 68 = 26$$

Bien entendu, ce qui est vrai pour la case 68 l'est pour n'importe quelle case : pour faire avancer le cheval de deux pas au nord suivi de deux pas à l'est, il faut ajouter 26 au numéro de cette case.

Les huit cas possibles de déplacement se traduisent donc pas l'ajout à AC au hasard d'un des huit nombres suivants :

$$10, 23, 25, 14, -10, -23, -25, -14.$$

G. Troisième problème : est-ce que la nouvelle position est acceptable ?

Lorsqu'on a obtenu le nouveau numéro de case, il faut vérifier qu'on ne tombe pas dans une des cases interdites du bord.

Si on tombe dans une case interdite, on fait un autre choix, jusqu'à ce qu'on arrive à une case autorisée.

Pour éliminer les cases interdites, on pourrait utiliser une collection d'instructions telles que :

```
300 IF N = 133 THEN 210
310 IF N = 134 THEN 210
320 IF N = 135 THEN 210
etc.
```

Ce ne serait pas très élégant. Éliminez les bandes d'une façon plus générale :

- élimination des bandes horizontales grâce à la division entière ;
- élimination des bandes verticales grâce au reste de la division entière.

H. Conclusion

- 1°) L'analyse du problème que nous faisons n'est peut-être pas la meilleure : proposez-nous la vôtre.
- 2°) Rédigez un programme résolvant le problème posé, MAIS **en** n'utilisant **que** les instructions et fonctions que vous avez apprises jusqu'à présent.
- 3°) Si vous possédez un ordinateur et une imprimante, pouvez-vous nous envoyer un listing de votre programme et deux ou trois exécutions (en remplaçant les PRINT par des LPRINT dans le programme, bien sûr) ?
- 4°) Ne trouvez-vous pas que en vous posant ce problème, nous avons été trop «bavards» ? N'est-il pas préférable de vous laisser chercher davantage, ou de vous poser des problèmes plus difficiles ?
- 5°) ET (SURTOUT !) donnez-nous des idées d'exercices de récapitulation résolubles uniquement avec les instructions qui ont été étudiées dans le cours.

COURS PRATIQUE DE MICROPROCESSEUR AVEC LE MICROPROFESSOR MPF-IB

SEPTIEME PARTIE

Le langage du Z80^R (3) (suite)

SOMMAIRE

I. INTRODUCTION

II. OPERATIONS DE BASE

- II. 1. Définition
- II. 2. Opérateur ET
- II. 3. Opérateur OU
- II. 4. Opérateur INVERSEUR
- II. 5. Opérateur OU INCLUSIF
- II. 6. Exemple 1
- II. 7. Exemple 2

III. ARITHMETIQUE BINAIRE

- III. 1. Notions de base
- III. 2. Système de numérotation binaire

IV. OPERATIONS LOGIQUES ET ARITHMETIQUES

- IV. 1. Introduction
- IV. 2. ET logique
- IV. 3. OU logique
- IV. 4. OU exclusif
- IV. 5. Addition arithmétique (sans report)
- IV. 6. Addition arithmétique (avec report)
- IV. 7. Positionnement de l'indicateur C
- IV. 8. Soustraction arithmétique (sans report)
- IV. 9. Soustraction arithmétique (avec report)
- IV. 10. Comparaison
- IV. 11. Opérations particulières sur l'accumulateur
- IV. 12. Incrémentation et documentation
- IV. 13. Opérations sur 16 bits

I. INTRODUCTION

Poursuivant notre étude du langage du microprocesseur, nous allons aborder dans cette septième partie, les opérateurs logiques et arithmétiques.

Les leçons 5 et 6 étaient consacrées aux mouvements de données internes dans le cas de transfert entre registres mouvements externes quand la donnée était le contenu d'un emplacement mémoire. Nous pouvons dire qu'il s'agissait d'«opérateurs neutres» dans la mesure où les **données ne sont pas modifiées mais transférées.**

Dans les instructions que nous allons étudier dès à présent, le résultat que nous obtiendrons est fonction d'une part de l'opérateur désigné mais aussi des données, celles-ci pouvant résulter d'un transfert.

Le résultat, lui aussi, peut être transféré dans un emplacement mémoire pour y être sauvegardé : ainsi opérateurs de transfert et opérateurs logiques et arithmétiques sont complémentaires pour obtenir le bon déroulement d'un programme.

Pour que tout soit bien clair, au risque «d'enfoncer une porte ouverte», nous rappelons brièvement les opérateurs de base dans le cadre «microprocesseur».

II. OPERATIONS DE BASE

II.1. Définition

Considérons le schéma électrique représenté par la figure 126.

Il comprend, montés en série :

- 1 générateur G
- 1 interrupteur I
- 1 poussoir P
- 1 lampe L

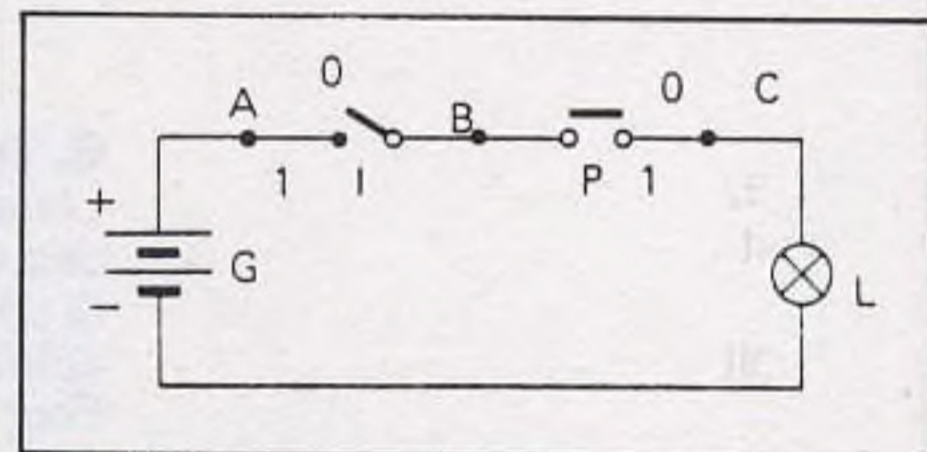


Fig. 126

L'interrupteur I peut prendre deux positions. Quand il est «OUVERT» (ou inactif) le courant ne peut circuler de A vers B. Par contre, lorsque I est «FERME» (ou actif) le courant peut circuler.

Le poussoir P peut prendre deux positions.

Quand P est :

- relâché (inactif), le courant ne peut pas circuler de B vers C
- enfoncé (actif), le courant peut circuler de B vers C.

La lampe L est soit «éteinte» soit «allumée».

L'interrupteur I et le poussoir P sont des variables logiques telles que nous les avons définies. En effet I est soit ouvert ou fermé ; P est soit relâché ou enfoncé ; la lampe est soit allumée ou éteinte.

Les trois variables I, P et L ne jouent pas le même rôle. L'état de la lampe (éteinte ou allumée) dépend de l'état des deux autres variables I et P. La lampe L sera allumée si l'interrupteur I ET le poussoir P sont l'un ET l'autre dans l'état actif.

Nous dirons que l'état de la varia-

ble de sortie L dépend de l'état des deux autres variables d'entrée I et P.

L'ensemble I et P, ainsi monté, réalise la fonction logique ET, c'est-à-dire que la lampe L sera allumée (état 1) quand l'une et l'autre des variables «I» ET «P» sont simultanément dans l'état actif (état 1).

Donc un «interrupteur» et un «poussoir» en série constituent un OPERATEUR qui réalise la fonction ET.

Modifions le circuit comme l'indique la figure 127.

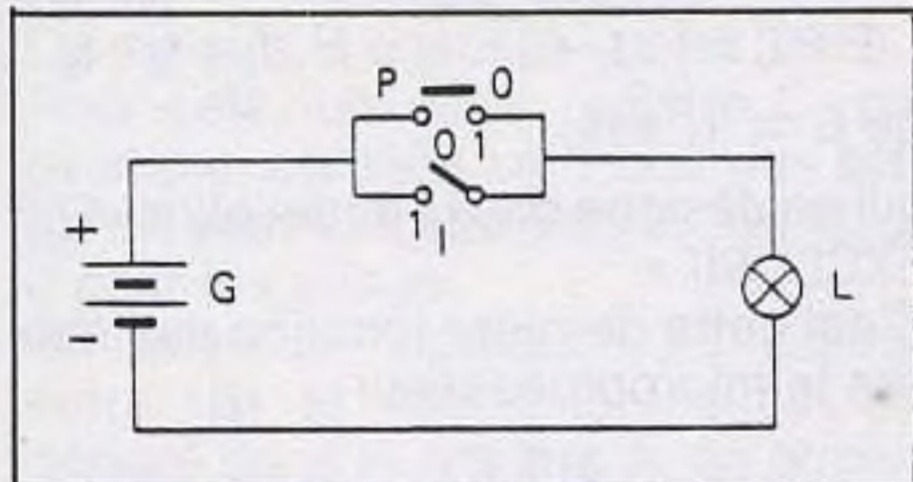


Fig. 127

Les deux commandes «I» et «P» sont en parallèle au lieu d'être en série. Le montage de I et P réalise un autre opérateur. Pour que la lampe L soit allumée, il faut que :

- I soit fermé
- OU
- P soit enfoncé.

Nous dirons que cet ensemble réalise la fonction logique «OU».

Examinons maintenant la figure 128.

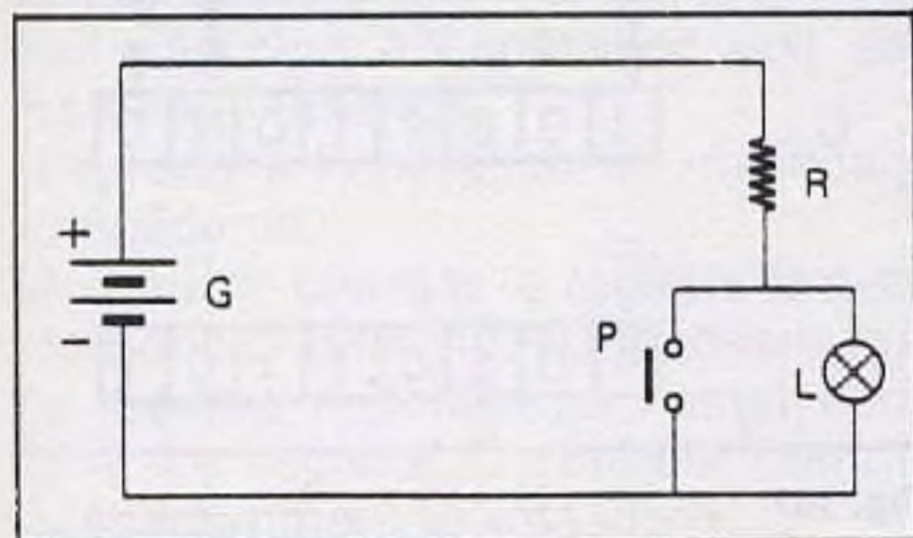


Fig. 128

Quand P est relâché (état 0), la lampe L est allumée (état 1).

Quand P est enfoncé (état 1), la lampe L est éteinte (état 0).

Il s'agit d'un autre opérateur puisque les états de la variable d'entrée et de la variable de sortie sont l'INVERSE l'un de l'autre. L'opérateur qui réalise cette fonction est un INVERSEUR.

II.2. Opérateur «ET»

Reprenons le schéma de la figure 126 et par convention nous désignons, mais arbitrairement, par «1» les états actifs des variables et par conséquent par «0» les états inactifs.

A l'aide de ces notations, nous pouvons établir les deux tableaux des

figures 129 et 129 bis. Tous deux donnent l'état de la variable de sortie (la lampe L) en fonction des positions des variables I et P.

Inter I	Poussoir P	Lampe L
Pos. 0	Relâché	Eteinte
Pos. 1	Relâché	Eteinte
Pos. 0	Enfoncé	Eteinte
Pos. 1	Enfoncé	Allumée

Fig. 129

I	P	L
0	0	0
1	0	0
0	1	0
1	1	1

Fig. 129 bis

Ces deux tableaux nous permettent de résumer la fonction réalisée par ce circuit de la manière suivante : **la lampe L est allumée quand l'interrupteur I ET le poussoir P sont, l'UN ET L'AUTRE dans l'état actif.**

Ce qui peut se traduire par l'équation logique :

$$L = I \times P$$

Le signe «X» (qu'il ne faut pas confondre avec le signe «multiplié») est celui de la fonction ET.

Que se passe-t-il dans un microprocesseur ?

La fonction ET entre le contenu de deux registres B et C par exemple, consiste à effectuer «bit à bit» l'opération ET entre chaque bit de même position des deux registres. Le résultat est généralement déposé dans l'accumulateur (figure 130).

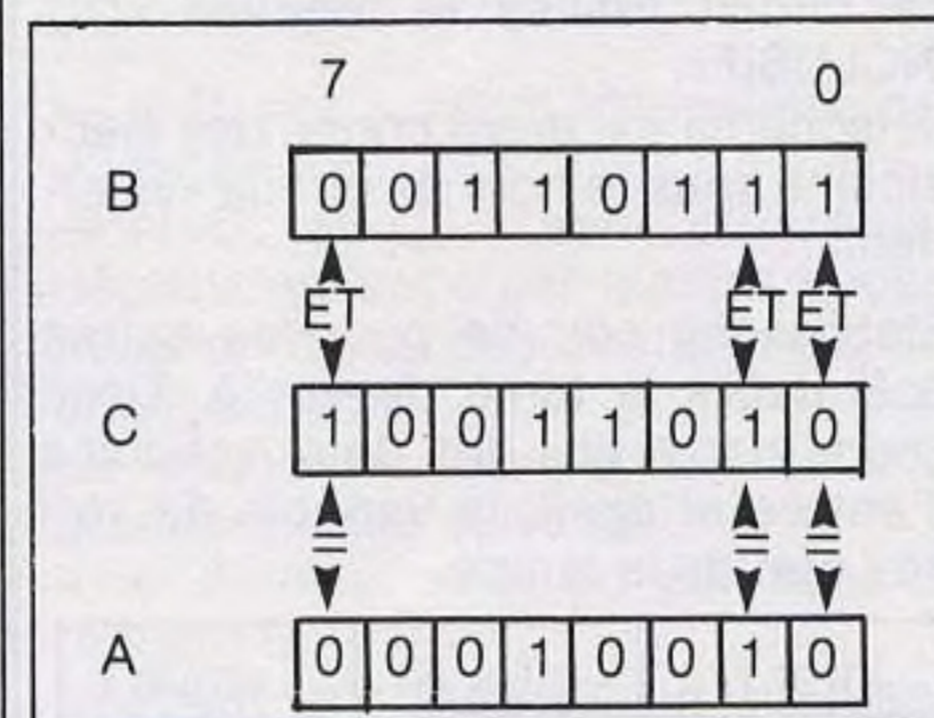


Fig. 130

Ainsi :

$$A_0 = B_0 \times C_0$$

ou $1 \times 0 = 0$

$$A_1 = B_1 \times C_1$$

ou $1 \times 1 = 1$

.....

$$A_7 = B_7 \times C_7$$

ou $0 \times 1 = 0$.

II.3. Opérateur «OU»

D'une manière analogue, nous pouvons établir les deux tableaux des figures 131 et 131 bis.

Tous deux donnent l'état de la variable de sortie en fonction des positions des variables I et P dans le cas d'un opérateur OU.

Inter I	Poussoir P	Lampe L
Pos. 0	Relâché	Eteinte
Pos. 1	Relâché	Allumée
Pos. 0	Enfoncé	Allumée
Pos. 1	Enfoncé	Allumée

Fig. 131

I	P	L
0	0	0
1	0	1
0	1	1
1	1	1

Fig. 131 bis

La lampe L est allumée quand l'interrupteur I OU le poussoir P est dans l'état actif.

Ce qui peut se traduire par l'équation logique :

$$L = I + P$$

Le signe «+» (qu'il ne faut pas confondre avec le signe addition) est celui de la fonction OU.

Etudions à nouveau ce qui se passe dans un microprocesseur (fig. 132).

La fonction OU entre les contenus de deux registres B et C, consiste à effectuer «bit à bit» l'opération OU entre chaque bit de même rang des deux registres. Le résultat est généralement déposé dans l'accumulateur.

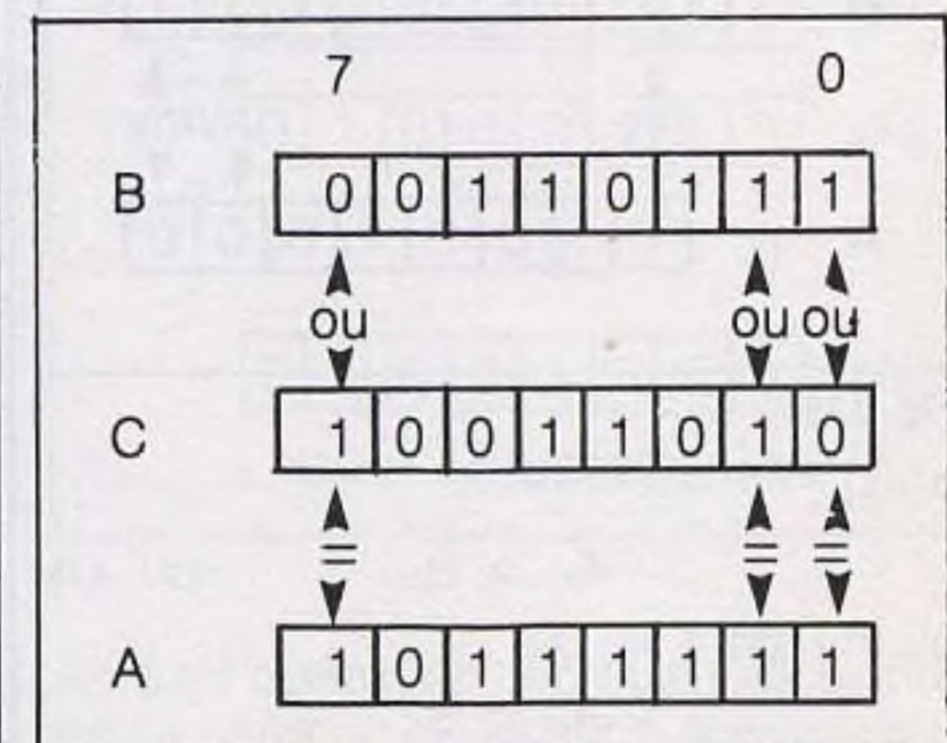


Fig. 132

Ainsi :

$$A_0 = B_0 + C_0$$

ou $1 + 0 = 1$

$$A_1 = B_1 + C_1$$

ou $1 + 1 = 1$

.....

$$A_7 = B_7 + C_7$$

ou $0 + 1 = 1$

II.4. Opérateur «inverseur»

Dans le cas de cet opérateur dit «inverseur» ou complément à 1, le tableau est réduit comme l'indiquent les figures 133 et 133 bis. Dans ce cas, l'état de la variable de sortie est inverse de celui de la commande.

Poussoir P	Lampe L
Relâché	Allumée
Enfoncé	Eteinte

Fig. 133

P	L
0	1
1	0

Fig. 133 bis

La lampe L est allumée quand le poussoir est relâché (inactif) et inversement, éteinte quand le poussoir est enfoncé (actif), ce qui peut se traduire par l'équation logique :

$L = \bar{P}$ où « $\bar{\quad}$ » indique l'état inverse de la variable.

Au niveau du microprocesseur, l'exécution d'une telle instruction consiste à inverser chaque bit du registre considéré et de déposer le résultat dans A.

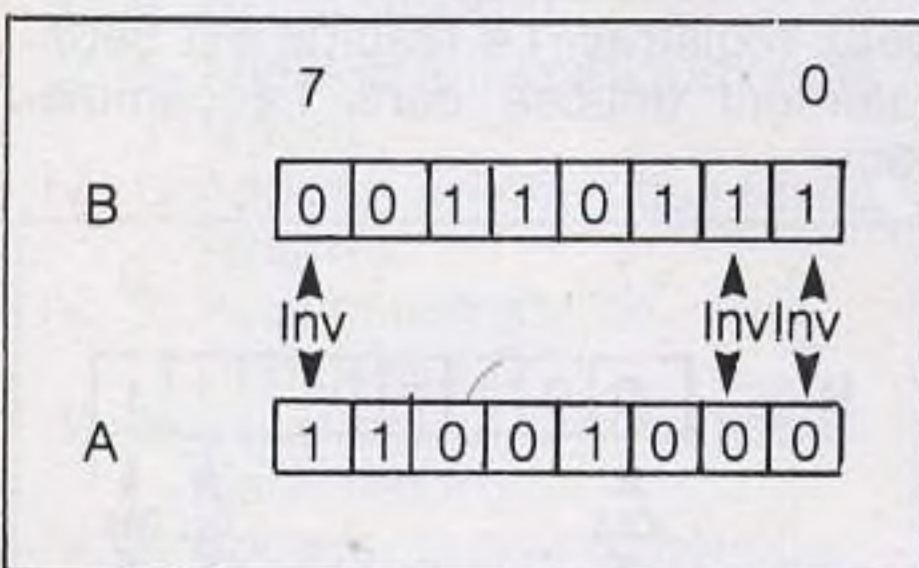


Fig. 134

Ainsi :

$$A_0 = \bar{B}_0$$

ou inv. de 1 = 0

$$A_1 = \bar{B}_1$$

ou inv. de 1 = 0

$$A_7 = \bar{B}_7$$

ou inv. de 0 = 1.

II.5. Opérateur «OU INCLUSIF»

Considérons le schéma de la figure 135. Le circuit comporte :

- un générateur G
- deux interrupteurs I_1 et I_2
- une lampe L.

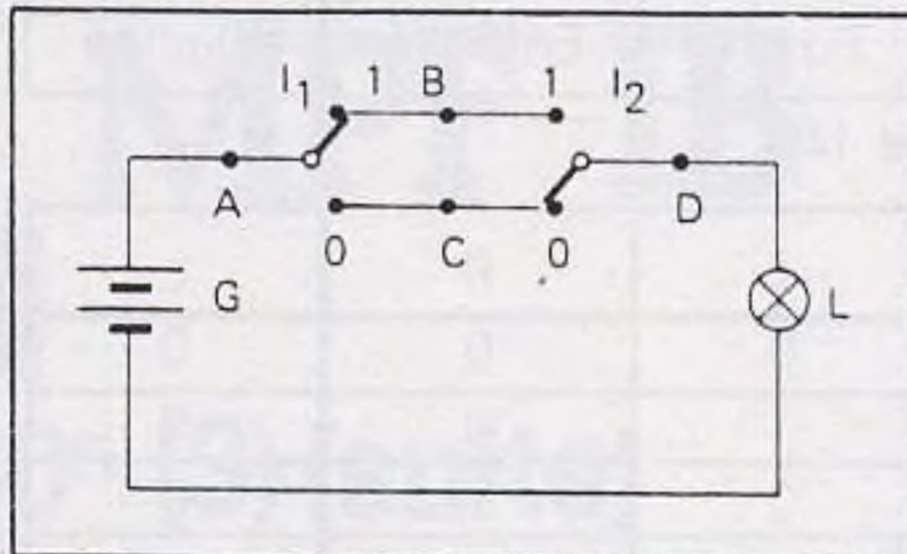


Fig. 135

Lorsque les interrupteurs I_1 et I_2 sont dans la position indiquée sur la figure (I_1 en 1, I_2 en 0), le courant ne circule pas : la lampe est éteinte.

Plaçons I_2 en position «1» sans changer I_1 , le courant circule au travers de la branche ABD.

Si maintenant, au lieu de modifier I_2 , nous avons placé I_1 en position «0» (I_2 étant resté en «0»), le courant circulerait au travers de la branche ACD.

En conclusion, **si les interrupteurs I_1 et I_2 sont tous les deux dans la position «1» (branche ABD) ou tous les deux dans la position «0» (branche ACD), la lampe L est allumée (ou position «1»).**

Par contre, si les variables I_1 et I_2 sont dans des états inverses «1» et «0» ou «0» et «1», la lampe L est éteinte (ou position «0»).

Ce circuit réalise la fonction «OU INCLUSIF».

Ce schéma est aussi connu des électriciens sous le nom de circuit «va-et-vient».

Etablissons comme pour les autres opérateurs la table de vérité. Désignons par I_1 et I_2 les deux variables d'entrée et par L la variable de sortie : état de la lampe.

Inter I_1	Inter I_2	Lampe L
Pos. 0	Pos. 0	Allumée
Pos. 1	Pos. 0	Eteinte
Pos. 0	Pos. 1	Eteinte
Pos. 1	Pos. 1	Allumée

Fig. 136

I_1	I_2	L	$E = L$
0	0	1	0
1	0	0	1
0	1	0	1
1	1	1	0

Fig. 136

ce qui se traduit par l'équation logique :

$$L = I_1 \cdot I_2 + \bar{I}_1 \cdot \bar{I}_2$$

En réalité, on lui préfère bien souvent l'opération inverse, telle que :

$$E = \bar{L} = I_1 \cdot I_2 + \bar{I}_1 \cdot \bar{I}_2 = I_1 \cdot \bar{I}_2 + \bar{I}_1 \cdot I_2$$

ou $E = I_1 \oplus I_2$

qui se désigne sous l'appellation «OU EXCLUSIF».

C'est cette dernière fonction que réalise le microprocesseur.

La fonction «OU exclusif» (ou «XOR») entre deux registres B et C par exemple, consiste à effectuer bit à bit l'opération OU exclusif entre chaque bit de même rang des deux registres.

Le résultat est généralement déposé dans l'accumulateur.

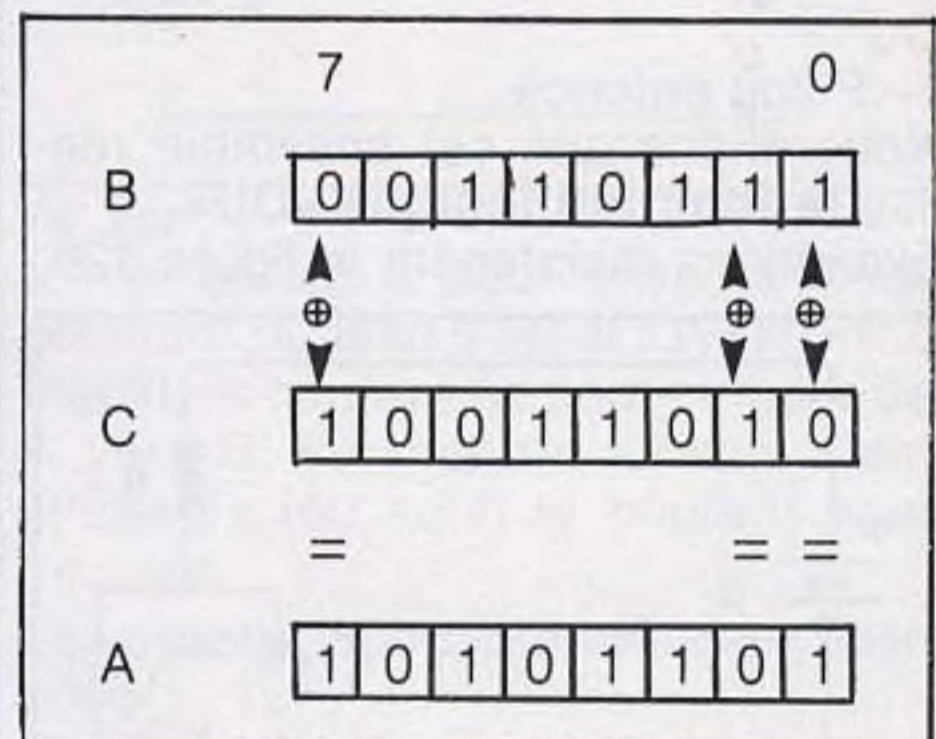


Fig. 137

Ainsi :

$$A_0 = B_0 \oplus C_0$$

ou $1 \oplus 0 = 1$

$$A_1 = B_1 \oplus C_1$$

ou $1 \oplus 1 = 0$

.....

$$A_7 = B_7 \oplus C_7$$

ou $0 \oplus 1 = 1$.

II.6. Exemple 1 : Opérateur «ET»

Nous allons montrer comment «isoler» un certain nombre de digits inclus dans un octet. Pour fixer les idées, on a besoin d'isoler les bits 2, 3 et 4 du registre B qui contient 0011 0111 (ou 37H).

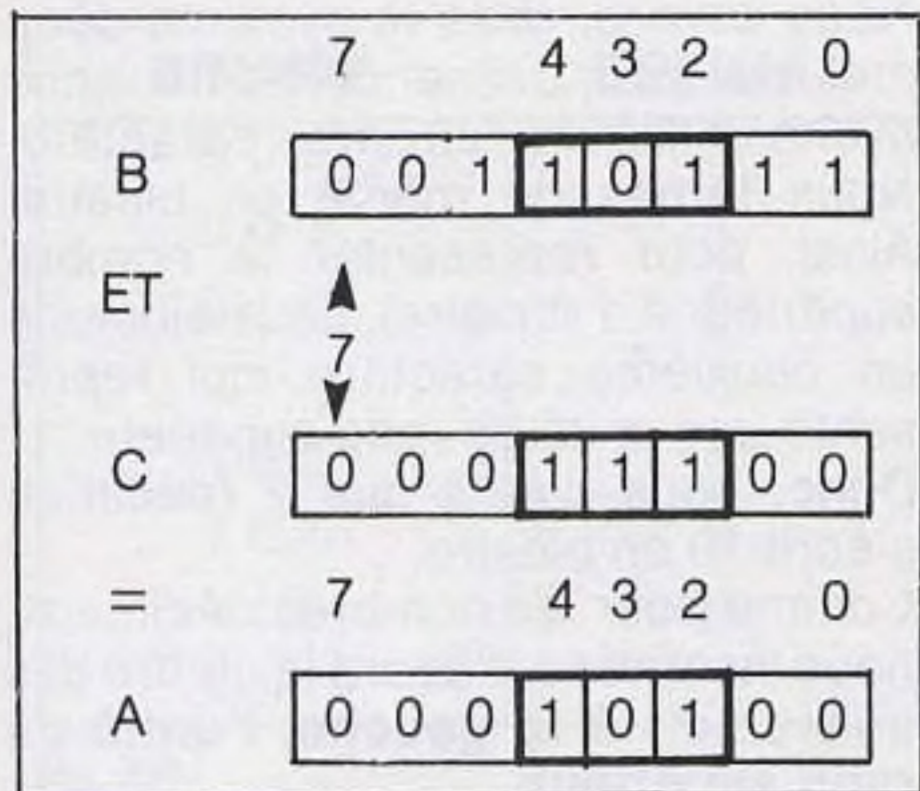


Fig. 138

Le registre B contient l'octet (37H). Pour cela, un autre registre C, par exemple, contient un 1 pour les bits qu'on veut extraire (bits 2, 3 et 4), un 0 pour les autres.

On effectue l'opération ET logique entre «B» et «C» et le résultat est déposé dans le registre A, on obtient ainsi les trois bits qu'on devait isoler (figure 138).

En réalité, dans les opérations logiques (et d'autres), le registre A joue un double rôle : il contient d'une part l'une des données et constitue aussi le registre destinataire de l'opération. Nous verrons ultérieurement comment cela est possible. Dans ce cas, la donnée qui était dans le registre A est détruite, puisque le contenu de celui-ci est remplacé par le résultat final. Par contre, le registre qui participe à l'opération voit son contenu inchangé.

Reprenons l'opération de masquage précédente.

Au lieu de charger le registre B avec l'octet d'origine, nous supposons que le registre A contienne l'octet complet. Le registre C contient comme précédemment le «masque».

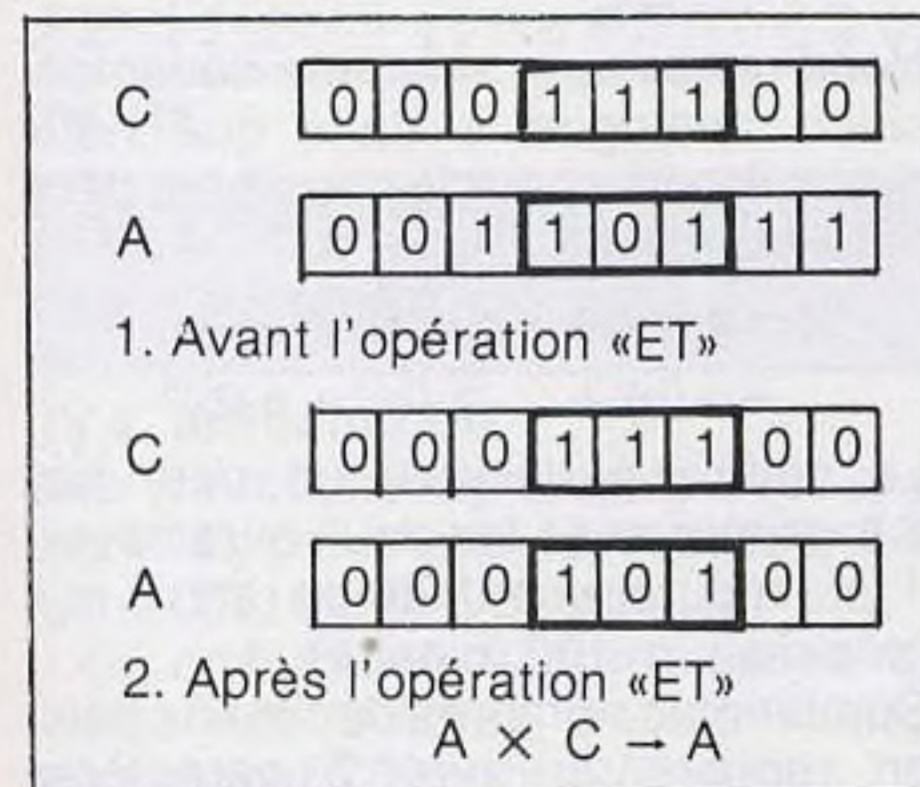


Fig. 139

Nota : X indique que l'opération est un ET.

Notons qu'il suffit :

— de charger le registre C avec le «masque»

— de charger le registre A avec l'octet d'origine

— d'effectuer l'opération ET logique entre A et C, il en résulte que :

• le contenu du registre A est la partie isolée

• le contenu du registre C est inchangé.

II.7. Exemple 2 : Opérateur «OU exclusif»

A l'aide d'un second exemple, nous allons montrer une application de la fonction «OU exclusif».

Soit un dispositif de sécurité ou d'alarme, qui comporte 8 contacts à surveiller. (On peut imaginer qu'il s'agit d'un dispositif de surveillance dans une maison ou un appartement).

Ceux-ci peuvent être soit ouverts (état «0») ou fermés (état «1»).

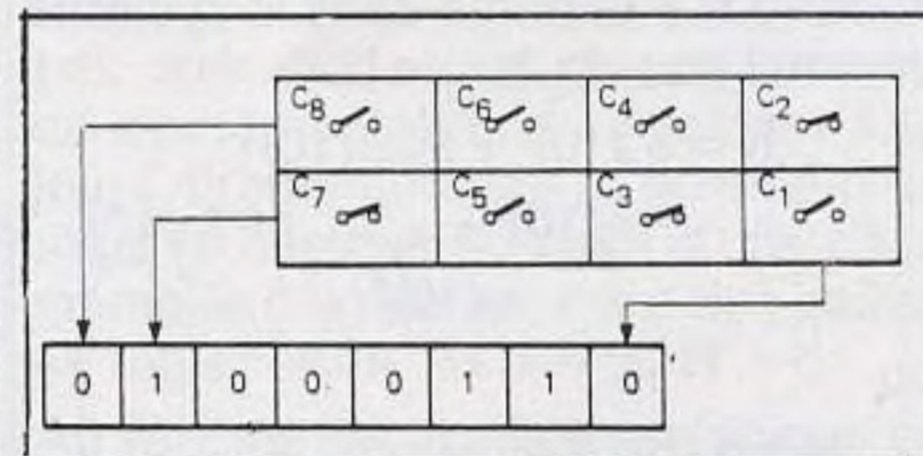


Fig. 140

Lorsque le système d'alarme est en fonctionnement, tout changement de l'état 1 ou plusieurs contacts, doit être détecté et les contacts identifiés.

Le rôle du microprocesseur est de «surveiller les contacts», d'effectuer périodiquement un relevé «état des contacts». Le système génère ainsi un mot de contrôle que nous désignerons par C_{n+1} et le compare au précédent C_n . Entre deux relevés successifs :

— si C_n et C_{n+1} sont identiques : pas d'alarme

— si C_n et C_{n+1} sont différents : déclenchement d'une alarme et identification du ou des contacts qui ont changé d'état.

La séquence (ou portion du programme) est la suivante :

a) le dernier octet C_n est dans le registre B

b) avant d'effectuer un nouveau relevé, le contenu de B est transféré dans A

c) le nouveau relevé C_{n+1} est chargé dans le registre B

d) l'opération $A \oplus B$ est effectuée, le résultat est déposé dans l'accumulateur A.

a) Sauvegarde de C_{n+1}

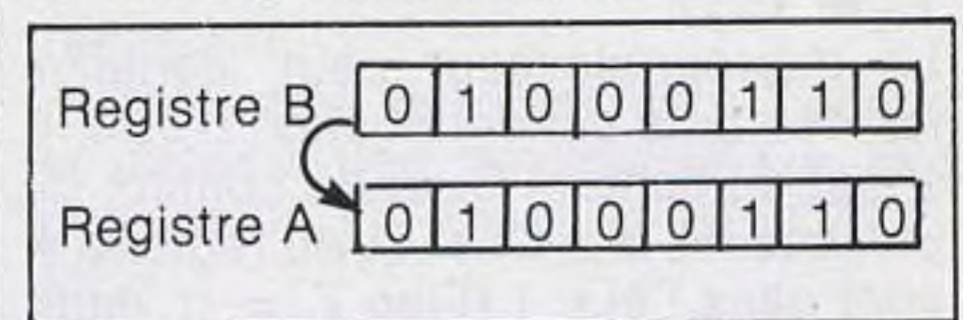


Fig. 141

b) Comparaison entre C_n et C_{n+1}

A partir de là, deux cas peuvent se présenter, comme l'illustrent les figures 142 et 143 ci-après.

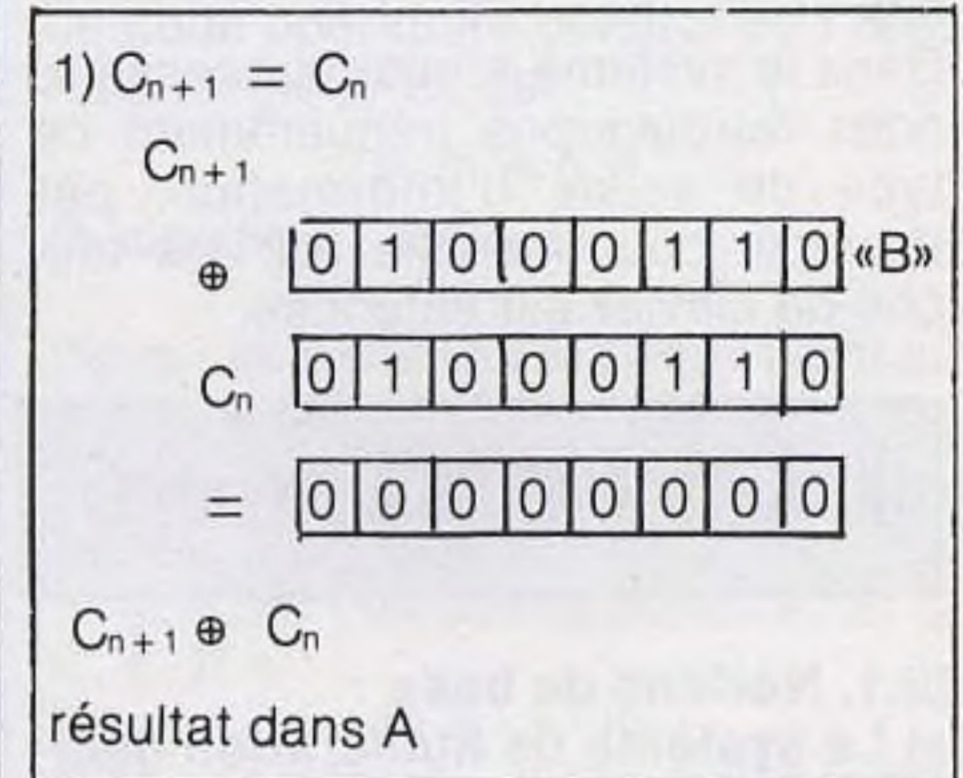


Fig. 142

Rappelons qu'avant l'opération logique «OU exclusif», le contenu du registre A était C_n , et celui du registre B : C_{n+1} . Après l'opération logique C_n est perdu, le contenu de A est le résultat $C_{n+1} \oplus C_n$. Le contenu du registre B est C_{n+1} , inchangé.

Dans ce cas, étant donné que $C_{n+1} = C_n$, le résultat de l'opération est une suite de 8 zéros : il n'y a eu aucun changement d'état entre les instants T_n et T_{n+1} . Pas d'alarme.

Nous verrons ultérieurement qu'une suite de 8 zéros dans le registre A résultant d'une opération logique (ou arithmétique) se détecte facilement (Flag Z en 1).

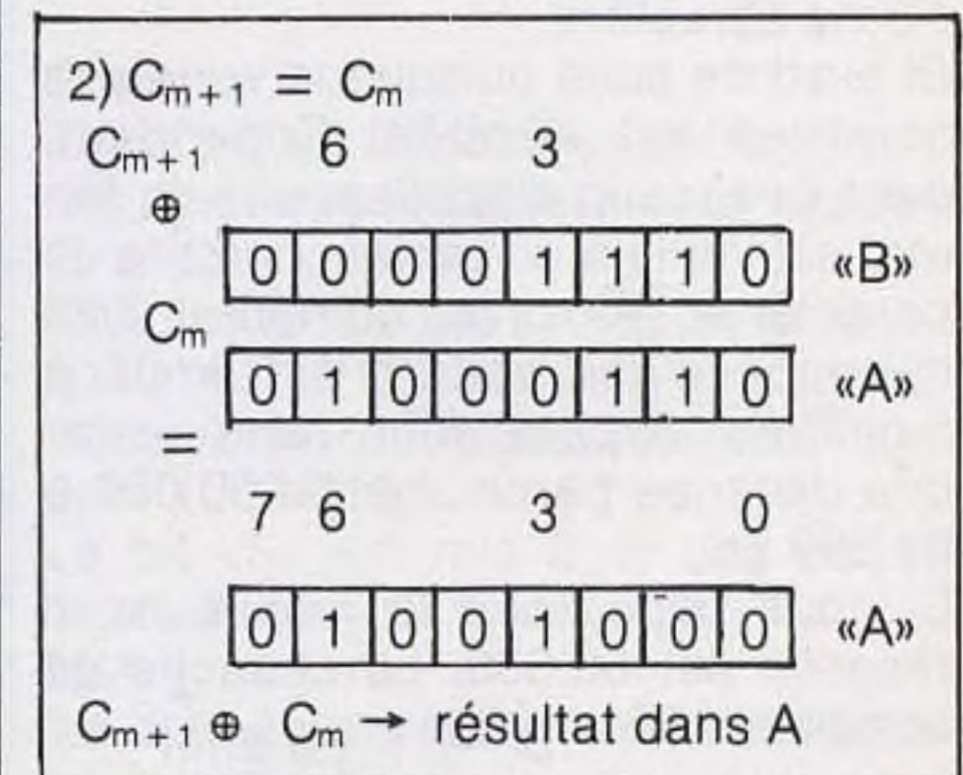


Fig. 143

Après l'opération logique OU exclusif, le contenu du registre A n'est plus «nul» : donc il y a au moins un des

contacts qui a changé d'état entre T_m et T_{m+1} .

Le microprocesseur peut identifier les contacts qui ont changé d'état, il s'agit du contact C_4 et du contact C_7 puisque les bits C_3 et C_6 du registre A sont dans l'état 1 (Flag Z = 0, dans ce cas).

En conclusion, la fonction «OU exclusif» a été mise à profit pour détecter d'une part et identifier d'autre part le ou les contacts qui ont changé d'états entre deux relevés successifs.

Dans le système à microprocesseur, nous retrouverons fréquemment ce type de saisie d'information, par exemple pour identifier qu'une touche de clavier est enfoncée.

III. ARITHMETIQUE BINAIRE

III.1. Notions de base :

a) Le système de numération décimale :

Notre façon de compter, le système de numération décimale nous est très familier. Nous en rappelons cependant les principes essentiels pour définir la notion de «base». Notre système de numération décimale, encore appelé **système à base DIX (10)** est composé de dix caractères nommés chiffres de 0 à 9. Pour représenter une quantité supérieure à 9, il faut faire appel à un deuxième caractère. On obtient ainsi après 9 le nombre 10 qui représente une unité de rang supérieur, appelée «dizaine».

Les nombres de deux chiffres permettent de compter jusqu'à 99. Pour aller au-delà, il faut utiliser un troisième caractère.

Et ainsi de suite puisque la suite des nombres est illimitée. Cependant, dans beaucoup d'applications, le format est limité à un certain nombre de caractères. Ainsi un compteur kilométrique d'une voiture est limité à 5 chiffres. Il peut donc représenter une distance parcourue de 00 000 à 99 999 km.

Si nous dépassons le maximum, il repasse par 00 000. La capacité du compteur est : $10^5 - 1$. En effet :

$$\begin{aligned} 10^5 &= 100\ 000 \text{ donc } 100\ 000 - 1 \\ &= 99\ 999 \\ \text{où } 10^5 - 1 &= 99\ 999 \end{aligned}$$

Exemple :

Pour représenter 2076, nous plaçons les uns à côté des autres et en commençant par la gauche, le chiffre 2 puis le chiffre 0, le 7 et pour finir le 6. Cette représentation signifie en réalité que le nombre 2076 se compose de la manière suivante.

2 unités de mille ou 2×1000 ou 2×10^3

0 unité de cent ou 0×100 ou 0×10^2

7 unités de dix ou 7×10 ou 7×10^1

6 unités ou 6×1 ou 6×10^0 .

(Rappel $A^0 = 1$)

et le nombre N peut s'écrire :

$$N = 2076 = 2 \times (10)^3 + 0 \times (10)^2 + 7 \times (10)^1 + 6 \times (10)^0$$

et en désignant par «b» la base (qui vaut 10 dans le système décimal)

$$N = 2076 = 2 \times (b)^3 + 0 \times (b)^2 + 7 \times (b)^1 + 6 \times (b)^0$$

D'une manière plus générale, un nombre N s'exprime dans le système décimal sous la forme :

$$\begin{aligned} N &= a_n(10)^n + a_{n-1}(10)^{n-1} \\ &+ \dots + a_i(10)^i + \dots + a_1(10)^1 \\ &+ a_0(10)^0 \end{aligned}$$

ou

$$N = a_n(b)^n + a_{n-1}(b)^{n-1} + \dots + a_i(b)^i + \dots + a_1(b)^1 + a_0(b)^0$$

que l'on écrit :

$n = a_n a_{n-1} \dots a_i \dots a_1 a_0$
expression dans laquelle chaque chiffre a_i ne peut que prendre une valeur entière comprise entre 0 et 9. La position d'un chiffre «a» représente un «poids» égal à $(10)^i$ ou $(b)^i$. Toutes ces notions de numération sont très importantes car elles restent **valables quelle que soit la base utilisée.**

Dans les systèmes logiques, on utilise la base $b = 2$, dans les calculateurs, ce sera la base $b = 8$ tandis que dans les systèmes microprocesseurs on utilise la base 16.

III.2. Le système de numération BINAIRE

La **base binaire** ne comporte que 2 caractères, représentés usuellement par «0» et «1» qu'il ne faut pas confondre avec les chiffres 0 et 1 du système décimal.

Dans la base binaire, le premier nombre est le nombre nul «0», le second est le nombre «1».

À partir du suivant, le problème se pose : comment peut-on représenter dans le système binaire l'équivalent du nombre 2 décimal ?

Nous savons, dans le système décimal, que pour passer de 9 à 10, nous avons utilisé un deuxième caractère. Nous ferons de même en binaire. Ainsi, pour représenter le nombre supérieur à 1 (binaire), nous ajoutons un deuxième caractère, qui représente une unité de rang supérieur.

Donc, nous dirons que 2 (décimal) s'écrit 10 en binaire.

Comme pour les nombres décimaux, nous inscrivons d'abord le chiffre des unités puis **à la gauche, l'unité de rang supérieur.**

Le chiffre 3 décimal s'écrit 11 en binaire puisqu'il suffit d'ajouter une unité à 10 (binaire) qui vaut 2 en décimal.

La quatrième unité est 8 et représente 2^3 ou $(b)^3$.

.....

La nième unité représente 2^n ou $(b)^n$. On appelle «POIDS» le nombre représenté par une unité quel que soit son rang.

Par exemple, le nombre 5 s'écrit 101 (binaire). Le premier caractère (le plus à gauche) représente les unités de «poids» le plus fort.

Dans le cas présent, la «présence» de ce poids (symbolisé par un 1) représente la quantité 4 en décimal.

Le second caractère «0» signifie que la quantité de poids est inférieure, dans ce cas, 2 n'est pas inclus dans le nombre. Par contre, le poids le plus faible (soit 1) est inclus.

En faisant la «somme» de tous ces «poids» (et on trouve une certaine analogie avec la balance de nos grands-mères), nous obtenons :

$$\begin{aligned} 1\ 0\ 1 &= 1 \times (2)^2 + 0(2)^1 + 1(2)^0 \\ &= 1 \times 4 + 0 \times 2 + 1 \times 1 \\ &= 4 + 0 + 1 \\ &= 5 \end{aligned}$$

Nous retrouvons ainsi une décomposition analogue à celle que nous avons étudiée pour les nombres décimaux, c'est-à-dire :

$$N = a_n(b)^n + a_{n-1}(b)^{n-1} + \dots + a_i(b)^i + \dots + a_1(b)^1 + a_0(b)^0$$

Le chiffre 4 décimal ne peut pas s'exprimer avec les deux caractères, il faut ajouter une unité de rang 3 et 4 (décimal) = 100 (binaire).

Combien de nombres décimaux peut-on représenter avec 3 caractères binaires ?

Etablissons le tableau (figure 144) qui donne l'équivalence d'un nombre binaire avec le nombre décimal qu'il représente.

BINAIRE	DECIMAL
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7

Fig. 144

La figure 144 nous montre qu'avec 3 bits on peut représenter les 8 premiers nombres décimaux (0 à 7) au-delà, il faut rajouter une quatrième unité, qui s'écrira 1000 en binaire et vaudra 8 en décimal.

En résumé, dans le système binaire :
— la première unité est 1 et représente 2^0 ou $(b)^0$

— la deuxième unité est 2 et représente 2^1 ou $(b)^1$

— la troisième unité est 4 et représente 2^2 ou $(b)^2$.

Le tableau de la figure 145 donne pour les dix premières puissances de «2» la représentation du nombre binaire et son équivalence décimale.

2^n	Nd	N en binaire
2^0	1	1
2^1	2	1 0
2^2	4	1 0 0
2^3	8	1 0 0 0
2^4	16	1 0 0 0 0
2^5	32	1 0 0 0 0 0
2^6	64	1 0 0 0 0 0 0
2^7	128	1 0 0 0 0 0 0 0
2^8	256	1 0 0 0 0 0 0 0 0
2^9	512	1 0 0 0 0 0 0 0 0 0

Fig. 145

IV. OPERATIONS LOGIQUES ET ARITHMETIQUES

IV.1. Introduction

Les opérations logiques de base exécutables par le microprocesseur Z80 sont : ET, OU et OU exclusif.

Les opérations arithmétiques de base exécutables par le microprocesseur Z80 sont :

- Addition avec ou sans report
- Soustraction avec ou sans report
- Comparaison
- Incrémenter et décrémentation.

Nous ne reviendrons pas sur l'aspect purement logique des opérateurs puisque les principes ainsi que les tables de vérité ont été rappelées dans le paragraphe 2.

Les opérations arithmétiques ne présentent aucune difficulté ; nous dirons simplement qu'«incrémenter» consiste à ajouter une unité et que «décrémenter» consiste à en retrancher une.

Une opération logique (ET, OU, OU exclusif) s'effectue toujours bit à bit entre un opérande désigné par «s» et le contenu de l'accumulateur A. Le résultat de cette opération est replacé dans l'accumulateur.

Une opération arithmétique (addition ou soustraction) s'effectue toujours entre la quantité contenue dans l'accumulateur et un opérande désigné par «S». Le résultat de cette opération est replacé dans l'accumulateur.

Les opérations d'incrémenter ou de décrémentation s'effectuent toujours directement sur la donnée d'un registre désigné ou celle d'une case mémoire pointée par l'une des paires de registres HL, IX + d ou IY + d.

Au résultat de l'opération logique ou arithmétique s'ajoute un second effet dont il faudra savoir tenir compte : c'est le **positionnement des indicateurs en fonction du résultat obtenu (registre F)**.

Pour les **trois opérations logiques**, les règles sont les suivantes :

— Indicateurs «C» (carry ou report) et «N» (indique que l'opération précédente était une soustraction) sont remis à «0».

— Indicateur «Z» (zéro) est mis à «1» quand le résultat de l'opération entraîne une mise à «0» de tous les bits de l'accumulateur.

— Indicateur «P/V» est mis à 1 quand le nombre de 1 de l'accumulateur est pair. Il est à 0 quand le nombre de 1 est impair (indicateur P). Fonction PARITE.

— Indicateur «S» (indicateur de signe) est la copie du bit 7. Dans les opérations sur les nombres relatifs, la mise en 1 de ce bit signifie que le nombre est négatif.

— Indicateur «H» est à «1» quand une retenue provient des 4 bits les moins significatifs.

Pour les **opérations arithmétiques**, les règles sont identiques excepté pour les indicateurs suivants :

— Indicateur «C» est mis à 1 quand

un report s'échappe du bit 7. Sinon remis à 0.

— Indicateur «P/V» est mis à 1 quand un dépassement de capacité survient (indicateur V). Fonction DEBORDEMENT.

— Indicateur «N» est mis à 1 dans le cas des opérations de soustraction. Dans l'étude des codes opérations, nous ne reviendrons pas sur ces règles concernant les indicateurs.

IV. 2. ET logique

Le code opératoire général de l'opération logique ET est :

$$A \leftarrow A \wedge s$$

(\wedge signifie ET logique)

(Nota : le signe « \wedge » est équivalent au symbole «X»)

dans lequel «s» désigne l'un des opérands suivants :

r : désigne l'un des registres A, B, C, D, E, H ou L

n : une donnée de 8 bits

(HL) : le contenu de l'emplacement pointé par HL

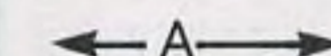
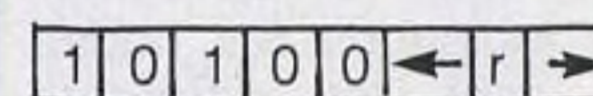
(IX + d) : le contenu de l'emplacement pointé par IX auquel s'ajoute le déplacement «d»

(IY + d) : le contenu de l'emplacement pointé par IY auquel s'ajoute le déplacement «d».

Suivant la provenance du byte que représente «s», les codes mnémoniques sont les suivants :

$$a) A \leftarrow A \wedge r$$

avec :
registre r



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

Exemple :

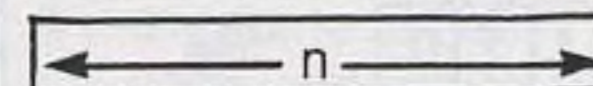
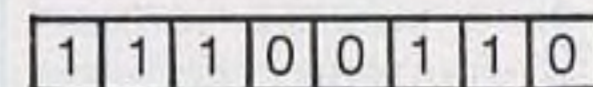
Effectuer un ET logique entre le contenu de l'accumulateur et le registre D, a pour code A2.

Si A contenait 1001 1101 et le registre D 1010 0111, après exécution de l'instruction A2, l'accumulateur contient 1000 0101.

Le bit «S» est mis à 1. Les autres remis à «0».

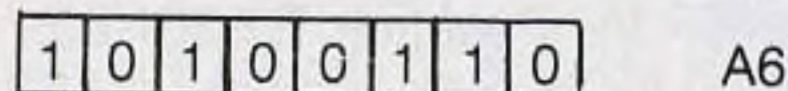
$$b) A \leftarrow A \wedge n$$

La donnée «n» suit immédiatement le code opératoire.



c) $A \leftarrow A \wedge (\text{HL})$

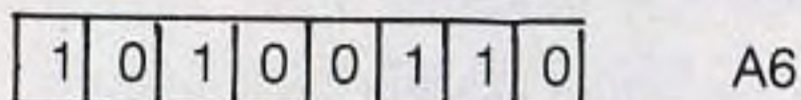
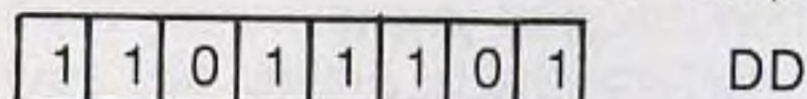
L'opérande «s» est la donnée contenue dans la case mémoire pointée par la paire de registres HL.



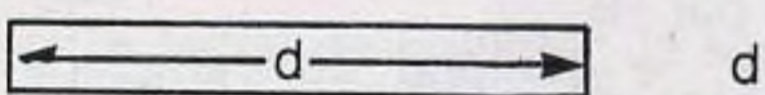
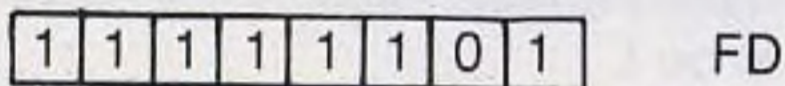
d) $A \leftarrow A \wedge (\text{IX} + d)$ ou $A \leftarrow A \wedge (\text{IY} + d)$

L'opérande «s» est la donnée contenue dans la case mémoire dont l'adresse est le contenu de la paire de registres IX et IY auquel est ajouté le déplacement «d» spécifié dans le code opératoire.

$A \leftarrow A \wedge (\text{IX} + d)$



$A \leftarrow A \wedge (\text{IY} + d)$



IV.3. OU logique

Le code opératoire général de l'opération logique OU est :

$$A \leftarrow A \vee s$$

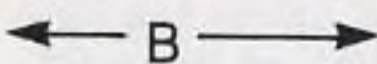
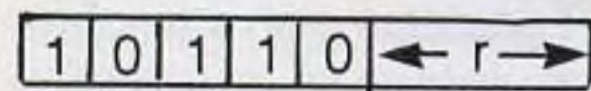
(V signifie OU logique)

(Nota : le signe «V» est équivalent au symbole «+»).

dans lequel «s» désigne l'un des opérandes comme dans le cas du ET logique. Nous indiquons les codes mnémoniques dans les différents cas :

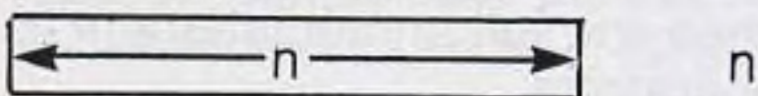
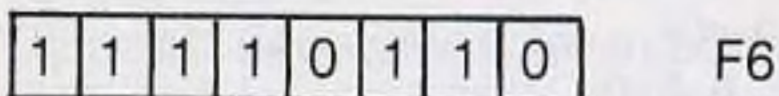
a) $A \leftarrow A \vee r$

avec :
registre r

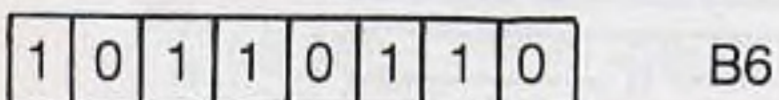


A	1 1 1
B	0 0 0
C	0 0 1
D	0 1 0
E	0 1 1
H	1 0 0
L	1 0 1

b) $A \leftarrow A \vee n$

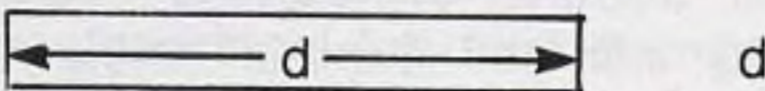
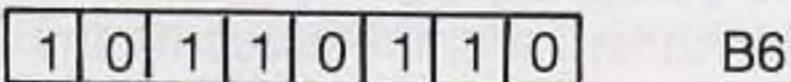
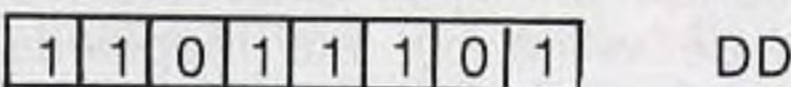


c) $A \leftarrow A \vee (\text{HL})$

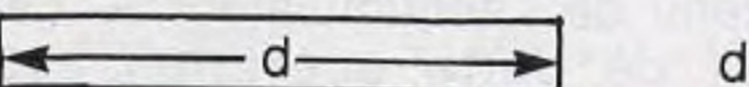
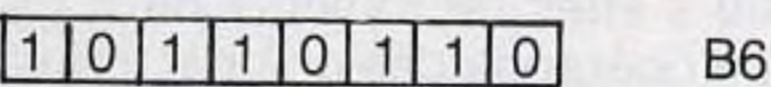


d) $A \leftarrow A \vee (\text{IX} + d)$ ou $A \leftarrow A \vee (\text{IY} + d)$

$A \leftarrow A \vee (\text{IX} + d)$



$A \leftarrow A \vee (\text{IY} + d)$



IV.4. OU exclusif

Le code opératoire général de l'opération logique OU exclusif est :

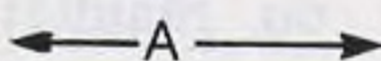
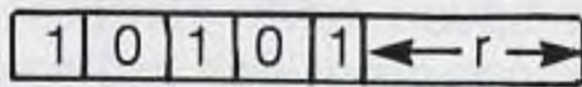
$$A \leftarrow A \oplus s$$

(\oplus signifie OU exclusif)

dans lequel «s» désigne l'un des opérandes comme dans le cas du ET logique. Nous indiquons les codes mnémoniques dans les différents cas.

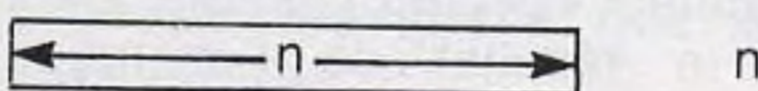
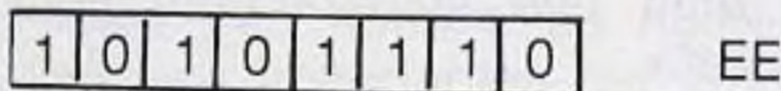
a) $A \leftarrow A \oplus r$

avec
registre r

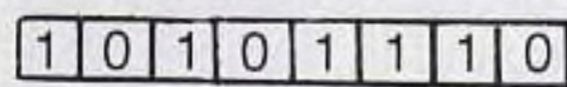


A	1 1 1
B	0 0 0
C	0 0 1
D	0 1 0
E	0 1 1
H	1 0 0
L	1 0 1

b) $A \leftarrow A \oplus n$

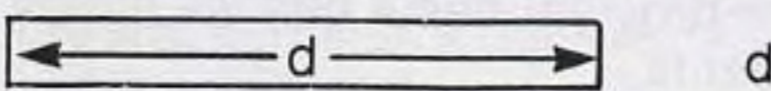
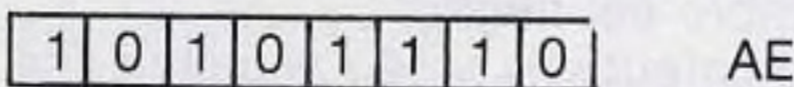
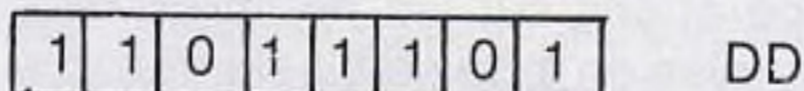


c) $A \leftarrow A \oplus (\text{HL})$

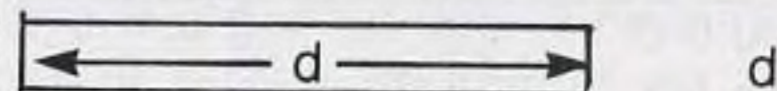
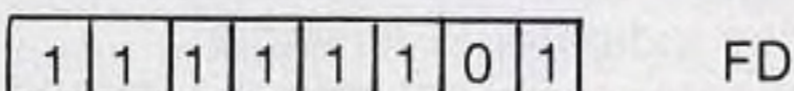


d) $A \leftarrow A \oplus (\text{IX} + d)$ ou $A \leftarrow A \oplus (\text{IY} + d)$

$A \leftarrow A \oplus (\text{IX} + d)$



$A \leftarrow A \oplus (\text{IY} + d)$



IV.5. Addition arithmétique (sans report)

Le code opératoire général de l'opération arithmétique addition est :

$$A \leftarrow A + s \text{ ou } \text{ADD } A, s$$

dans lequel «s» désigne l'un des opérandes suivants :

r : désigne l'un des registres A, B, C, D, E, H ou L

n : une donnée de 8 bits

(HL) : le contenu de l'emplacement pointé par (HL)

(IX + d) : le contenu de l'emplacement pointé par IX auquel s'ajoute le déplacement «d»

(IY + d) : le contenu de l'emplacement pointé par IY auquel s'ajoute le déplacement «d».

Suivant la provenance du byte que représente «s», les codes mnémoniques sont les suivants :

a) $A \leftarrow A + r$ ou $\text{ADD } A, r$

1 0 0 0 ← r →	A	1 1 1
← 8 →	B	0 0 0
	C	0 0 1
	D	0 1 0
	E	0 1 1
	H	1 0 0
	L	1 0 1

Exemple :

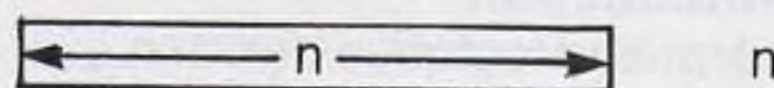
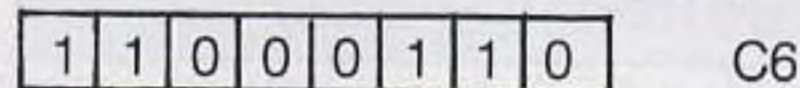
L'instruction «additionner au contenu de l'accumulateur A le contenu du registre H» a pour code 84.

Si A contient 3 BH (59)_d et le registre H contient 79 H (121)_d après exécution de l'instruction 84, l'accumulateur contient B4 H (180)_d.

L'indicateur «S» est à 1 tandis que tous les autres sont à «0».

b) $A \leftarrow A + n$ ou $\text{ADD } A, n$

La donnée «n» à additionner suit immédiatement le code opération.



Exemple :

Le registre A contient 43 H (67)_d, après exécution de C6 1E, le registre A contient 61 H (97)_d. Aucun indicateur n'est à 1.

c) $A \leftarrow A + (\text{HL})$ ou $\text{ADD } A, (\text{HL})$

La donnée à additionner est le contenu de l'emplacement pointé par HL. Le code opération est :

1 0 0 0 0 1 1 0

86

Exemple :

Le registre A contient DDH (221)_d, le contenu de la paire de registres HL est 18FE H et le contenu de l'emplacement 18FE H est 4B H.

Après exécution de l'instruction 86, le contenu de A est de 2 BH (43)_d.

Les indicateurs P/V et C sont mis à 1.

d) $A \leftarrow A + (IX + d)$ ou $A \leftarrow A + (IY + d)$
La donnée à additionner est le contenu de l'emplacement pointé par le contenu de la paire de registres IX ou IY auquel est ajouté le déplacement «d».

Les codes opératoires sont :

$A \leftarrow A + (IX + d)$

1 1 0 1 1 1 0 1

DD

1 0 0 0 0 1 1 0

86

← d →

d

$A \leftarrow A + (IY + d)$

1 1 1 1 1 1 0 1

FD

1 0 0 0 0 1 1 0

86

← d →

d

Exemple :

L'accumulateur contient 32 H (50)_d, le registre d'index IY contient 18 00 et l'emplacement mémoire 18 40 H contient 19 H ; après exécution de FD 86 40 ou $A \leftarrow A + (IY + 40)$ l'accumulateur contient 4B H. Tous les indicateurs sont à «0».

IV.6. Addition arithmétique avec report

Le code opératoire général de l'opération arithmétique addition avec report (c'est-à-dire en tenant compte du bit C du registre F) est :

$A \leftarrow A + s + Cy$

L'opération ADC (addition avec report) est identique à l'opération ADD, si ce n'est que dans la première il est tenu compte du bit C qui est 0 ou 1.

Nous présentons sous la forme d'un tableau comparatif les codes mnémoniques des différentes instructions.

Remarque :

Cette représentation a l'avantage de faire ressortir les différentes similitudes dans l'élaboration des codes mnémoniques. Essayez de les trouver vous-même.

ADD A, s $A \leftarrow A + s$	ADC A, s $A \leftarrow A + s + Cy$
ADD A, r 10 000 ← r →	ADC A, r 10 001 ← r →
ADD A, n 11 000 110 ← n →	ADC A, n 11 001 110 ← n →
ADD A, (HL) 10 000 110	ADC A, (HL) 10 001 110
ADD A, (IX+d) 11 111 101 10 000 110 ← d →	ADC A, (IX+d) 11 111 101 10 001 110 ← d →
ADD A, (IY+d) 11 111 101 10 000 110 ← d →	ADC A, (IY+d) 11 111 101 10 001 110 ← d →

Tableau I

IV.7. Positionnement de l'indicateur C

Nous venons de voir dans les additions avec report que l'indicateur C (registre F) pouvait entrer dans l'exécution des opérations arithmétiques entre autres. Il faut donc pouvoir contrôler son contenu. Pour cela le Z80 possède deux instructions :

- a) mise à «1» de l'indicateur «C»
- b) complémentation de l'indicateur «C».

a) Mise à 1 de l'indicateur C «SCF»

Le mnémonique de cette instruction est «SCF» ou C←1 et son code opération : 37.

Description :

L'indicateur de report C (bit 0 du registre F) est mis à 1 après exécution de cette instruction.

b) Complémentation de l'indicateur «C»

Le mnémonique de cette instruction est «CCF» ou C←C et son code opération : 3F.

Description :

Le bit 0 du registre F (indicateur C) est remplacé par son complément. S'il était «1» avant l'exécution de l'instruction, il devient «0». S'il était «0», il devient «1».

IV.8. Soustraction arithmétique (sans report)

Le code opératoire général de l'opération arithmétique soustraction est :

$A \leftarrow A - s$ ou SUB A, s

dans lequel «s» désigne l'un des opérandes comme indiqué dans l'addition arithmétique.

Les règles de positionnement des indicateurs sont identiques à l'exception de l'indicateur N qui est systématiquement mis à 1 puisqu'il s'agit d'une opération de soustraction.

Les codes mnémoniques sont donnés par le tableau II.

IV.9. Soustraction arithmétique (avec report)

Le code opératoire général de l'opération arithmétique soustraction avec report est :

$A \leftarrow A - s - Cy$ ou SBC A, s

dans lequel «s» désigne l'un des opérandes.

Les règles de positionnement des indicateurs sont identiques à l'opération soustraction arithmétique.

Les codes mnémoniques sont donnés par le tableau I.

SUB A, s $A \leftarrow A - s$	SBC A, s $A \leftarrow A - s - Cy$
SUB r 1001 0 ← r →	SBC r 1001 1 ← r →
SUB n 1101 0110 ← n →	SBC n 1101 1110 ← n →
SUB (HL) 1001 0110	SBC (HL) 1001 1110
SUB (IX+d) 1101 1101 1001 0110 ← d →	SBC (IX+d) 1101 1101 1001 1110 ← d →
SUB (IY+d) 1111 1101 1001 0110 ← d →	SBC (IY+d) 1111 1101 1001 1110 ← d →

Tableau II

Remarque :

Comme dans la présentation des opérations d'addition, ce tableau permet de faire ressortir quelques similitudes dans la constitution des codes. Nous invitons vivement le lecteur à effectuer un parallèle avec le tableau I.

Exemples :

1) L'accumulateur A contient 85 H (133)_d et le registre E 48 H (75)_d, après exécution de l'instruction 93 ($A \leftarrow A - E$), le contenu de A sera 3A H (58)_d. Les indicateurs N et H sont à «1», les autres à «0».

2) L'accumulateur A contient 3A (186)_d, l'indicateur C est à 1, la paire de registres HL contient 20 FE H et la case mémoire 20 FE, 55 H, après exécution de l'instruction 9E, le contenu de l'accumulateur sera 64 H (100)_d. Seul l'indicateur N sera positionné.

IV.10. Comparaison

Cette instruction de «comparaison» permet d'établir si la quantité désignée par l'opérande «s» est identique à celle contenu dans l'accumulateur. Le code opératoire de cette opéra-

tion est : A - s dans lequel «s» désigne l'un des opérandes comme indiqué dans l'addition arithmétique.

Cette opération est similaire à la soustraction, à ceci près que le résultat n'apparaît pas dans l'accumulateur, mais dans le registre F. Si la comparaison est vraie, le bit Z est 1 (puisque A - s donne 0) tandis que si la comparaison est différente Z = 0 ; c'est cet indicateur qui donne le résultat de la comparaison.

Les autres indicateurs sont positionnés comme dans la soustraction.

Les codes mnémoniques sont donnés par le tableau suivant :

CP A-s		
CP r	1 0 1 1 1 ← r →	B
CP n	← n → 1 1 1 1 1 1 1 0	FE
CP (HL)	1 0 1 1 1 1 1 0	BE
CP (IX+d)	1 1 0 1 1 1 0 1 ← d →	DD BE
CP (IY+d)	1 1 1 1 1 1 0 1 ← d →	FD BE

Tableau III

Exemple :

L'accumulateur A contient FA H et le registre B contient 86 H, après exécution de l'instruction B8 (CP A et B), tous les indicateurs sont à 0 sauf N.

IV.11. Opérations particulières sur l'accumulateur

Nous allons décrire trois instructions particulières qui ne portent que sur le contenu de l'accumulateur et qui facilitent les opérations arithmétiques.

1. Ajustement décimal de l'accumulateur «DAA»

Le mnémorique de cette instruction est DAA et son code opération : 27.

Description :

Pour bien comprendre cette instruction, nous allons traiter un exemple. Jusqu'à présent nous avons dit que les opérations arithmétiques ne pouvaient s'effectuer que sur des quantités exprimées en binaire.

L'instruction DAA permet de réaliser des opérations arithmétiques sur des quantités exprimées en B.C.D (Données et résultat).

Soit à additionner les nombres BCD 59 et 22, l'arithmétique décimale donne :

$$\begin{array}{r} 59 \\ + 22 \\ \hline 81 \end{array}$$

L'arithmétique binaire donne :

$$\begin{array}{r} 01011001 \\ + 00100010 \\ \hline 01111011 = 7B \end{array}$$

Le résultat 7B est équivoque : il fait apparaître notamment un caractère (B) qui est «interdit» en BCD.

Ajoutons 06 (0000 0110) au résultat. Ce qui donne :

$$\begin{array}{r} 01111011 \\ + 00000110 \\ \hline 10000001 = 81 \end{array}$$

Le résultat (code BCD) 81 est maintenant correct.

Conclusion :

L'instruction DAA (Decimal Adjust Accumulator) a pour but d'effectuer conditionnellement les corrections nécessaires sur le contenu de l'accumulateur après les opérations arithmétiques (addition et soustraction) pour obtenir un résultat en BCD.

La valeur corrective ainsi ajoutée au cours de l'instruction DAA, dépend des 2 demi-octets (ou quartets) contenus dans A, comme l'indique le tableau IV.

Application :

1800	Load A, 59 H	3E	59
1802	ADD A, 22 H	C6	22
1804	DAA	27	
1805	HALT	76	

Après l'exécution de ce programme, le contenu de A est 81.

2. Valeur de signe opposé dans l'accumulateur

Le mnémorique de cette instruction est NEG et son code opération est : ED 44.

Description :

Cette instruction forme le complément à deux (ou la valeur opposée) de la quantité contenue dans l'accumulateur. Le résultat est déposé dans l'accumulateur.

L'instruction réalise l'opération A - 0 - A qui est une autre représentation.

Exemple :

AVANT

A : C4

APRES

A : 3C

Opération	Indic. C Avant DAA	Valeur du quartet de poids fort	Indice H Avant DAA	Valeur du quartet de poids faible	Quantité corrective ajoutée à «A»	Indice C Après DAA
	0	0-9	0	0-9	00	0
	0	0-8	0	A-F	06	0
ADD	0	0-9	1	0-3	06	0
ADC	0	A-F	0	0-9	60	1
INC	0	9-F	0	A-F	66	1
	0	A-F	1	0-3	66	1
	1	0-2	0	0-9	60	1
	1	0-2	0	A-F	66	1
	1	0-3	1	0-3	66	1
SUB	0	0-9	0	0-9	00	0
SBC	0	0-8	1	6-F	FA	0
DEC	1	7-F	0	0-9	A0	1
NEG	1	6-F	1	6-F	9A	1

Tableau IV

3. Complémentation de l'accumulateur

Le mnémorique de cette instruction est CPL et son code opération est : 2 F.

Description :

Cette instruction forme le complément à un (ou inversion des bits) du contenu binaire de l'accumulateur. Le résultat est déposé dans l'accumulateur.

L'instruction réalise l'opération A - A qui est une autre représentation.

Exemple :

AVANT

C4

1 1 0 0 0 1 0 0

APRES

3B

0 0 1 1 1 0 1 1

IV.12. Incrémentation et décrémentation

Les opérations «INC» et «DEC» ne sont en fait que des cas particuliers des opérations arithmétiques ADD et SUB dans lesquelles la quantité à additionner (INC) ou à retrancher (DEC) est la valeur unitaire : 1.

Le code général de l'instruction est :

INC m ou DEC m

dans lequel m désigne l'un des opérandes suivant : soit l'un des registres soit l'un des emplacements mémoires.

Les codes mnémoniques sont donnés par le tableau suivant :

a) «m» représente une quantité de 1 octet

INC m m ← m + 1		DEC m m ← m - 1		
INC r	00 ← r → 100	DEC r	00 ← r → 101	
INC (HL)	0011 0100	34	DEC (HL) 0011 0101	35
INC (IX+d)	1101 1101 0011 0100 ← d →	DD	DEC (IX+d) 1101 1101 0011 0101 ← d →	DD
INC (IY+d)	1111 1101 0011 0100 ← d →	FD	DEC (IY+d) 1111 1101 0011 0101 ← d →	FD

b) Registres 16 bits :

Incréméntation		Décréméntation		
INC «ss»	00 ss 0011	DEC «ss»	00 ss 1011	
INC IX	1101 1101 0010 0011	DD	DEC IX 1101 1101 0010 1011	DD
INC IY	1111 1101 0010 0011	FD	DEC IY 1111 1101 0010 1011	FD

Tableau V

avec «ss» : BC = 00 HL = 10
DE = 01 SP = 11

IV.13. Opérations sur 16 bits

Dans les paragraphes IV.5, 6, 8 et 9, nous avons étudié les opérations arithmétiques sur 8 bits. Le résultat était ensuite placé dans l'accumulateur.

Il est possible de réaliser des opérations arithmétiques sur 16 bits entre paires de registres. Dans ce cas, le résultat n'est plus déposé dans l'accumulateur (8 bits) mais dans la paire de registres HL ou l'un des registres index IX ou IY.

1. Addition sur 16 bits sans report

0 0 r r 1 0 0 1

avec rr : BC = 00 IX = 10
DE = 01 SP = 11

Indicateurs :

C est positionné par le report du bit 15

H est positionné en fonction du bit 7.

IY ← IY + rr ou ADD IY, rr

Description :

Le contenu de la paire de registres «rr» (BC, DE, IY ou SP) est ajouté au contenu du registre index IY et le résultat déposé dans IY.

Le code opératoire est :

1 1 1 1 1 1 0 1 FD

0 0 r r 1 0 0 1

avec rr : BC = 00 IY = 10
DE = 10 SP = 11

Indicateurs :

C est positionné par le report du bit 15.

H est positionné en fonction du bit 7.

a) HL ← HL + ss ou ADD HL, ss

Description :

Le contenu de la paire de registres «ss» (BC, DE, HL ou SP) est ajouté au contenu de la paire de registres HL et le résultat est déposé dans HL.

Le code opératoire est :

0 0 s s 1 0 0 1

avec ss : BC = 00 HL = 10
DE = 01 SP = 11

Indicateurs :

L'indicateur C est positionné par le report du bit 15 sinon remis à 0.

L'indicateur H est positionné en fonction du bit 7.

b) IX ← IX + rr ou ADD IX, rr

Description :

Le contenu de la paire de registres «rr» (BC, DE, IX ou SP) est ajouté au contenu du registre index IX et le résultat déposé dans IX.

Le code opératoire est :

1 1 0 1 1 1 0 1 DD

2. Addition sur 16 bits avec reports. ADC HL ← HL + rr + C

Description :

Le contenu de la paire de registres «ss» (BC, DE, HL ou SP) est ajouté au contenu de la paire de registres puis la valeur de l'indicateur C. Le résultat est ensuite déposé dans HL.

Le code opératoire de HL ← HL + ss + C est :

1 1 1 0 1 1 0 1 ED

0 1 s s 1 0 1 0

avec ss : BC = 00 HL = 10
DE = 01 SP = 11

Les indicateurs S, Z, P/V et C sont positionnés en fonction du résultat. N est à zéro. H est positionné en fonction du bit 11.

3. Soustraction sur 16 bits avec report

Description :

Le contenu de la paire de registres «ss» (BC, DE, HL ou SP) auquel est ajoutée la valeur de l'indicateur C est soustrait du contenu de la paire de registres HL.

Le résultat est ensuite déposé dans HL.

Le code opératoire de HL ← HL - ss - C.

1 1 1 0 1 1 0 1 ED

0 1 s s 0 0 1 0

avec ss : BC = 00 HL = 10
DE = 01 SP = 11

Les indicateurs S, Z, P/V et C sont positionnés en fonction du résultat. N est à 1, H est positionné en fonction du bit 12.

Philippe Duquesne





UTILITAIRES

PHOTO D'ECRAN

QUELLE PHOTO ?

Mais non, il ne s'agit pas de prendre votre Polaroid, de vous installer en face de votre micro-ordinateur et de mitrailler à tout va. Nous ne doutons pas du côté artistique de cette expérience mais l'intérêt informatique est plus incertain...

La PHOTO d'ECRAN, ou HARDCOPY, fait partie des fonctions de base que devraient posséder tous les micros qui se respectent. Ce n'est pas le cas et loin s'en faut.

A cette carence, il y a trois réponses :

- s'en passer ;
- créer une fonction en ASSEMBLEUR ;
- créer un programme en langage évolué (BASIC...).

La première est à exclure sinon cet article n'aurait pas de raison d'être, et puis vous semblez avoir de l'ambition.

La deuxième est au-delà de notre sujet (pour le moment).

Nous allons donc mettre au point ce programme qui vous donnera par la suite de grandes satisfactions.

BUT DE L'UTILITAIRE

L'intérêt de cette étude ne se limite pas aux possesseurs d'imprimantes... Nous allons utiliser les commandes PEEK POKE... et espionner la mémoire d'un micro ; pensez à tout ce qui devient possible dès que vous saurez communiquer en direct avec la mémoire de votre ordinateur !!

Souvent dans le déroulement de jeux, de projets scientifiques ou de gestion, on désire conserver une trace sur papier de l'état de l'écran, à un moment donné de la procédure.

Vous avez sans doute déjà eu ce désir... Dans ce cas vous faites un sous-programme qui vous transfère les résultats sur l'imprimante, ou bien vous prenez des notes. Mais ceci est parfois plus difficile, particulièrement lorsque vous avez des tableaux, des courbes ou des représentations graphiques plus complexes, voire des dessins.

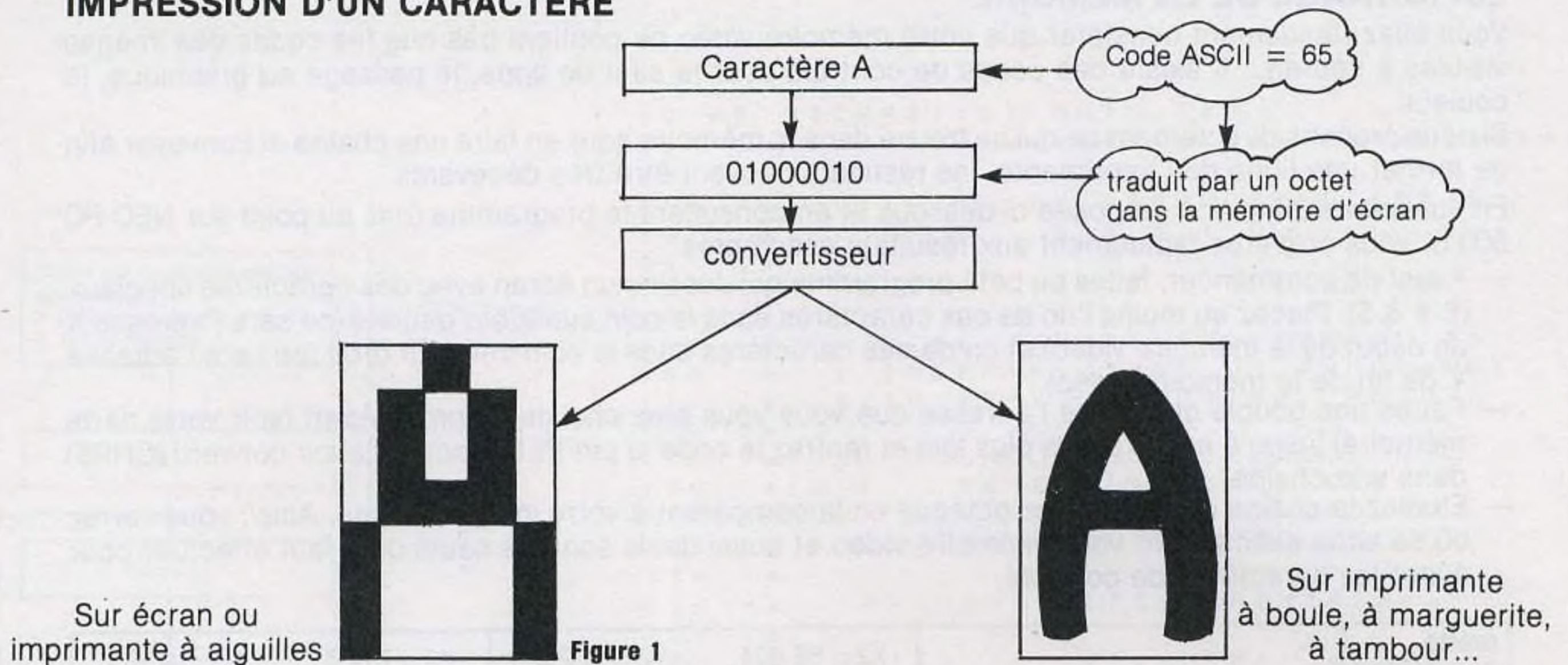
Cet utilitaire va donc recopier l'image qui se trouve à l'écran pour l'envoyer vers l'imprimante. Vous pourrez aussi conserver les données extraites de la mémoire vidéo et ainsi en fin d'exécution appeler les différents tableaux que vous aurez mémorisés... De toute façon, l'utilisation aisée de la mémoire est à la base de tous les utilitaires que nous mettrons au point dans les prochains articles.

ATTENTION !

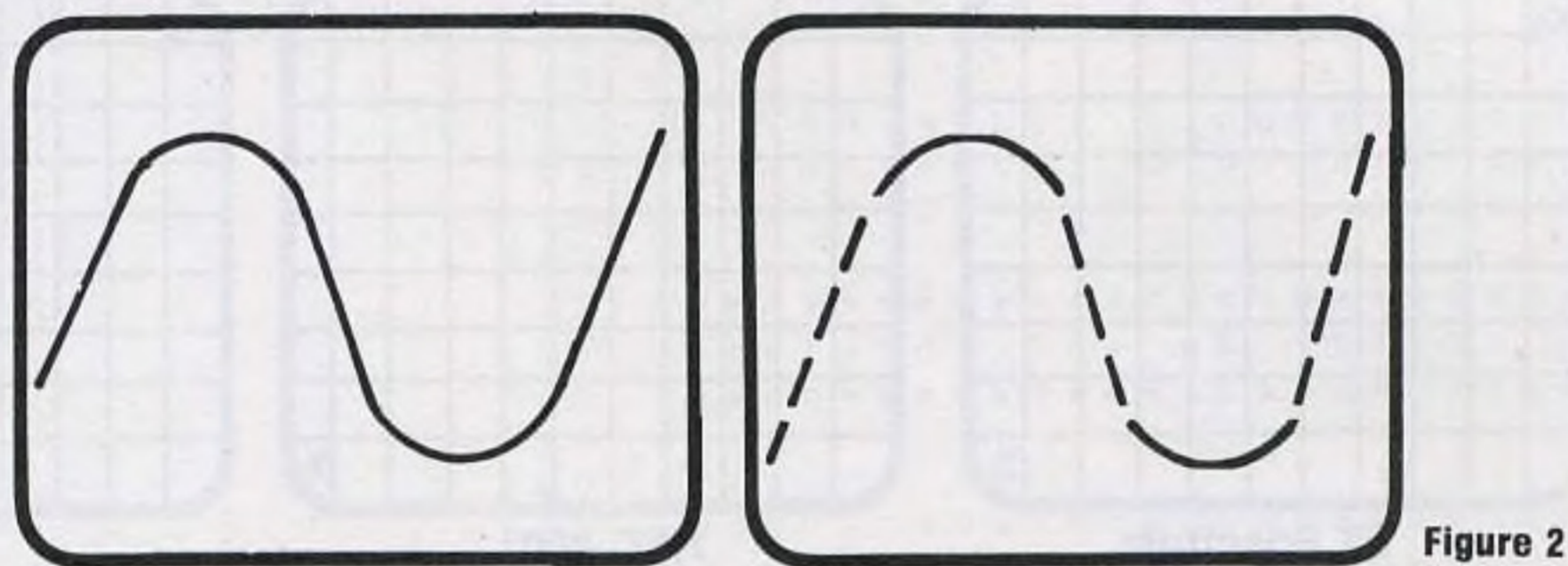
Il se peut que la PHOTO D'ECRAN sur papier présente quelques différences avec l'image cathodique. deux raisons principales à ces « anomalies » :

- Les caractères de l'imprimante (boule, cylindre...) ne sont pas identiques aux matrices de l'écran (voir figure 1 page ci-contre).
- L'interligne de l'écran ne correspond pas toujours au saut de ligne de l'imprimante (figure 2 page ci-contre).

IMPRESSION D'UN CARACTERE



IMPRESSION D'UN DESSIN



Une courbe continue sur l'écran...

... apparaîtra hâchurée par les interlignes dus aux sauts de l'imprimante.

RAPPEL SUR LES PEEK, POKE ET CHR\$

- PEEK** : C'est une commande qui va lire ce qui se trouve à une certaine adresse mémoire. Si vous faites PEEK (nnnn), vous obtenez le code du caractère qui se trouve à l'adresse nnnn. Ce code, comme nous l'avons vu au-dessus, est un nombre compris entre 0 et 255 (bornes incluses).
- POKE** : Cette commande envoie à une certaine adresse mémoire un code caractère. Si l'adresse mémoire correspond à une position de votre écran, vous obtiendrez sur l'écran, à une place déterminée, l'image du code que vous avez envoyée.
Exemple : POKE nnnn, 077 stockera le nombre 077 à l'adresse nnnn. Comme 077 est le code ASCII du M majuscule et si nnnn est une adresse de l'espace vidéo, vous verrez apparaître le M majuscule à la position nnnn.
- CHR\$** : C'est une commande de conversion qui transforme un nombre (de 0 à 255) en l'image qui lui correspond. Pour certains micros qui possèdent un synthétiseur, ce code peut correspondre à un son.
Exemple : M = CHR\$(77)
Vous faites Z\$ = CHR\$(PEEK (nnnn)), vous obtiendrez dans la chaîne Z\$ l'image qui correspond au code trouvé à l'adresse nnnn.

L'utilitaire PHOTO D'ECRAN emploiera des PEEK, des CHR\$, des PRINT, des LPRINT, la concaténation (voir LED-MICRO n° 13 §3.15.6.B page 18) et des boucles FOR... NEXT..., que tous les fortiches connaissent.

ESPIONNAGE DE LA MEMOIRE

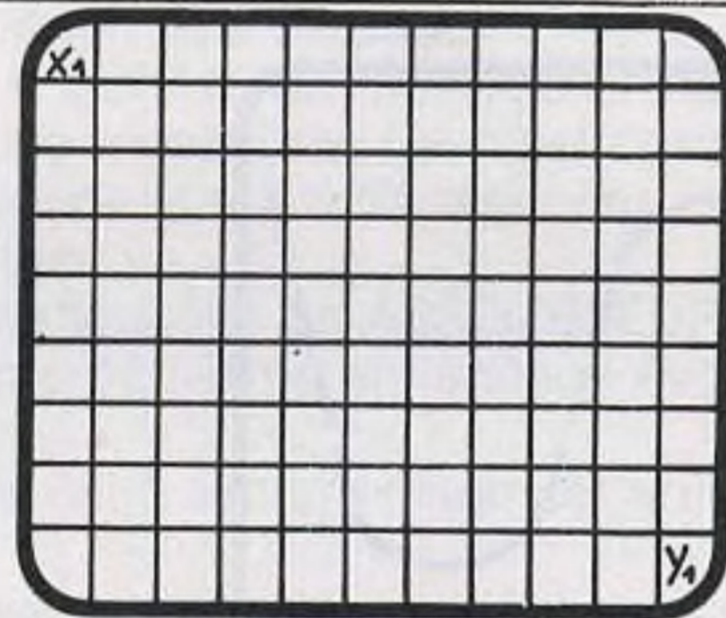
Vous allez rapidement constater que votre mémoire vidéo ne contient pas que les codes des images visibles à l'écran... il existe des codes de contrôle pour le saut de ligne, le passage au graphique, la couleur...

Si nous prenons directement ce qui se trouve dans la mémoire pour en faire une chaîne et l'envoyer afin de former une ligne de l'imprimante, les résultats peuvent être très décevants.

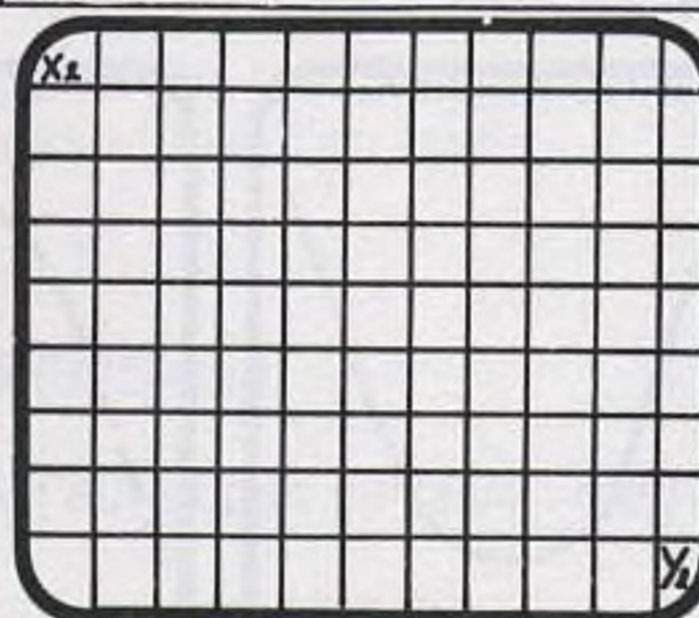
En suivant l'expérience proposée ci-dessous et en consultant le programme (mis au point sur NEC PC 8001), vous arriverez rapidement aux résultats escomptés.

- Avant de commencer, faites un petit programme qui dessine un écran avec des caractères spéciaux (£ # & \$). Placez au moins l'un de ces caractères dans le coin supérieur gauche (ce sera l'adresse X de début de la mémoire vidéo) et un de ces caractères dans le coin inférieur droit (ce sera l'adresse Y de fin de la mémoire vidéo).
- Faites une boucle qui va lire l'adresse que vous vous êtes choisie comme départ (voir votre carte mémoire) jusqu'à nn adresses plus loin et rentrez le code lu par PEEK, après l'avoir converti (CHR\$) dans une chaîne.
- Etudiez la chaîne de caractères obtenue en la comparant à votre image d'écran. Ainsi, vous verrez où se situe exactement votre mémoire vidéo et aussi quels sont les sauts qu'il faut effectuer pour éviter les caractères de contrôle.

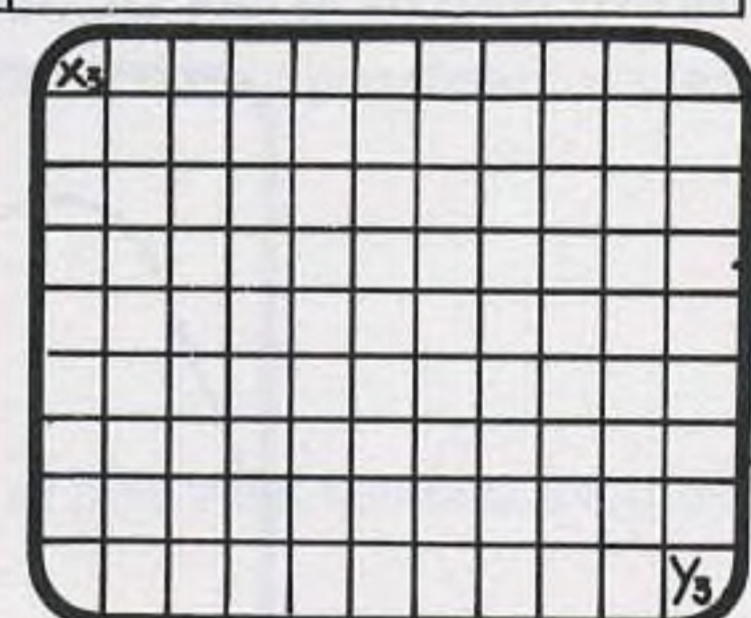
mode caractère		X2 = 59 904	Y2 = 62208	X3 = 7 680	Y3 = 8 185	
mode graphique ou couleur	X1 = 16 384	Y1 = 22 528	X2 = 59 904	Y2 = 65 160	X3 = 38 400	Y3 = 38 905



ZX Spectrum



NEC 8001

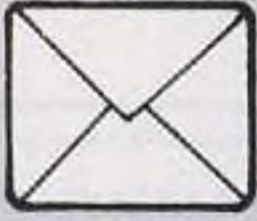


VIC 20

COMMENTAIRES SUR UN EXEMPLE (figures ci-contre)

- 410 Affectation au paramètre DE de l'adresse de départ
- 420-450 Le NEC peut travailler en plusieurs configurations d'écran (colonnes : 40, 72 ou 80 ; lignes : 20 ou 25). Ici on fait des tests pour savoir dans quelle configuration on se trouve, cela influe sur les sauts pour éviter les codes de contrôle et bien sûr la longueur de la chaîne qui correspond à une longueur d'une ligne.
- 460 Envoie sur l'imprimante du titre qui délimite la photo d'écran (facultatif) C = nb colonnes, L = nb lignes.
- 470 Début du balayage des lignes de l'écran du haut vers le bas.
- 480 Mise à blanc de la chaîne qui contient une ligne.
- 490 Début de la boucle de balayage de toutes les colonnes d'une ligne.
- 500 Si on trouve le code 0 dans la case mémoire on passe en 130.
- 510 Concaténation de la chaîne Z\$ avec la chaîne de longueur 1 qui contient l'image extraite de la case mémoire.
- 520 Saut.
- 530 Concaténation de Z\$ avec la chaîne contenant un blanc (code 0).
- 540 Fin de boucle sur toutes les colonnes d'une ligne.
- 550 La ligne 1 est entièrement rentrée dans la chaîne Z\$, on l'envoie à l'imprimante.
- 560 Fin de boucle sur toutes les lignes de l'écran.
- 570 Affichage de la fin.
- 580 Réinitialisation de l'écran (s'il avait été modifié en ligne 450).

Jean Yrytov



NOS LECTEURS NOUS ECRIVENT

■ ... Encore une fois bravo pour vos cours. Simples, clairs, précis et détaillés (...)

J'ai lu que des notions de maths seraient données dans le cours : parfait car je suis nulle. Pourquoi ne pas remplacer les cours en langage machine par une initiation aux mathématiques pour informaticiens (...).

Vous devriez paraître deux fois par mois, ou avoir un contenu double (...)

Ne cherchez pas à tout faire (articles, essais...). C'est ce que font les autres et au bout du journal, on ne sait rien de plus.

(...)

Vous dites (LED-MICRO n° 14 paragraphe 3.15.16, bas de la page 19) qu'on ne peut pas faire :

AGE ? NOM ?

(et c'est vrai car on n'a pas vu encore les CHR\$(30) ni les TAB) mais voici quand même une solution à ce problème :

```
10 INPUT "AGE"; A ; ? TAB(9)
CHR$(30) ; : INPUT "NOM"; N$
```

(CHR\$(30) étant \ ou cursor UP).

D'autre part, je vous joins mes réponses à vos exercices et un projet d'exercice.

Catherine B. 69001 Lyon

Vous posez plusieurs problèmes :

Problème 1 : le cours de maths

Je me suis mal fait comprendre. Nous ne donnerons pas de cours de formation en mathématiques mais simplement nous proposerons quelques exercices de programmation destinés à aider les matheux. En effet, il serait dommage de ne pas leur donner les

moyens d'utiliser l'ordinateur pour faciliter leurs études ou résoudre leurs problèmes et ce, avec suffisamment d'exemples et d'exercices.

C'est ainsi que nous rédigerons quelques pages permettant aux lecteurs d'écrire un programme de résolution de l'équation du troisième degré (prochain LED-MICRO) parce que ce programme est un excellent exemple de la sélection multiple, et sera très utilisé comme sous-programme dans nos futures applications C.A.O.

Inversement, LED-MICRO n'est pas destiné aux docteurs es-Sciences mathématiques : tout le monde doit pouvoir le lire.

D'où :

1. Nous distinguerons trois niveaux de connaissances en mathématiques

— sans astérisque (le plus souvent 95 % des textes et exercices) niveau : classe de troisième ;

— une astérisque : niveau de la première S ;

— deux astérisques : niveau Mathématiques Générales.

2. Les lecteurs auxquels les maths ont toujours fait pousser des boutons pourront se dispenser de lire des paragraphes du cours et de faire les exercices précédés d'un (ou de deux) astérisque(s) : ce cours est rédigé de façon à ce qu'ils ne perdent rien d'essentiel.

Problème 2 : le contenu

Paraître deux fois par mois... avoir un contenu double... Tout est théoriquement possible. Mais notre Directeur a défini et expliqué la structure qu'il donnait à sa revue dans l'éditorial du numéro 12. Veuillez vous y reporter.

Problème 3 : séries de PRINT et de INPUT sur une même ligne

Bien sûr, je pourrais vous répondre «les lecteurs n'ont pas étudié ni CHR\$(30) ni TAB(X), donc ils ne peuvent pas utiliser votre proposition».

Mais ne soyons pas hypocrites. Votre exemple est excellent et me montre que j'ai eu tort de ne pas parler de ces fonctions au moment de l'étude du PRINT : avec ce tout petit «plus» on permettait à l'élève de faire dès maintenant des présentations impeccables de programmes conversationnels sans avoir besoin d'attendre l'étude des INKEY\$ et autres GET A\$. (Il est vrai que Bruno Lilamand vous a permis d'obtenir ce résultat de façon plus puissante avec son gérant de formulaire : LED-MICRO n° 12).

Je vais me rattraper en utilisant des CHR\$(28), CHR\$(30), etc. dans des exemples de boucles. J'aurais dû également donner au moins un exemple d'une ligne comportant un «point-virgule» suivi d'un «deux points». Merci de l'avoir fait pour moi !

Encore une fois : bravo pour cette question. Une ligne beaucoup plus précieuse que des pages de philosophie.

Problème 4 : réponse à l'exercice d'application R9

Ce n'est pas ce que j'ai demandé. Je n'ai peut-être pas été assez clair.

Problème 5 : votre proposition d'exercice

Original et intéressant. Nous publierons peut-être. Attendons de recevoir d'autres propositions.

■ ... Dans vos numéros de septembre et octobre 1984, vous nous demandez notre avis sur l'environnement

de la micro-informatique (...).

Anglais

(...) S'il faut accepter de force les termes anglais pour la programmation, je m'élève énergiquement contre les notices postées en anglais ou remises dans les magasins français.

Canon X07 - ALICE

J'ai acheté un Canon X07 pour m'initier à l'informatique. Je regrette que la maison Canon impose une interface coûtant 2 300 F + 500 F d'adaptateur, soit 2 800 F pour brancher ce micro sur la télévision couleur familiale. L'interface liant le micro français «ALICE» à un téléviseur sans prise Péritel coûte 500 F et son alimentation moins de 100 F.

Olivetti

J'ai, depuis 1979, une machine à écrire à mémoire Olivetti TES 401 qui m'a rendu et me rend encore de grands services. Je peux obtenir la césure lorsque je demande une mise en page. Sauf chez IBM (et ses compatibles) je n'ai pas trouvé de traitement de texte avec césure. Par contre, le contrat de maintenance d'Olivetti n'est pas très franc : leur garantie (théoriquement de 6 mois) ne porte en fait que sur 3 mois.

(...)

SYBEX

J'ai acheté chez SYBEX deux livres :

— d'une part le «BASIC par la pratique» (60 exercices)

— d'autre part «Exercices en BASIC sur l'ordinateur Personal IBM.

Or j'ai constaté que le texte est identiquement le même, sauf quatre pages. C'est un scandale de vendre

deux livres comportant des couvertures et des titres différents avec les mêmes exercices.

LED-MICRO

(...) Pour faciliter le classement des articles, je détache les pages de chaque article et je les classe par famille. Je suis obligé de faire des photocopies pour les pages dont un côté correspond à un article et l'autre à un autre article. Ne vous serait-il pas possible de placer la publicité sur une des pages.

Pierre L. 92400 Courbevoie

Anglais

Eh oui... et pourtant il existe une loi imposant la francisation des matériels et notices.

Canon - Olivetti - Sybex

Si les firmes que vous citez nous donnent des explications, nous les publierons.

Led-Micro

Pas de chance : dans mon cours de programmation, je me donne un mal fou pour faire exactement le contraire de ce que vous souhaitez ! J'essaie de faire en sorte que le lecteur ait toutes les informations nécessaires dans une «unité pédagogique» composée des deux pages gauche-droite se faisant vis-à-vis. Ainsi le lecteur n'a pas à tourner les pages et revenir en arrière pour retrouver la figure correspondant au texte.

Un gagnant à un concours... avant la publication du règlement

Monsieur René SIPRA (il nous a autorisé à publier son nom) nous a envoyé une solution à notre exercice de récapitulation R8 absolument excellente. Je l'ai publiée dans mon cours de programmation à la place de ma rédaction.

Ce lecteur :

— d'une part gagne un abonnement à Led-Micro pour une personne de son choix (c'est la moindre des choses !)

— d'autre part il est inscrit à l'avance comme participant à notre (futur) concours du meilleur exercice pédagogique... concours dont nous donnerons le règlement début 85.

Un grand merci au samideano René Sipra (samideano signifie «personne qui a la même idée», en Esperanto).

A propos des exercices de récapitulation

+ J'ai reçu un assez grand nombre de solutions aux exercices de récapitulation (que je proposai dans le cours de programmation du numéro précédent). Merci et continuez.

+ En particulier, j'ai reçu un assez grand nombre de programmes «contrôle de vraisemblance d'une date». Là, il semble y avoir malentendu : à la page 5 de Led-Micro n° 14, je vous citai les exercices que j'avais prévus (contrôle de vraisemblance d'une date, équation du 3^e degré, etc.) en sous-entendant «envoyez-moi des sujets d'exercices de préférence autres que ceux-ci».

+ Soyez gentils avec moi : si vous n'avez pas d'imprimante, envoyez-moi vos programmes écrits en majuscules d'imprimerie et (si possible) sur du papier quadrillé. J'ai reçu quelques programmes intéressants mais que je devrais entièrement réécrire pour les comprendre. Pitié !

4 collections



Aux libraires

Nous sommes heureux de porter à votre connaissance la création d'un département ouvrages à part entière au sein des Editions Fréquences et dont le développement sera accéléré et désormais productif.

Ce département aura pour mission (dans le cadre de nos spécialités : Electronique générale, Haute Fidélité, Son Pro, Micro-informatique, Audiophilie, Photo Vidéo) d'amplifier une activité restée jusqu'alors secondaire aux Editions Fréquences et par conséquent offrant des créations peu nombreuses.

La publicité ci-contre situe les différentes collections et leur vocation.

Jusqu'à présent, notre modeste production d'ouvrages était diffusée par les Editions RADIO -SECF.

Désormais, ce sont les Editions Fréquences qui en assureront directement la distribution.

N'hésitez pas à nous contacter (même vous qui avez déjà peut-être sur vos rayons nos premières créations ?).

La plupart de nos lecteurs, nous en sommes persuadés, seront ravis de faire leurs achats comme par le passé chez leurs fournisseurs habituels.

Soyez assez aimable de nous retourner le bon ci-dessous qui nous permettra de vous expédier par retour notre catalogue ainsi que nos conditions.

Librairie :

Responsable :

Adresse :



Collection rouge "Led loisirs" (135x210)

Conseils et tours de main en électronique J. Hiraga 160 p.
Lexique de l'électronique anglais-français J. Hiraga 100 p.

EN PREPARATION... Les montages électronique "simples" B. Duval - Les montages électroniques "évolués" B. Duval - Filtres actifs et passifs pour enceintes acoustiques Ch.-H. Delaleu - Les amplis à transistors G. Le Doré - Les amplis à tubes J. Hiraga - Les circuits intégrés les plus utilisés Ch.-H. Delaleu - Le compact disque "soft et hard" G. Le Doré, C. Dartevelle, J.-L. Macia. - Les installations hifi les plus folles de 1960 à aujourd'hui J. Hiraga - Les accessoires hifi audio - La commande à distance - Les alarmes - Le guide pratique des microphones - La prise de son vidéo pour professionnels - Alignement des têtes de magnétophones - Installez votre studio de prise de son - Calcul des pavillons.

Collection verte "week-end" (135x210)

Week-end photo P. Folie-Dupart 160 p.

EN PREPARATION : Week-end Labo O. Gouriet

Week-end vidéo P. Folie-Dupart

Week-end prise de son P. Folie-Dupart

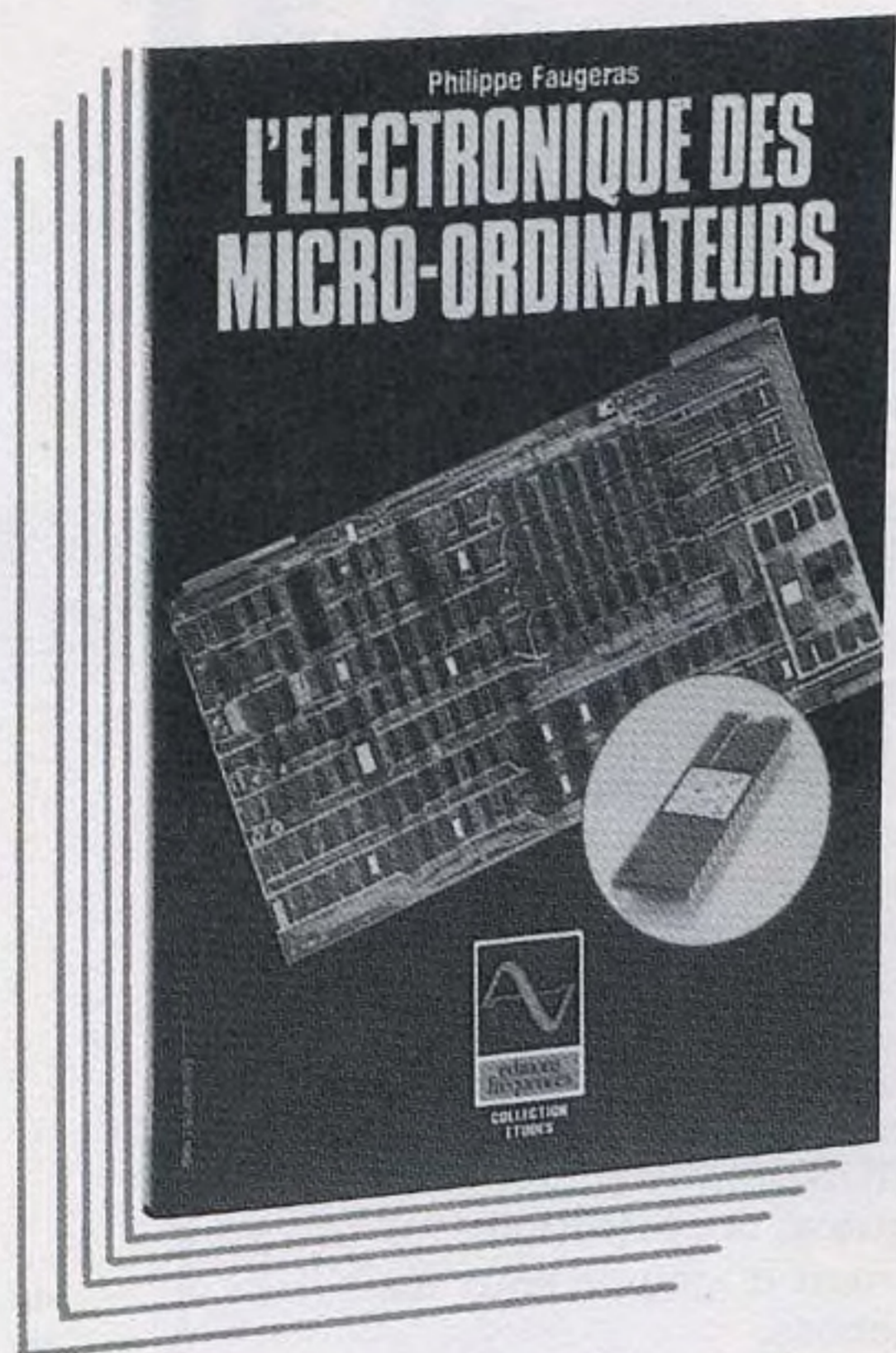


Collection jaune "pédagogique"

(210x270)
Led-Robot - Collectif d'auteurs C. Polgar

EN PREPARATION
Cours de programmation en micro-informatique C. Polgar
Cours d'électronique digitale P. Duquesne
Cours d'initiation à l'électronique des amplificateurs BF
Cours d'électronique générale

«éditions fréquences»



■ Collection noire «études»

(165 x 240)

Les magnétophones C. Gendre - L'enregistrement et les enregistreurs

Les magnétoscopes et la télévision - C. Gendre - Un livre qui fait référence dans la vidéo

Les haut-parleurs - J. Hiraga - Histoire et panorama complet des transducteurs et enceintes acoustiques

L'optimisation des haut-parleurs et enceintes acoustiques - Ch.-H. Delaleu - Le haut-parleur et l'enceinte acoustique moderne expliquée. Réalisation accessible à tous

Introduction à l'audio numérique - J.-P. Picot - La révélation d'une nouvelle technique d'enregistrement

L'électronique des micro-ordinateurs - P. Faugeras - Un ouvrage très complet accessible à tous

EN PREPARATION

Sélection de l'Audiophile - Volumes 1 et 2

Les principales études parues dans "l'Audiophile" depuis le n° 1 jusqu'au n° 15, regroupées par thèmes :

Volume 1 : L'électronique - **Volume 2 : Les transducteurs Les techniques du son** - **Volumes 1, 2 et 3**

Volume 1 : les notions fondamentales - **Volume 2** : le matériel - **Volume 3** : l'exploitation.

Collection «Led loisirs»

Les sujets abordés dans cette collection sont traités avec un souci pédagogique et dans un but de vulgarisation, pour répondre à un large public d'amateurs pour qui l'électronique dans sa pratique est souvent un barrage nécessaire à franchir pour mieux servir leur hobby.

Collection «week-end»

Cette collection vise essentiellement la pratique qui doit conduire à de bonnes photos, de bons films vidéo, etc., etc. Elle se veut claire et chaque thème est traité par des auteurs passionnés eux-mêmes par le sujet.

Collection «pédagogique»

La collection s'ingéniera à aborder des sujets souvent délaissés ou non actualisés. Led-Robot en est un exemple immédiat.

Toujours dans un esprit de clarté pratique et d'enseignement vivant, cette collection s'adresse aussi bien aux débutants qu'aux initiés même chevronnés.

Collection «études»

Les sujets abordés dans la collection "études" sont traités le plus largement possible par les meilleurs spécialistes actuels, la plupart du temps rédacteurs dans nos propres revues.

➔ **Votre libraire ne possède pas le titre qui vous intéresse, deux solutions :**

- Vous faites votre commande chez votre libraire que nous livrerons immédiatement ;
- Vous nous passez commande directement avec le bon de commande ci-dessous du ou des ouvrages qui vous intéressent, et vous les recevez par la poste.

BON DE COMMANDE

Pour recevoir l'un ou plusieurs des ouvrages ci-contre, cocher le carré correspondant au(x) titre(s) désiré(s) et remplir la case «MONTANT TOTAL».

Envoyer ce bon aux EDITIONS FREQUENCES, 1 boulevard Ney 75018 Paris, accompagné de votre règlement.

Nom :

Prénom :

Adresse :

Règlement effectué

par CCP par chèque bancaire par mandat

Collection étude	Titre désiré	Prix TTC	Frais de port	Prix port compris
LES HAUT-PARLEURS	<input type="checkbox"/>	165 F	15 F	180 F
LES MAGNETOPHONES, C. Gendre	<input type="checkbox"/>	92 F	8 F	100 F
LES MAGNETOSCOPES ET LA TELEVISION C. Gendre	<input type="checkbox"/>	145 F	15 F	160 F
L'OPTIMISATION DES H.P. ET ENCEINTES ACOUST. C.-H. Delaleu	<input type="checkbox"/>	154 F	8 F	162 F
L'INTRODUCTION A L'AUDIO NUMERIQUE J.-P. Picot	<input type="checkbox"/>	155 F	10 F	165 F
L'ELECTRONIQUE DES MICRO-ORDINATEURS P. Faugeras	<input checked="" type="checkbox"/>	150 F	15 F	165 F
Collection Led pédagogique				
LED-ROBOT - INITIATION A LA ROBOTIQUE C. Polgar	<input type="checkbox"/>	115 F	10 F	125 F
Collection Led loisirs				
CONSEILS ET TOURS DE MAIN J. Hiraga	<input type="checkbox"/>	68 F	7 F	75 F
LEXIQUE DE L'ELECTRONIQUE FR.-ANGL. J. Hiraga	<input type="checkbox"/>	65 F	7 F	72 F
MONTANT TOTAL				



LIBRE PROPOS

Il y a de plus en plus de concurrence sur le marché des micro-ordinateurs et les fabricants n'hésitent pas à utiliser des superlatifs pour glorifier leurs enfants. Toutefois, si cela est acceptable pour une majorité d'utilisateurs potentiels, il existe de plus en plus de petits «malins» qui indiquent dans leurs documentations que leur micro-ordinateur présente des dessins en trois dimensions. S'il existe un métier où en ce moment on se permet de raconter n'importe quoi, c'est bien en micro-informatique. En effet, les logiciels qui permettent le graphisme en trois dimensions sont très rares en micro-informatique, et surtout très onéreux. Quant aux langages qui autorisent la 3D directement, ils sont quasi inexistantes. Alors, pourquoi parler de trois dimensions si, dans 95 % des cas (et c'est sûrement plus) elles n'existent pas ? C'est très facile car on fait passer un dessin réalisé en deux dimensions représentant une perspective. L'acheteur, en voyant cela, est persuadé qu'il pourra réaliser d'aussi beaux dessins que sur un Computervision ou autre Tektronic. Mais voilà, le malheur veut que pour parler de trois dimensions, un système réponde à certains algorithmes. Les algorithmes les plus utiles en 3D sont : la rotation dans les trois axes (X, Y, Z), la translation, le zoom, la projection orthographique, la projection oblique et la projection perspective. A ces algorithmes de base, il convient d'ajouter pour les systèmes plus élaborés la gestion des faces cachées.

La rotation, comme son nom l'indique, permet de tourner la représentation du dessin sur les trois axes. La translation autorise un mouvement du dessin sur l'écran ou en dehors. Le zoom permet l'étude de parties précises du dessin, la projection orthographique de dessiner l'objet représenté comme sur un plan normalisé (vue de face, de dessus, de côté). La projection oblique représente le dessin incliné. Enfin, la projection perspective présente le dessin sous une simulation perspective. En effet, il s'agit de simulation perspective car contrairement aux autres possibilités, cette fonction n'est pas réversible. La seule possibilité pour revenir en arrière lorsqu'on vient de réaliser une perspective d'un dessin 3D est de le sauvegarder avant la perspective. Tous les autres concepts sont réalisables et réversibles sans astuces.

En fait, le nerf de la guerre en graphisme trois dimensions consiste en un stockage des données sur d'énormes matrices qui renferment les milliers de coordonnées X, Y, Z du dessin. Mais si chaque point est sauvegardé en fonction de sa position dans «l'espace», il convient d'ajouter aux coordonnées d'autres informations. Ainsi chaque position d'un point dans l'espace sera accompagné de trois valeurs (X, Y, Z) et de la position de la plume (crayon) : levé ou baissé, du choix de plume (couleurs). Pour réaliser un dessin complexe, il faudra donc utiliser une énorme matrice. La mémoire vive pour des applications correctes devra se situer entre 500 Koctets et 1 Mégaoctet. De plus, il sera nécessaire d'avoir un processeur très rapide car il n'est pas question d'attendre quinze jours pour que le dessin arrive, il doit être affiché instantanément. Conclusion, à chaque fois, vérifier ce que votre interlocuteur vous raconte.

Charles-Henry Delaleu

SOMMAIRE

UN PETIT VOYAGE AU PAYS DES BOUTIQUES PARISIENNES DE MICRO-INFORMATIQUE : TRÈS INTÉRESSANT ET INSTRUCTIF **P. 50.** LA CONTRE-MESURE DU MOIS : UN NOUVEAU JAPONAIS AU STANDARD MSX, LE YASHICA YC-64 **P. 56.** POUR JOUER AU PÈRE NOËL, UNE SÉLECTION DE PRODUITS MATÉRIELS ET LOGICIELS **P. 60.** UNE SÉLECTION DE LIVRES A LIRE ET A OFFRIR **P. 64.**

D'UNE BOUTIQUE A L'AUTRE sourire, compétence et incompétence

Un après-midi de novembre. Un temps doux, propice à la promenade, idéal pour faire un tour dans les boutiques de micro. Intéressant et instructif ce petit voyage au pays de la micro. Ce n'est pas une enquête mais un croquis pris sur le vif qui permet de dégager quelques clés pour bien acheter.

Nous sommes parties sans idées préconçues, limitant notre domaine de chasse micro au cœur de Paris. Et bien évidemment, nous n'avons pu rendre visite qu'à un petit nombre. Suffisant, néanmoins pour se faire une idée.

HBN : LE TOUT ELECTRONIQUE

Première étape, HBN, boulevard Magenta. Un magasin tout neuf, sans luxe inutile mais bien agréable. Ici la micro fait bon ménage avec l'électronique. Paradis du bricoleur, ce magasin fait partie d'une chaîne de boutiques installées un peu partout en France et qui vendent des composants électroniques, des haut-parleurs, des appareils de mesures, enfin tout ce qui est nécessaire pour effectuer des réalisations électroniques. Si vous voulez faire votre propre décodeur pour Canal-Plus, vous trouverez chez HBN tout ce qu'il faut, sauf le plan, bien entendu. Mais tel n'est pas l'objet de notre visite. Nous étions entrées dans ce magasin parce qu'en vitrine étaient présentés des micro-ordinateurs Thomson.

En attendant le vendeur, nous avons passé quelques minutes à feuilleter les

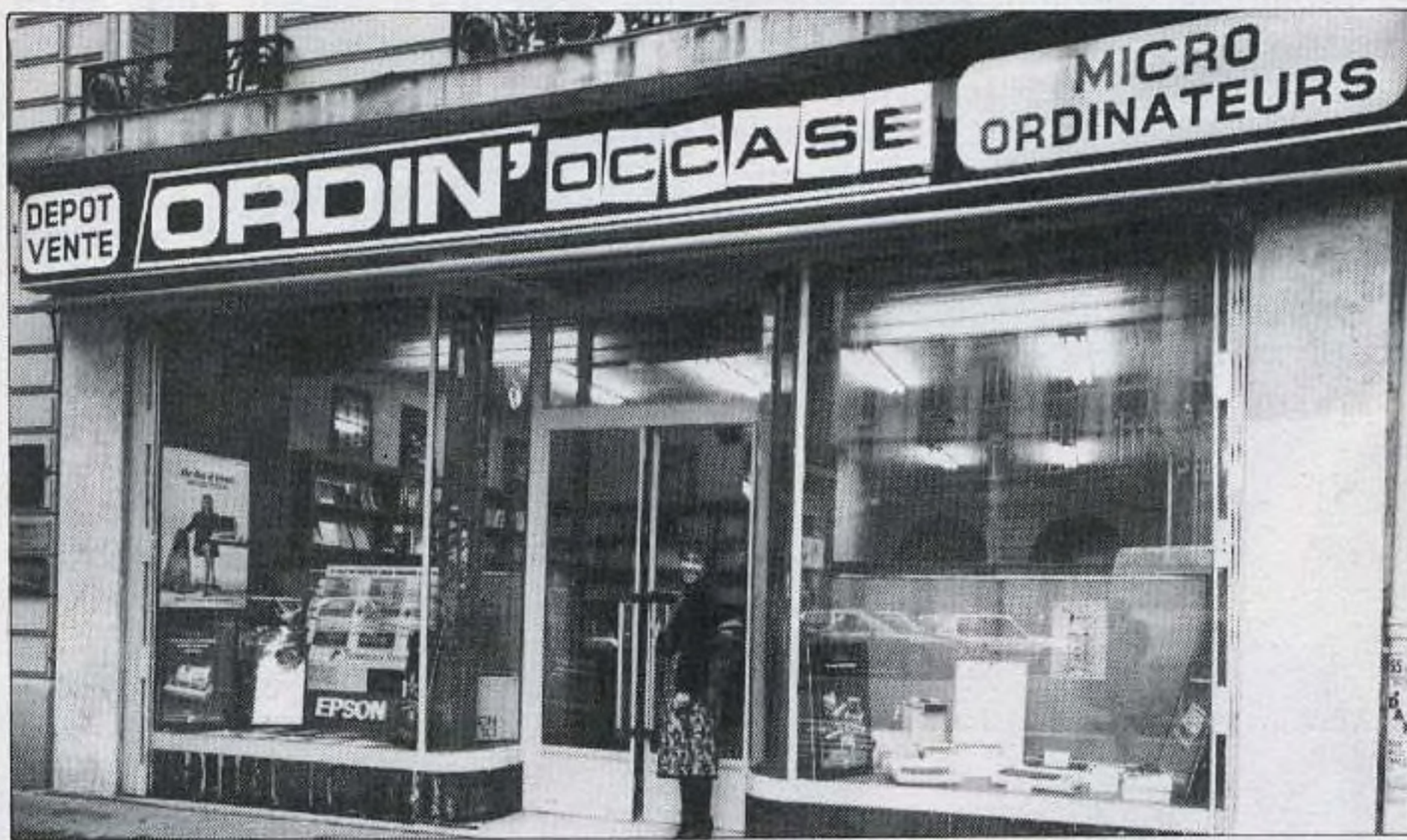


ouvrages du rayon librairie. Un rayon bourré d'ouvrages techniques qui, pour les néophytes que nous sommes, étaient hors de portée de notre compréhension, (il faut vous dire que nous sommes, pour la circonstance, deux femmes en quête d'un micro pour leur neveu et que nous avons fort peu de connaissances en micro-informatique). Le vendeur nous oriente tout de suite vers un M05 nous expliquant simplement que cette machine convenait parfaitement à un adolescent qui veut s'initier à la micro. Pas tout à fait innocemment, nous nous fixons sur le Guépard (ce micro maison dont nous vous avons déjà parlé). Et le vendeur de nous démontrer que cette machine très performante pour son prix con-

vient certes, fort bien à un amateur mais faut-il encore qu'il ait déjà un bon niveau de connaissances. Il vaut mieux, dit-il, commencer par une machine simple, car il n'est pas dit que votre neveu se passionne vraiment pour la micro. Un sage conseil. Nous reviendrons.

POURQUOI PAS L'OCCASION

Un peu plus loin, nous nous sommes arrêtées à Ordin'Occase. Cette boutique avait attiré l'attention sur elle à l'occasion du Sicob, en distribuant aux visiteurs de la «boutique micro» des stylos portant son adresse. Ordin'Occase pratique le dépôt-vente en se ménageant une commission de



20 % sur le montant de la revente et ne rachète jamais de matériel. Lorsque la machine n'est plus commercialisée, Ordin'Occase pratique une décote de 40 à 50 % sur le prix du neuf. En dépôt, on trouvait des Apple, des Oric-Atmos et des Commodore 64 et Vic 20. Ces machines restent peu en dépôt, moins d'une semaine en moyenne. Il est également possible de réserver une machine par téléphone. Dans ce cas, elle est immédiatement retirée de la vente.

Constatant notre incompetence en micro-informatique, la responsable du magasin nous a annoncé que prochainement, elle proposerait des cours d'initiation pour parents dépassés ! Une bonne idée, il y a tant de parents dans ce cas. Ordin'Occase vend également du matériel neuf, l'Epson 0X10, une machine professionnelle qui ne correspond pas à notre projet.

ble des micros présents, nous expliquant les possibilités de chacun. Ainsi, nous partons du VG5000 de Philips puis successivement nous sont présentés l'Excel 100, l'Atmos d'Oric, le Commodore 64, le QL de Sinclair (il nous a assuré en avoir une dizaine en



stock, sans aucun doute importée d'Angleterre, car la filiale française de Sinclair ne commercialisera pas avant le premier trimestre 85, la version francisée du QL). Peu de détails sur les

Spectravidéo au standard MSX, tout nouvellement arrivé. Pour expliquer l'Amstrad, une comparaison imagée : il y a la même différence qu'entre un camping car et qu'une caravane. L'Amstrad, comme le camping car, regroupe tout (magnétocassette, moniteur), le micro classique, comme une caravane, s'attelle à la voiture de votre choix, se connecte à la télé ou à un moniteur de votre préférence.

Le vendeur nous a conseillé un Atmos (2 490 F), un micro évolutif ou mieux encore le Commodore 64, une valeur sûre qui bénéficie d'une librairie de logiciels importante. En revanche, pour lui, l'Aquarius et le Laser 200 sont des machines dépassées. Au passage, il n'a pas hésité à nous donner quelques informations sur le Macintosh (environ 25 000 F).

Que retenir de cette visite ? un vendeur aimable qui prend le temps d'expliquer, qui ne pousse pas à l'achat et qui a un honnête niveau de connaissances. Une adresse à retenir.

DOMESTIC COMPUTER : UNE AMBIANCE AGREABLE

Arrêt suivant : Domestic Computer, rue de Clichy. Un petit détail, ce magasin porte toujours l'enseigne International Computer. Ici l'ambiance est chaude, sans doute en raison de la décoration dans les tons de marron et or qui tranche sur celle des boutiques micro assez sévères généralement. La vendeuse a une démarche logique et demande dès le début de l'entretien le but — initiation ou jeu — et le budget. Pour elle, l'Excel 100 est avant tout

JCR : UN VASTE ASSORTIMENT

Poursuivant notre périple, nous nous sommes arrêtées chez JCR, rue Notre-Dame-de-Lorette. Une boutique vivante où les micros foisonnent comme les livres et les logiciels. Un client pianote sur un Amstrad. Nous attendons quelques minutes et un vendeur nous aborde. Devant notre incompetence avouée, il n'hésite pas à nous faire une présentation de l'ensem-



BOUTIQUE

une machine de jeux grâce à la reconnaissance et plus destinée aux jeunes enfants. En initiation, elle propose l'Atmos, le Commodore 64 parce que ces machines ont de nombreux logiciels et périphériques qui ne limitent pas leurs utilisations. A noter également qu'elle n'a pas omis de nous demander si le téléviseur familial disposait d'une prise Péritel, sinon il nous en coûtera 550 F supplémentaires pour une interface télévision. Elle nous a également proposé l'Alice dans sa valise rouge, sans insister d'ailleurs. Dans ce magasin, nous avons également vu le Memotech (3 400 F dans sa version 64 K et 4 300 F dans sa version 128 K). Une machine, nous a dit la vendeuse, qui a l'avantage de pouvoir évoluer en fonction des besoins et des disponibilités financières et qui se place en concurrence de l'Apple II dans sa version 128 K (vendu également par le magasin). Là encore, en fonction de notre budget — environ 3 000 F — on nous a conseillé l'Atmos et le Commodore 64.

Décidément, les magasins parisiens font dans le classique. Les dernières nouveautés sont présentées, mais sans plus.

SIVEA : UNE OFFRE REDUITE

Avant-dernière étape, Sivéa, boulevard de Batignolles. Un magasin dépeuplé, en machines et en clients. Un vendeur présentait à un client une gamme Apple avec force détails. Aussi quelle ne fut pas notre surprise d'entendre le discours tenu par notre vendeuse. Elle nous propose un T07,

en promotion, en nous disant que le magasin arrêta la vente des machines Thomson parce que cette société avait une politique commerciale déplorable (entre autre que la garantie porte uniquement sur les pièces et non sur la main-d'œuvre). Avec désinvolture, elle a ajouté que pour la maintenance, nous n'aurions qu'à nous adresser à un



confrère. Autre, l'Atari 800XL en promotion à 8 000 F avec ses périphériques : table graphique, imprimante, lecteur de cassette, le Commodore 64 (3 900 F) en version Pal auquel il faut ajouter une interface Pal/Secam (550 F). Une offre très réduite à laquelle il faut adjoindre l'Oric 1 (dont chacun sait qu'il est remplacé par l'Atmos), le Laser 200 à 2 495 F avec le lecteur de cassettes. A notre question : avez-vous l'Excel 100 dont une publicité passe à la télévision ? C'est quoi, un jeu, moi je ne sais pas. Et le MXS ? Connais pas. Surprenant pour le moins. Si nous avons commencé notre périple par ce magasin, nul doute

que nous n'aurions eu guère envie d'acheter, ou même d'aller plus avant.

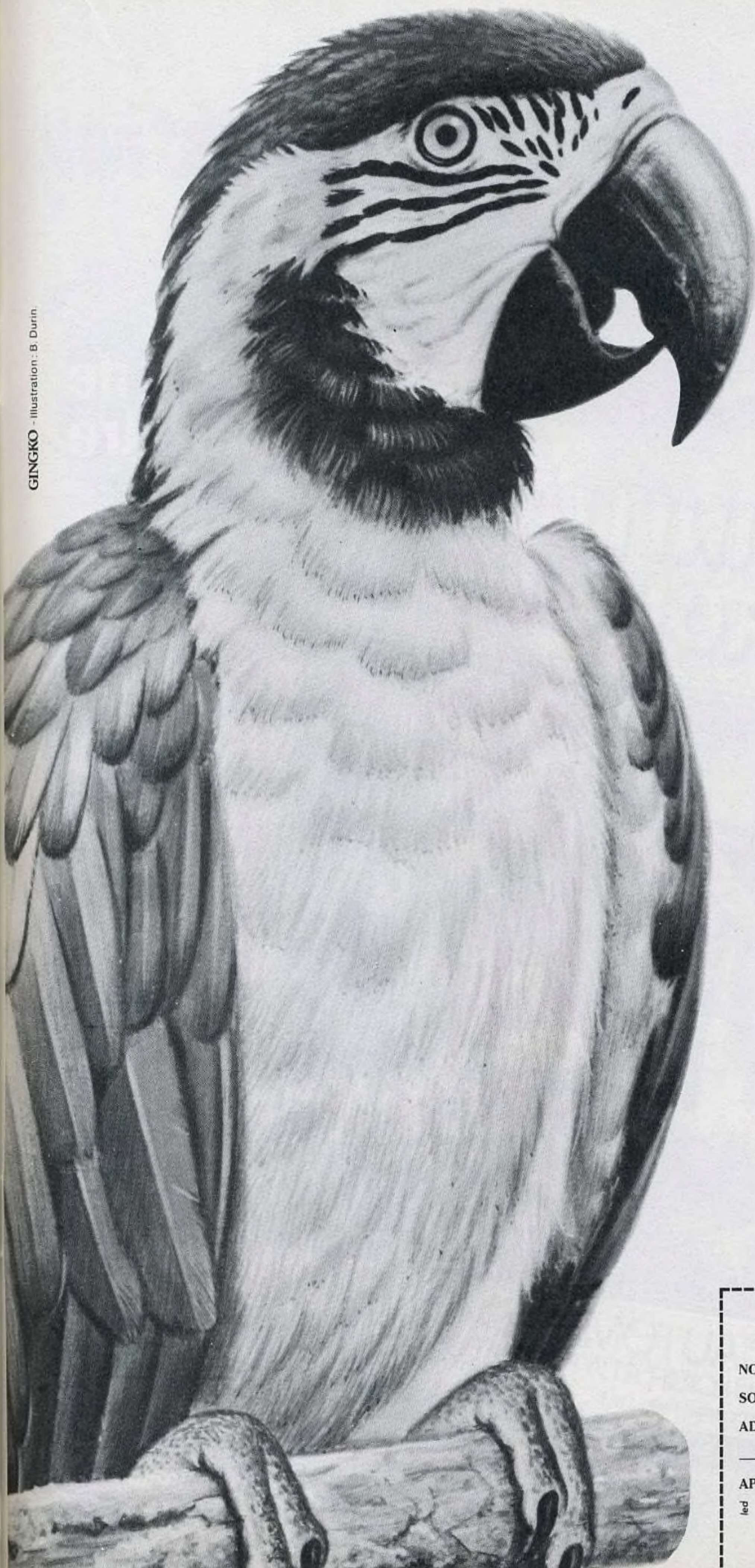
ESPACE MICRO : EINSTEIN ET AUTRES

Dernière étape, Espace Micro, rue de Chateaudun. Un magasin clair, très simple, sans décoration : les micros sont présentés branchés sur une longue étagère courant le long du mur, avec en face de chacun une chaise. Le choix est large : deux MSX, le Sanyo à 2 990 F, le Yeno à 3 450 F, l'Excel 100 à 3 190 F, voici pour les nouveaux. Et puis, l'Atmos, le Dragon, le Commodore 64, les Thomson T07-70, M05, le Spectrum. N'oublions pas l'Einstein pour amateur éclairé et à la bourse bien garnie. Une machine qui, incontestablement, offre de vastes possibilités. A signaler que ce magasin n'hésite pas à proposer un crédit même sur un M07. Pendant notre visite, un client intéressé par l'Excel 100 s'est vu proposé un crédit, solution qui, d'ailleurs, l'a séduite et a fait craquer ses dernières hésitations.

Nous avons constaté que, d'un magasin à l'autre, les prix se tiennent ; les différences sont très faibles. Le discours des vendeurs diffère peu. Ils expliquent peu les caractéristiques techniques et semblent tous croire que les clients savent ce qu'est un octet. Ils se bornent à savoir qu'elle est l'utilisation recherchée et le budget. Ce qui les différencie : l'amabilité et la capacité à répondre aux questions.

Lorsque vous partez en quête d'un micro, un conseil, sachez bien ce que vous voulez. Faites-vous une liste de questions (capacité mémoire, possibilités d'extension, logiciels disponibles, garantie, service après-vente, etc.). N'hésitez pas à demander une documentation. Nous avons remarqué que les vendeurs ne la donnent pas systématiquement. N'oubliez pas que l'unité centrale n'est qu'un élément et que vous aurez besoin assez rapidement de périphériques comme le lecteur de cassettes (certains micros comme le Commodore 64 exigent un lecteur spécifique à la marque), d'une imprimante. Tenez-en compte lors de l'établissement de votre budget, de même que du coût du crédit.





EXCUSEZ-MOI DE ME RÉPÉTER MAIS LE VICTOR S1 PARLE TOUTES LES LANGUES.

Rarement on aura vu aussi doué pour les langages informatiques les plus courants. Muni d'un haut-parleur, il est doté de possibilités vocales.

De plus, il donne la possibilité d'interconnecter d'autres postes de travail Victor et périphériques.

Mieux encore : il peut communiquer avec les gros systèmes (IBM, DEC, BULL...).

Renseignez-vous, comparez-nous. Nous sommes sûrs que nous sommes appelés à nous rencontrer.

VICTOR TECHNOLOGIES



**JE VOUDRAIS QUE VOUS ME PARLIEZ
PLUS LONGUEMENT DU VICTOR S1.**

NOM _____

SOCIÉTÉ _____ FONCTION _____

ADRESSE _____

TÉL. _____

APPLICATION _____

led

VICTOR TECHNOLOGIES
Tour Horizon - 52, quai de Dion-Bouton - 92800 Puteaux
Tél. : 778.14.50

VOICI
LA PREMIÈRE PIERRE
D'UN DOMAINE
ENCORE INEXPLORÉ...

Vous étiez

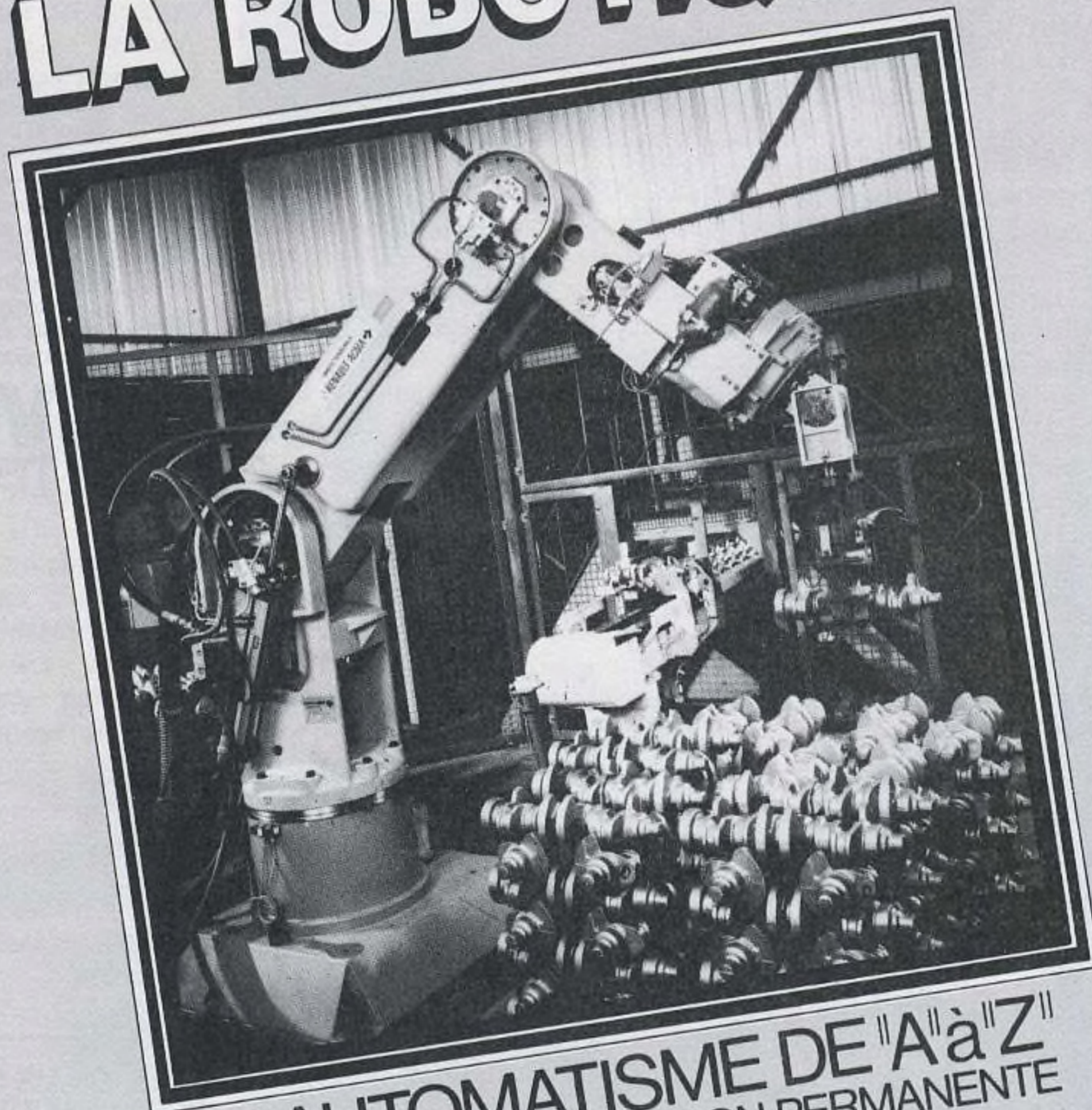
hors série

MOUS D'AUJOURD'HUI

Led

ROBOT INITIATION A LA ROBOTIQUE

**vient de
paraître**



L'AUTOMATISME DE "A" à "Z"
DU LOISIR A LA FORMATION PERMANENTE
DECEMBRE 1984 - 115 F

si nombreux à l'attendre !

L'ouverture au monde passionnant de la robotique, dans un style simple et direct, travail d'un collectif de spécialistes animé par Claude Polgar.

Format 21 x 27, 100 pages, plus de 130 schémas et illustrations.

Le sommaire : une somme !

- **La grande relève des hommes par les robots**
- **L'anatomie de HERO 1** : bras, jambes, ouïe, vue, télémétrie, détection de mouvements.
- **Inventeurs et inventions** : ne confiez pas vos inventions avant de vous être protégé.
- **Cours de conception mécanique** : vocabulaire et notion de base - Ajustement, tolérance, excentricité, etc.
- **Cours de logique générale** : schémas et symboles
- **Electronique industrielle** : du circuit au démultiplexeur.
- **Vie industrielle** : la CAO, assistante de la création.
- **Conception et construction** : de la tortue au robot.
- **Modules fonctionnels** : construction de la carte de départ pour commander les moteurs pas à pas à partir de votre micro.
- **Maquettes et modélisme** : le modélisme ferroviaire se renouvelle grâce à la micro-informatique.
- **Analyses et méthodes** : les rosaces d'évaluation.

BON DE COMMANDE



Je désire recevoir Led-Robot «INITIATION A LA ROBOTIQUE» (attention, cet ouvrage n'est pas vendu en kiosque) au prix de 125 F (port compris).

Nom : Prénom :

Adresse :

ATTENTION - Si je suis abonné soit à LED, soit à LED-MICRO, je bénéficierai d'une réduction de 20 % sur le prix de l'ouvrage, et je ne le paierai que 100 F (port compris).

Je vous note, dans le cadre, mon numéro d'abonné

Ci-joint un chèque bancaire chèque postal mandat .

Adresser votre commande et votre règlement aux
EDITIONS FREQUENCES, 1 boulevard Ney, 75018 Paris.

yashica YC-64

Dans ces conditions, le lecteur désirent acquérir un YC-64 devra s'informer sur la documentation finale telle qu'elle sera réellement lors de l'achat (documentation générale, manuels d'utilisation), ainsi que sur les accessoires accompagnant ce micro-ordinateur.

Le Yashica YC-64 est l'un des premiers ordinateurs ludiques au format MSX à passer aux contre-mesures dans Led-Micro. Avant d'aller plus loin, signalons que le modèle qui nous a été livré est un prototype sans sa documentation. Pourquoi en parler ici : nous avons succombé à la tentation du standard MSX. En effet, le Yashica est le premier modèle à être présenté à la vente (sauf erreur !), et le MSX étant la référence pour les prochaines années, il fallait vite se faire une opinion.

ans ces conditions, le lecteur désirent acquérir un YC-64 devra s'informer sur la documentation finale telle qu'elle sera réellement lors de l'achat (documentation générale, manuels d'utilisation), ainsi que sur les accessoires accompagnant ce micro-ordinateur.

LE MSX

Le MSX est un nouveau standard en informatique. Si les grands classiques en matière de système d'exploitation sont MS-DOS, CP/M, UNIX, etc., il n'existait pas jusqu'à présent une référence en matière de micro-ordinateur à usage familial.

Les possesseurs de petites machines (Oric, ZX, M05, etc.) savent qu'il est impossible d'échanger les programmes. En 1983, le Ministère du Commerce Extérieur japonais (MITI), prenant conscience de l'énorme évolution du marché micro-informatique lança un appel d'offres international afin d'étudier un système d'exploitation qui serait commun aux différents modèles de micros proposés sur le marché. Ce fut la société américaine Microsoft qui proposa la solution la plus intéressante. Il s'agissait d'un système

d'exploitation type associé à une configuration de base pour la machine. L'intérêt devenait évident et permettait de réaliser les logiciels types associés à une portabilité sur tous les modèles MSX. De même, la configuration minimale étant identique sur tous les MSX, les problèmes d'interfaçage et de périphériques se trouvaient dès lors simplifiés.

LE HARD MSX

Le microprocesseur choisi pour le standard MSX est le Z80 A ou un compatible. La fréquence d'horloge est de 4 MHz. La mémoire morte est de 32 Koctets, la mémoire vive devra être d'un minimum de 8 Koctets. Toutefois, 16 Koctets est recommandé. La partie vidéo est composée d'un éditeur de texte de 24 lignes de 40 caractères, la résolution graphique est de 256x192 points. Le standard impose 16 couleurs. L'interface cassette est de type FSK avec une vitesse de 1 200/2 400 bauds. La partie son est composée de huit octaves et trois voies. Il y a, suivant les pays, plusieurs possibilités de clavier. Le format AZERTY est recommandé pour la France. Le MSX autorise deux slots minimum, un pour



les cartouches de jeux, un pour le bus d'extension. Il est possible de connecter un ou deux joysticks.

Il est recommandé pour l'Europe et les Etats-Unis de porter la mémoire vive à un total de 64 Koctets. En option, il est souhaitable d'avoir un éditeur 80 colonnes, une interface RS232, une interface parallèle 8 bits, une autre pour raccorder des disques souples au format MS-DOS, ainsi qu'une batterie pour sauver l'horloge.

Si le processeur est imposé, il en va de même des circuits principaux, ainsi on retrouvera dans chaque MSX, comme contrôleur d'écran un TMS-9918A ou un compatible, comme synthétiseur musical, un AY-3-8910 ou un compatible, comme gestionnaire des périphériques un 8255 Intel ou un compatible. Le clavier doit posséder les touches ESC, CLS, HOME, INS, DEL, et les quatre flèches : haut, bas, droite et gauche. On retrouvera en haut du clavier les touches de fonction F1 à F10 ainsi que Select et Stop.

LE MSX BASIC

Le MSX Basic est une version adaptée du Basic Microsoft 4.5, incluant le graphisme, les fonctions musicales, et certaines fonctions pour périphérique.

Comme la majorité des Basics, le MSX

Basic commence chaque ligne de programmes par une numérotation de ligne et un Carriage Return en fin de chaque ligne composée d'un maximum de 255 caractères.

On retrouvera la majorité des caractères désormais classiques aux micro-ordinateurs.

Les nombres entiers sont compris entre -32768 et 32767. La virgule flottante possède une précision de 10 puissance -64 à 10 puissance 63.

Les variables sont équivalentes au Basic MS/DOS. Les opérateurs sont soit arithmétiques, relationnels, logiques ou fonctionnels.

L'éditeur MSX permet d'imprimer des caractères existants, de supprimer des caractères par la droite ou par la gauche de chaque ligne, ainsi que l'insertion.

Les dix clés de fonction sont pré-programmées sur les ordres suivants :

F1	COLOR [b]
F2	AUTO [b]
F3	GOTO [b]
F4	LIST [b]
F5	RUN [CR]
F6	COLOR 15, 4, 7
F7	CLOAD "
F8	CONT [CR]
F9	LIST [CR] [4] [4]
F10	[CLS] RUN [CR]

Elles peuvent aussi, dans certains cas

commander : ON KEY GOSUB et KEY ON/OFF/STOP.

Ces clés sont, bien sûr, programmables par MSX-Basic afin de les utiliser comme clés de fonctions personnelles. Du côté des ordres, on notera la présence de AUTO, DELETE et RENUM qui sont bien pratiques lors de l'écriture d'un programme.

L'ordre ON ERROR sera pratique si des incertitudes sont possibles dans vos logiciels, il ne manque plus que ON TIME DATE (gestion du temps) et ON TIME OUT (gestion des interfaces).

Les amateurs de graphisme apprécieront des ordres SPRITE, CIRCLE, mais surtout DRAW qui, dans MSX-BASIC est particulièrement élaboré car il regroupe DRAW, MOVE et PLOT.

PLAY qui programme les sons musicaux n'a rien à envier à DRAW car cet ordre est, lui aussi, très performant et permet avec un seul ordre de commander les trois voies.

Les quinze couleurs sont :

0	transparent	8	rouge
1	noir	9	rouge clair
2	vert	10	jaune foncé
3	vert clair	11	jaune clair
4	bleu foncé	12	vert foncé
5	bleu clair	13	magenta
6	rouge foncé	14	gris
7	cyan	15	blanc

CONTRE-MESURES

LE YC-64

D'un poids acceptable pour sa taille, ce micro-ordinateur est vêtu d'un châssis plastique au dessus rouge et dessous noir. Le clavier est bien repéré, touches grises pour les fonctions et les ordres pré-programmés, blanches pour les autres. Le clavier de l'appareil testé était de type QWERTY, mais une lettre de l'importateur nous assure que les modèles livrés seraient de type AZERTY. Cette même lettre nous précise la présence d'une sortie AUDIO. Le Yashica devrait de même être livré avec un manuel en français, la prise Péritel et un cordon pour magnétophone.

Nous n'aborderons pas aujourd'hui les parties concernant le système d'exploitation et la documentation, le nécessaire ayant été fait en première partie des contre-mesures de ce mois.

En ce qui concerne la partie hardware, elle est conforme à la norme MSX. Segimex annonce la sortie prochaine d'un éditeur 80 colonnes sur 24 lignes, la fonction horloge avec batterie de sauvegarde, l'interface RS 232, un lec-

teur de disquettes, enfin un crayon optique.

L'utilisation du YC-64 ne pose aucun problème si ce n'est un fait important à indiquer aux lecteurs de Led-Micro. Lors de l'essai du Yashica, nous n'avions pas à notre disposition de moniteur couleur ni de téléviseur avec prise Péritel. Nous avons donc interfacé le YC-64 avec un téléviseur couleur ancienne norme. Pour ce faire, nous avons utilisé un adaptateur CGV. Ce produit nous a donné une image correcte mais nous fûmes, ô combien, surpris après quelques hésitations à la mise en route, de nous apercevoir que la prise alimentation qui se raccorde au secteur par une petite alimentation standard (6-9-12 V) est connectée en opposition de polarité inverse par rapport à ce que font 99,99 % des constructeurs. Visiblement le service études de CGV a commis là une énorme gaffe. (Avis aux amateurs !)

Le clavier est assez souple à utiliser et les touches de fonctions et préprogrammées facilitent le travail. Les quatre touches de curseur sont pratiques

en édition. La vitesse de la machine est satisfaisante. La trappe du ROM-PACK (pack de jeux) de l'appareil que nous avons en démonstration était hors service. Espérons que la série ne pose pas de problème au niveau du rappel du ressort qui permet à la trappe d'être fermée lorsqu'il n'y a pas de pack enfiché.

Le MSX YC-64 ne propose aucune révolution en matière de micro-informatique. En effet, cette machine ne présente pas d'innovations remarquables. Son grand avantage sera, bien sûr, son appartenance au clan encore bien trop restreint des MSX.

C.-H. Delaleu

LE COUT DU YC-64

Yashica	3 950 F
Buro Joker	1 590 F
Moniteur Philips	1 105 F

FICHE SIGNALÉTIQUE

Processeur	Z 80 A
ROM	: 32 Ko.
RAM	: 64 Ko.
Affichage écran	24 lignes, 40 colonnes en 16 couleurs 24 lignes, 32 colonnes en 16 couleurs
Graphique	16 couleurs
Résolution	256 x 192 points
Langage standard	MSX-BASIC
Système d'exploitation	MSX (Microsoft)
Langages optionnels	Assembleur, Forth, Logo, Cobol, Fortran
Clavier	73 (2 x 5 touches de fonction)
Effets musicaux	8 octaves sur 3 voies + 1 voie pour gérer les effets spéciaux
Interface	Cartouche ROMPACK Cassette Manettes de jeux Entrées-sorties Z 80 A Imprimante

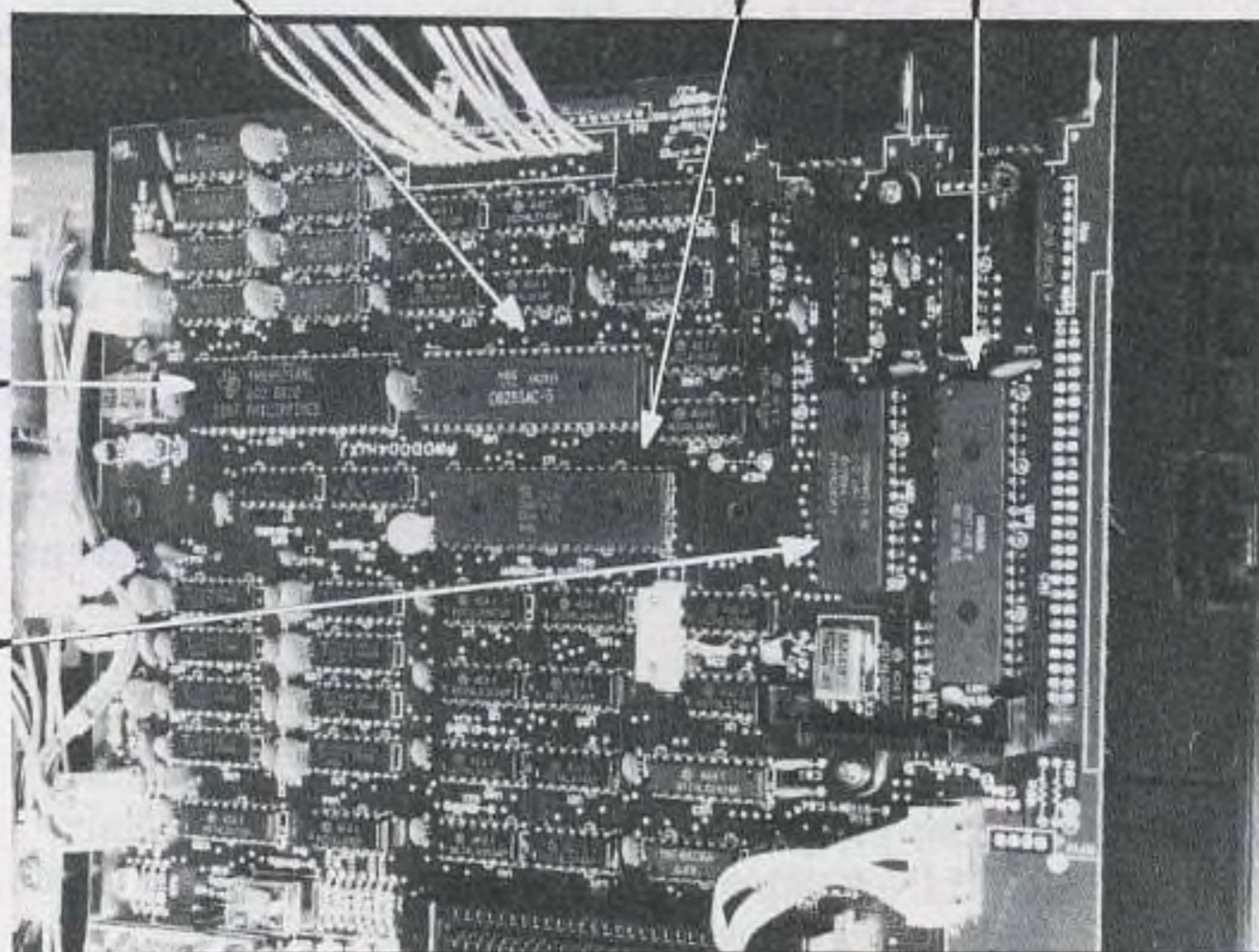
Le 8255 : le gestionnaire de périphériques

Le processeur Z80

Le synthétiseur musical

Le contrôleur d'écran TMS 9918A

Le SED Microsoft





et la pratique ?

Une initiative que devait attendre un grand nombre d'entre vous dans le cadre de budgets très raisonnables !

La formation continue à la micro-informatique

10 stages sur mesure

- | | |
|--|---|
| A. Initiation informatique 2 jours : 14 heures | G. Initiation Pascal 5 jours : 35 heures |
| B. Initiation Basic 5 jours : 35 heures | H. Travaux pratiques club
Led-Micro cours du soir : 30 heures |
| C. Initiation Basic cours du soir : 35 heures | I. Travaux pratiques club
Led-Micro samedi matin : 30 heures |
| D. Fichiers en Basic 2 jours : 14 heures | J. Initiation à l'informatique
industrielle 10 jours : 70 heures |
| E. Graphique I 2 jours : 14 heures | |
| F. Graphique II 2 jours : 14 heures | |

Un troisième cours

A ses cours, Led-Micro a voulu ajouter une formation essentiellement à base de pratique dans des délais de temps courts sur l'initiation à l'informatique. N'hésitez pas à nous consulter en adressant le bon ci-dessous à l'EMIA (Ecole de micro-informatique appliquée), nous vous retournerons une documentation.



ECOLE DE MICRO-INFORMATIQUE APPLIQUEE

DEMANDE DE DOCUMENTATION

Je désire recevoir une documentation sur l'EMIA. Ci-joint 2 timbres à 2,10 F pour frais de poste.

Nom Prénom

Adresse

.

à retourner à : Stages de formation EMIA 1 Bd Ney, 75018 Paris



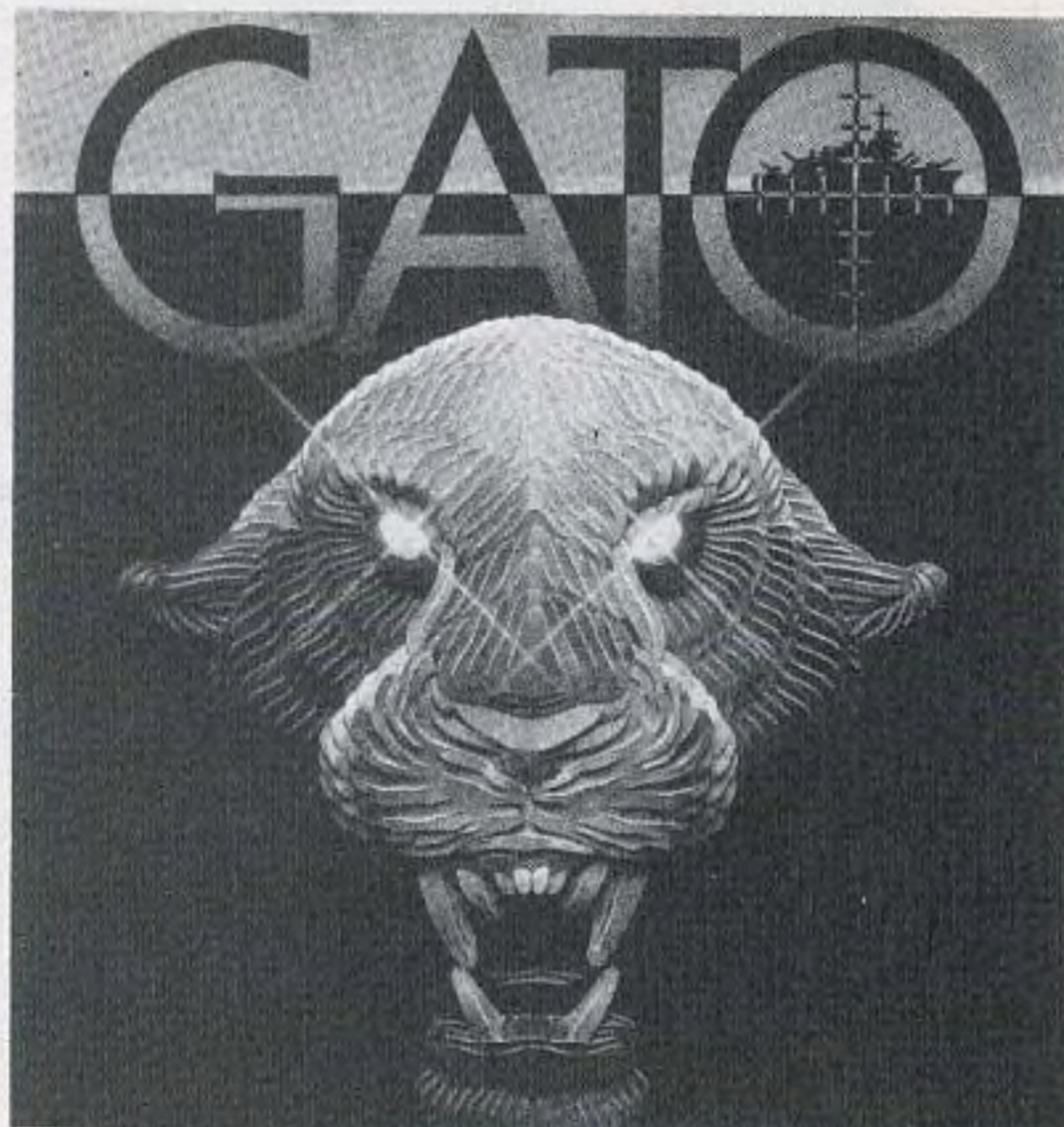
Dans moins de quinze jours, Noël. Autrefois, il n'y a pas bien longtemps, le bonhomme Noël remplissait les chaussures bien rangées devant la cheminée, de livres et de jouets. Aujourd'hui sa hotte déborde de micros et de jeux électroniques. Pour l'aider et vous aider à choisir, voici les toutes dernières nouveautés et celles qui le sont un peu moins. Vous aurez l'embarras du choix à condition d'avoir le portefeuille bien garni. Eh oui, il faut bien compter entre 2 500 et 3 500 F pour un micro en cette fin 84.



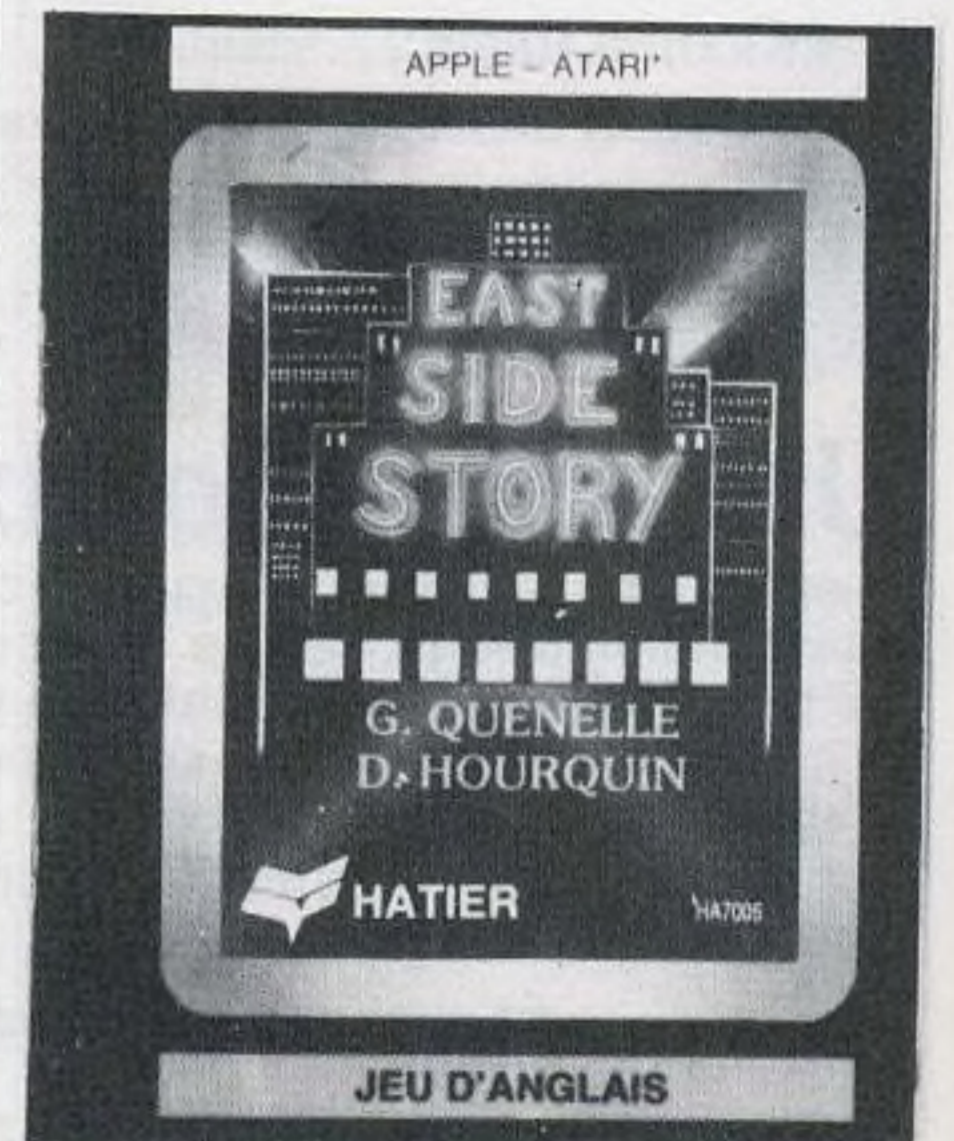
GATO : LE TIGRE DU PACIFIQUE

Ce jeu vous place aux commandes d'un sous-marin de la seconde guerre mondiale. Sonore, en couleur et en trois dimensions, Gato vous projette dans l'univers des combats navals de la seconde guerre mondiale dans le Pacifique. Les itinéraires, la position du

sous-marin et de tous les navires sont calculés en permanence et simulent une situation réelle. La perspective en trois dimensions rend ce jeu spectaculaire. Pour pouvoir l'utiliser, il faut posséder un IBM PC ou XT, ou un micro-ordinateur compatible possédant 128 k Ram, une carte graphique et une unité de disques souples. Gato est importé des Etats-Unis où il a été vendu à plus de 10 000 exemplaires.



HATIER : 12 LOGICIELS EDUCATIFS



Hatier n'est pas seulement les livres scolaires de notre enfance, mais également tout une multitude de «produits» pédagogiques qui font référence. Cette collection que lance Hatier est composée de jeux pédagogiques faisant appel à la mémoire virtuelle ou à la logique pour apprendre en s'amusant. Ils sont

destinés aux enfants à partir de sept ans et valent entre 160 et 220 F. Pour les plus petits : Ludomatic (exercice de mémoire visuelle) sur cassette. A partir de 9 ans : Orthocrack (exercices de grammaire), le Minotaure (entraînement au calcul mental), le Compte est rond (entraînement à l'enchaînement des opérations). East Side Story (sur Apple) est destiné aux adolescents ayant trois années d'anglais. Ce logiciel développe les compétences linguistiques et culturelles.

Ces logiciels sont disponibles sur les machines les plus largement diffusées : T07, M05, Oric, Atari et Apple.

TOUJOURS PLUS

Sinclair propose Spectrum +, une version améliorée de Spectrum qui remplace à terme le Spectrum. Le design s'inscrit dans le style de celui du QL que l'on verra enfin en début de l'année prochaine. L'amélioration la plus marquante se situe au niveau du clavier : le Spectrum + a un vrai clavier de machine à écrire. A noter également deux petites innovations très utiles : une touche de réinitialisation qui évite d'avoir à débrancher puis rebrancher en cas de blocage du programme et des pieds rétractables pour incliner le clavier. Un manuel d'utilisation et une



cassette de démonstration illustrant les instructions du manuel page à page le complète. Et pour les fêtes, il est livré avec un coffret de 8 cassettes de jeux et d'éducation. Il est compatible avec le Spectrum ce qui lui donne accès à une très vaste bibliothèque de programme. Il est vendu 2 590 F avec la prise Péri-tel et 2 230 F en version Pal.

L'ELECTRONIQUE DES MICRO-ORDINATEURS PAR PHILIPPE FAUGERAS

Cet ouvrage est destiné aux micro-informaticiens qui désirent posséder une connaissance des composants de leur micro-ordinateur.

Le sujet est vaste, compte tenu de la diversité et du nombre de ces composants. L'auteur a choisi, avec juste raison, d'articuler son étude autour d'un microprocesseur connu et répandu, puisqu'il s'agit du Z-80, sans pour autant se limiter à la description de la famille de circuits qui lui sont spécifiquement associés (P10 Z-80 etc...).

Cependant, tant pour les informaticiens, gens du soft, que pour les électroniciens, un rappel de l'architecture interne du Z-80 était nécessaire et constitue le deuxième chapitre.

Le degré de connaissances requis pour aborder la suite et en tirer parti n'est pas spécialement élevé : la maîtrise complète du jeu d'instruction du Z-80 n'est pas nécessaire impérieuse, et c'est là le grand mérite de l'ensemble de l'ouvrage. Par contre, quelques connaissances de base en logique séquentielle et combinatoire, voire une première expérience en matière de circuits TTL sera bienvenue. A partir de cela,



l'auteur envisage une mise en œuvre de base du Z-80, par l'adjonction de quelques boîtiers TTL (tampons d'entrée/sortie, élaboration de signaux de contrôles pratiques), afin de réaliser concrètement une unité centrale.

Dès ces premiers chapitres apparaît la structure de chacun des suivants et, en quelque sorte la teneur de l'ouvrage : description fonctionnelle des éléments, centraux (microprocesseur) ou périphériques, choix du composant adéquat parmi les sources disponibles,

exemple(s) de connexion à l'unité centrale et d'application.

La petite pilule à avaler, inévitable, se trouve au chapitre cinq, traitant des interruptions. Ces événements, fondamentaux dans le fonctionnement d'un système, se devaient d'être décrits d'une manière assez précise ; ce qui est le cas. Cependant, il était nécessaire, pour mener à bien cette tâche un peu ingrate, de parler logiciel(s), ce qui effrayera un peu l'électronicien pur et dur.

Retour en force du «hard», au chapitre sept, avec les claviers, puis au chapitre huit, fort intéressant (avec le «quatre» sur les mémoires, c'est le meilleur) traitant d'interfaces vidéo, de génération de signaux de service associés : sujet copieux, à dévorer, mais qui, malgré l'approche des dernières pages du livre, laisse un peu sur sa faim... de réalisation ! Voilà bien l'esprit de ce bouquin : on analyse, on fait la synthèse, puis un projet, qui grandit à chaque chapitre, c'est stimulant !

En bref, l'électronique des micro-ordinateurs a bien atteint son but : présenter l'aspect matériel de la micro-informatique. Mais il réussit également à susciter chez le lecteur électronicien, une envie de création certaine.

125 pages, 240x165. Collection Etudes aux Editions Fréquences.

UN GUEPARD REMOIS

Conçu par la société HBN de Reims, ce micro-ordinateur s'adresse aussi bien aux professionnels qu'au grand public. Il est construit autour d'un Z80 et a 64 Ko de mémoire vive. Il fonctionne sous CP/M 3.0 et New Dos 2.0 et présente une certaine compatibilité avec le Tandy TRS 80 modèle I et III. HBN estime qu'aujourd'hui encore un 8 bits n'est pas dépassé et répond aux besoins des petites entreprises et des particuliers qui peuvent concevoir leurs programmes ou utiliser des logiciels standard comme Multiplan ou Wordstar en investissant au minimum : 20 000 F TTC imprimante comprise. La version de base du guépard vaut 13 900 F HT.



MADE IN BRITAIN



Electron d'Acorn, un micro qui se veut être un pro. 64 Ko de mémoire, graphique 640 x 256, couleurs, clavier mécanique mais Qwerty. Ce micro offre de nombreuses possibilités : en outre la diversité des langages, Basic d'Electron, Assembleur, Pascal, Forth, Lisp et la possibilité de lui adjoindre un boîtier d'extension Plus 1 qui apporte les entrées analogiques pour joystick, la sortie imprimante et les emplacements pour deux cartouches de programmes en Rom. Avec la prise Péritel, il est commercialisé un peu moins de 3 000 F.

LE MUST DE SEIKO

La montre-ordinateur est déjà vendue par Seiko dans le circuit des horlogers-bijoutiers. Cette montre est connectable à un micro-ordinateur de style Commodore 64, Apple II ou même IBM PC. Elle possède 2 Ko de mémoire vive et 7,5 Ko de mémoire morte et s'adapte à un petit clavier permettant des notes (1 800 F l'ensemble), ou à un ordinateur comportant Basic, imprimante et cartouches de jeux en mémoire morte (3 200 F au total).



LE SPECTRAVIDEO NOUVEAU EST ARRIVE



Le Spectravideo 728 est un MSX disposant de 64 Ko de mémoire. Il est vendu 3 500 F environ. Sinon, que dire, qu'il a toutes les caractéristiques d'une machine au standard MSX dont nous avons parlées dans un précédent numéro. La différence de prix s'explique par le fait qu'il a 64 Ko de mémoire.

Il est distribué en France par un nouvel importateur : Serepe (103 rue Charles-Michel, 93200 ZAC de Saint-Denis. Tél : 243.36.22).

A LIRE

dBase II sans embûches.

G. Grigorieff - Eyrolles

Une des principales applications professionnelles de la micro-informatique concerne les bases de données (par définition une base de données est une collection d'informations ayant un ou plusieurs sujets). Que ce soit en comptabilité, gestion de stocks ou de fichiers un utilisateur a besoin de créer, manipuler ou imprimer des bases de données.

Des logiciels spécialisés (en informatique on parle alors de «progiciels») ont donc été développés afin d'aider ces nouveaux utilisateurs. Parmi les plus connus, citons Wordstar ou dBase II.

Ce nouveau livre de G. Grigorieff est une présentation de dBase II, progiciel qui se présente à la fois comme un programme «clé en main» interactif mais aussi comme un langage de programmation très puissant. Ce livre de 170 pages est divisé en trois parties.

Tout d'abord, l'auteur analyse les principales commandes de dBase II qui permettent de créer, gérer, modifier, imprimer des bases de données. Chaque commande est illustrée à l'aide d'un petit listing qui décrit les différentes étapes du dialogue utilisateur-micro-ordinateur.

Dans une seconde partie, l'aspect langage de programmation de dBase II est abordé. Outre les commandes clavier (thème de la première partie) dBase II permet d'accéder à des instructions d'entrées-sorties (INPUT, READ) de branchements conditionnels et inconditionnels (DO/IF) ainsi qu'à des boucles. Des programmes d'applications sont décrits ainsi que la façon de créer et d'utiliser des menus.

La troisième partie de cet ouvrage est consacrée à une comparaison entre le Basic et dBase II.

Pratique des MS-DOS et PC-DOS

H. Lilen - Editions Radio

Grâce à l'IBM PC, MS-DOS (créé par Microsoft) est devenu peu à peu un standard comme système d'exploitation. On le trouve sur l'IBM PC mais aussi sur tous les compatibles qui utilisent le même microprocesseur que le PC, le 8088 d'Intel.

Un système d'exploitation (ou DOS, Disk Operating System en anglais) est un programme qu'on vient charger (à partir d'une disquette) à chaque initialisation ou mise en route du micro-ordinateur. La fonction de ce programme est multiple : en premier lieu, il gère tous les périphériques reliés à l'unité centrale (clavier, écran, mémoire de masse...) mais aussi il synchronise les différents échanges entre l'utilisateur et le micro-ordinateur.

Dans ce livre, divisé en trois parties, H. Lilen nous propose de découvrir tous les secrets de ce système d'exploitation et sa mise en œuvre. Dans une première partie, l'auteur effectue une revue de l'IBM PC et de ses principaux composants et périphériques... Une bonne révision pour tous ceux qui ne sont pas familiarisés avec le «hardware» et le vocabulaire micro-

PRATIQUE DES MS-DOS et PC-DOS

H. LILEN



dBASE II

sans
embûches

G. GRIGORIEFF



informatique.

L'étude du MS-DOS proprement dite commence à partir de la seconde partie avec une description de la mise en œuvre de MS-DOS. Les premières opérations à effectuer lors de l'acquisition d'un IBM PC sont la création de la disquette DOS national (adaptation du PC aux normes françaises) et la réalisation des sauvegardes de toutes ses disquettes (une mauvaise manipulation est si vite arrivée !). Toutes ces opérations sont décrites à l'aide de nombreux exemples où tous les cas de figure sont envisagés.

La troisième partie est consacrée aux différents commandes et utilitaires disponibles sous MS-DOS. La syntaxe et la fonction de chaque commande sont décrites à l'aide de petits programmes. Le dernier chapitre de ce livre traite d'un utilitaire bien particulier qui est l'éditeur de lignes EDLIN.

Dico-Flash

DICTIONNAIRE DE LA MICRO- INFORMATIQUE

l'explication de tous les mots de base

Un livre sans surprise où on retrouve tous les ingrédients : clarté, simplicité qui font le succès d'H. Lilen.

A noter un livre traitant du même sujet chez Eyrolles «Le système d'exploitation MS-DOS versions 1 et 2» - R. Politis, B. Vanryb.

Communiquez avec votre ZX81 ou avec votre Timex 1000.

Denis Bonomo, Eddy Dutertre - Editions Soracom

L'éditeur Soracom, plus connu pour ses revues consacrées aux ondes courtes (Mégahertz), se lance à son tour dans la micro-informatique. Parmi les nombreux livres reçus, notre attention a été attirée par un ouvrage sur le ZX81. Première surprise, un schéma complet du ZX81 est donné en première partie suivi de nombreuses informations (extension, périphériques, microprocesseur Z80). Vérification faite, tous les livres de cet éditeur traitant d'un micro-ordinateur particulier sont accompagnés des différents schémas électroniques. Il faut saluer cette initiative qui intéressera beaucoup d'électroniciens.

La seconde partie de ce livre traite de l'utilisation du ZX81 dans différentes applications. De nombreux programmes sont commentés comme la réalisation d'un desassembleur. Une attention toute particulière est donnée à l'emploi d'un ZX81 dans un système radio (MIRE, MORSE, RTTY...). La dernière partie de cet ouvrage est consacrée à la réalisation d'un coupleur d'entrées-sorties (circuit 6820 Motorola) et à son utilisation (programmeur EPROM...). A noter aussi chez

INFORMATIQUE
le **ZX 81**
à l'**ÉCOLE**



Soracom et suivant un principe similaire :
Communiquez avec votre Oric 1 et votre ATMOS Denis Bonomo, Eddy Dutertre
Les Mystères du Laser Denis Bourquin.

Le ZX81 à l'école

Coopératives scolaires de l'école du quai Farconnet, Tournon-sur-Rhône.

Led-Micro, revue pédagogique est à l'affût de toutes les initiatives des enseignants et des écoles. Aussi, c'est avec plaisir que nous parlerons aujourd'hui de ce fichier pédagogique réalisé par un groupe de professeurs de l'école de Tournon-sur-Rhône. Ce cours se présente sous la forme de 50 fiches individuelles traitant des différents aspects de la micro-informatique. Dans une première partie (19 fiches) réservée aux enseignants, les auteurs développent des thèmes généraux, sur la façon d'aborder (et donc d'enseigner) la micro-informatique. La seconde partie, destinée aux élèves, est un cours de Basic. Chaque fiche analyse les différentes instructions Basic (syntaxe, fonction) et leur utilisation dans un programme. De nombreux exemples sont développés



permettant d'illustrer les notions abordées. Le matériel servant de support à ces fiches est le ZX81 (choisi pour son faible coût), mais il est très facile de transposer ce cours sur un autre matériel utilisant un basic standard.

Ce livre (les 50 fiches sont réunies dans un classeur au format d'un livre de poche) est à conseiller à tous les enseignants ou animateurs de club qui désirent développer une expérience micro-informatique. Ce fichier est disponible par correspondance à l'Ecole Publique Mixte, quai Farconnet, 07300 Tournon-sur-Rhône. (Prix : 60 francs + 12 francs pour le port).

ATMOS à la conquête des jeux
Jean-Yves Astier - Ed. Eyrolles

Les jeux et les micro-ordinateurs : un sujet inépuisable qui fait couler beaucoup d'encre. Aujourd'hui, c'est l'ATMOS qui est pris comme base de cet ouvrage proposé par Eyrolles et J.-Y. Astier. Ce livre est composé de deux parties indépendantes. Dans une première partie, l'auteur analyse quinze jeux différents parmi lesquels on retrouve la Tour de Hanoï, Master Mind et les terribles... Gloutons. Chaque jeu est accompagné par de nombreuses explications ainsi que d'un mode d'emploi complet. Il est donc facile à tous les programmeurs introduits dans le «milieu Basic» de décomposer le listing et pourquoi pas de faire des modifications.

La seconde partie de ce livre traite de la façon d'écrire ses propres jeux. Pour ce faire, il faut tout d'abord bien connaître l'organisation (mémoire, entrées-sorties) et le fonctionnement de son micro-ordinateur. De plus, l'optimisation d'un programme passe souvent par l'utilisation du langage machine associé au microprocesseur (6502). Toutes ces notions sont développées à l'aide de nombreux exemples.

Personnellement, je pense que les jeux peuvent être un excellent thème pour s'initier à la programmation. Ce livre en est la preuve flagrante.

Dans le cadre ATMOS/ORIC en tout genre, une sélection :
Pratique de l'Oric ATMOS et 36 programmes H. Lilen, J. Benard - Ed. Radio
Faites vos jeux avec ATMOS C. Delannoy - Ed. Eyrolles
Oric ATMOS votre micro-ordinateur M. Bussac - Ed. Cedic Nathan.

Editions Cedic/Nathan

Thomson Nathan, une association qui se porte bien ? On peut le croire avec la sortie de ces trois nouveaux livres ayant pour support le M05 et le T07.
Logo Manuel de référence D. Avram.

Initiation à Logo D. Avram, M. Weidenfeld.

Initiation au Forth SEFI.

Rappelons que LOGO est un langage de la nouvelle génération, à vocation pédagogique, faisant appel essentiellement aux graphiques.

Quant à FORTH «langage plus proche de la machine» (exécution plus rapide et encombrement mémoire plus faible), il permet ce qu'on appelle la programmation structurée. Chaque utilisateur crée lui-même ses fonctions de base (primitives), ce qui rend ce langage très extensible.

Dictionnaire de la micro-informatique

Editions Marabout

L'informatique a inclus dans le langage courant de nombreux mots qui restent incompréhensibles pour les non-initiés.

Le but de ce petit livre (format Marabout) est de «démystifier ce nouveau langage» et de donner la définition des principaux mots rencontrés en informatique. Chaque terme analysé est accompagné de sa traduction en anglais et d'une explication qui le restitue dans son contexte informatique.

P.F.

P.A. GRATUITES

Vends Sharp PC 1500 + impri. + interface 2 K7 + nbr. prog. et manuels + ext. 8 Ko, prix 1 850 F. Tél. : 071.66.56.

Vends collection complète des revues ABC Informatique, list, SVM, VO (pas de détail) à moitié prix. Tél. : 79/75.04.43.

AV interface ZP82 pour ZX81, état neuf, 600 F + extension mémoire 16 K, état neuf : 260 F. Guy Brothier, 1 rue St Michel, 17100 Saintes - (46) 74.28.22.

ECHANGE : traducteur de langue CRAIG-3000 mots, possibilité 10 langues dont 3 simultanément, branchement micro possible. PLUS Slenderton (8 plaques), CONTRE micro, branchement télé direct ou moniteur vidéo plus lecteur disquette, 2 drives. Etudie toute autre proposition. Tél. : 203.59.87 Yannick.

Vds compatible Apple carte mère avec Z80, coffret et alim., 4 150,00. Imprimante Line Printer V 6 000,00. Ti58 + 1 module 450,00. Appareil photo OM1, 1 000,00. Grobos L., 14 rue des Prés, Fay-les-Nemours, 77167 Bagneaux/Loing.

Professeur de mathématiques souhaite échanger expériences du Sanyo PHC 25 et Assembleur Z80. Guy Vantorre, 35 rue de la Plaine à Joncs, 59610 Fourmies.

Vends Spectrum 48 K garanti (4-84) + Péritel + 1 livre + 3 cassettes de jeux. Le tout 1 500 F. Tél. : 657.23.14 après 19 h.

Possesseur de ZX81, Spectrum, je peux vous faire gagner de l'argent. Pour en savoir plus, envoyez-moi vos coordonnées ainsi qu'un chèque ou mandat-lettre de 20 F pour frais d'envoi et gestion. A S.A.N.S., BP 16, 50320 La Haye-Pesnel.

Etudiant en première année de mécanique automatisme, recherche pour débiter en informatique un aimable donateur de Commodore 64 ou d'équivalent. Merci - Roussel Pascal, 116 rue de la République, 62700 Bruay-en-Artois.

Vends ordinateur TRS80 modèle 3, état neuf. Prix 4 000 F. Tél. : (81) 39.22.78, après 20 heures. Forner J.-F. 17 bis, rue de Mor-teau, Pontarlier.

Vds Multitech MPF-II + 64 K RAM + Basic Applesoft + logi-ciels + manuel + écran : 3 000 F (10/83). Tél. : (3) 991.20.33 (avant 21 h) Stéphane Henry.

Echanges nombreux programmes (utilitaires et jeux) pour Oric 1 et Atmos. Mr Beugnies Jacky, 15 rue de la 32^e D.I. 59229 Teteg-hem - Tél. : (28) 61.84.48.

Vends machine à écrire à mémoire Olivetti TES 401, 1979, mar-guerite avec 400 disquettes de 7,5 K, 5 000 F. P. Lamy 12 r. Car-peaux 92400 Courbevoie - Tél. : 333.38.83.

Pour ZX81 vends extension 16 K + K7 Pendu, 250 F + 7 livres sur ZX81, 300 F. L'ensemble 500 F. Tél. au 985.22.39 après 18 h.

Vends lecteur de disquettes 5" double face, très bon état. Prix 1 300 F. Clavier ASCII professionnel, 92 touches, neuf. Prix 800 F. pièces : ventilateurs, transfos, moteurs, connecteurs. Téléphoner le soir à Toulouse : (61) 70.61.40.

Affaire exceptionnelle : TRS80 M1 LII + 2 drives + doubleur denstié + orchestra 80 New-Dos/L-Dos/Visicalc/Scripts/Super Utility sur disquettes d'origine + 200 programmes cassettes et disquettes + importante bibliothèque. Pierre Campmai : (35) 91.11.47.

Index des annonceurs

Editions Fréquences, p. 5-46-47-54-55 - Electropuce, p. 63 - EMIA, p. 59 - Texas Instruments, p. 4 - Victor, p. 53-68 - Vidéo Technologie, p. 67 - Yashica, p. 1 - ZMC, p. 2

**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

LE LASER 200

UN MICRO ORDINATEUR COULEUR SECAM

VRAIMENT TRÈS ÉTONNANT.



1290 F TTC

Microprocesseur Z 80 A • Langage Microsoft Basic • Affichage direct
 antenne télé SECAM • Clavier 45 touches pleine écriture, + clef d'entrée,
 + graphismes, + bip sonore anti-erreurs... • Texte + graphismes mixables
 9 couleurs • Edition et correction plein écran • Son incorporé

• Toutes options : extension + 16 K + 64 K,
 interface imprimante, imprimante,
 stylo optique, manettes,
 jeux, modem,
 disquettes...



**VIDEO TECHNOLOGIE
FRANCE**

19, rue Luisant - 91310 Montlhéry
 Tél. (6)901.93.40
 Télex SIGMA 180114

BON DE COMMANDE
 A retourner à : VIDEO TECHNOLOGIE - 19, rue Luisant - 91310 Montlhéry
 Tél. (6)901.93.40 - Télex SIGMA 180114

Je désire recevoir :
LASER 200 SECAM comprenant :
 Le LASER 200 avec son modulateur SECAM
 incorporé se branchant directement sur l'antenne
 du téléviseur.
 + Câble de liaison fiches jack pour lecteur de K7
 + Câble de liaison micro/télé ou moniteur
 + Livre technique (150 pages) de BASIC
 + Livret d'exercices
 + Manuel de mise en route
 + Casette de démonstration en français
 + Garantie **1290 F TTC**

EXTENSION-PERIPHERIQUES-
INTERFACES LASER 200
 Extension mémoire 16K 590 F TTC
 Extension mémoire 64K 1.190 F TTC
 Lecteur pré-réglé de cassettes
 type DR 10 570 F TTC
 Paire de manettes de jeux
 avec son interface 320 F TTC
 Interface d'imprimante "Centronic
 parallèle" 320 F TTC
 Imprimante 4 couleurs
 papier standard 2.190 F TTC
 Interface disquette (en préparation) ... N.C.
 Stylo optique (en préparation) N.C.

LOGICIELS LASER 200
 Cassettes avec programmes 4K ou 16K...79 F TTC
 (Voir liste détaillée constamment augmentée)

TOTAL DE MA COMMANDE :

Je choisis de payer le total de ma commande :
 Au comptant, par CCP, chèque bancaire, ou mandat,
 à l'ordre de VIDEO TECHNOLOGIE FRANCE
 Contre-remboursement au transporteur,
 moyennant une taxe de 60 F.

Nom _____
 Prénom _____
 N° _____
 Rue _____
 Ville _____
 Code Postal _____

Signature _____

LM

Liste de plus de 100 revendeurs, sur simple demande

chaque mois dans...

MICRO MAGAZINE

La revue de la micro-informatique
professionnelle et de l'utilisateur Victor



le dossier

l'actualité

les applications

la technique

les fiches cuisine

le catalogue
des progiciels

les logiciels

EDITE PAR SITTELLE CONSEIL, 32, RUE WASHINGTON 75008 PARIS - (1) 359.68.34

CHEZ VOTRE MARCHAND DE JOURNAUX