

LOISIRS
hors série

LES TECHNIQUES D'AUJOURD'HUI

Leed MICRO

APPRENDRE • La programmation, cours de Claude Polgar • Le Microprofessor MPF 1B, cours pratique de microprocesseur par Philippe Duquesne. • Les fortiches : éditeur de tableau. MAGAZINE • Le traitement de texte • La contre-mesure du Victor S-1 • Shopping.

DE L'INITIATION A LA PRATIQUE DE L'INFORMATIQUE

COURS
N°16



LE GUÉPARD

ISSN 0757-6889

M 1988 - N° 16 - 16 F

MENSUEL JANVIER 1985 BELGIQUE 111,15 FB/CANADA 3,75 \$/SUISSE 6,75 FS.

UN PREMIER LEXIQUE ANGLAIS-FRANÇAIS VRAIMENT PRATIQUE ET TRÈS COMPLET + de **1500** termes!

- Index français-anglais
- Lexique des termes anglais et américains avec explication en français
- Tables de conversion

JEAN HIRAGA

lexique de l'électronique anglais-français



Pour la première fois en électronique, un lexique anglais-français présenté sous forme pratique avec en plus des explications techniques succinctes mais précises.

En vente
chez votre
libraire
et aux
Editions
Fréquences

Pour recevoir
directement
ce livre chez vous,
remplissez le bon
de commande de
la page 62



éditions fréquences
COLLECTION **Led** LOISIRS

112 pages
PRIX: 65 F

LO...QUES D'AUJOURD'HUI

hors série

Led

MICRO

JANVIER 85

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

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

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

**Services Rédaction-Publicité-
Abonnements :**

1, bd Ney, 75018 Paris
Tél. : (1) 607.01.97
Lignes groupées

Rédaction

Rubriques pédagogiques

Chef de rubriques :

Claude Polgar,
Ont collaboré à ce numéro :
Philippe Duquesne, Bruno
Lilamand

Rubriques Magazine

Chef de rubrique :

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

41

LE COIN DES FORTICHES

Editeur de tableau

par **Bruno Lilamand**

48

LE COURRIER DES LECTEURS

Claude Polgar répond

50

LIBRES PROPOS

Réflexions sur la micro-informatique

RUBRIQUES MAGAZINE

52

LE TRAITEMENT DE TEXTE

56

LES CONTRE-MESURES

Le Victor S-1



62

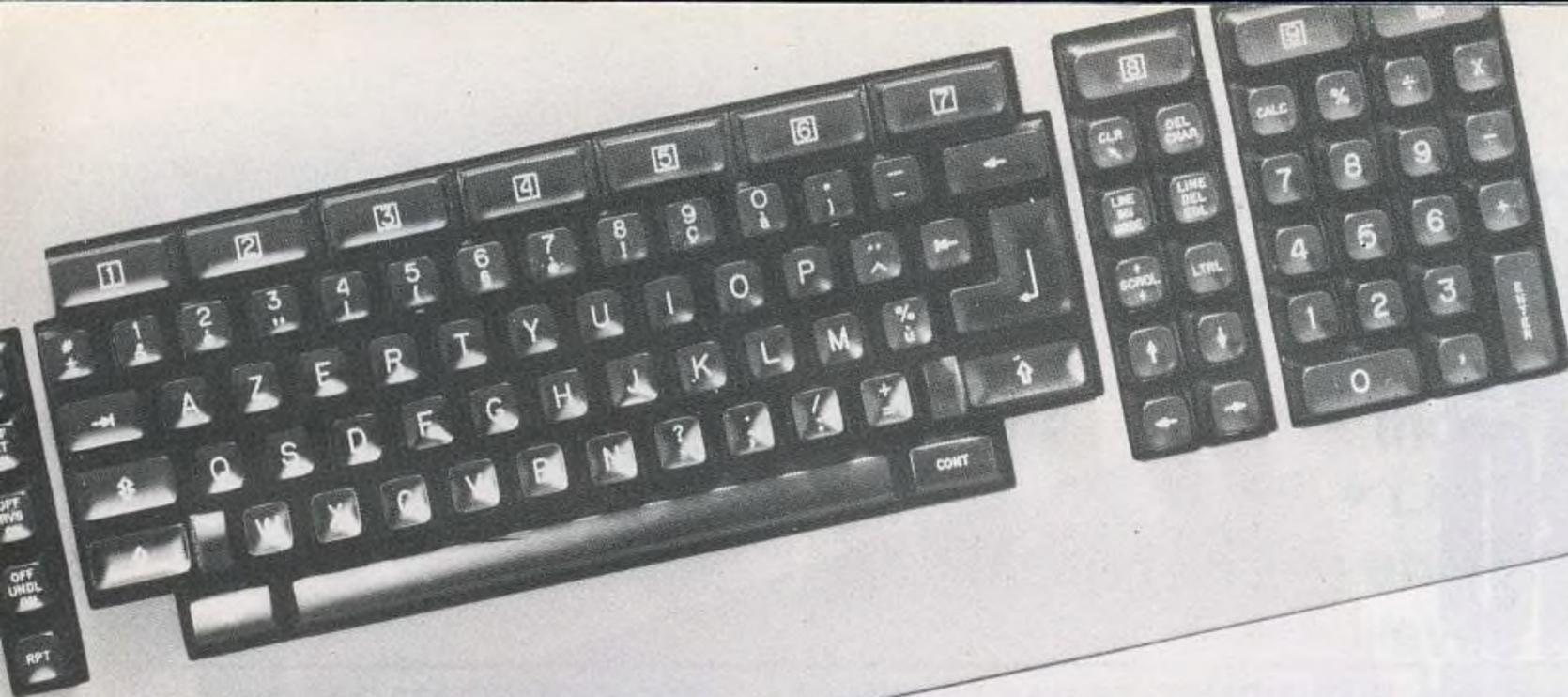
LES INFOS ET LES PRODUITS

64

BIBLIOGRAPHIE

A lire

NOTRE COUVERTURE : Le Guépard de chez HBN Computer. Cet appareil fera l'objet d'une contre-mesures dans notre numéro de février 85.



et la pratique ?

Une initiative que devait attendre un grand nombre d'entre vous

La formation continue à la micro-informatique

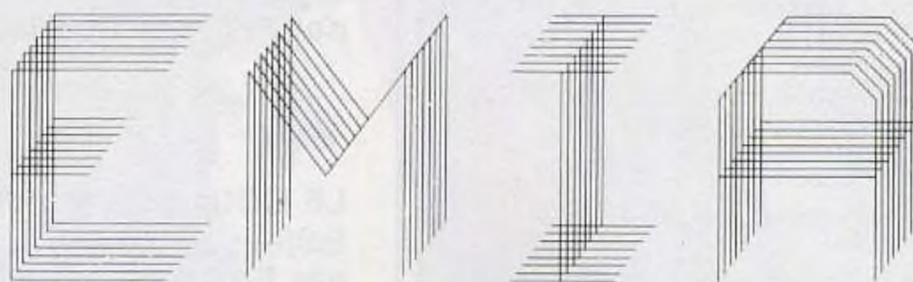
10 stages sur mesure

- | | |
|--|--|
| 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 EMIA cours du soir : 30 heures |
| C. Initiation Basic cours du soir : 35 heures | I. Travaux pratiques club EMIA 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

En complément au cours théorique que vous suivez assidûment dans LED MICRO, EMIA 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



COURS DE PROGRAMMATION(16)

NOTRE STOCK D'INSTRUCTIONS AU 15 JANVIER 1985

Voici le «stock d'instructions» que les lecteurs du «Cours de Programmation en BASIC» ont étudié. Dans vos réponses à nos exercices de récapitulation, essayez de n'employer que ces instructions.

BASIC «de Référence» : le BASIC Microsoft du PC d'IBM.

Opérateurs

=	+	-	/
*	↑	()
>	<	<>	<=
=>	AND	OR	NOT

Instructions - Commandes - Fonctions

ABS(X)	LOG(X)
ASC(X\$)	LPRINT
ATN(X)	NEW
CHR\$(X)	ON...GOTO
CLS	ON ERROR...GOTO
END	PRINT
EXP(X)	REM
GOTO	RIGHT\$(X\$,n)
IF...THEN	RUN
IF...THEN...ELSE	SIN(X)
INPUT	SGN(X)
INT(X)	STR\$(X)
INPUT	TAN(X)
INT(X)	TIMES
LEFT\$(X\$,n)	VAL(X\$)

APPEL AUX SPECIALISTES

Votre serviteur (Claude Polgar) vous fournira dans son cours des exemples d'exercices programmés sur IBM-PC ou sur un compatible (en principe : le PAP de TOSHIBA quand on abordera le graphisme... à moins que d'ici là on trouve sur le marché une merveille qui ridiculise le PAP !)

Nous sommes en train de constituer une équipe de «spécialistes» tels que :

— Bruno Lilamand est en train de le devenir pour le Commodore 64.

Nous espérons avoir également des spécialistes ZX Spectrum, Oric, Apple II, Dragon, Laser, Goupil, TRS80, Sanco...

Peut-être pourriez-vous devenir un «spécialiste» de l'un de ces appareils. Pour cela, il vous faut :

— disposer de l'un de ces appareils (avec imprimante)

— habiter (si possible) en Région Parisienne

— ne pas être un débutant (une connaissance minimum de l'assembleur est nécessaire)

— vous tenir au courant de tout ce qui concerne «votre» produit.

Nous vous demanderions :

— de traduire certains exercices sur «votre» système, et (lorsqu'il est nécessaire d'effectuer des modifications par rapport à la version «de référence») de nous en fournir un listing accompagné (éventuellement) de commentaires.

— d'assister à une réunion technique, une fois tous les deux mois (un jeudi soir à 18 heures au 1 boulevard Ney)

— de répondre aux lecteurs qui me posent des «colles» sur leur appareil.

Nous vous offririons :

— une «pige» (= rémunération à la page)... mais au tarif habituel. Ne comptez donc pas sur ces revenus pour changer de voiture.

— (peut-être ?) un contact avec les techniciens du constructeur (contact enrichissant).

Cette procédure va se mettre en place petit à petit. En cas de succès, nous espérons que bientôt nos lecteurs pourront travailler leurs cours de LED-MICRO sur leur ordinateur, quel qu'il soit. Quand exactement, me demanderez-vous ? Je n'ose plus faire de prévision depuis mes mésaventures avec LED-ROBOT (délai annoncé 3 mois. Délai respecté 10 mois). Je vise le mois de mai 1985.

LE CORRIGE DES EXERCICES DE RECAPITULATION

Au moment de boucler le numéro 15 de LED-MICRO, je n'avais reçu qu'une seule réponse à mes exercices de récapitulation : celle de M. Sipra qui était excellente et que j'ai publiée «in extenso». Mais quelques jours après, j'ai reçu d'autres réponses de très bonne qualité (mais moins explicites). En particulier, de MM. Michel P. (47300 Villeneuve-sur-Lot), Michel R. (Bourg-St-Andéol). La nécessité de boucler une revue en temps voulu et celle de tenir compte de toutes les réponses sont contradictoires !

Même chose ce mois-ci avec l'exercice sur le Cheval Errant. Je viens de recevoir des solutions intéressantes et différentes de la mienne.

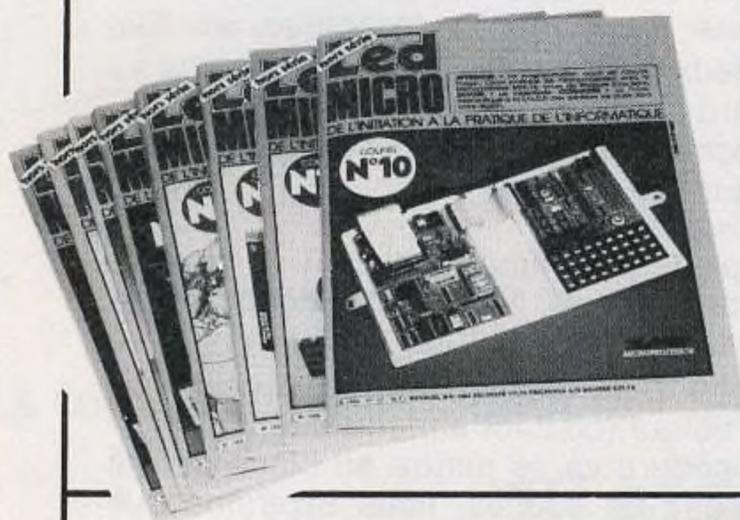
Voici donc comment j'envisage de procéder

— Le mois N, je propose aux lecteurs 2 ou 3 exercices de récapitulation

— Le mois N + 1, je publie «ma» solution

— Le mois N + 2, je publie (s'il y a lieu) les solutions différentes proposées par les lecteurs (avec des commentaires).

Vous découvrez Led-Micro avec ce n°16
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, numériser, photostyle, souris • Opérateurs de base
- N°4** Langages compilés et interprétés - Les systèmes d'exploitation - Les logiciels - 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 • Arithmétique binaire
- N°7** Déroulement d'un programme - Représentation des nombres • Les bascules
- N°8** Calculs en BASIC - Les registres - Les compteurs
- N°9** Notion de format - Le NEWDOS • Architecture d'un système à microprocesseur
- N°10** Le NEWDOS (fin) - Le CP/M80 - Les registres du Z80 - Déroulement d'un programme - L'U.A.L.
- N°11** Utilisation d'un fichier enregistré en MBASIC (sous CP/M) - Le formatage • Le hardware du MPF-1B
- N°12** L'affectation - Variables chaînes booléennes - Le langage du Z80^R
- N°13** L'affectation (fin) : INPUT - La sélection (1^{re} partie : Sélection simple).
- N°14** La sélection - Le langage Z80^R (suite).
- N°15** Complément sur les chaînes de caractères - Le langage du Z80^R (suite).

Vous désirez un ou plusieurs numéros qui vous manquent (de 1 à 15) : 18 F par numéro PORT COMPRIS

BON DE COMMANDE

à retourner aux EDITIONS FRÉQUENCES 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

Nom : prénom :

Adresse :

Ville Code postal

COURS DE PROGRAMMATION(16)

TROISIEME PARTIE (SUITE)

<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. Modifions 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
<p>3.16. La sélection (fin) La sélection multiple</p>	LED-MICRO n° 16

Corrigé de l'exercice de récapitulation R12 : le cheval errant

Une erreur dans l'organigramme proposé

L'organigramme que nous vous avons proposé (page 28 de LED-MICRO n° 15) n'affichait que la position initiale du cheval ! Il fallait faire remonter les flèches de boucle **avant** l'affichage. Le croquis n° 1 (page ci-contre) vous fournit le «vrai» organigramme. Vous y remarquerez aussi que les cases initiales et finales s'appellent maintenant 26 et 117 (au lieu de 27 et 118) : encore une étourderie de ma part, que je vous expliquerai un peu plus loin.

L'important et l'accessoire

Un problème de «méthodologie» : faut-il commencer par chercher à traduire le numéro de la case en langage échiquéen ? Notre réponse est : «Non. Ce n'est pas par là qu'il faut commencer. C'est tout à fait accessoire. Cherchons d'abord à réaliser un programme «qui tourne», ensuite nous améliorerons sa présentation». Ce «principe» est tout à fait général.

Donc commençons par faire tourner notre programme en lui laissant afficher des numéros de cases en chiffres.

Petit rappel de mathématiques très élémentaires

Je considère deux nombres $A = 89$ et $B = 12$ que je vais diviser de façon «classique» :

$$\begin{array}{r|l} 89 & \\ \hline 12 & 7 \\ \hline & 5 \end{array}$$

Ce qui s'écrit :

$$89 = 8.25 \star 12$$

c'est-à-dire

$$A = C \star B$$

ou

$$C = A / B$$

Si je décide de ne pas continuer la division après la virgule, j'obtiendrai

$$\begin{array}{r|l} 89 & \\ \hline 12 & 7 \\ \hline & 5 \end{array}$$

On dira :

la division entière de 89 par 12 est $Q = 7$

le reste de cette division entière est $R = 5$

Ce qui s'écrira

$$A = B \star Q + R$$

On obtiendra alors :

$$Q = \text{INT}(A/B)$$

puis

$$R = A - B \star Q$$

Des lignes et des colonnes

Nous vous avons demandé (troisième problème - paragraphe G - page 29) de vérifier qu'on ne tombe pas dans les «cases interdites» en éliminant les bandes horizontales et verticales indésirables grâce à une «division entière».

L'élimination des «cases interdites» est un excellent exemple de l'emploi de la division entière.

Supposons que notre cheval se trouve sur la position A du damier (voir figure 2 page ci-contre). On pourra définir la ligne Q et la colonne R dans laquelle il se trouve en effectuant la division entière de A par 12

$$A = Q \star 12 + R$$

Mais c'est faux !

J'avoue que j'ai été tenté d'écrire :

«Voici une petite complication qui va nous fournir l'occasion de réfléchir à...»

Ne soyons pas hypocrites ! Je n'ai pas réfléchi suffisamment en rédigeant le texte de mon exercice. Tirons-en une leçon. Rockefeller a dit : « Chaque fois qu'il m'est arrivé un ennui, je me suis efforcé de transformer cette difficulté en une opportunité. C'est mon seul secret ». Utilisons ce secret de Rockefeller : quand nous aurons à numéroter des cases en quadrillage, pensons à commencer la numérotation à 0 plutôt qu'à 1, cela facilitera souvent les calculs. Alors numérotions les cases de notre échiquier comme l'indique la figure 3.

L'organigramme modifié

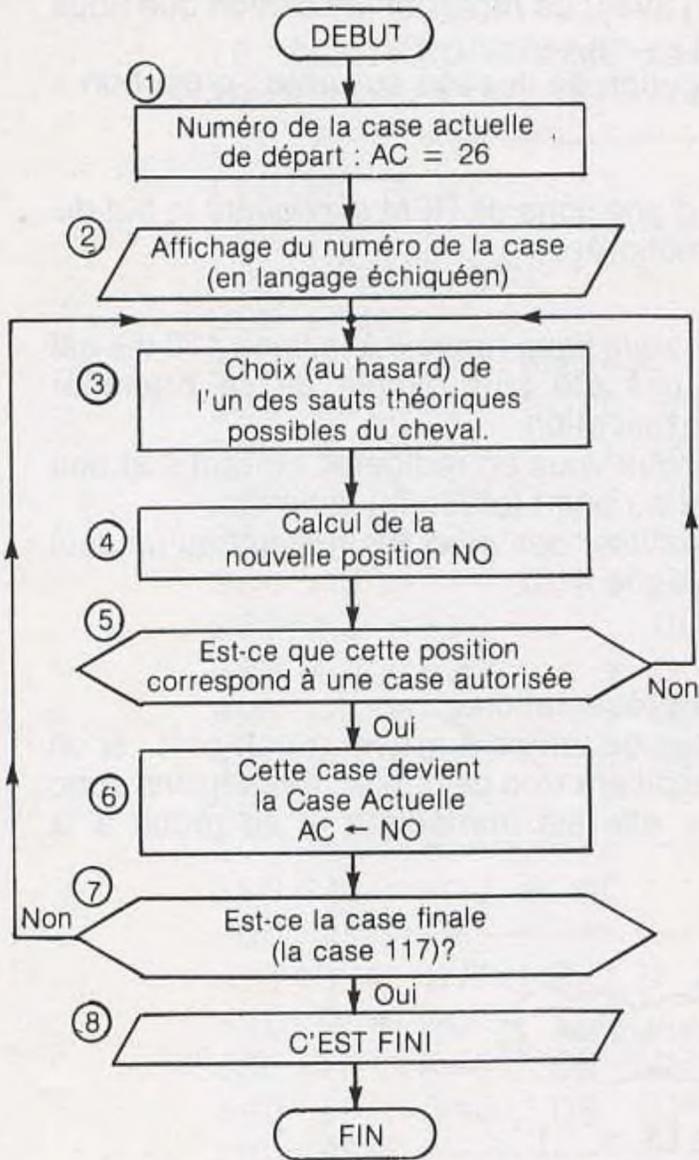


Figure 1

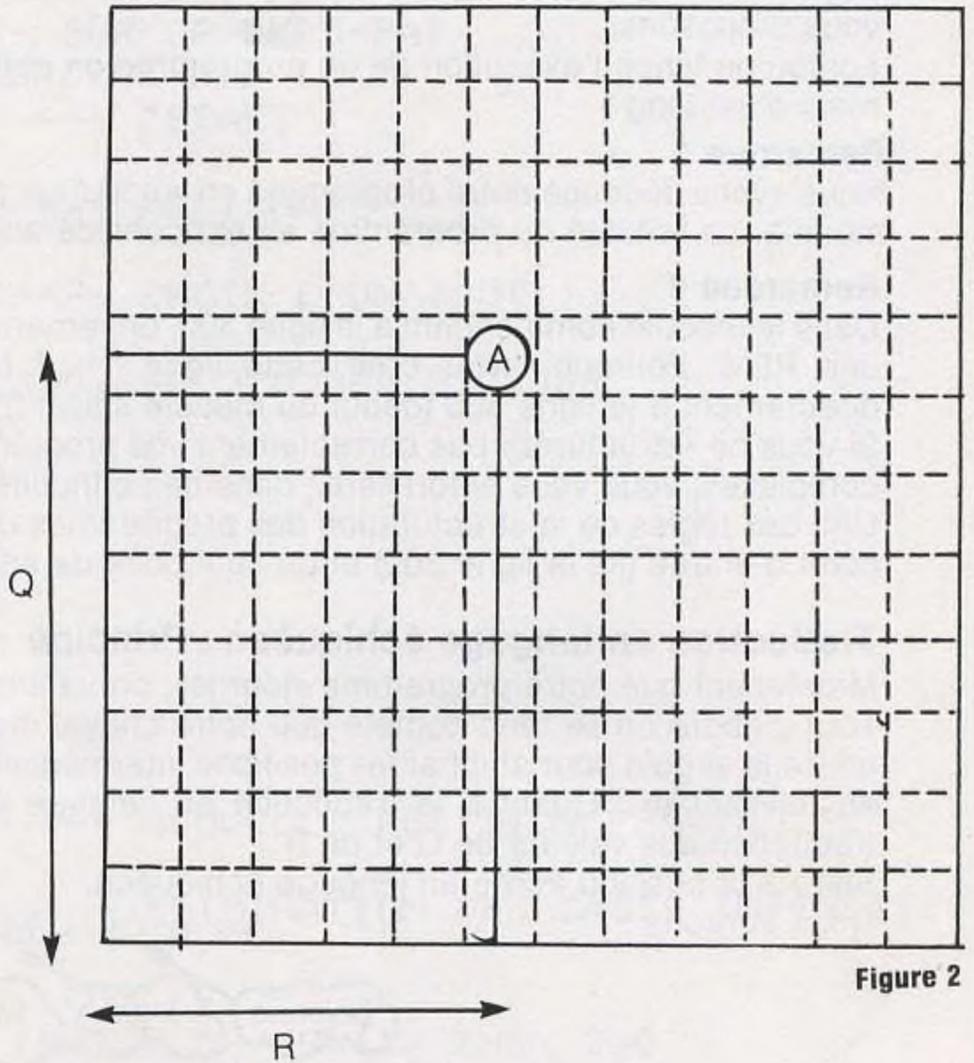


Figure 2

Figure 3

11	132	133	134	135	136	137	138	139	140	141	142	143
10	120	121	122	123	124	125	126	127	128	129	130	131
9	108	109	110	111	112	113	114	115	116	117	118	119
8	96	97	98	99	100	101	102	103	104	105	106	107
7	84	85	86	87	88	89	90	91	92	93	94	95
6	72	73	74	75	76	77	78	79	80	81	82	83
5	60	61	62	63	64	65	66	67	68	69	70	71
4	48	49	50	51	52	53	54	55	56	57	58	59
3	36	37	38	39	40	41	42	43	44	45	46	47
2	24	25	26	27	28	29	30	31	32	33	34	35
1	12	13	14	15	16	17	18	19	20	21	22	23
0	0	1	2	3	4	5	6	7	8	9	10	11

R = ∅ 1 2 3 4 5 6 7 8 9 10 11

Figure 3

Corrigés des exercices R10 et R11

R10 : microtraitement de texte (LED-MICRO n° 15, page 17)

J'ai mal rédigé l'énoncé de cet exercice. Tel quel, vous ne pouvez pas le faire puisque vous n'êtes pas censés connaître les INKEY\$ et GET A\$.

Attendez donc pour travailler sur cet exercice que vous ayez étudié :

— les INKEY\$ et GET A\$ (mais Bruno Lilamand vous en a déjà un peu parlé : LED-MICRO n° 12 page 42).

— les boucles (de façon à tester l'un après l'autre les caractères pour savoir où il y a des «espaces» qu'on peut augmenter pour «ajuster».

Cependant, je recevrai avec plaisir des solutions émanant de «fortiches» qui sont déjà au courant.

Dans ce cas, s'il vous plait :

— Précisez sur quel ordinateur, vous avez mis votre programme au point.

— Si c'est possible, envoyez-moi un listing et vos commentaires.

— Ne me proposez pas un programme complet de traitement de texte : limitez-vous à ce problème.

R11 : contrôle de validité d'une date (LED-MICRO n° 15 page 24)

Je manque de place. Mes excuses aux lecteurs qui m'ont envoyé leur solution. Nous en reparlerons dans un prochain numéro.

Il n'y a plus qu'à...

Il n'y a plus qu'à traduire en BASIC chacune des cases de l'organigramme. C'est ce que nous faisons page ci-contre. Essayez d'effectuer vous-même cette traduction avant de regarder la solution que nous vous proposons.

Lorsqu'on lance l'exécution de ce programme on obtient l'exécution de la page suivante : c'est bon... mais c'est long !

Remarque 1

Nous avons découpé notre programme en «modules» précédés d'une ligne de REM expliquant le but du module. La lisibilité du programme en est considérablement améliorée.

Remarque 2

Dans le module commençant à la ligne 300, on remarque que chaque ligne renvoie à la ligne 410 qui est une REM. Pourquoi avoir créé cette ligne 410 ? N'aurait-il pas été plus simple de se brancher directement à la ligne 500 (début du module suivant) ? Voici l'explication.

Si vous ne «structurez» pas correctement vos programmes dès que vous en rédigez un tant soit peu complexes, vous vous entortillerez dans des difficultés de mise au point (presque) sans fin.

Une des règles de la structuration des programmes consiste à utiliser des modules n'ayant qu'un seul point d'entrée (ici la ligne 300) et un seul point de sortie (ici la ligne 410).

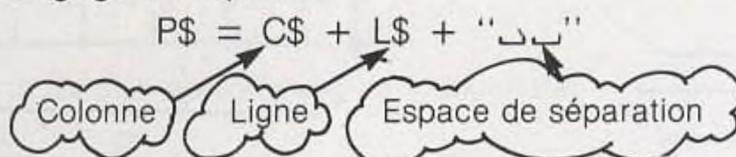
Traduction en langage échiquéen : Principe

Maintenant que notre programme «tourne», on va améliorer sa présentation.

Tout d'abord on se rend compte que notre cheval met beaucoup de temps à arriver à bon port : si on utilise la virgule pour afficher les positions intermédiaires, on gaspillera trop de papier. Resserrons donc en utilisant le ; Quant à la traduction en langage échiquéen, elle est immédiate et se réduit à la traduction des valeurs de Q et de R.

Soit P\$ la position écrite en langage échiquéen.

Nous écrivons :



Avec :

Si R = 2 Alors C\$ = "A"

Si Q = 2 Alors L\$ = "1"

Si R = 3 Alors C\$ = "B"

Si Q = 3 Alors L\$ = "2"

On pourrait ainsi obtenir C\$ et L\$ avec une collection de IF... THEN... Ce ne serait pas très élégant. Essayons autre chose.

Obtention de L\$

La correspondance :

Q	2	3	4	5	6	7	8	9
Ligne	1	2	3	4	5	6	7	8

se traduit par :

$$\text{Ligne} = Q - 1$$

Mais, comme il faut obtenir L\$ sous forme de chaîne de caractères, donc

$$L\$ = \text{STR\$}(Q - 1)$$

Obtention de C\$

Les correspondances

R	2	3	4	5	6	7	8	9
Colonne	A	B	C	D	E	F	G	H

et :

Lettre	A	B	C	D	E	F	G	H
Code ASCII	65	66	67	68	69	70	71	72

se résumant en

R	2	3	4	5	6	7	8	9
Code	65	66	67	68	69	70	71	72

c'est-à-dire : C\$ = CHR\$(R + 63)

Le programme...

```

10 REM PROGRAMME CHEVAL23- SUR CP-DK05-PAP
20 REM CHEVAL ERRANT - EXERCICE R12
100 REM ----- DEPART
110 AC = 26
120 REM ----- AFFICHAGE
130 PRINT AC,
300 REM ----- CHOIX D'UN SAUT
310 V = INT(RND*8)
320 ON V+1 GOTO 330,340,350,360,370,380,390,400
330 AJ = 10 : GOTO 410
340 AJ = 23 : GOTO 410
350 AJ = 25 : GOTO 410
360 AJ = 14 : GOTO 410
370 AJ = -10 : GOTO 410
380 AJ = -23 : GOTO 410
390 AJ = -25 : GOTO 410
400 AJ = -14 : GOTO 410
410 REM
500 REM ----- NOUVELLE POSITION PROVISOIRE
510 NO = AJ + AC
600 REM ----- ELIMINATION DES CASES INTERDITES
610 Q = INT(NO/12)
620 R = NO - 12*Q
630 IF (R=0) OR (R=1) OR (R=10) OR (R=11) THEN 300
640 IF (Q=0) OR (Q=1) OR (Q=10) OR (Q=11) THEN 300
700 REM ----- CE DEVIENT LA NOUVELLE POSITION
710 AC = NO
800 REM ----- UN AUTRE SAUT
810 IF AC<>117 THEN 120
900 REM ----- FIN
910 PRINT "ON EST ARRIVE"
990 END

```

... et son execution

26	40	54	64	50
27	52	38	28	42
67	90	67	57	67
42	32	55	69	92
102	92	78	103	80
66	80	66	41	27
52	66	80	57	80
103	78	55	32	57
80	103	93	68	78
88	98	75	65	51
41	31	45	55	80
55	41	55	78	53
43	53	39	29	43
66	76	101	91	101
91	77	91	101	91
116	93	79	104	90
65	90	104	90	65
79	56	33	43	57
32	42	32	57	32
etc...				

Le programme complet

Vous trouverez ci-après la version «complète» de ce programme et de son exécution sur un compatible IBM-PC.

Ce programme me conduit à faire quelques commentaires.

1°) Structuration

Du fait que le programme simple de départ (CHEVAL 23) est bien présenté, l'introduction de la traduction en langage échiquéen (lignes 700 à 750) s'effectue sans problème.

2°) Un plège

Je souhaitais que les positions des cases soient indiquées par une lettre suivie d'un chiffre mais sans espace entre le chiffre et la lettre.

Lorsque j'ai lancé l'exécution de mon programme : ô surprise ! un espace vient s'intercaler ! J'en étais resté à «mon» BASIC qui n'affichait pas d'espace devant une chaîne de caractères mais qui en affichait un devant un nombre positif (pour laisser la place de mettre un - pour les nombres négatifs). Pour remédier à cette (vilaine) présentation, il faudrait intercaler dans la ligne 740 un PRINT CHR\$(30).

```
10 REM PROGRAMME CHEVAL24- SUR CP-DK05-PAP
20 REM CHEVAL ERRANT - EXERCICE R12
100 REM ----- DEPART - ET AFFICHAGE
110 AC = 26
120 LPRINT "A 1 ";
300 REM ----- CHOIX D'UN SAUT
310 V = INT(RND*8)
320 ON V+1 GOTO 330,340,350,360,370,380,390,400
330 AJ = 10 : GOTO 410
340 AJ = 23 : GOTO 410
350 AJ = 25 : GOTO 410
360 AJ = 14 : GOTO 410
370 AJ = -10 : GOTO 410
380 AJ = -23 : GOTO 410
390 AJ = -25 : GOTO 410
400 AJ = -14 : GOTO 410
410 REM
500 REM ----- NOUVELLE POSITION PROVISOIRE
510 NO = AJ + AC
600 REM ----- ELIMINATION DES CASES INTERDITES
610 Q = INT(NO/12)
620 R = NO - 12*Q
630 IF (R=0) OR (R=1) OR (R=10) OR (R=11) THEN 300
640 IF (Q=0) OR (Q=1) OR (Q=10) OR (Q=11) THEN 300
700 REM ----- CE DEVIENT LA NOUVELLE POSITION - AFFICHAGE
710 AC = NO
720 L$ = STR$(Q-1)
730 C$ = CHR$(R+63)
740 P$ = C$ + L$ + " "
750 LPRINT P$;
800 REM ----- UN AUTRE SAUT
810 IF AC<>117 THEN 300
900 REM ----- FIN
910 LPRINT "ON EST ARRIVE"
990 END
```

A 1	C 2	E 3	C 4	A 3	B 1	C 3	A 2	C 1	E 2	F 4	E 6	F 4	H 3	F 4	E 2
G 1	F 3	H 4	G 6	E 7	G 6	E 5	F 7	G 5	E 4	G 5	E 4	D 2	B 1	C 3	E 4
G 5	H 3	G 5	F 7	E 5	F 3	G 1	H 3	G 5	F 7	H 6	G 4	E 5	C 6	A 7	B 5
D 4	B 3	D 2	F 1	H 2	F 3	G 5	F 3	D 2	F 3	E 5	D 3	F 2	D 3	B 2	D 1
F 2	E 4	C 5	D 7	F 6	D 7	F 6	D 5	F 6	D 7	F 6	G 8	H 6	F 5	G 7	E 6
D 4	E 6	G 7	E 6	D 4	F 5	G 3	H 1	F 2	H 3	G 1	E 2	G 1	H 3	G 1	E 2
G 3	E 2	C 3	D 1	F 2	E 4	D 6	E 8	F 6	E 8	G 7	F 5	D 6	F 7	G 5	H 3
F 2	D 1	E 3	C 2	B 4	D 3	F 4	H 3	F 4	H 5	G 7	E 6	C 5	A 6	C 7	D 5
B 4	A 6	C 7	E 6	C 5	E 4	G 3	E 2	C 1	A 2	B 4	C 2	E 1	F 3	D 4	B 5
C 3	B 1	C 3	A 2	C 1	D 3	B 4	A 6	C 7	E 8	F 6	H 7	F 6	E 4	F 6	G 4
E 5	D 7	B 8	A 6	C 5	A 4	C 5	E 4	C 5	B 7	D 8	B 7	D 8	B 7	D 6	E 4
G 5	H 3	G 1	E 2	F 4	G 2	E 1	C 2	E 1	C 2	B 4	D 5	E 3	D 5	E 3	C 2
E 1	G 2	E 1	D 3	B 4	C 6	E 5	G 4	F 6	E 4	G 3	H 1	G 3	E 2	G 3	E 4
D 2	C 4	E 3	G 2	E 1	G 2	F 4	G 2	H 4	F 5	H 4	G 2	F 4	E 6	F 8	D 7
C 5	A 6	C 5	E 6	D 8	B 7	D 6	F 7	D 6	B 5	A 3	B 5	D 4	B 3	D 2	C 4
D 2	F 1	E 3	F 1	E 3	G 2	H 4	F 3	G 1	H 3	F 2	H 1	G 3	H 1	F 2	E 4
G 5	F 7	G 5	F 3	E 1	F 3	H 2	F 3	E 1	C 2	D 4	C 2	D 4	E 6	C 5	A 6
C 7	E 8	C 7	A 6	C 5	E 6	F 8	E 6	F 4	G 6	F 4	D 3	B 2	C 4	E 3	G 2
E 3	F 1	D 2	B 3	A 5	B 7	D 8	E 6	F 8	G 6	F 8	G 6	F 8	H 7	F 8	H 7
F 6	E 8	C 7	B 5	C 7	E 8	G 7	F 5	D 6	E 4	C 3	D 5	C 7	D 5	F 4	H 5
G 7	H 5	F 4	H 5	F 4	E 6	F 4	G 6	H 4	G 6	E 7	F 5	E 3	F 1	G 3	E 2
C 3	D 1	B 2	D 1	B 2	D 1	C 3	D 1	C 3	D 5	B 6	D 5	F 6	G 8	F 6	H 5
G 7	E 8	D 6	E 4	G 3	E 4	D 2	B 1	C 3	A 2	B 4	D 3	B 2	D 1	B 2	D 1
B 2	A 4	B 2	D 1	C 3	B 1	C 3	A 4	C 5	B 7	D 6	E 4	D 2	C 4	D 2	B 1
D 2	E 4	F 2	E 4	C 5	E 6	F 4	E 6	G 5	E 6	F 4	G 2	E 1	F 3	D 4	F 3
E 1	F 3	G 1	H 3	G 1	H 3	F 4	G 6	F 8	G 6	H 8	ON EST ARRIVE				

Variante au programme du cheval errant

Au moment d'envoyer mon manuscrit à la photocomposition, je reçois plusieurs solutions à ce «cheval errant», dont certaines sont assez différentes de celle que je viens de vous proposer :

- une solution qui calcule les positions du cavalier directement en coordonnées X, Y et sur un échiquier limité à 8 cases × 8 cases
- une solution d'un lecteur «en avance» sur notre cours (utilisant un GOSUB) et qui prévoit ce que nous vous proposerons plus tard (partir de n'importe quelle case de l'échiquier pour arriver à n'importe quelle autre). Nous en reparlerons dans le prochain numéro.

Puisque beaucoup de nos lecteurs semblent très en avance sur notre cours, y en aurait-il un qui serait encore plus en avance et pourrait nous proposer une analyse et une solution pour le problème suivant :

«On donne (au hasard) une case de départ, et (au hasard) une case d'arrivée. On voudrait que le cheval arrive dans un nombre minimum de coups. Je n'ai pas beaucoup réfléchi à ce problème (et je vais peut-être engager nos lecteurs dans des voies sans issue ?) mais a priori je vois deux approches :

- l'une dans laquelle c'est le programmeur qui trouve et qui définit la règle (la «stratégie»);
- l'autre dans laquelle l'ordinateur cherche «à apprendre» et à trouver sa stratégie (ça doit être ça l'intelligence artificielle ? Non ?)

3.16.12. La sélection multiple avec des IF... THEN...

A. Sélection simple

Les instructions

IF... THEN...

ou

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

permettent d'effectuer un choix entre deux possibilités (le IF) puis un aiguillage vers l'un ou l'autre des deux traitements (le THEN... ELSE). Ce choix entre deux possibilités et deux possibilités seulement est symbolisé par les organigrammes des figures 1 et 2.

La figure 1 (représentation «étalée») est plus lisible que la figure 1 (représentation «linéaire»), mais la figure 2 est plus proche de sa traduction en programme.

B. Sélection multiple

Il est clair qu'en «empilant» plusieurs sélections simples on peut effectuer des choix entre 3, 4, 5, 6... possibilités et d'aiguiller vers le traitement correspondant. Nous avons déjà appliqué cette sélection multiple avant de savoir comment on l'appelait. Ce choix peut s'effectuer de deux façons.

— En parcourant successivement toutes les possibilités

Ceci est illustré par l'organigramme de la figure 4.

Nous en avons eu un exemple dans notre exercice de récapitulation R7 (les capitales de l'Amérique du Sud).

Lorsque les traitements à effectuer sont très courts, on obtient une présentation très lisible en rédigeant le traitement sur la même ligne que le IF... THEN... : voir la solution CAPITA02 de cet exercice R7.

— En utilisant une structure «arborescente» représentée par l'organigramme de la figure 5.

Encore un peu d'anglais informatique

true : vrai

false : faux

to jump : sauter

to choose : choisir

pattern : schéma

one way selection : sélection simple

two way selection : sélection double

multiway selection : sélection multiple

flow chart : organigramme

less than : plus petit que

equal to : égal à

greater than : plus grand que

in other case : dans l'autre cas

goes to ligne 200 : va à la ligne 200

The GOTO statement provide a unconditional branching by transferring control to a specified line.

L'instruction GOTO fournit un branchement inconditionnel en transférant la commande à la ligne désignée.

The ON... GOTO (computed GOTO) statement allows control to be transferred to...

L'instruction ON... GOTO (GOTO calculée) permet de transférer la commande...

The relation «less than or equal to» is denoted by not greater.

La relation «moins que ou égal à» est appelée «pas plus grand que».

Sélection simple

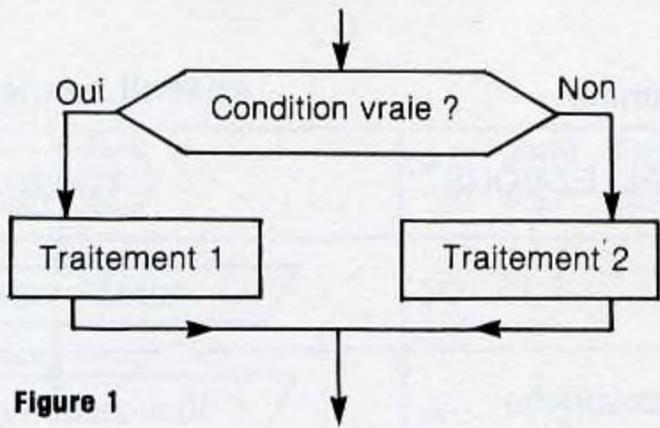


Figure 1

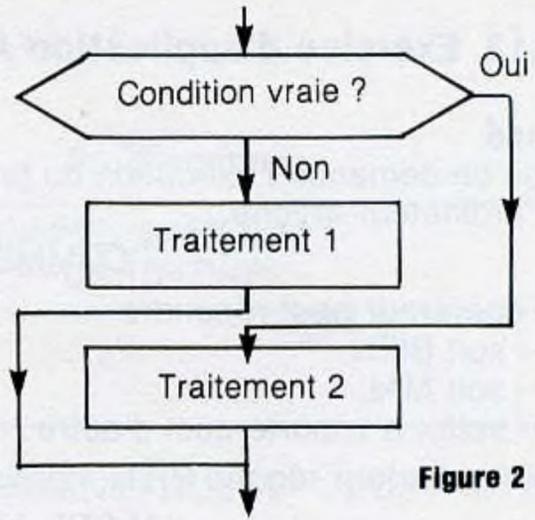


Figure 2

Sélection multiple

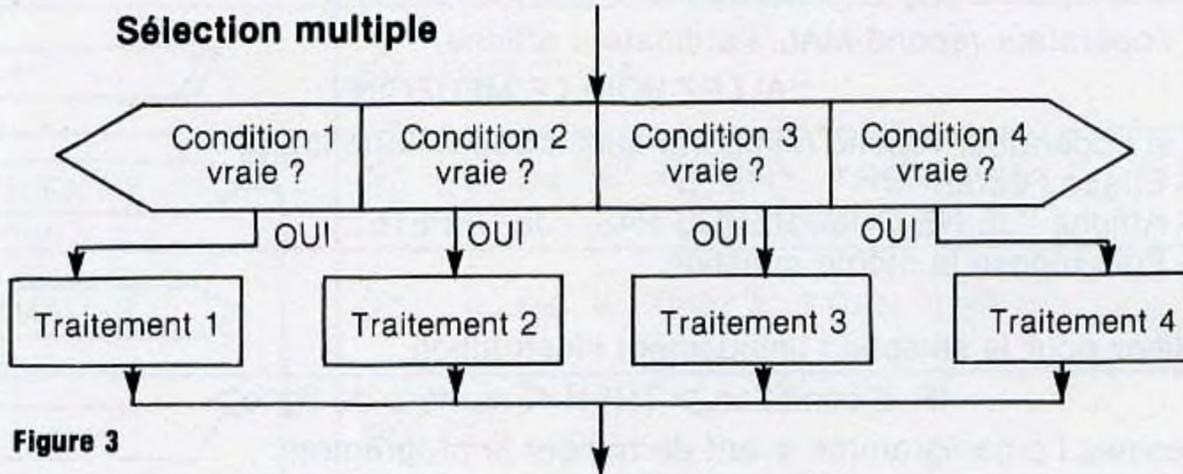


Figure 3

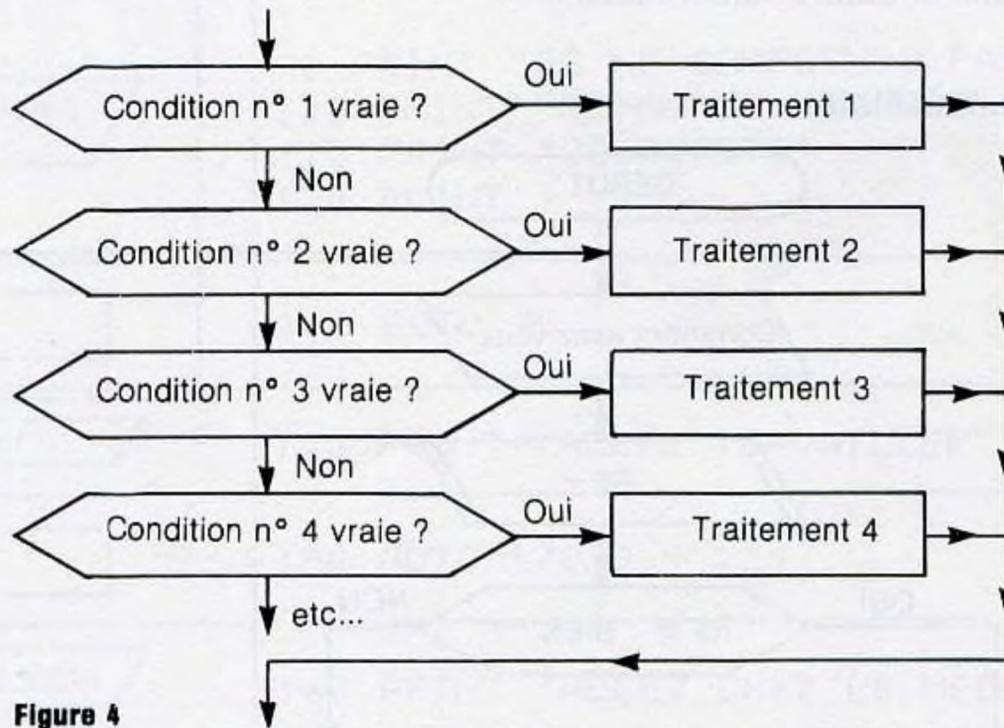


Figure 4

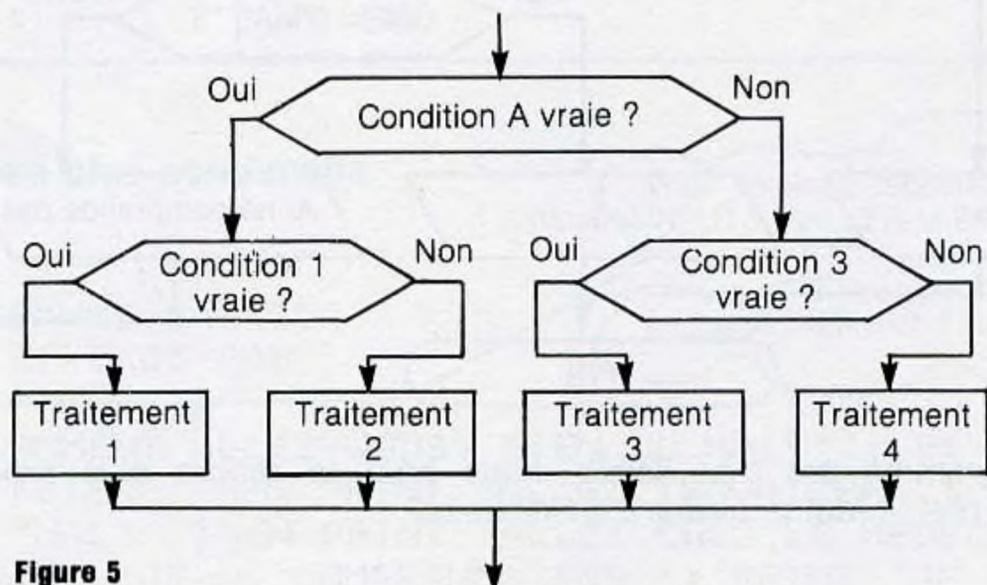


Figure 5

3.16.13. Exercice d'application A47 : Comment allez-vous ?

Énoncé

Lorsqu'on demande l'exécution du programme

1°) L'ordinateur affiche

"COMMENT ALLEZ-VOUS ?"

2°) L'opérateur peut répondre

— soit BIEN

— soit MAL

— soit... n'importe quoi d'autre.

3°) Si l'opérateur répond BIEN, l'ordinateur affiche

"ALORS AU TRAVAIL... ET VITE"

4°) Si l'opérateur répond MAL, l'ordinateur affiche

"ALLEZ VOIR LE MEDECIN"

5°) et si l'opérateur répond n'importe quoi d'autre, l'ordinateur

— Efface l'écran

— Affiche "JE NE COMPRENDS PAS... JE REPETE"

— Puis repose la même question.

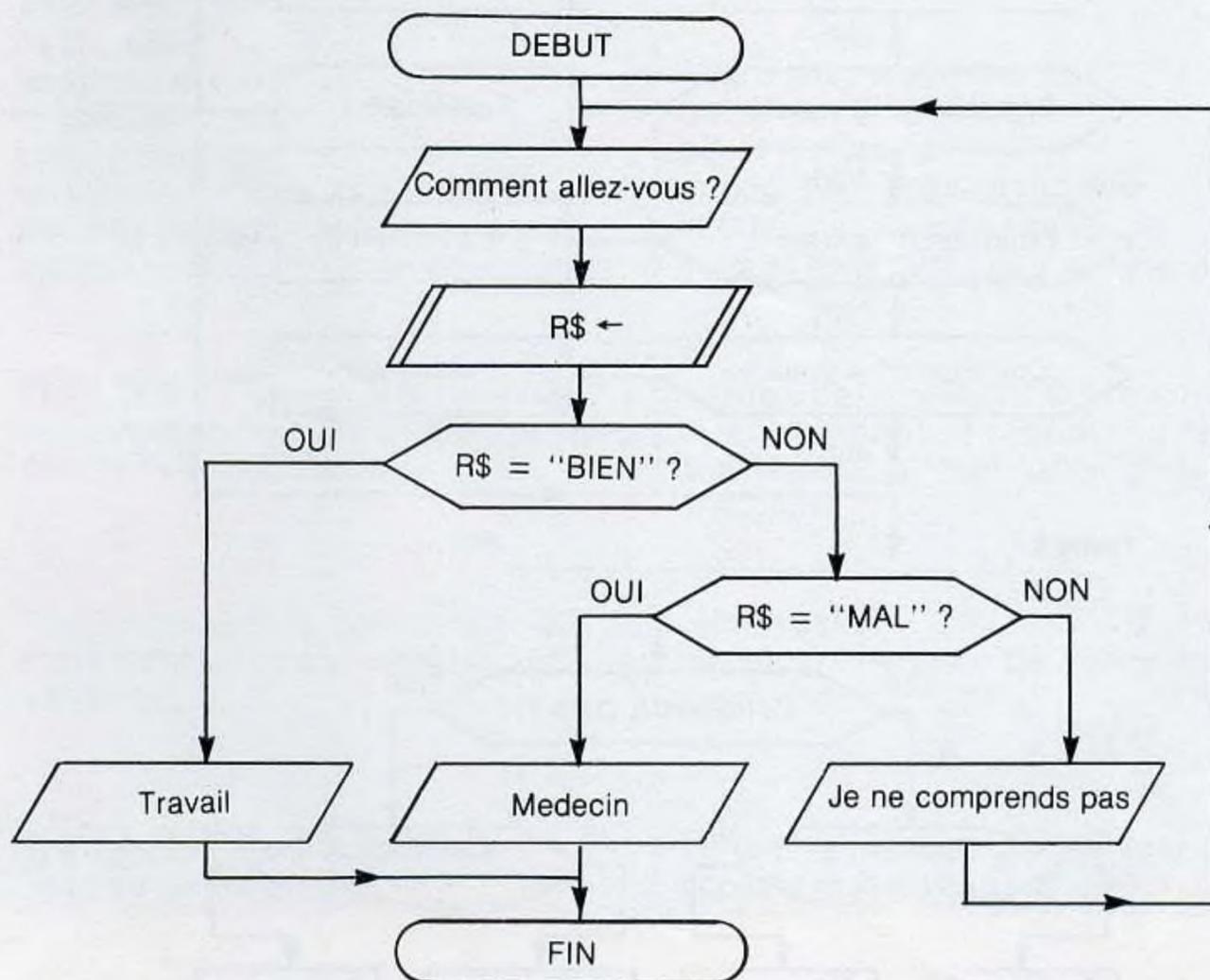
⊗ Utilisez pour la sélection uniquement l'instruction :

IF <condition> THEN <numéro de ligne>

⊗ Dessinez l'organigramme avant de rédiger le programme.

⊗ Utilisez une structure «arborescente».

Organigramme «étalé»



Cet organigramme est très lisible. Mais pour le rendre plus proche du programme réel, il faut le traduire en «linéaire».

Une solution de A47

Organigramme linéaire

Programme

DEBUT	10 REM PROGRAMME SANTE01 20 REM SUR CP-DK05-PAP
Effacer l'écran	30 CLS
Comment allez-vous ?	40 PRINT "COMMENT ALLEZ-VOUS? BIEN OU MAL ?"
R\$ ←	50 INPUT R\$
R\$ = "BIEN" ?	60 IF R\$ = "BIEN" THEN 140
R\$ = "MAL" ?	70 IF R\$ = "MAL" THEN 160
Effacer l'écran	80 CLS
Je ne comprend pas Je répète	90 PRINT "JE NE COMPRENDS PAS" 100 PRINT "REPONDEZ MOI BIEN OU MAL" 110 PRINT "JE REPETE" 120 PRINT
	130 GOTO 40
Allez travailler	140 PRINT "ALLEZ TRAVAILLER"
	150 GOTO 170
Allez chez le médecin	160 PRINT "ALLEZ CHEZ LE MEDECIN"
FIN	170 END

Une solution plus condensée

```

10 REM PROGRAMME SANTE02
20 REM SUR CP-DK05-PAP
30 CLS
40 INPUT "COMMENT ALLEZ-VOUS? BIEN OU MAL " ; R$
50 IF R$ = "BIEN" THEN PRINT "ALLEZ TRAVAILLER" : GOTO 80
60 IF R$ = "MAL" THEN PRINT "ALLEZ CHEZ LE MEDECIN" : GOTO 80
70 CLS : PRINT "JE NE COMPRENDS PAS " : PRINT "JE REPETE" : GOTO 40
80 END
    
```

Vous ne vous rappelez pas cet emploi de INPUT ? Voir LED-MICRO n° 14 page 18.

Inutile de mettre un ? : le INPUT s'en charge.

★ 3.16.14. Exercice d'application A48 : Equation du second degré

Enoncé

1°) L'ordinateur affiche le texte

Equation du second degré
 $AX^2 + BX + C = 0$

Entrez successivement les valeurs de A, B et C

2°) L'opérateur entre les valeurs de A, B et C
(utilisez une seule instruction)

3°) L'ordinateur effectue le calcul

4°) Puis affiche le résultat sous une forme telle que

Deux racines :

$X_1 = 2$

$X_2 = -1$

(ou un texte du même genre)

5°) Et demande à l'utilisateur s'il veut calculer une autre équation.

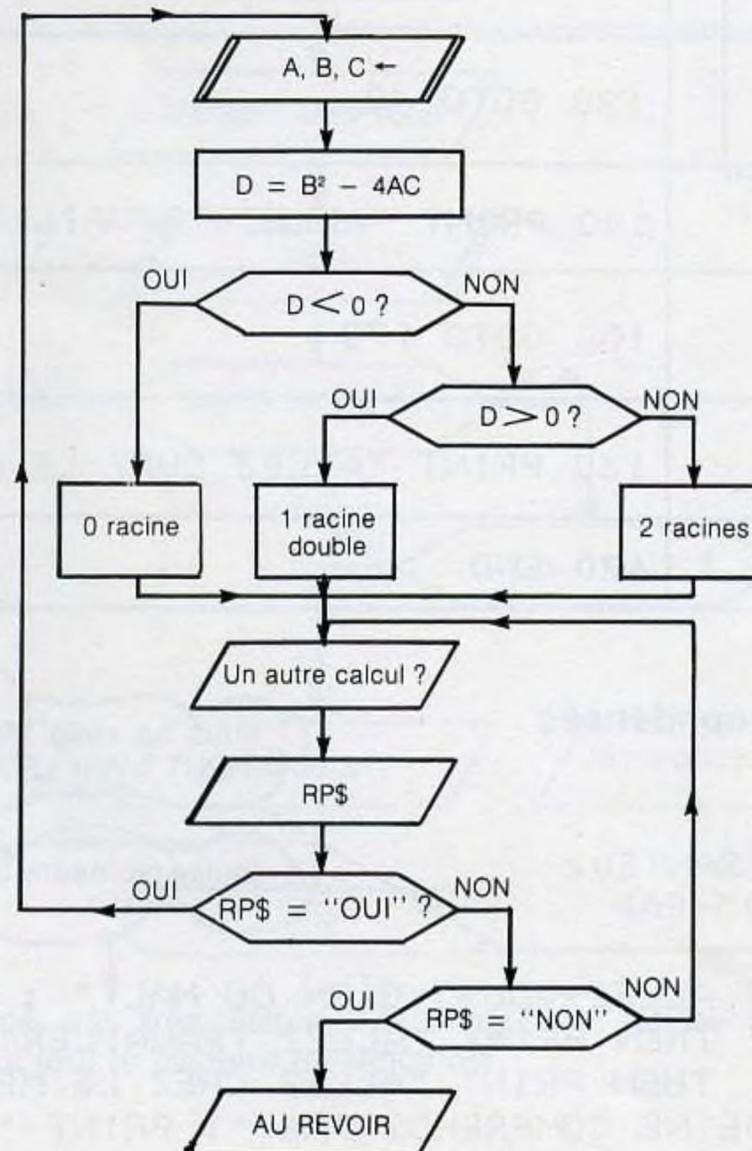
6°) Si l'utilisateur veut continuer, on repose le même problème. Sinon l'ordinateur lui dit «au revoir» et s'arrête (revenant au mode commande).

Remarque

Cet exercice est précédé d'une astérisque, ce qui signifie «nécessite des connaissances en mathématiques supérieures à la classe de 3^e». Si ce n'est pas (ou si ce n'est plus) votre cas : vous pouvez sauter ce paragraphe sans grand inconvénient.

Organigramme simplifié et «étalé»

Supposons $A = 0$ (voir page ci-contre pour un organigramme plus complet)



Un exemple de programme

Bien sûr, le discriminant ne peut pas s'appeler Δ, car le BASIC ne connaît pas le grec.

Vous pouvez le réduire à une seule ligne :
50 INPUT "ENTREZ LES COEFFICIENTS" ; A, B, C

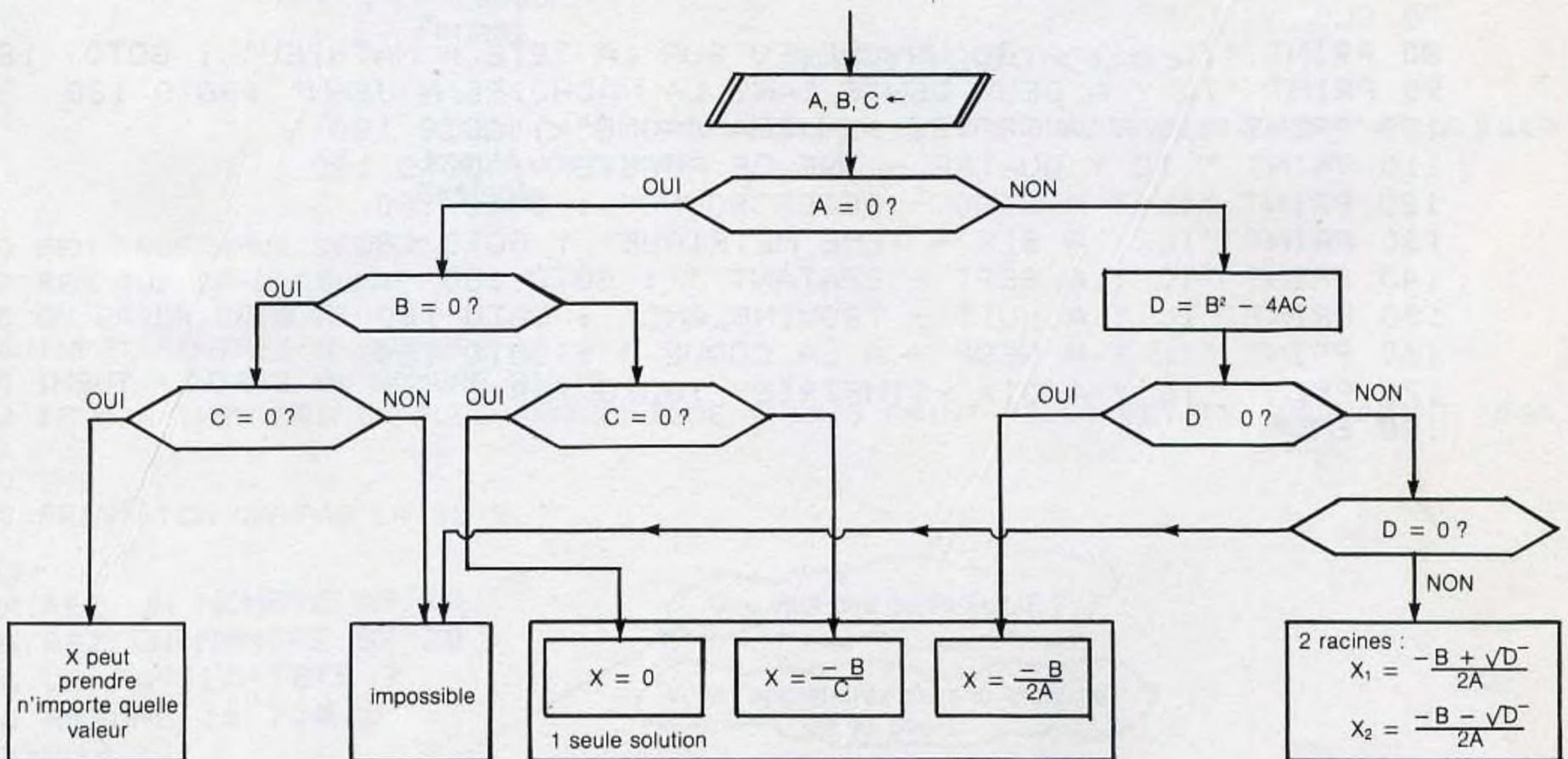
```

10 REM PROGRAMME EQUAT12
20 REM SUR CP-DK05-PAP
30 CLS
40 PRINT "EQUATION DU 2 DEGRE : AX2 + BX + C = 0"
50 INPUT "ENTREZ LA VALEUR DE A " ; A
60 INPUT "ENTREZ LA VALEUR DE B " ; B
70 INPUT "ENTREZ LA VALEUR DE C " ; C
80 D = B*B - 4*A*C
90 IF D < 0 THEN PRINT "PAS DE SOLUTION" : GOTO 150
100 IF D = 0 THEN PRINT "UNE RACINE DOUBLE = " ; (-B)/(2*A) : GOTO 150
110 E = SQR(D)
120 PRINT " 2 RACINES "
130 PRINT " X1 = " ; (-B + E)/(2*A)
140 PRINT " X2 = " ; (-B - E)/(2*A)
150 PRINT
160 INPUT "UN AUTRE CALCUL (OUI/NON) " ; RP$
170 IF RP$ = "OUI" THEN 30
180 IF RP$ = "NON" THEN 200
190 PRINT "JE NE COMPRENDS PAS JE REPETE" : GOTO 160
200 CLS
210 PRINT "AU REVOIR"
220 END
    
```

Une présentation habituelle pour préciser comment on souhaite que l'utilisateur réponde.

Utilisez des noms de variables qui rappellent leur objet : RP = abréviation de REPONSE.

Un organigramme plus complet



3.16.15. L'instruction ON... GOTO...

A. Définition

Lorsqu'on a à programmer une sélection multiple comportant un choix entre 3, 4, 5, 6.. cas, au lieu d'utiliser une cascade de IF... THEN..., on a généralement intérêt à utiliser l'instruction

ON... GOTO... , ... , ... ,

Raisonnons sur un exemple.

L'instruction :

```
50 ON N GOTO 200, 250, 310, 140, 200, 400
```

signifie :

Si N = 1 → allez vous brancher à l'instruction 200

Si N = 2 → allez vous brancher à l'instruction 250

Si N = 3 → allez vous brancher à l'instruction 310

Si N = 4 → allez vous brancher à l'instruction 140

Si N = 5 → etc...

B. Format

Apprenons à utiliser le jargon des formats (voir LED-MICRO n° 9 §3.12.3 page 15).

avec : ON <n> GOTO <ligne> [, <ligne>] ...

<n> : expression numérique arrondie à un nombre entier si nécessaire. Le résultat doit être compris entre 0 et 255.

<ligne> : numéro de ligne auquel l'ordinateur ira se brancher. La valeur de n détermine le numéro de la liste.

C. Exemple

```
10 REM PROGRAMME COMPT12
20 REM SUR CP-DK05-PAP
30 CLS
40 PRINT "DONNEZ UN NOMBRE COMPRIS ENTRE 1 ET 10"
50 INPUT N
60 ON N GOTO 80,90,100,110,120,130,140,150,160,170
70 CLS
80 PRINT "IL N'Y A QU'UN CHEVEU SUR LA TETE A MATHIEU" : GOTO 180
90 PRINT "IL Y A DEUX DENTS DANS LA MACHOIRE A JEAN" :GOTO 180
100 PRINT "IL Y A TROYES EN CHAMPAGNE" : GOTO 180
110 PRINT " IL Y QUATRE - INE DE RUSSIE" : GOTO 180
120 PRINT "IL Y A CINQ - PETERSBOURG" : GOTO 180
130 PRINT "IL Y A SIX - TEME METRIQUE" : GOTO 180
140 PRINT "IL Y A SEPT - EPATANT " : GOTO 180
150 PRINT "IL Y A HUIT - TROVINBLANC" : GOTO 180
160 PRINT "IL Y A NEUF - A LA COQUE " : GOTO 180
170 PRINT "IL Y A DIX -SIMETRIE" :GOTO 180
180 END
```

3.16.16. L'Instruction ON ERROR GOTO...

Une instruction qui serait commode... si elle existait

Certains BASICs (dont celui de Hewlett-Packard, si mes souvenirs sont exacts) permettent d'utiliser une instruction IF... THEN... ELSE... à branchement multiple
IF... THEN... ELSE... ELSE... ELSE...

Essayons donc le programme ci-dessous :

```
10 REM PROGRAMME SELECO5
20 REM SUR CP-DK05-PAP
30 INPUT "ENTREZ UN NOMBRE A" ; A
40 INPUT "ENTREZ UN NOMBRE B" ; B
50 IF A>B PRINT "PLUS GRAND" ELSE IF A<B PRINT "PLUS PETIT" ELSE PRINT "EGAL"
60 END
```

Résultat :

```
RUN
ENTREZ UN NOMBRE A? 10
ENTREZ UN NOMBRE B? 20
Syntax error in 50
OK
```

Je me fais insulter. «Mon» BASIC (Microsoft) n'aime pas l'instruction

```
50 IF A>B PRINT "PLUS GRAND" ELSE IF A<B PRINT "PLUS PETIT" ELSE PRINT "EGAL"
```

Une ambition limitée

Lorsque nous nous mettrons à rédiger de longs programmes, il est vraisemblable que nos programmes comporteront de nombreuses erreurs. La partie n° 4 de ce cours sera consacrée

— à l'art de réduire ses erreurs de programmation (grâce à la programmation structurée, en particulier)

— et à l'art de «mettre au point» (c'est-à-dire d'éliminer les erreurs restantes).

Pour ce faire, nous étudierons (et pratiquerons) des instructions BASIC plus spécialement destinées à la mise au point : TRON, TROFF, STOP, CONT, RESUME. Contentons-nous pour le moment de citer simplement l'une de ces instructions de mise au point : ON ERROR GOTO.

ON ERROR GOTO

Effet

L'instruction ON ERROR GOTO

— effectue un déroutement en cas d'erreur

— signale le numéro de la première ligne où a eu lieu l'erreur.

Format

ON ERROR GOTO <ligne>

avec :

<ligne > = numéro de la ligne où l'ordinateur devra se brancher dès qu'il aura détecté une erreur.

Exemple

```
10 REM PROGRAMME SELECO6
20 REM SUR CP-DK05-PAP
25 ON ERROR GOTO 70
30 INPUT "ENTREZ UN NOMBRE A" ; A
40 INPUT "ENTREZ UN NOMBRE B" ; B
50 IF A>B THEN PRINT "PLUS GRAND" ELSE IF A<B PRINT "PLUS PETIT" ELSE PRINT "EGAL"
60 END
70 PRINT "CA VA PAS LA TETE ?"
```

```
RUN
ENTREZ UN NOMBRE A? 10
ENTREZ UN NOMBRE B? 20
CA VA PAS LA TETE ?
No RESUME in 70
OK
```

Que veut dire ce RESUME ?

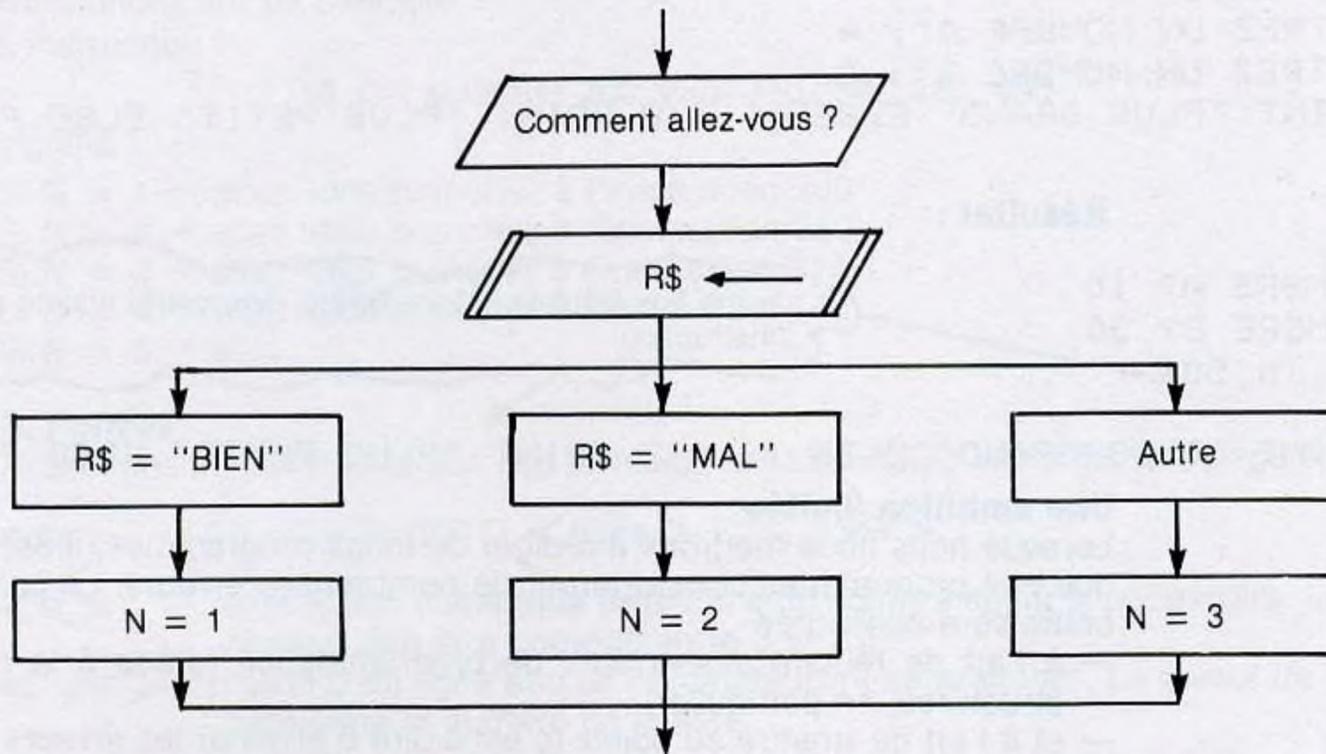
Vous l'apprendrez dans la 4^e partie de ce cours.

3.16.17. Exercices d'application A49 et A50

Enoncé de A49

Exactement identique à celui de l'exercice A47 (Comment allez-vous?). Mais à rédiger en utilisant une seule instruction ON... GOTO... à la place des IF... THEN... Chercher avant de lire les solutions ci-dessous.

Une solution peu élégante



```
10 REM PROGRAMME SANTE03
20 REM SUR CP-DK05-PAP
30 CLS
40 INPUT "COMMENT ALLEZ-VOUS? BIEN OU MAL " ; R$
50 IF R$ = "BIEN" THEN N = 1
60 IF R$ = "MAL" THEN N = 2
70 IF (R$ <> "BIEN") AND (R$ <> "MAL") THEN N = 3
80 ON N GOTO 90,100,110
90 PRINT "ALLEZ TRAVAILLER" ; GOTO 120
100 PRINT "ALLEZ CHEZ LE MEDECIN" ; GOTO 120
110 PRINT "JE NE COMPRENDS PAS JE REPETE " ; GOTO 40
120 END
```

Vous comprenez ?
L'instruction 70 veut dire :
Si R\$ n'est égal NI à
"BIEN" NI à "MAL"

Enoncé de A50

L'exemple ci-dessus n'est guère convaincant de l'intérêt de ON.. GOTO : l'introduction de la variable N n'y est qu'un intermédiaire inutile.

Essayez de supprimer les lignes 50 à 70 en remplaçant N par une expression booléenne dans 80.

Cherchez un peu avant de regarder la solution (page ci-contre).

Une solution pour A50

Pour faciliter les explications, supposons, pour commencer, que nous travaillons en APPLESOFT, avec lequel :

(condition vraie) = + 1

(condition fausse) = 0

Dans ce cas, les trois lignes 50, 60 et 70 pourront être remplacées par :

$$N = \underbrace{1}_{\text{terme 1}} + \underbrace{(R\$ = \text{"BIEN"})}_{\text{terme 2}} + \underbrace{2 \star (R\$ = \text{"MAL"})}_{\text{terme 3}}$$

+ Si l'opérateur tape n'importe quoi, par exemple R\$ = "GLOP", on a
la condition (R\$ = "BIEN") est fausse donc terme 2 = 0
la condition (R\$ = "MAL") est fausse donc terme 3 = 0
et $N + 1 = 0 + 0 = 1$

+ Si l'opérateur tape "BIEN" :
la condition (R\$ = "BIEN") est vraie donc terme 2 = 1
la condition (R\$ = "MAL") est fausse donc terme 3 = 0
et $N = 1 + 1 + 0 = 2$

+ Si l'opérateur tape "MAL" :
la condition (R\$ = "BIEN") est fausse donc terme 2 = 0
la condition (R\$ = "MAL") est vraie donc terme 3 = $2 \star 1 = 2$
et $N = 1 + 0 + 2 = 3$

Avec les Basics Microsoft, la condition vraie se traduit par - 1.

Donc pour obtenir les valeurs 1, 2 et 3 pour N, il faut écrire :

$$N = 1 - (R\$ = \text{"BIEN"}) - 2 \star (R\$ = \text{"MAL"})$$

Comme nous vous avons demandé de remplacer les 4 lignes 50 à 80 par une seule, on aboutit finalement au programme ci-dessous.

```
10 REM PROGRAMME SANTE04
20 REM SUR CP-DK05-PAP
30 CLS
40 INPUT "COMMENT ALLEZ-VOUS? BIEN OU MAL " ; R$
80 ON 1 -(R$ = "BIEN") - 2*(R$ = "MAL") GOTO 110,90,100
90 PRINT "ALLEZ TRAVAILLER" : GOTO 120
100 PRINT "ALLEZ CHEZ LE MEDECIN" : GOTO 120
110 PRINT "JE NE COMPRENDS PAS JE REPETE " : GOTO 40
120 END
```

3.16.18. Exercice de récapitulation R13 : les Impôts

Y-a-t'il parmi nos lecteurs un inspecteur des impôts ou un expert fiscal qui pourrait nous fournir le montant imposable en fonction

- des revenus
- du nombre de personnes à charge
- de... je ne sais pas moi : c'est vous l'expert fiscal.

Essayez de rédiger ce programme de façon à ce qu'il pose des questions faciles à l'utilisateur (par exemple : combien avez-vous d'enfants mineurs, plutôt que «nombre de parts»).

Cherchez à présenter le programme en le décomposant en modules précédés d'un REM d'explications (comme dans nos programmes CHEVAL 23 et CHEVAL 24).

P.S. - Si un lecteur pouvait nous imaginer un programme nous aidant à payer moins d'impôts (en nous rappelant quelles sont les déductions fiscales auxquelles nous avons droit, par exemple), il serait le bienvenu. Même si, pour rédiger ce programme il utilisait des instructions non encore étudiées !

3.16.19. Les menus - Exercices R14 et R15

Nous vous avons déjà parlé plusieurs fois des «menus» qui permettent de réaliser des progiciels très «conviviaux» (faciles à utiliser pour ceux qui préfèrent le français simple au français informatique). La figure ci-dessous vous représente le «menu général» du progiciel de patrons personnalisés ALAMOD.

M E N U G E N E R A L

POUR :

TAPEZ :

1	obtenir le CATalogue de la disquette fichiers..CAT	
2	lancer l'EXEcution du modèle en mémoire centr..EXE	
3	Créer un MoDèle.....KMD	-
4	cHercher un MoDèle existant.....HMD	
5	Créer un MoRceau.....KMR	-
6	cHercher un MoRceau existant.....HMR	
7	Créer un MANnequin.....KMA	-
8	cHercher un MANnequin existant.....HMA	
9	Créer des paramètres de SOrtie.....KSO	
10	cHercher des paramètres de SOrtie existants....HSO	
11	Créer un PLacement du modèle.....KPL	
12	cHercher un PLacement existant.....HPL	
13	Créer des Conditions de Tracé morceau.....CCT	
14	cHercher des Conditions de Tracé existtant....HCT	
15	ARReter de travailler.....ARR	

Exercice de récapitulation R14

Rédigez un programme qui, lorsque l'utilisateur tape un chiffre de 1 à 15 (puis frappe ENTER) aiguille le traitement vers une ligne donnée du programme.

(par exemple : 1.000 s'il tape 1, 2.000 s'il tape 2, etc.)

Arrêtez-vous à 6 : ça suffit pour le principe. Utilisez uniquement des ON... GOTO...

Exercice de récapitulation R15

Rapprochons-nous un peu plus du programme ALAMOD «réel» :

Pour créer un modèle (abréviation KMD – K = créer – MD = modèle) l'utilisateur peut taper :

- soit le chiffre 3
- soit le sigle KMD

Pour définir les mensurations d'une cliente (ou d'une taille standard), l'utilisateur pourra taper soit le chiffre 7 soit le sigle KMA. (Ceci a pour but de permettre à un utilisateur exercé de «court-circuiter» l'enchaînement des menus. Nous y reviendrons).

Remarque

Bien sûr, le «vrai» programme ALAMOD comporte des menus beaucoup plus élaborés (utilisant des ON... GOSUB, traitant les erreurs de frappe, etc.). Nous y reviendrons. Notons notre parfaite conformité de vue avec le lecteur qui nous demande de prendre un petit exemple simple (sa feuille de paie) et de le perfectionner au fur et à mesure du progrès de nos connaissances.

3.16.20. Exercice de récapitulation n° R16 : le mot le plus long

L'exercice que nous vous proposons ci-dessous est fortement inspiré de l'émission de jeux télévisés «Des chiffres et des lettres». La règle du jeu est un peu différente : peu importe, notre but est de vous exercer à la programmation. Ce jeu peut se jouer à 2, 3 ou 4 personnes.

L'arbitre (ou l'un des joueurs) tape sur le clavier de son ordinateur un mot de 7 lettres (par exemple : SOIGNER) Dès qu'il a fini de taper ce mot (c'est-à-dire dès qu'il a confirmé en tapant sur ENTER), l'ordinateur

- efface l'écran ;
- vérifie que le mot contient bien 7 lettres (ni plus, ni moins, pas de chiffres, pas de signe, pas d'espace) ;
- prend les 7 lettres de ce mot pour les mélanger de la façon la plus abominable ;
- et affiche ce mot ainsi caché.

Dans le cas présent, l'ordinateur affichera (par exemple) : EGINORS.

A ce moment, c'est au joueur n° 1 de chercher quel est le mot caché. S'il trouve le mot caché (SOIGNER) ou un autre mot de 7 lettres (par exemple : GERIONS ou IGNORES ou REGIONS), l'ordinateur lui compte 7 points (+ une prime de 3 points pour avoir utilisé toutes les lettres).

S'il ne trouve qu'un mot de 5 lettres, l'ordinateur ne lui compte que 5 points.

Nous vous demandons de programmer deux variantes :

- Variante 1 : le mot caché est présenté par l'ordinateur simplement en classant les lettres dans l'ordre alphabétique (c'est le cas de EGINORS).
- Variante 2 : le mot caché est obtenu en mélangeant les lettres à l'aide de la fonction aléatoire RND.

Essayez de rédiger un programme «professionnel»

- bien présenté pour l'utilisateur (mais n'allez pas trop loin : inutile d'expliquer au départ la règle du jeu) ;
- bien présenté pour le lecteur des programmes (avec des REM là où il faut).

Si vous utilisez ce jeu côté arbitre, voici une série de mots de 7 lettres comportant chacun au moins trois solutions complètes (de 7 lettres) :

- CANARDS	- CARNAGE	- RELACHA	- ECLAIRA	- ASPIRER
- ATTACHE	- CAMEL	- CARESSA	- CHATAIN	- AJUSTER
- CANARIS	- SALADES	- LAINAGE	- MANGERA	- ROBUSTE
- SALAIRE	- AMIANTE	- AMATEUR	- PARAPET	- RENONCE
- SATRAPE	- ANGLAIS	- ARTISAN	- ABORDER	- ROTONDE
- LIBERAL	- SABLIER	- ETABLIS	- ISOBARE	- SURGELE
- SALUBRE	- ESCADRE	- RADOUCI	- GERANCE	- INTERET
- HARCELE	- ETANCHE	- RELANCE	- CASERNE	- SENTEUR
- ANCETRE	- ESPACER	- GLACIER	- CHANTER	- VELOURS
- LANCIER	- RACISME	- RACINES	- SAUCIER	- PIASTRE
- RACONTE	- CARNETS	- CASSURE	- CAISSON	- ASSUMER
- RETARDE	- GARDIEN	- VEINARD	- DRAPIER	- DISCRET
- ENGAGER	- GALERIE	- MESANGE	- ETRANGE	- CUISSON
- ASPERGE	- MANIERE	- SATIETE	- ALERTER	- DOMINER
- SULFATE	- FAUSSER	- SIGNALE	- SAIGNER	- TREILLE
- GRANITE	- VITRAGE	- MOLLAIRE	- PLAINT	- REPRISE
- VARIOLE	- SPIRALE	- NOTAIRE	- FERTILE	- SERVEUR
- RETRAIT	- ARTISTE	- REVENTE	- MEUNIER	- TONSURE
- PASTEUR	- BOULIER	- ENLISER	- MESURER	- ENFOUR
- RECITER	- CIREUSE	- SOULEVE	- INSULTE	- TREPIED
- DINETTE				

Ceux de nos élèves qui sont en avance sur notre cours pourront inclure cette liste dans des DATA, afin que l'ordinateur choisisse lui-même le mot à cacher. Ce n'est pas ce que nous vous demandons aujourd'hui.

**** 3.16.21. L'équation du 3^e degré - Exercices de récapitulation R17 et R18**

A. But du programme

Le programme que nous vous proposons de rédiger (et de nous envoyer), a pour but immédiat de faire résoudre par votre ordinateur toutes les équations du 3^e degré que vous aurez (certainement !) à résoudre dans votre vie quotidienne.

La résolution de ce programme vous exercera :

- à manipuler les fonctions mathématiques (voir LED-MICRO n° 8) ;
- à utiliser la sélection multiple.

D'autre part, ce petit programme pourra être réutilisé (plus tard !) lorsque vous entrerez dans le détail des logiciels utilisés en C.A.O.

L'étude de l'équation réduite à sa forme canonique ($Y^3 + PY + Q = 0$) vous entraînera à jongler avec les fonctions mathématiques du BASIC en déjouant certains pièges pratiques tels que :

- Votre BASIC ne connaît peut-être pas les arcs sinus
- Votre BASIC ne sait pas prendre la racine cubique d'un nombre négatif

Vous trouverez dans les pages suivantes une analyse du problème et le programme BASIC que nous avons rédigé pour vous.

Soyez fair-play : cherchez par vous-mêmes avant de lire notre solution.

Attention : le programme que nous vous proposons est un simple programme d'étude : si vous lui demandez de calculer une valeur avec un $P = 0$, il vous insultera d'un «division by zero» à la ligne 120.

B. Exercice de récapitulation R17

Pour transformer le programme d'étude que nous vous proposons en un programme «professionnel», il faudra :

- 1°) Ne faire d'impasse sur aucune hypothèse, donc utiliser un organigramme plus complet (comme celui que nous vous avons rédigé à la fin du § 3.16.14 pour l'équation du second degré)
- 2°) Partir de l'équation générale du 3^e degré (avec un terme du second degré et un coefficient quelconque pour le terme du troisième degré).

Pour me faciliter le travail, je demande aux lecteurs qui voudront bien m'envoyer leur solution, de l'envoyer sous forme de listing (ce qui élimine les lecteurs n'ayant pas d'ordinateur sous la main. Excusez-moi pour cette fois). Je les prie également d'utiliser les notations suivantes :

1°) Coefficients

A0, A1, A2, A3 et A4

2°) Différents cas

- Cas 0 : problème impossible (aucune racine ni réelle ni imaginative)
- Cas 1 : aucune racine réelle
- Cas 2 : une seule racine réelle simple
- Cas 3 : une seule racine réelle double
- Cas 4 : deux racines réelles distinctes
- Cas 5 : une racine réelle simple et une racine double
- Cas 6 : trois racines réelles distinctes
- Cas 7 : problème indéterminé (nombre infini de solutions)

C. Exercice de récapitulation R18

Si vous êtes un peu en avance (sur notre cours de programmation) rédigez un programme qui obtient le même résultat mais en travaillant par approximations successives et comparez le résultat des deux méthodes (temps de calcul).

Vous trouverez un exposé complet (et bien rédigé) de résolution par la méthode de Newton, dans le livre «programmes en BASIC pour scientifiques et ingénieurs», auteur : Alan R. Miller, éditeur Sybex.

Mais attention ! pour que la comparaison soit honnête et (surtout) pour que «votre» programme soit utilisable dans les futures applications à la C.A.O. (programme de travail : horizon 1986 !) il faut que vous trouviez toutes les racines des équations !

Rappels théoriques sur l'équation du troisième degré

Résolution de l'équation

$$A_0X^3 + A_1X^2 + A_2X + A_3 = 0 \quad [A]$$

1. Elimination des cas de dégénérescence

- si $A_0 = 0$ [A] se réduit à une équation du second degré ;
- si $A_3 = 0$ [A] comporte la solution $X = 0$ et les éventuelles racines d'une équation du second degré.
- si $A_0 = 0$ et $A_1 = 0$ [A] se réduit à une équation du premier degré
- si $A_0 = 0$ et $A_1 = 0$ et $A_2 = 0$ et $A_3 = 0$ [A] a une infinité de solutions
- si $A_0 = 0$, $A_1 = 0$, $A_2 = 0$ et $A_3 \neq 0$ [A] n'a pas de racine.

A partir de maintenant, nous supposons que nous avons à faire à une «vraie» équation du troisième degré, c'est-à-dire que $A_0 \neq 0$.

2. Réduction de l'équation à sa forme canonique

On va commencer par effectuer le changement de variable :

$$X = Y - \frac{A_1}{3A_0} \quad [B]$$

afin d'obtenir une équation du troisième degré en Y sans terme du deuxième degré, c'est-à-dire une équation de la forme

$$Y^3 + PY + Q = 0 \quad [C]$$

(dont nous verrons la solution plus loin).

$$A_0 \left[Y - \frac{A_1}{3A_0} \right]^3 + A_1 \left[Y - \frac{A_1}{3A_0} \right]^2 + A_2 \left[Y - \frac{A_1}{3A_0} \right] + A_3 = 0$$

Ainsi [B] devient, en ordonnant par rapport à Y :

$$A_0Y^3 + \left[\frac{-A_1^2 + 3A_0A_2}{3A_0} \right] Y + \left[\frac{2A_1^3 - 9A_0A_1A_2 + 27A_0^2A_3}{27A_0^2} \right] = 0$$

En posant :

$$P = \frac{-A_1^2 + 3A_0A_2}{3A_0^2} \quad [D]$$

$$Q = \frac{2A_1^3 - 9A_0A_1A_2 + 27A_0^2A_3}{27A_0^3} \quad [E]$$

la résolution de l'équation [A] en X se ramène à la résolution de l'équation [C] en Y.

3. Distinction des différents cas

Considérons dans le plan [YOZ] la courbe d'équation :

$$Z = Y^3 + PY \quad [F]$$

Les racines de l'équation [C] sont les abscisses des points communs à la droite d'équation :

$$Z = -Q \quad [F]$$

et à cette courbe.

1^{er} cas : $P \geq 0$:

Z étant la somme de deux fonctions continues et croissantes est elle-même continue et croissante.

Lorsque Y varie de $-\infty$ à $+\infty$, le Z de [F] varie également de $-\infty$ à $+\infty$. Z prendra donc une fois et une seule la valeur $-P$ et l'équation [C] admet toujours une racine réelle et une seule.

2^e cas : $P < 0$:

Construisons la courbe [E]. Pour ce faire, on calcule la dérivée :

$$Z' = 3Y^2 + P$$

qui s'annule pour deux valeurs de Y

$$Y_1 = -\text{SQR}(-P/3)$$

$$Y_2 = +\text{SQR}(-P/3)$$

On calcule facilement les valeurs Z1 et Z2 de Z correspondant à ces valeurs Y1 et Y2 de Y :

$$Z_1 = -(2 \star P/3) \star \text{SQR}(-P/3)$$

$$Z_2 = +(2 \star P/3) \star \text{SQR}(-P/3)$$

Ce qui permet de dresser le tableau de variation :

Y	$-\infty$	Y_1	Y_2	$+\infty$
Z'	$+\infty$	+	0	-
Z	$-\infty$	↑	Z1	↓
			Z2	↑
				$+\infty$

L'équation [C] a alors trois racines réelles et distinctes si $-Q$ est compris entre le maximum Z1 et le minimum Z2, c'est-à-dire si on a :

$$Q^2 < -4 \star P^3/27$$

qui peut s'écrire :

$$4P^3 + 27Q^2 < 0$$

En résumé, l'équation [C] a

1) si $P \geq 0$

une seule racine réelle

2) si $P < 0$

- si $4P^3 + 27Q^2 < 0$ 3 racines réelles
- si $4P^3 + 27Q^2 = 0$ 2 racines réelles dont une double
- si $4P^3 + 27Q^2 > 0$ 1 seule racine réelle.

4. Résolution dans le cas où $P \geq 0$

Nous venons de voir que cette équation admet une racine réelle Y_1 et une seule. Nous allons chercher une expression de Y_1 sous la forme :

$$Y_1 = K (\text{tg} A - \text{cotg} A) \quad [H]$$

Trouver la racine Y_1 se réduira à trouver une valeur de K et une valeur de A.

Commençons par élever [H] au cube :

$$Y^3 = K^3 (\text{tg}^3 A - 3\text{tg}^2 A \text{cotg} A + 3\text{tg} A \text{cotg}^2 A - \text{cotg}^3 A)$$

$$Y^3 = K^3 (\text{tg}^3 A - \text{cotg}^3 A) - 3 K^3 [\text{tg} A - \text{cotg} A]$$

$$Y^3 = K^3 (\text{tg}^3 A - \text{cotg}^3 A) - 3 K^2 Y$$

soit

$$Y^3 + 3K^2 Y - K^3 (\text{tg}^3 A - \text{cotg}^3 A) = 0 \quad [I]$$

En identifiant [J] et [C], on obtient :

$$3K^2 = P \quad [J]$$

$$-K^3 (\text{tg}^3 A - \text{cotg}^3 A) = Q \quad [K]$$

[J] fournit immédiatement la valeur de K

$$K = \sqrt[3]{\frac{P}{3}}$$

K étant connu, [K] fournit la valeur de A par résolution de l'équation trigonométrique :

$$\text{tg}^3 A - \text{cotg}^3 A = -Q/K^3 \quad [M]$$

Pour ce faire, nous allons tout d'abord chercher une expression plus simple de

$$E = \text{tg}^3 A - \text{cotg}^3 A \quad [N]$$

en posant

$$\text{tg} B = \text{tg}^3 A \quad [O]$$

[N] s'écrit alors :

$$E = \text{tg} B - \text{cotg} B \quad [P]$$

Comme :

$$\text{tg} 2B = \frac{2\text{tg} B}{1 - \text{tg}^2 B} = \frac{2}{\frac{1}{\text{tg} B} - \text{tg} B} = \frac{2}{\text{cotg} B - \text{tg} B}$$

[P] peut s'écrire :

$$E = \text{tg} B - \text{cotg} B = -\frac{2}{\text{tg} 2B} \quad [Q]$$

D'où la suite des calculs :

1. On calcule K grâce à [M]

$$K = \sqrt[3]{\frac{P}{3}}$$

2. On calcule la valeur de $\text{tg}^3 A - \text{cotg}^3 A$ tirée de [M] et [N]

$$E = \text{tg}^3 A - \text{cotg}^3 A = -\frac{Q}{K^3}$$

3. D'où la valeur de B tirée de [Q] :

$$\text{tg} 2B = -\frac{2}{E}$$

d'où :

$$2B = \text{arctg} \left(-\frac{2}{E} \right)$$

soit :

$$B = \frac{1}{2} \text{arctg} \left(-\frac{2}{E} \right)$$

4. D'où la valeur de A tirée de [O]

$$\text{tg} A = \sqrt[3]{\text{tg} B} = G$$

5. et finalement la valeur de la racine Y_1 d'après [I]

$$Y_1 = K \left[G - \frac{1}{G} \right]$$

Une petite variante pour éviter d'élever un nombre négatif à la puissance 1/3 et on obtient les lignes de BASIC suivantes :

- 110 K = SQR(P/3)
- 120 E = -Q/(K*K*K)
- 130 B = 0.5 * (ATN(-2/E))
- 140 IF B >= 0 THEN G = TAN(B) * (1/3)
- 150 IF B < 0 THEN G = -(ABS(TAN(B)) * (1/3))
- 160 Y1 = K * (G - 1/G)

Vérification

$$Y^3 + 3Y - 4 = 0$$

c'est-à-dire : P = 3 et Q = -4

On obtient successivement :

$$K = 1 \quad E = 4 \quad B = -0.2318238 \quad G = -0.618034$$

d'où $Y_1 = 1$

5. Résolution dans le cas où $P < 0$ et $4P^3 + 27Q^2 < 0$

Nous savons que dans ce cas, l'équation [C] admet trois racines Y_1, Y_2 et Y_3 . Nous allons d'abord rechercher une de ces racines Y_1 que nous supposons être de la forme

$$Y_1 = K \sin A \quad [R]$$

On se rappelle la formule de Moivre :

$$(\cos A + i \sin A)^n = \cos nA + i \sin nA$$

Pour $n = 3$

$$\cos^3 A + 3\cos^2 A \sin A i - 3\cos A \sin^2 A - \sin^3 A = \cos^3 A + i \sin^3 A$$

donc

$$(\cos^3 A - 3\cos A \sin^2 A) + (3\cos^2 A \sin A - \sin^3 A) i = \cos^3 A + i \sin^3 A$$

L'égalité des parties imaginaires donne :

$$3\cos^2 A \sin A - \sin^3 A = \sin^3 A$$

ou : $3(1 - \sin^2 A) \sin A - \sin^3 A = \sin^3 A$

ou : $3\sin A - 4\sin^3 A = \sin^3 A$

d'où : $\sin^3 A = \frac{1}{4} [3\sin A - \sin^3 A]$

donc en élevant les deux termes de [R] au cube

$$Y_1^3 = K^3 \sin^3 A = K^3 \frac{1}{4} [3\sin A - \sin^3 A]$$

soit :

$$Y_1^3 - K^2 \times \frac{3}{4} \times K \sin A + \frac{K^3}{4} \sin^3 A = 0$$

ou

$$Y_1^3 - \frac{3}{4} K^2 \times Y + \frac{K^3}{4} \sin^3 A = 0 \quad [T]$$

En identifiant [T] à [C], on obtient le système d'équations qui permet de calculer K et A en fonction de P et Q :

$$-\frac{3}{4} K^2 = P \quad [U]$$

$$+\frac{K^3}{4} \sin^3 A = Q \quad [V]$$

De [U] on tire :

$$K = 2 \sqrt{\frac{-P}{3}} \quad [W]$$

En portant cette valeur dans [V]

$$\sin^3 A = \frac{4Q}{K^3} = \left| \frac{Q}{2} \times \sqrt{\frac{27}{-P^3}} \right| \quad [X]$$

(qui est < 1 car $4P^3 + 27Q^2 < 0$)

Pour garder des notations ressemblant à celles du cas où $P > 0$, on posera

$$E = \sin^3 A$$

d'où :

$$3A = \arcsin(E)$$

et $A = \frac{1}{3} \arcsin(E)$

La plupart des BASICs ne connaissent pas l'arc sinus, mais seulement l'arc tangente. D'où le petit calcul intermédiaire :

si $E = \sin x$

$$T = \operatorname{tg} x = \frac{\sin x}{\cos x} = \frac{\sin x}{\sqrt{1 - \sin^2 x}} = \frac{E}{\sqrt{1 - E^2}}$$

d'où : $\arcsin(E) = \arctan\left(\frac{E}{\sqrt{1 - E^2}}\right)$

$$A = \frac{1}{3} \arctan\left(\frac{E}{\sqrt{1 - E^2}}\right)$$

d'où la suite des calculs permettant d'obtenir Y_1

1°) $K = 2 \sqrt{\frac{-P}{3}}$

2°) $E = \frac{4Q}{K^3}$

3°) $A = \frac{1}{3} \operatorname{arctg}\left(\frac{E}{\sqrt{1 - E^2}}\right)$

4°) $G = \sin A$

5°) $Y_1 = K \times G$

On peut obtenir les deux autres racines (Y_2 et Y_3) en considérant les trois valeurs possibles pour A :

$$(A_1 = A); (A_2 = A + \frac{2\pi}{3}); (A_3 = A + \frac{4\pi}{3})$$

Nous pouvons raisonner autrement :

Connaissant une racine Y_1 , on peut diviser l'équation de départ [C] par $Y - Y_1$ de façon à obtenir une équation du second degré dont la résolution fournira Y_2 et Y_3 :

La division des polynômes fournit ainsi :

$$(Y^3 + PY + Q) = (Y - Y_1)(Y^2 + Y_1 Y + Y_1^2 + P)$$

Le binôme $Y^2 + Y_1 Y + (Y_1^2 + P)$ a pour déterminant :

$$\Delta = Y_1^2 - 4(Y_1^2 + P) = -3Y_1^2 - 4P$$

ou comme $Y_1 = K \times G = K \sin A = 2 \sqrt{\frac{-P}{3}} \times \sin A$

$$\Delta = -\frac{3 \times 4}{3} (-P \sin^2 A) - 4P = -4P [1 - \sin^2 A]$$

$$= -4P \cos^2 A \text{ (qui est toujours } > 0 \text{ car } P < 0)$$

d'où les racines Y_2 et Y_3 :

$$Y_2 = \frac{-Y_1 + 2 \cos A \sqrt{-P}}{2}$$

$$Y_3 = \frac{-Y_1 - 2 \cos A \sqrt{-P}}{2}$$

Finalement, ce calcul se traduit en BASIC par :

```
190 K = 2 * SQR(-P/3)
200 E = 4 * Q / (K * K * K)
210 A = ATN(E / SQR(1 - E * E)) / 3
220 G = SIN A
230 H = SQR(1 - G * G)
240 J = 2 * H * SQR(-P)
250 Y1 = K * G
260 Y2 = (-Y1 + J) / 2
270 Y3 = (-Y1 - J) / 2
```

Vérifications

$$Y^3 - 7Y + 6 = 0$$

c'est-à-dire :

$$P = -7 \text{ et } Q = 6$$

On obtient successivement :

$$K = 3.05505 \quad E = 0.8416976 \quad A = 0.3334732$$

$$G = 0.3273269 \quad H = 0.9449111 \quad J = 5$$

$$Y_1 = 1 \quad Y_2 = 2 \quad Y_3 = -3$$

6. Résolution dans le cas $P < 0$ et $4P^3 + 27Q^2 > 0$

Ce cas ne nous donnera qu'une seule solution réelle. Le changement de variable est pratiquement le même que celui du paragraphe 4 :

$$Y = K (\operatorname{tg} A + \operatorname{cotg} A)$$

On aboutit ainsi à l'équation :

$$Y^3 - 3K^2 Y - K^3 (\operatorname{tg}^3 A + \operatorname{cotg}^3 A) = 0$$

En identifiant avec [C], on obtient le système :

$$P = -3K^2$$

$$Q = -K^3 (\operatorname{tg}^3 A + \operatorname{cotg}^3 A)$$

dont la résolution donne :

$$K = \sqrt{\frac{-P}{3}}$$

$$Q = \frac{-Q}{K^3}$$

Le même deuxième changement de variable :

$$\operatorname{tg} B = \operatorname{tg}^3 A$$

$$E = \operatorname{tg}^3 A + \operatorname{cotg}^3 A$$

fournit

$$E = \operatorname{tg} B + \operatorname{cotg} B$$

Le troisième changement de variable est un peu différent :

On pose :

$$\sin 2B = \frac{2}{E} = -2 \frac{K^3}{Q} = -\frac{2}{Q} \sqrt{\frac{-P^3}{27}}$$

On a le droit d'effectuer ce changement de variables dans notre hypothèse car :

$$\left[-\frac{2}{Q} \sqrt{\frac{-P^3}{27}} \right]^2 = \frac{4}{Q^2} \left(\frac{-P^3}{27} \right) < 1$$

et $P < 0$

Comme notre BASIC ne connaît pas l'arc sinus, il faudra passer par l'intermédiaire de l'arc tangente, de la même façon qu'au § 5, c'est-à-dire :

remplacer : $\arcsin x$ par : $\arctan\left(\frac{x}{\sqrt{1-x^2}}\right)$

d'où la suite de calculs (que nous rédigeons tout de suite en BASIC), en laissant nombre de parenthèses inutiles afin de faciliter la lecture :

```
300 K = SQR(-P/3)
310 E = -Q / (K * K * K)
320 F = SQR(1 - (2 / E) * (2/E))
330 B = 0.5 * ATN((2/E) / F)
340 IF B >= 0 THEN G = (TAN(B)) ^ (1/3)
350 IF B < 0 THEN G = - (ABS(TAN(B))) ^ (1/3)
360 Y1 = K * (G + 1/G)
```

Le programme d'étude...

```
10 REM PROGRAMME EQUAT50 - SUR CP-OK05-PAP
20 REM EQUATION DU 3 DEGRE
30 REM
40 INPUT "P = ";P
50 INPUT "Q = ";Q
60 A = 4*P*P*P + 27*Q*Q
70 IF P=> 0 THEN Z = 1 : PRINT "CAS 1" : GOTO 100
80 IF P<0 AND A <= 0 THEN Z = 2 : PRINT "CAS 2 " : GOTO 180
90 IF P<0 AND A>0 THEN Z= 3 : PRINT "CAS 3 " : GOTO 290
100 REM ----- CAS 1
110 K = SQR(P/3) : PRINT " K = ";K
120 E = -Q/(K*K*K) : PRINT "E = "; E
130 B = .5*ATN(-2/E) : PRINT "B = "; B
140 IF B>=0 THEN G = (TAN(B))^(1/3) : PRINT "G = "; G
150 IF B< 0 THEN G = -(ABS(TAN(B)))^(1/3) : PRINT "G = "; G
160 Y1 = K*(G-1/G) : PRINT "Y1 = "; Y1
170 GOTO 370
180 REM ----- CAS 2 : P<0 ET A <= 0
190 K = 2*SQR(-P/3) : PRINT "K = "; K
200 E = (4*Q)/(K*K*K) : PRINT "E = "; E
210 A = ATN(E/SQR(1-E*E))/3 : PRINT "A = "; A
220 G = SIN(A) : PRINT "G = "; G
230 H = SQR(1-G*G) : PRINT "H = "; H
240 J = 2*H*SQR(-P) : PRINT "J = "; J
250 Y1 = K*G : PRINT "Y1 = ";Y1
260 Y2 =(-Y1+J)/2 : PRINT "Y2 = "; Y2
270 Y3 =(-Y1-J)/2 : PRINT "Y3 = "; Y3
280 GOTO 370
290 REM ----- CAS 3 : P<0 ET A >0
300 K = SQR(-P/3) : PRINT "K= ";K
310 E = -Q/(K*K*K) : PRINT "E= ";E
320 F = SQR(1 - (2/E)*(2/E)) : PRINT "F= "; F
330 B = .5*ATN((2/E)/F) : PRINT "B= "; B
340 IF B >=0 THEN G =(TAN(B))^(1/3) : PRINT "G = "; G
350 IF B < 0 THEN G =-(ABS(TAN(B))^(1/3)) : PRINT "G = "; G
360 Y1 = K*(G + 1/G) : PRINT "Y1 = ";Y1
370 END
```

... Et quelques exécutions :

```
RUN
P = ? 3
Q = ? -4
CAS 1
K = 1
E = 4
B = -.2318238
G = -.618034
Y1 = 1
OK
```

```
RUN
P = ? -7
Q = ? 6
CAS 2
K = 3.05505
E = .8416976
A = .3334732
G = .3273269
H = .9449111
J = 5
Y1 = 1
Y2 = 2
Y3 = -3
OK
```

```
RUN
P = ? -1
Q = ? 6
CAS 3
K= .5773503
E= -31.17692
F= .9979402
B= -3.209705E-02
G = -.3178372
Y1 = -2
OK
```

COURS PRATIQUE DE MICROPROCESSEUR AVEC LE MICROPROFESSOR MPF-IB

HUITIEME PARTIE

Les modes d'adressage

SOMMAIRE

I. INTRODUCTION

II. MODE D'ADRESSAGE

- II. 1. Adressage immédiat
- II. 2. Adressage étendu immédiat
- II. 3. Adressage étendu
- II. 4. Adressage par registre
- II. 5. Adressage indirect par registre
- II. 6. Adressage indexé
- II. 7. Adressage implicite
- II. 8. Adressage relatif
- II. 9. Adressage modifié en page zéro
- II.10. Exercices

III. INSTRUCTIONS DE SAUT, D'APPEL ET DE RETOUR

- III. 1. Introduction
- III. 2. «Instructions de saut»
- III. 3. Sauts conditionnels
- III. 4. Sauts en adressage étendu immédiat
- III. 5. Sauts en adressage relatif
- III. 6. Instructions de sauts en adressage relatif
- III. 7. Comparaison entre les sauts adressage immédiat et relatif
- III. 8. Instructions de saut en adressage par registres
- III. 9. Instruction «DJNZ»
- III.10. Exemple d'application
- III.11. Instructions d'«APPEL» et de «RETOUR»
- III.12. Instruction «RESTART»
- III.13. Exercices

I. INTRODUCTION

Avant de poursuivre l'étude proprement dite de la programmation d'un microprocesseur, nous allons présenter les divers modes d'adressage. Pour définir une instruction, il ne suffit pas de **déterminer l'opération à réaliser**, il faut bien souvent **indiquer les adresses de la donnée d'origine ainsi que celle de la destination** du résultat.

D'autre part, nous étudierons dans la prochaine leçon, les instructions de «sauts de programme». Selon **les directives du programmeur** et si certaines conditions sont remplies, on assiste à des «déroutements» de programme. Le microprocesseur **effectue un «saut» à une adresse différente du déroulement classique**. L'instruction à exécuter n'est plus la suivante, mais **au contraire elle se situe à un tout autre emplacement**. Dans ce cas, le compteur de programme est «chargé» avec une nouvelle adresse.

Nous reviendrons sur cette technique ultérieurement.

II. MODE D'ADRESSAGE

Le microprocesseur Z-80 ne comporte pas moins de 10 modes d'adressage. En réalité, nous les avons presque tous rencontrés dans la leçon précédente, mais sans y faire allusion directement.

II.1. Adressage immédiat

L'adresse qui contient l'opérande est

celle qui suit **«immédiatement»** le code opération.

Exemple :

Additionner au contenu de l'accumulateur la quantité «3A».

PC (adresse du programme)	Contenu
1800	C6
→ 1801	3A
1802	

L'adresse de l'instruction C6

(ADD A ← A + n)

est 1800, celle de l'opérande «n» est 1801 : elle **suit immédiatement celle de l'instruction**.

II.2. Adressage étendu immédiat

Dans ce cas, l'opérande n'est plus constituée du seul octet qui **suit l'instruction**, mais des **«deux octets» consécutifs**.

Exemple 1

Charger la paire de registres HL avec 04 FE.

PC	Contenu
1800	21
← 1801	FE (adresse de l'octet de poids faible)
← 1802	04 (adresse de l'octet de poids fort)

A noter que l'adresse (1801 dans l'exemple) qui suit immédiatement celle de l'instruction est celle de l'octet de poids faible de l'opérande et la suivante (1802) celle de l'octet de poids fort.

Exemple 2

Charger le registre d'index IX avec 30 00.

PC	Contenu	
1800	FD	
1801	21	
← 1802	00	(adresse de l'octet de poids faible)
← 1803	30	(adresse de l'octet de poids fort)

Dans cet exemple, l'instruction
LD IY ← 30 00

est constituée de 2 bytes : FD et 21. Dans ce cas les adresses de l'opérande sont 1802 et 1803.

III.3. Adressage étendu

Dans ce mode d'adressage, les deux octets consécutifs qui suivent immédiatement le code opératoire (1 ou 2 bytes) constituent **une adresse de 16 bits**. (Le poids faible étant le premier octet, le poids fort est le suivant). Deux cas se présentent :

1) Quand il s'agit **d'une instruction de saut**, les 2 octets représentent **l'adresse à partir de laquelle le programme se poursuit**.

2) Quand il s'agit **d'une donnée**, ils représentent l'adresse de **l'emplacement mémoire qui contient l'opérande** (1 octet).

Il ne faut pas confondre l'adressage étendu avec l'adressage étendu immédiat.

Exemple 1

Charger le contenu de l'accumulateur dans la case d'adresse 2300 H.

PC	Contenu	
1800	32	
→ 1801	00	(octet de poids faible de l'adresse)
→ 1802	23	(octet de poids fort de l'adresse)

ce qui s'écrit :

LD (2300 H) ← A

Exemple 2

Effectuer un saut à 1900 H.

PC	Contenu	
1800	C3	
→ 1801	00	(octet de poids faible de l'adresse)
→ 1802	19	(octet de poids fort de l'adresse)

Dans ce cas l'instruction qui sera exécutée après 1802 ne sera pas celle contenue dans 1803 **mais celle**

contenue dans 1900 H ce qui s'écrit :

JP 1900 H

On notera dans ce dernier cas **l'absence de parenthèses**.

II.4. Adressage par registre

Dans ce mode d'adressage, l'instruction contient non seulement le code opération mais aussi **le registre concerné** qui peut être aussi bien **la source de l'opérande que la destination du résultat de l'opération**.

Exemple 1

Effectuer un OU logique entre le contenu de l'accumulateur et le contenu du registre E.

Ce qui s'écrit :

B3 ou A ← A + E (+ : ET logique)

Exemple 2

Décrémenter le contenu du registre L.

Ce qui s'écrit :

2 D ou L ← L - 1

Exemple 3

Charger le registre C avec le contenu du registre E.

Ce qui s'écrit :

4 B ou E ← C

II.5. Adressage indirect par registre

Dans ce mode d'adressage, l'adresse de l'opérande est le **contenu d'une paire de registres spécifiée** (16 bits). Ce qui revient à dire que la paire de registres (BC, DE ou HL) se comporte comme un «pointeur» d'adresse de la mémoire.

Exemple :

Comparer le contenu de l'accumulateur A avec le contenu de la case mémoire pointée par (HL).

Ce qui s'écrit :

BE ou CP A, (HL)

On notera la **présence de parenthèses** pour (HL).

II.6. Adressage indexé

L'adresse de l'opérande est dans ce cas constituée, comme dans le cas précédent, **du contenu d'un registre** (ici IX ou IY) **au contenu duquel on ajoute un déplacement «d»**.

L'instruction contient une valeur arithmétique «d» qui est ajoutée au contenu du registre index spécifié durant l'exécution du programme, **pour constituer l'adresse de l'opérande**.

Exemple :

Le registre d'index IY contient 20 00 H, l'instruction

LD (IY + 10 H), FE

chargera l'emplacement mémoire 20 00 + 10 = 20 10 H avec la quantité FE.

PC	Contenu	
1800	FD	
1801	36	
1802	10	déplacement
1803	FE	

II.7. Adressage implicite

Dans ce mode d'adressage, **l'un des registres du CPU** bien souvent l'accumulateur **est automatiquement impliqué à être la destination** du résultat.

Exemple :

Retrancher 23 H du contenu de l'accumulateur s'écrit :

D 6 23 H ou A ← A - 23 H

II.8. Adressage relatif

Le mode d'adressage relatif est essentiellement utilisé dans les instructions de branchement (conditionnel ou inconditionnel).

Pour que l'adressage soit valide, une règle limitant la distance du déplacement de l'instruction de branchement à la destination doit être respectée.

Cette règle est que **l'adresse de branchement doit être comprise dans les limites suivantes :**

$$(PC + 2) - 128 \leq D \leq (PC + 2) + 127$$

On notera que **la valeur du déplacement «D» est un nombre signé, donc compris entre +127 ou -128**.

II.9. Adressage modifié en page zéro

Comme dans le mode précédent, il s'agit d'un saut de programme. Il existe dans le Z-80 huit instructions d'un seul octet qui permettent de charger le compteur de programme avec l'un des huit octets qui constitue **le byte de poids faible d'une adresse en page zéro**. C'est-à-dire que le premier octet (poids fort) est 00 tandis que le second est spécifié dans l'instruction. Ainsi quand le programme exécute une instruction RESTART (comme on les appelle), il peut effectuer en un temps très court un saut à la première adresse d'un sous-programme.

Nous reviendrons dans cette leçon sur ces deux modes d'adressage.

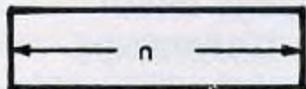
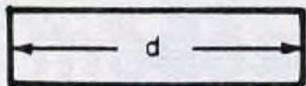
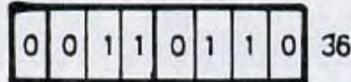
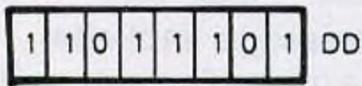
II.10. Instructions de transfert par registres INDEX

Nous pouvons maintenant étudier les instructions de chargement de 1 octet par registre Index.

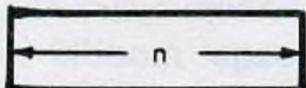
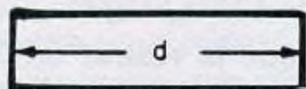
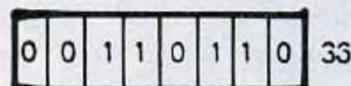
a) LD (IX + d), n ou LD (IY + d), n
Le contenu de l'opérande «n» est chargé à l'adresse mémoire constituée par le contenu du registre index IX ou IY auquel on ajoute le déplacement «d».

Les codes binaire et hexadécimal sont :

LD (IX + d) ← n



LD (IY + d) ← n



Exemple :

LD (IX + 20 H), 05

s'écrit :

DD 36 20 05

Si IX contient 840 H, après l'exécution de cette instruction, l'emplacement mémoire

840 + 20 = 860 H,

contient la donnée 05 H.

b) LD (IX + d), r ou LD (IY + d), r

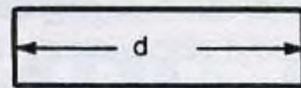
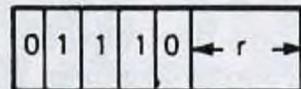
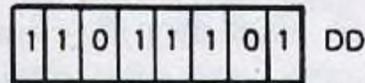
Le contenu du registre «r» spécifié dans l'instruction est chargé à l'adresse mémoire constituée par le contenu du registre index IX ou IY auquel on ajoute le déplacement «d». Les codes binaire et hexadécimal sont : (voir page suivante)

Exemple :

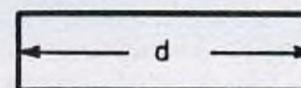
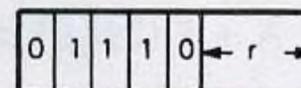
LD (IY + 10 H), D

s'écrit FD 72 10.

LD (IX + d) ← r



LD (IY + d) ← r



avec r :

A = 111 D = 010 L = 101

B = 000 E = 011

C = 001 H = 100

Si IY contient 21 00 H, après exécution de cette instruction, l'emplacement mémoire

21 00 + 10 = 21 10 H

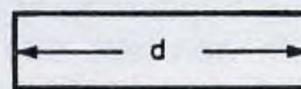
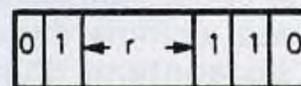
est chargé par le contenu du registre D.

c) LD r, (IX + d) ou LD r, (IY + d)

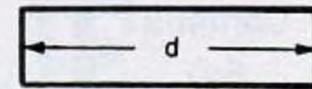
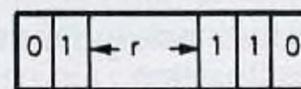
Le contenu de la mémoire d'adresse constituée par le contenu du registre index IX ou IY auquel on ajoute le déplacement «d» est transféré dans le registre «r» désigné par l'instruction.

Les codes binaire et hexadécimal sont :

LD r ← (IX + d)



LD r ← (IY + d)



Exemple : LD L ← (IX + 03), s'écrit DD 6E 03

Si IX contient 18 000 H, après exécution de cette instruction, l'emplacement mémoire

18 00 + 03 = 18 03 H

est chargé dans le registre «L».

II.10. Exercices

1. A quel mode d'adressage appartiennent les instructions suivantes :

- $A \leftarrow A + B + C$
- $A \leftarrow A - D$
- $HL \leftarrow HL + BC + C$
- $A \leftarrow A (HL)$
- $IX \leftarrow IX + SP$
- Incrém. (IX + d)
- EX (SP), HL
- Load HL, (nn)
- Load IX, nn
- Load E, (IY + d)

2. Donner le code machine des instructions suivantes :

- LD (IX + 3E), 50 d
- Charger B avec le contenu adresse par (IY + 10 H)
- Transférer le contenu de D dans (IX + 3 F)
- LD A, (IY + 7)
- Charger (IX + d) avec 95 d

III. INSTRUCTIONS DE SAUT, D'APPEL ET DE RETOUR

III.1. Introduction

En guise d'introduction de ces nouvelles instructions, nous écrivons un petit programme pour bien comprendre le problème tel qu'il se pose.

Il s'agit d'inscrire FF H dans les 16 cases mémoires de la RAM à partir de l'adresse 18 00 H à 18 0F H. L'écriture de ce programme commençant à l'adresse 18 50 H.

(Rappel : à la mise sous tension, le contenu d'une mémoire vive est a priori quelconque).

En examinant attentivement le répertoire d'instruction du Z 80, quoi que fort abondant, nous n'y trouvons pas une instruction qui puisse «charger directement une case d'adresse donnée par un octet». Une telle instruction s'écrirait :

LD (nn), n

ou charger la case d'adresse nn par la quantité «n».

Par contre, nous pouvons charger une case donnée «nn» par le contenu de l'accumulateur A :

Load (nn), A ou 32 nn

et charger A par une donnée FF :

Load A, FF ou 3 E FF

D'où l'écriture du programme «PO».

Pour obtenir le même résultat qu'avec le programme PO, nous n'avons eu à utiliser que 24 H soit 36 instructions au lieu de 50. C'est un peu mieux.

En examinant attentivement la deuxième version du programme, nous remarquons que nous avons répété 16 fois le même couple d'ins-

Adresse	Mnémonique	Codes hexadécimaux
1850	LD A, FF	3E FF
1852	LD (1800), A	3A 00 18
1855	LD (1801), A	3A 01 18
1858	LD (1802), A	3A 02 18
.....		
187C	LD (180E), A	3A 0E 18
187F	LD (180F), A	3A 0F 18
1882	Halt (FIN)	76

L'écriture de ce programme ne comporte pas moins de 50 instructions (1882 H - 1850 H = 32 H soit 50 d) pour charger 16 cases mémoires avec FF ! Heureusement qu'il ne faut pas ainsi remplir les 2 K (2048) octets de la RAM !

Dans une première étape, nous allons mettre à profit ce que nous venons d'étudier à propos de l'adressage pour réduire la longueur de notre programme.

Réécrivons le programme en utilisant l'adressage par la paire de registres HL.

Le registre A contient toujours la donnée FF, mais le transfert de la donnée s'effectue dans la case mémoire pointée par la paire de registres HL. Il suffit d'initialiser HL en chargeant ce registre double avec l'adresse de départ, et après chaque opération de transfert d'incrémenter son contenu. On obtient ainsi : Programme P1

tructions :

LD (HL), A et INC HL

d'où l'idée de chercher si dans le répertoire instructions du Z 80^R, il n'existe pas une instruction ayant pour objet : «Recommencer à exécuter la dernière ou les n dernières instructions».

Une telle instruction n'existe pas, ou tout au moins sous cette forme. Par contre, il existe tout un ensemble d'instructions qui permettent d'effectuer des «sauts» aussi bien en arrière

Programme P2

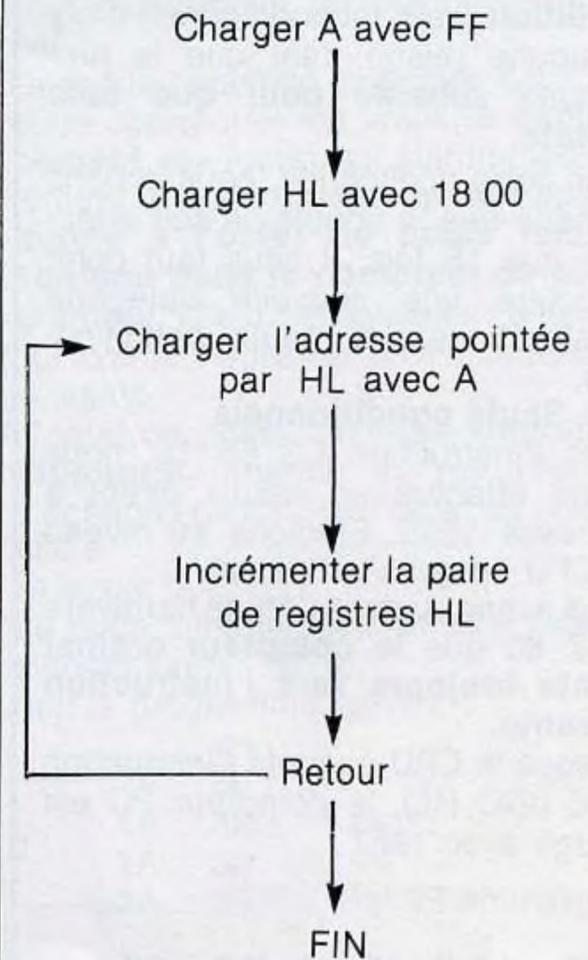
Adresse	Mnémonique	Codes hexadécimaux
1850	LD A, FF	3E FF
1852	LD HL, 1800	21 00 18
1855	LD (HL), A	77
1856	INC HL	23
1857	Retour à 1855	C3 55 18
185A	Halt (FIN)	

Adresse	Mnémonique	Codes hexadécimaux
1850	LD A, FF	3E FF
1852	LD HL, 1800	21 00 18 (HL) = 1800
1855	LD (HL), A	77
1856	INC HL	23 (HL) = 1801
1857	LD (HL), A	77
1858	INC HL	23 (HL) = 1802
.....		
1871	LD (HL), A	77
1872	INC HL	23 (HL) = 180F
1873	LD (HL), A	77
1874	Halt (FIN)	76

qu'en avant. C'est ce que nous étudierons maintenant.

III.2. «Instructions de saut»

Reprenons le programme P1 et présentons-le sous la forme d'un organigramme.



Ce qui conduit au programme suivant : P2

Nous venons ainsi d'introduire une nouvelle instruction «Saut incondi-

tionnel» à une adresse donnée. Le code opératoire est C3 suivi des deux octets d'adresse à laquelle le programme doit «sauter».

Comme d'habitude, **l'octet de poids faible suit le code opératoire et vient ensuite l'octet de poids fort.** Ainsi, d'un programme de 50 instructions puis de 36, nous obtenons maintenant un programme de... 10 instructions.

Que le lecteur ne se précipite pas sur son MPF-1B s'il en possède un, pour essayer ce dernier programme.

L'objectif ne sera-t-il pas atteint ? Bien sûr que oui, mais comment le

programme s'arrêtera-t-il? Car a priori l'instruction «Halt» ne pourra jamais être exécutée.
 En effectuant à l'adresse 1857 un retour à 1855, nous effectuons une «boucle de programmation» (ou LOOP en anglais). Etant donné que **cette instruction est réalisée sans condition** (saut inconditionnel) il n'y a aucune raison, tant que le programme subsiste **pour que cela s'arrête.**

Dans notre application, nous aurions souhaité que la boucle ne soit effectuée que 16 fois, il nous faut donc introduire une nouvelle catégorie d'instructions, les sauts conditionnels.

III.3. Sauts conditionnels

Dans l'instruction C3 55 18, nous avons effectué un «saut» direct à l'adresse 1855. Etudions au niveau du CPU ce qui s'est passé.

Nous avons vu dans l'étude hardware du Z 80 que **le compteur ordinal pointe toujours vers l'instruction suivante.**

Lorsque le CPU exécute l'instruction 1856 (INC HL), le compteur PC est chargé avec 1857.

Programme P2

```

1850 LD A, FF      3E FF
1852 LD HL, 1800  21 00 18
1855 LD (HL) A    77
1856 INC HL      23
1857 Retour à 1855 C3 55 18
185A Halt (FIN)
    
```

ou

```

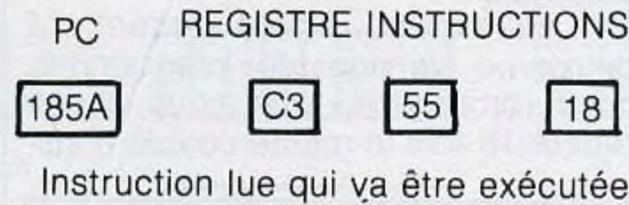
1850 3E Instruction 1
1851 FF
1852 21 Instruction 2
1853 00
1854 18
1855 77 Instruction 3
1856 23 Instruction 4
1857 C3 Instruction 5
1858 55
1859 18
185A 76 Instruction 6
    
```

Après avoir effectué +1 dans HL, le CPU lit l'instruction suivante (5) composée de 3 octets (C3, 55 et 18), puis le compteur PC s'incrémente : il contient 185A (instruction 6).

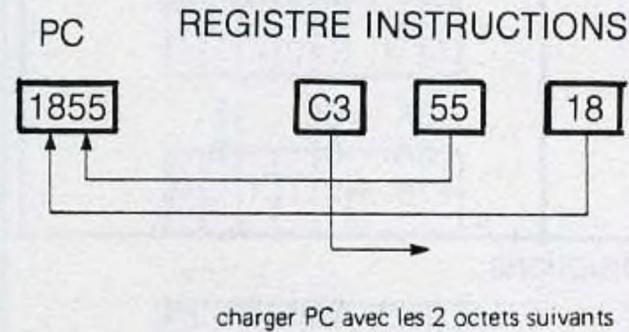
L'exécution de l'instruction (5) : (C3 55 18) charge le compteur PC avec l'adresse 1855 et par le fait même «écrase» le contenu précédent (185A) : le programme exécutera l'instruction (3) (contenue dans 1855) au lieu de l'instruction (6) (contenue dans 185A).

D'où le schéma :

Avant l'exécution de l'instruction (5) [1857]



Après l'exécution de l'instruction (5) [1857]



En n'introduisant aucune condition, le programme reviendra (tant qu'il subsiste) à 1855 après l'instruction 1857 : ce ne sera plus 16 cases mémoires qui contiendront FF, mais toute la mémoire vive ! En réalité, notre programme sera détruit par lui-même avant.

Nous allons donc modifier notre programme en utilisant un compteur. Nous utiliserons le registre B comme compteur.

Nous comptabiliserons dans B le nombre de fois que nous avons exécuté la «boucle» :

(LD (HL), A et INC HL)

Quand 16 sera atteint, nous arrêterons le programme.

En réalité, il est plus aisé de détecter le «passage à zéro», aussi nous chargerons B avec 16 que nous décrémenterons (DEC B) à chaque «passage» et nous surveillerons le passage à «0».

Ainsi, tant que le contenu de B est différent de 0, nous répétons la boucle.

Programme P3

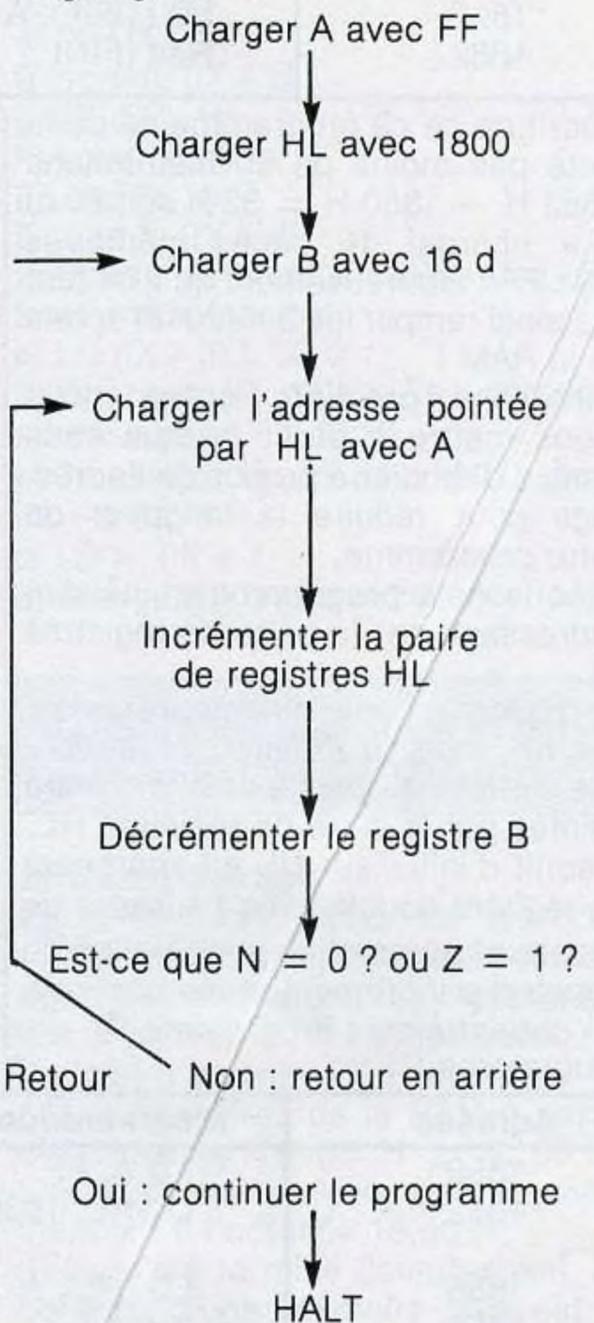
Adresse	Mnémonique	Codes hexadécimaux
1850	LD A, FF	3E FF
1852	LD HL, 1800	21 00 18
1855	LD B, 0FH	06 0F
1857	LD (HL), A	77
1858	INC HL	23
1859	DEC B	05
185A	Saut à 1857 si Z = 0	C2 57 18
185D	Halt (FIN)	76

Quand B = 0, nous effectuons l'instruction suivante, c'est-à-dire HALT (FIN).

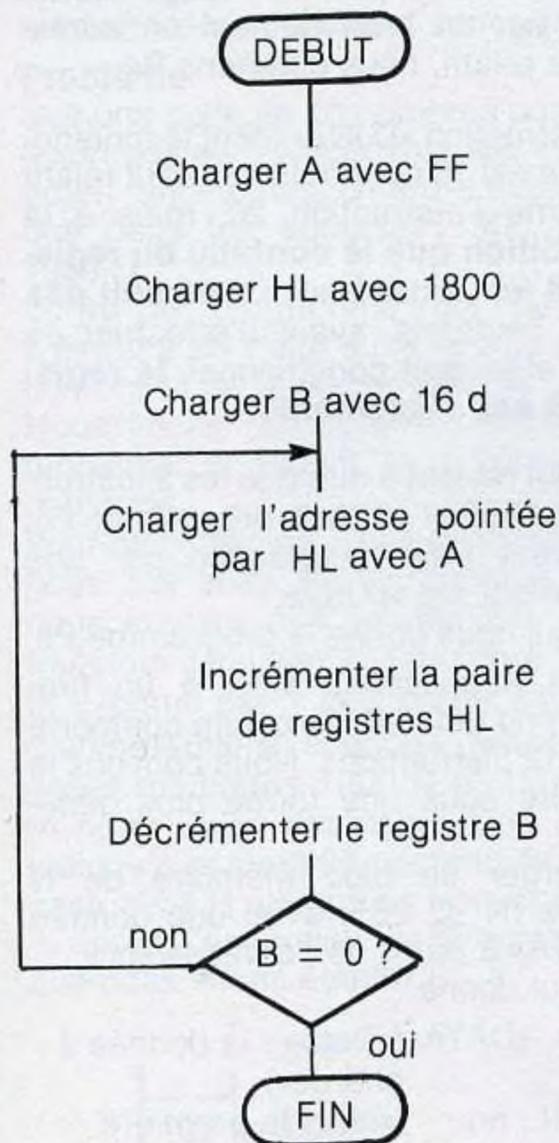
Reste à détecter le «passage à zéro» de B. Nous avons vu dans les instructions «INC» et «DEC» que les indicateurs (Registre F) étaient positionnés en fonction du résultat. Lorsqu'on effectue une décrément de B (05), l'indicateur Z sera à 1 quand le contenu de B sera nul.

En conclusion, la boucle sera effectuée tant que Z = 0 (donc B ≠ 0). L'instruction «saut à nn» si Z = 0 est C2.

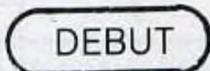
Nous obtenons ainsi un nouveau programme et un nouvel organigramme. Organigramme



Notre programme P3 comporte 14 instructions et exécute ce qui est demandé en s'arrêtant sitôt l'exécution terminée : c'est exactement ce que nous souhaitons. Nous allons présenter l'organigramme avec les éléments de programmation.

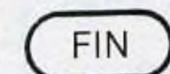


Symbole pour indiquer le début d'un programme

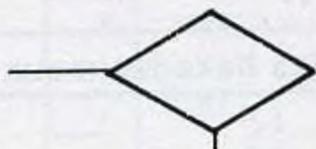


Symbole pour indiquer la fin d'un programme.

Une instruction comme Halt (76) doit terminer un programme.



Symbole de test. Il y a deux alternatives possibles et seulement deux.



Nota : Que se passe-t-il quand $Z = 1$? (ou $B = 0$).

Au moment de l'exécution de l'instruction 185A, le compteur ordinal PC contient 185D. Seulement au lieu d'être modifié avec 1857 (quand $Z =$

0), il n'y a **pas de changement de PC** : l'instruction suivante est alors 185D.

III.4. Sauts en adressage étendu immédiat

Les instructions de sauts en adressage étendu immédiat se présentent toujours sous la forme d'une instruction de 3 octets. Le premier contient la condition (qui peut être incondionnelle), le second indique le poids faible de l'adresse de branchement et le troisième représente l'octet de poids fort.

Hormis l'instruction C3 nn, saut incondionnel, la condition de branchement est l'état «1» ou «0» de l'un des indicateurs du registre F. Ce qui conduit au tableau suivant :

Code opératoire	Condition de saut
C3 nn	Saut incondionnel
DA nn	Saut à nn si C = 1
D2 nn	Saut à nn si C = 0
CA nn	Saut à nn si Z = 1
C2 nn	Saut à nn si Z = 0
EA nn	Saut à nn si parité impaire
E2 nn	Saut à nn si parité paire
FA nn	Saut à nn si signe négatif
F2 nn	Saut à nn si signe positif

A noter qu'aucun indicateur n'est affecté par ces instructions.

A noter qu'aucun indicateur n'est affecté par ces instructions.

III.5. Sauts en adressage relatif

Dans ce type de branchement, il ne s'agit plus de faire un saut à une adresse quelconque du programme, mais d'effectuer **un déplacement en avant ou en arrière par rapport à l'instruction** qui détermine le branchement.

L'instruction de saut relatif est constituée de 2 octets. Le premier indique la condition requise pour que le saut ait lieu (le «saut» peut être incondionnel). Le second indique la **valeur algébrique** du déplacement.

Comme la valeur du déplacement est un «nombre signé», si le bit 7 (poids le plus fort) est «0», le déplacement est positif. Si le bit 7 est «1», le déplacement est négatif, c'est un retour en arrière.

Ainsi, le déplacement e est tel que :

$$-128 \leq e \leq +127 \text{ (décimal)}$$

$$1111\ 1111 \leq e \leq 0111\ 1111 \text{ (en binaire)}$$

$$FF \leq e \leq 7F \text{ (en hexadécimal)}$$

Que se passe-t-il au niveau du CPU et plus exactement du compteur ordinal ?

Quand la condition (spécifiée dans le code opératoire) est vraie, le **déplacement «e»** (octet qui suit immédiatement le code opératoire) **est additionné à l'octet de poids faible contenu dans le compteur de programme**. C'est ce nouveau contenu qui pointe l'adresse de l'instruction suivante.

L'octet de poids fort reste **toujours inchangé**, même si un report s'échappe du bit 7 de l'octet de poids faible.

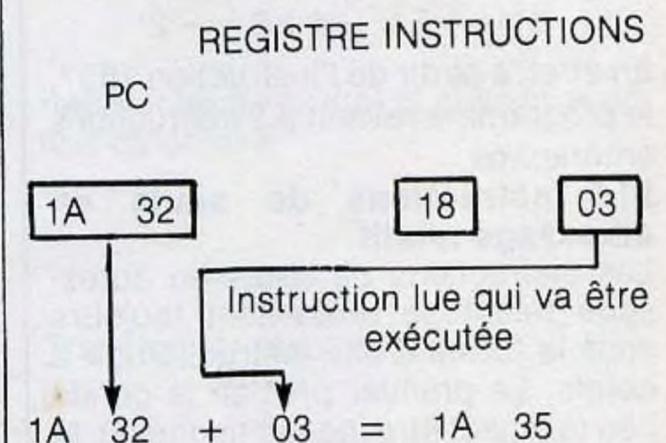
Exemple 1

Le code mnémorique du saut relatif incondionnel est 18.

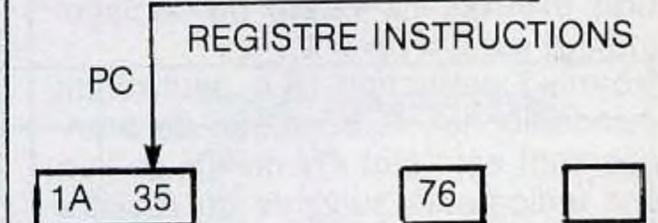
Soit le programme suivant :

1A	30	JR + 3	18 03
1A	32	—	
1A	33	—	
1A	35	Halt	76

Avant l'exécution de 1A 30



Après l'exécution de 1A 30



La quantité 03 est additionnée à l'octet de poids faible du compteur PC

La prochaine instruction qui va être exécutée est celle contenue dans l'emplacement 1A 35.

La valeur du déplacement

$$d = 1A\ 35 - 1A\ 30 = 5 \text{ ou } d = e + 2$$

Adresse	Mnémonique	Codes hexadécimaux
1850	LD A, FF	3E FF
1852	LD HL, 1800	21 00 18
1855	LD (HL), A	77
1856	INC HL	23
1857	Retour «d»	18 e
1859	Halt (FIN)	76

Exemple 2

Reprenons le programme P2 dans lequel nous remplaçons le saut inconditionnel en adressage immédiat étendu par un saut en adressage relatif.

Au moment de l'exécution de l'instruction 1857 (ici il n'y a que 2 octets), le contenu de PC est 1859. Pour que la prochaine instruction soit 1856, il faut retrancher 4 au contenu de PC, soit additionner (-4) ou en valeur hexadécimale signée FCH. En effet :

$$\begin{array}{r} 59 \text{ H} \\ \text{FC H} \\ \hline (1) \quad 54 \text{ H} \end{array}$$

Le report qui s'échappe de l'octet de poids faible (1) est perdu.

L'adresse de l'instruction suivante est 1854 H.

Le déplacement

$$d = e + 2 = -4 + 2 = -2$$

En effet, à partir de l'instruction 1857, le programme revient à 2 instructions antérieures.

III.6. Instructions de sauts en adressage relatif

Les instructions de sauts en adressage relatif se présentent toujours sous la forme d'une instruction de 2 octets. Le premier précise la condition (qui peut être inconditionnelle), le second indique la quantité «e» signée qui sera ajoutée au compteur ordinal. La valeur du déplacement d est $d = \text{«e»} + 2$.

Hormis l'instruction 18 d, saut relatif inconditionnel, la condition de branchement est l'état «1» ou «0» de l'un des indicateurs suivants du registre F : C (Carry) ou Z (Zéro).

Ce qui conduit au tableau suivant :

Code opératoire	Condition de saut
18 d	Saut inconditionnel
38 d	Saut à PC + d si C = 1
30 d	Saut à PC + d si C = 0
28 d	Saut à PC + d si Z = 1
20 d	Saut à PC + d si Z = 0

III.7. Comparaison entre les sauts en adressage immédiat et relatif

Avantages du saut en adressage relatif :

- l'instruction ne comporte que 2 octets
 - le programme est «transposable».
- Le programme précédent de l'exemple 2, qui est stocké arbitrairement à partir de l'adresse 1850, pourrait tout aussi bien être écrit à partir de 1950, sans changer quoi que ce soit à son contenu. Ce qui n'est pas le cas dans l'adressage immédiat puisque l'adresse de branchement est précisée (saut à 1857).

Inconvénients :

- le champ de branchement est limité $-129 \leq d \leq +127$
- les indicateurs P et S ne peuvent intervenir comme condition.

III.8. Instructions de saut en adressage par registres

Il existe un ensemble de trois instructions qui ne permettent d'effectuer que des sauts inconditionnels à l'adresse dans l'une des paires de registres HL, IX ou IY.

Les codes mnémoniques sont :

E9 PC ← HL : le compteur ordinal est

chargé par le contenu de la paire de registres HL

DD E9 PC ← IX : le compteur ordinal est chargé par le contenu du registre ou FD E9 PC ← IY : Index IX ou IY.

III.9. Instruction «DJNZ»

Reprenons le programme P3 que nous allons modifier en remplaçant le branchement par adressage immédiat par un branchement en adressage relatif, nous obtenons P4.

L'instruction «DJNZ» (dont le mnémonique est 10 e) effectue un saut relatif comme l'instruction 20, mais à la condition que le contenu du registre B (et uniquement lui) ne soit pas nul. Toutefois, avant d'effectuer ce test et le saut conditionnel, le registre B est décrémenté.

Ce qui revient à dire que les 2 instructions 1859 et 185A du programme P4, peuvent être remplacées par une seule qui est «DJNZ».

Ce qui nous donne le programme P5.

Nous aboutissons ainsi à un programme définitif P5, qui ne comporte que 12 instructions. Nous pouvons le décrire sous une forme plus générale :

«Charger un bloc mémoire de N cases ($N \leq 255$) avec une donnée «DATA» à partir de l'adresse «nn».

Ce qui donne :

LD A, «DATA» («Data» : la donnée à charger)

LD HL, nn («nn» : la première adresse)

Programme P4

Adresse	Mnémoniques	Codes hexadécimaux
1850	LD A, FF	3E FF
1852	LD HL, 1800	21 00 18
1855	LD B, 0F H	06 0F
1857	LD (HL), A	77
1858	INC HL	23
1859	DEC B	05
185A	JR NZ, -3	20 FB
185C	Halt (FIN)	76

Programme P5

Adresse	Mnémoniques	Codes hexadécimaux
1850	LD A, FF	3E FF
1852	LD HL, 1800	21 00 18
1855	LD B, 0F	06 0F
1857	LD (HL), A	77
1858	HL, HL + 1	23
1859	DJNZ, -2	10 FC
185B	Halt (FIN)	76

LD B, N («N» : le nombre de cases mémoires)
 LD (HL), A
 INC HL
 DJNZ, -2
 Halt (FIN)

III.10. Exemple d'application

Les connaissances ainsi acquises sont largement suffisantes pour résoudre des problèmes concrets.

Problème

Soit une suite de 10 nombres positifs (exprimés en hexadécimal sur 1 seul octet) rangés dans un ordre quelconque dans les adresses 1900 H à 1909 H.

Ecrire le programme qui permet d'extraire de cette suite le plus petit nombre et de le placer dans A.

Nous allons résoudre ensemble ce problème et établir le programme demandé.

Pour bien fixer les idées, établissons-nous une suite de 10 nombres que nous exprimerons en décimal sans toutefois dépasser 127 (nombre positif exprimé par 1 seul octet), rangée d'une manière aléatoire entre les cases mémoires 1900 H et 1909 H. (A noter qu'un examen rapide nous indique que c'est 08 contenu dans la case 1904 H qui doit se trouver dans A après l'exécution du programme que nous allons établir).

	18FF
109	1900
76	1901
89	1902
49	1903
08	1904
17	1905
120	1906
29	1907
33	1908
12	1909
	190A

Le principe adopté est de comparer le contenu du registre B avec successivement chacun des nombres N de la suite.

Si le résultat est négatif ($N > B$), le registre B contient un nombre plus petit que N, on ne change rien dans B.

Si le résultat est positif ($N < B$), N est plus petit que B et l'on place N dans B.

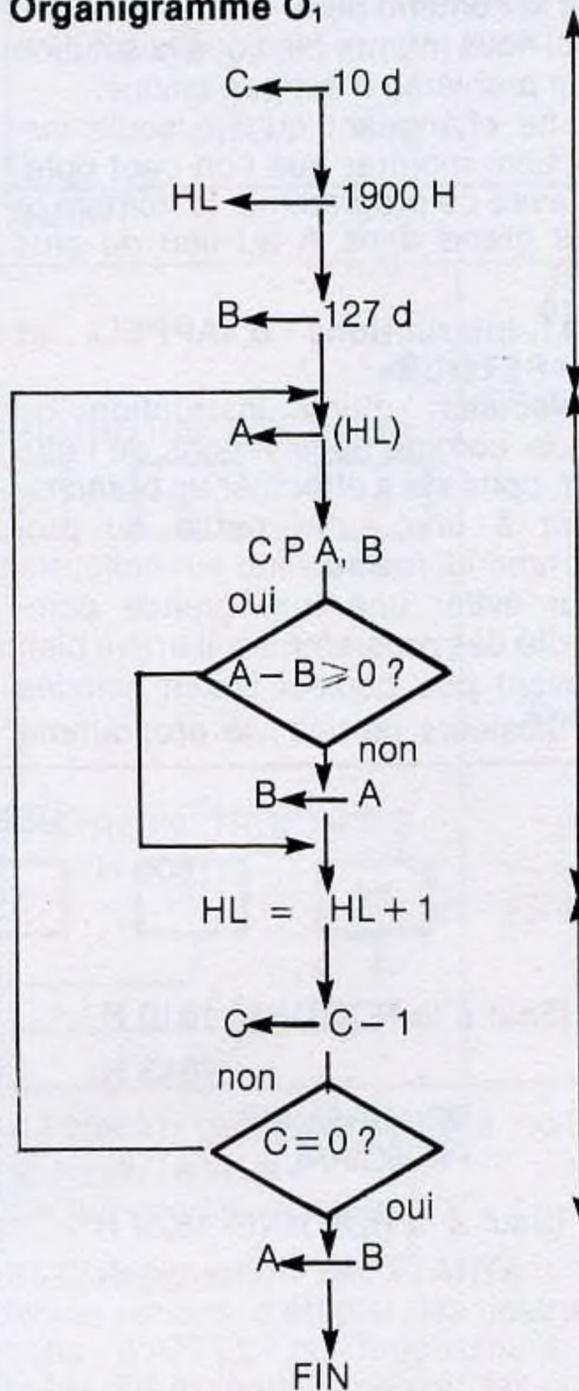
Lorsque les 10 nombres auront ainsi été examinés, il ne restera plus qu'à transférer le contenu du registre B

dans A.

Nous allons suggérer un premier organigramme basé sur ce principe. Nous ne saurions trop insister pour que le lecteur établisse lui-même le programme en «code machine» correspondant et l'exécute avec son MPF-1B.

Le programme est écrit à partir de l'adresse 1800 H.

Organigramme O₁



Initialisation :

nombre de chiffres, adresse de départ et valeur maximale

Boucle (Loop) de comparaison

si $A \geq B$, B inchangé
 si $A < B$, B chargé avec A

Recherche du chiffre N suivant avec test du dernier

Transfert du résultat dans A

Fin du programme

Ecriture du programme Pap 1 :

Adresse	Mnémoniques	Codes hexadécimaux
1800	LD C, 10 d	0E 0A
1802	LD HL, 1900 H	21 00 19
1805	LD B, 127 d	06 7F
1807	LD A, (HL)	7E
1808	CP A, B	B8
1809	JP, S	F2 0D 18
180C	LD B, A	47
180D	INC HL	23
180E	DEC C	0D
180F	JR, Z	20 F6
1811	LD A, B	78
1812	Halt (FIN)	76

Questions :

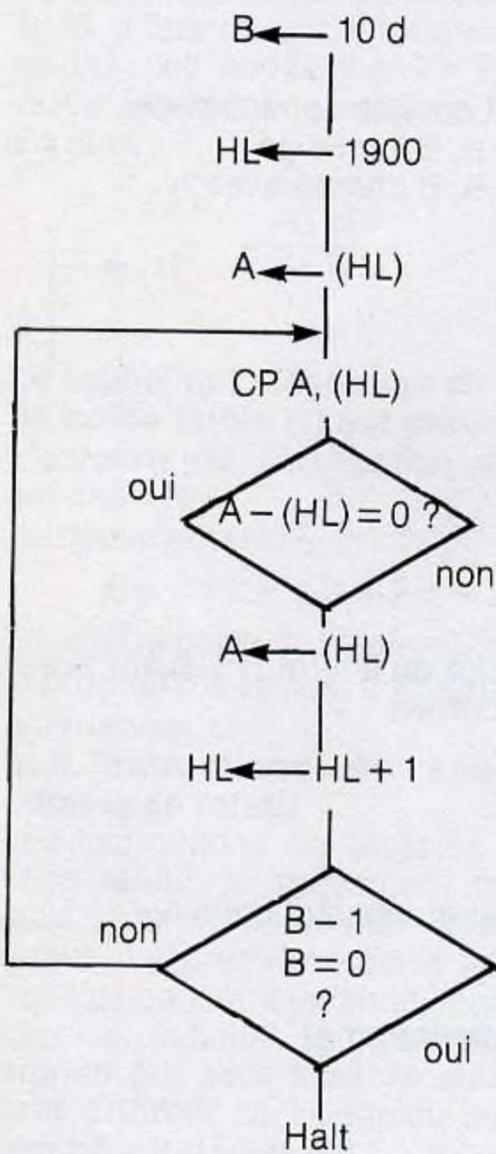
- a) Le programme ainsi écrit est-il transposable ?
- b) Quelles suggestions peut-on faire pour le simplifier ? (Notamment pour réduire le nombre d'instructions)
- c) En modifiant l'organigramme de départ, établir un autre programme (plus simple si possible)

Nota :

Si des erreurs sont apparues dans l'élaboration du programme, nous vous conseillons de vous reporter à l'étude de l'instruction correspondante.

Nous présentons ici une seconde solution :

Organigramme O₂



Programme Pap 2

Adresse	Mnémoniques	Codes hexadécimaux
1800	LD B, 10 d	06 0A
02	LD HL, 1900	21 00 19
05	LD A, (HL)	7E
Loop 06	CP A, (HL)	BE
07	JP,	FA 0B 18
0A	LD A, (HL)	7E
0B	INC HL	23
0C	DJNZ, (Loop)	10 F8
0E	Halt (FIN)	76

Nous obtenons un programme de 15 instructions au lieu de 19.

Les remarques que l'on peut faire sont :

- a) on considère la première valeur comme la plus petite
- b) on utilise l'adressage par HL pour la comparaison
- c) on utilise DJNZ, pour le comptage
- d) on utilise le fait que CP ne détruit pas le contenu de A.

Ceci nous montre bien que la solution à un problème n'est pas unique.

En ne changeant qu'une seule instruction, montrer que l'on peut obtenir avec ce programme, le nombre le plus grand dans A au lieu du plus petit.

III.11. Instructions d'«APPEL» et de «RETOUR»

L'exécution d'une instruction de «saut» comme nous venons de l'étudier, consiste à effectuer un branchement à une autre partie du programme lui-même.

Pour éviter une trop grande complexité des programmes, il arrive bien souvent que ceux-ci soient scindés en plusieurs parties : le programme

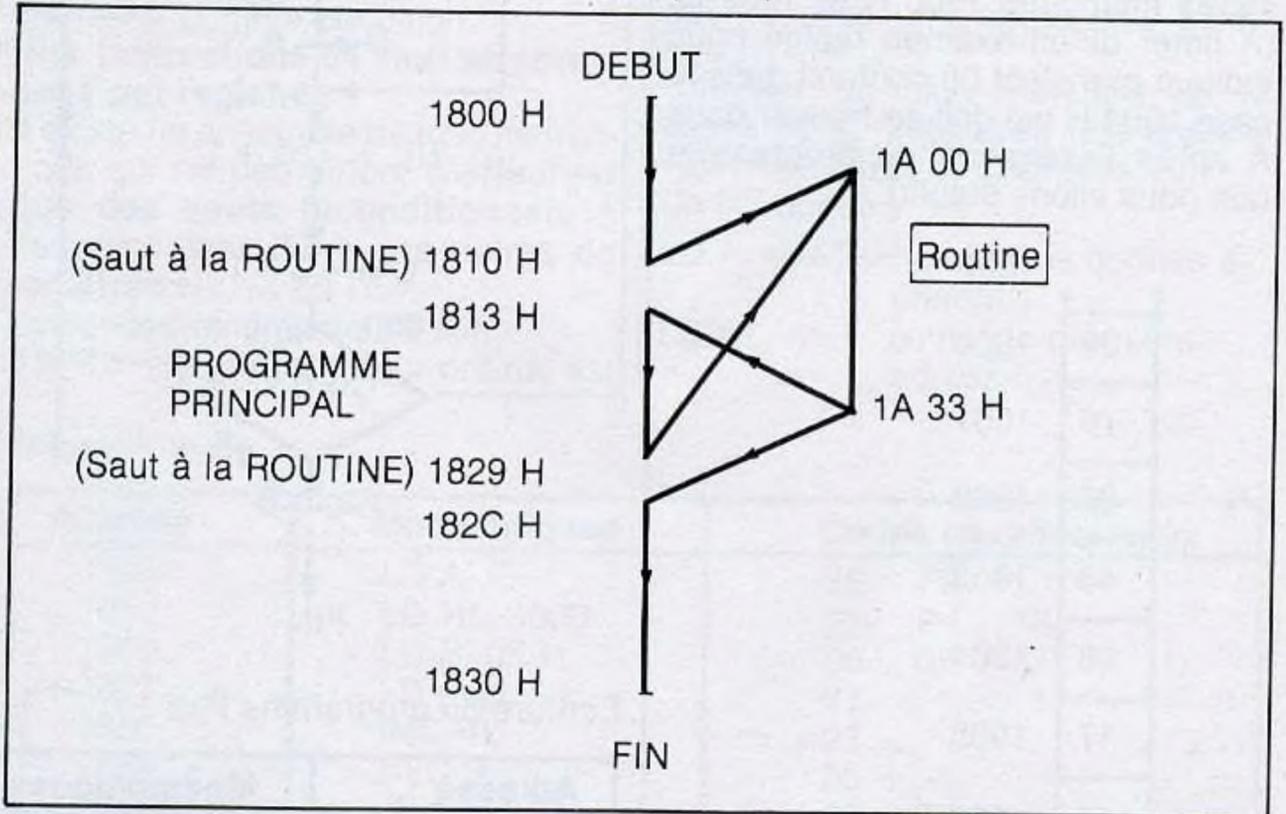
principal par lui-même et un ou plusieurs sous-programmes.

Un «sous-programme» est un programme par lui-même, mais qui effectue une (ou un ensemble de) tâche bien précise, à laquelle le programme principal est souvent amené à faire «APPEL».

Par exemple, si le programme principal doit effectuer des opérations arithmétiques sur des quantités introduites en BCD, il faudra avant tout les convertir en «binaire». De même le résultat qui apparaîtra en «binaire» devra être transcodé en «BCD».

Il est bien évident que les deux séquences de transcodage peuvent être introduites dans le programme principal. Seulement chaque fois que le programmeur voudra les utiliser, il devra les réintroduire intégralement. Ceci risque fort d'alourdir sérieusement son travail.

On préfère de beaucoup utiliser des «sous-programmes» ou «routines» que l'utilisateur peut placer là où il veut dans le champ adressable de la mémoire. Généralement, on les place avant ou après le programme principal.



L'exécution d'une instruction d'«APPEL» **suspend momentanément le déroulement du programme** en cours, qui sera repris sitôt la routine exécutée.

La figure 5 schématise le déroulement d'un programme avec deux branchements à la routine qui commence en 1A00 pour se terminer en 1A33.

Supposons que l'instruction soit «APPEL inconditionnel», le code opératoire est analogue à l'instruction de

branchement en adressage immédiat. Le code hexadécimal est D3 suivi de la première adresse de la routine.

Ainsi, nous aurons en 1810 H :

D3 00 1A

Comme il s'agit d'une instruction de 3 bytes, avant l'exécution de 1810 H, le contenu du compteur ordinal sera

1813 H (1810 H + 3)

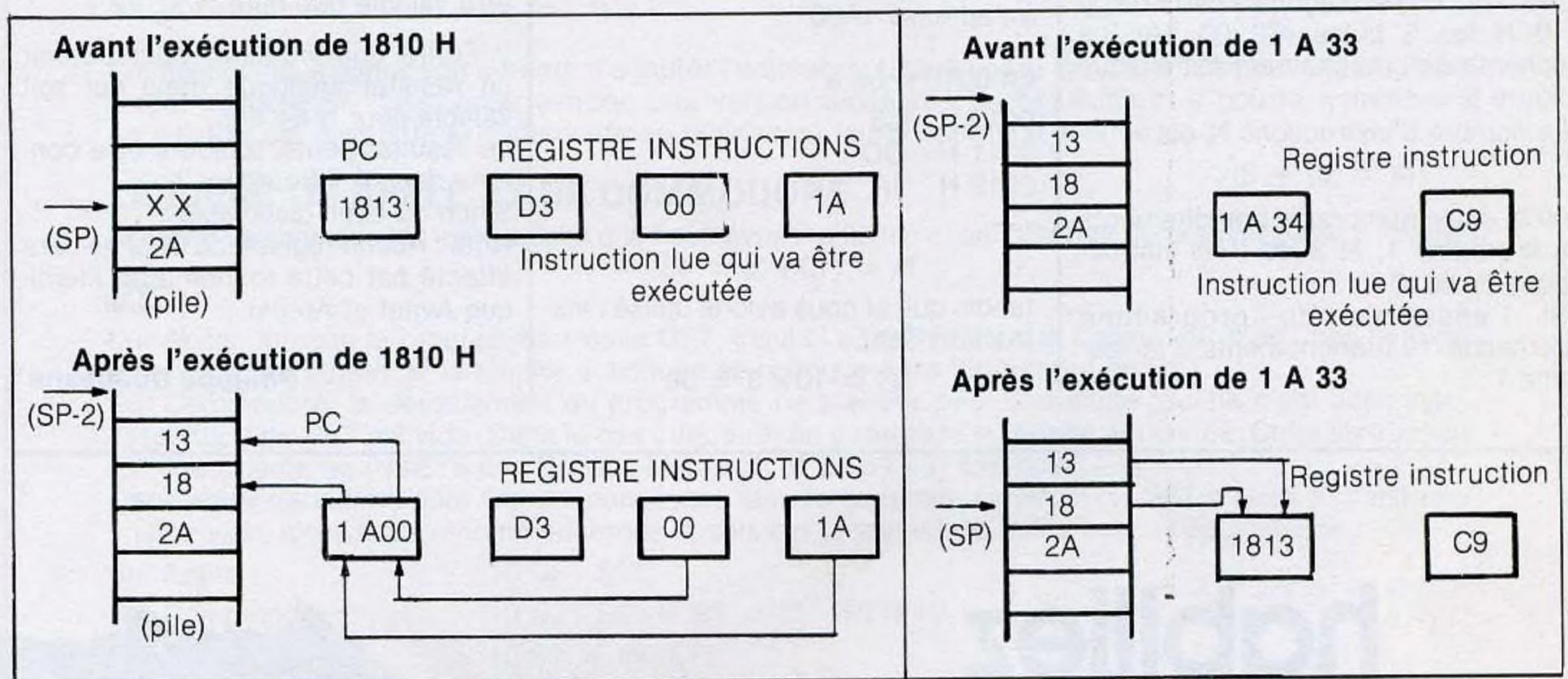
Après l'exécution de la routine, le programme principal doit être repris, là même où il a été suspendu.

D'où le schéma de l'exécution de

D3 00 1A.

CODE OPERATOIRE		CONDITION
APPEL	RETOUR	
CD n n	C9	Inconditionnel
DC n n	D8	Si C = 1
D4 n n	D0	Si C = 0
CC n n	C8	Si Z = 1
C4 n n	C0	Si Z = 0
EC n n	E8	Si parité impaire
E4 n n	E0	Si parité paire
FC n n	F8	Si signe négatif S = 1
F4 n n	F0	Si signe positif S = 0

Tableau I



Avant de charger PC avec le contenu de l'opérande (1A00), le contenu du compteur ordinal est sauvegardé dans la pile. Le registre pointeur de pile SP est décrémenté de 2.

Le programme saute à 1A00 H pour exécuter la routine. Celle-ci doit impérativement se terminer par une instruction de retour. Ainsi, si en 1A33, fin de la routine, nous avons une instruction de retour (sous-entendu) au programme suspendu, le schéma d'exécution de cette instruction est représenté ci-dessous.

Le code hexadécimal de retour inconditionnel est C9.

L'instruction de retour charge le compteur ordinal avec l'adresse du programme suspendu qui avait été sauvegardée dans la pile.

Les instructions d'APPEL sont toujours effectuées en adressage immédiat étendu (2 octets).

Les instructions d'APPEL et de RETOUR peuvent être conditionnées

à l'état «1» ou «0» de l'un des indicateurs du registre F.

Ce qui conduit au tableau I.

III.12. Instruction «RESTART»

Nous venons d'étudier les instructions d'APPEL qui nécessitent 3 bytes dus essentiellement au fait qu'il s'agit d'un adressage immédiat étendu.

Certains microprocesseurs et notamment le Z 80 possèdent des instruc-

tions dites «restart» d'un seul octet qui permettent d'effectuer un saut inconditionnel avec sauvegarde du contenu de PC dans la pile à l'une des huit adresses comme il l'est indiqué sur le tableau II.

Pendant l'exécution d'une **instruction restart**, le contenu du **compteur ordinal est d'abord sauvegardé dans la pile.**

Le PC est chargé pour l'octet de poids fort avec «OOH» et pour l'octet

Mnémonique	Code opération	Adresse
RST «0»	C7	00 00 H
RST «8»	CF	00 08 H
RST «16»	D7	00 10 H
RST «24»	DF	00 18 H
RST «32»	E7	00 20 H
RST «40»	EF	00 28 H
RST «48»	F7	00 30 H
RST «56»	FF	00 38 H

Tableau II

de poids faible l'un des octets comme l'indique le tableau en fonction du code opération.

Exemple :

Supposons que le programme principal effectué au cours de son déroulement des Appels à une routine écrite à partir de 1A00.

Nous avons étudié dans le paragraphe précédent comment l'instruction CALL (CD 00 1A) était utilisée dans ce cas. Chaque fois qu'il sera nécessaire de faire appel à la routine, nous devons utiliser 3 bytes dans le programme.

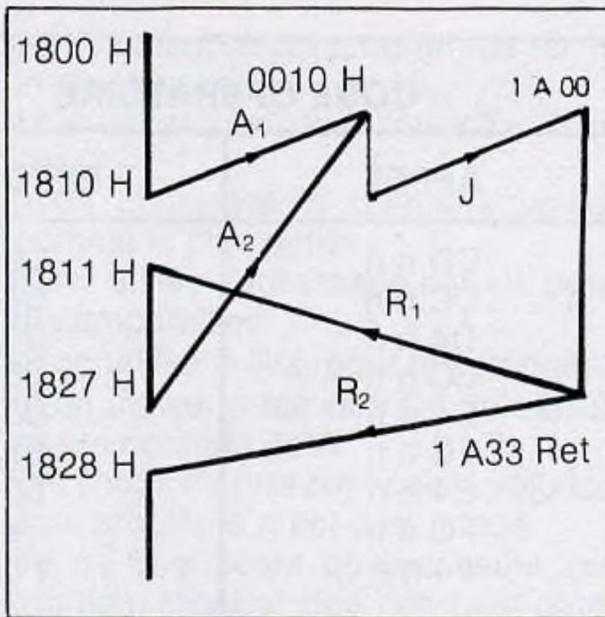
Remplaçons l'instruction CALL du programme principal par Restart (RST 16 : D7) et plaçons à partir de 00 10 H les 3 bytes (C3 00 1A). Le schéma de branchement est indiqué figure 8.

Le nombre d'instructions N_i est :

$$N_i = (b_i + 3)$$

où b_i est le nombre de branchements à la routine 1, et 3 les trois instructions en 0010 H.

Si l'ensemble du programme demande 10 branchements à la routine 1 :



Saut inconditionnel :
à l'adresse 1A00

En 0010 H on a :

0010 H C3
0011 H 00
0012 H 1A

$$N_i = (10 + 3) = 13$$

tandis que si nous avons utilisé l'instruction Call :

$$N'_i = 10 \times 3 = 30$$

Dans cet exemple, le gain est de 17 instructions.

A noter qu'en 0010 H, l'instruction est un branchement inconditionnel et non pas un « Appel ».

III.13. Exercices

Nota : Prendre comme première adresse de vos programmes 1800 H

1. Ecrire un programme simple qui additionne les N premiers nombres entiers et place le résultat dans l'accumulateur.

Prendre N = 12.

Pour N = 18, que faut-il faire ?

Pourquoi ce programme ne peut-il être valable que pour $N \leq 22$?

2. Ecrire une « routine » pour obtenir un résultat analogue mais qui soit valable pour $N \leq 255$.

Le résultat peut-il toujours être contenu dans A ? Pourquoi ?

Sinon où est-il disponible ?

Nota : Aucun registre du CPU ne sera affecté par cette routine (Etat identique Avant et Après).

Philippe Duquesne

habilitez votre collection



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

Venez chercher votre (vos) exemplaires, ou envoyez ce bon de commande, accompagné de votre règlement à :

EDITIONS FREQUENCES
1, boulevard Ney, 75018 Paris

Nom

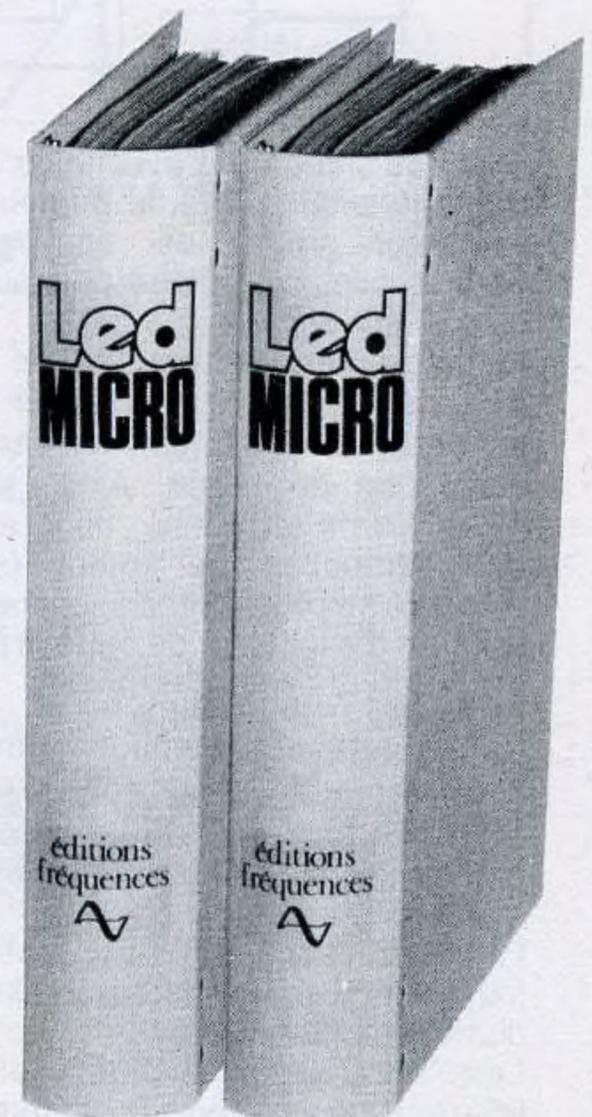
Adresse

Ci-joint le montant de

CCP Chèque bancaire Mandat

Led MICRO

avec
une
superbe
reliure
toilée
jaune



UN EDITEUR DE TABLEAU

Ce mois-ci, nous allons voir comment adapter l'éditeur du tableau paru dans le n° 13 pour l'Apple II sur le Commodore 64, puis, nous verrons une version améliorée de cet éditeur. Il pourra notamment enregistrer sur disquette le contenu de ce tableau ou faire l'opération inverse.

LE BASIC DE APPLE ET DE COMMODORE

Nous ne verrons que les instructions que nous avons utilisées et qui diffèrent du premier basic à l'autre.

GET

Sur Apple, lorsque le programme trouve GET, celui-ci s'arrête tant que l'opérateur n'agit pas sur une touche. Le caractère de la touche actionnée se retrouve dans l'argument de GET.

Sur Commodore, le déroulement du programme ne s'arrête pas. Si aucune touche n'est actionnée, l'argument de GET est vide. Dans le cas contraire, on y retrouve la touche actionnée. Cette instruction est équivalente au INKEY\$ des BASICs Microsoft et ZX 81 par exemple.

Dans notre traduction pour Commodore, nous testerons ce que l'instruction GET a saisi. Si c'est une chaîne vide, alors nous recommencerons la saisie d'un caractère, nous avons l'équivalence :

```
Sur Apple :          10 GET S$
Sur Commodore :     10 GET S$ : IF S$ = "" THEN 10
Sur Sinclair :      10 S$ = INKEY$
                   1$ IF S$ = "" THEN GOTO 10
```

GESTION DU CURSEUR ECRAN

C'est ici que commencent à apparaître les problèmes sérieux.

Sur Apple nous disposons des instructions suivantes :

- Home : efface écran
- HTAB I : positionne le curseur dans la lème colonne
- VTAB J : positionne le curseur dans la Jème colonne

Sur Commodore, nous avons :

- TAB I : positionne le curseur dans la lème colonne

Pour les autres opérations, nous devons envoyer un code spécial qui aura une action particulière. Cette table nous donne la valeur décimale du code à envoyer pour différentes actions :

Effacement de l'écran	147
Le curseur écran en haut à gauche	19
Le curseur écran sur la ligne d'en dessous	145
Le curseur écran sur la ligne d'en dessus	17
Le curseur écran à droite	29
Le curseur écran à gauche	157

Par exemple, lorsque le programme rencontre l'instruction PRINT CHR\$(147) ; l'écran est effacé et le curseur se trouve dans le coin en haut à gauche.

Dans le programme pour l'Apple, nous avons aux lignes

```
HTAB P(XT) + C - 1
VTAB YT - YR + 1
```

Dans le programme **pour le Commodore**, nous plaçons d'abord le curseur dans le coin en haut à gauche en envoyant le code 19 :

```
9930 PRINT CHR$(19) ;
```

Le point virgule est indispensable. Si celui-ci n'était pas présent, le curseur se trouverait sur la deuxième ligne et la première colonne car l'absence de ponctuation après l'instruction PRINT génère un retour chariot vers l'écran.

Nous voulons écrire un caractère sur la $(YT - YR + 1)$ ème ligne, nous devons donc sauter $(YT - YR)$ ligne (si nous n'avons pas mis de ponctuation après l'instruction PRINT de la ligne 9930, nous aurions dû sauter $(YT - YR - 1)$ ligne). Pour cela, nous utilisons une boucle FOR/NEXT aux lignes 9931 et 9932. Le curseur écran se trouve alors sur la $(YT - YR + 1)$ ème ligne mais sur la 1ère colonne.

Pour l'amener sur la $P(XT) + C - 1$ ème colonne, nous utilisons une instruction TAB.

Il faut se rappeler que PRINT TAB(0) place le curseur dans la 1ère colonne, il y a un décalage d'une unité par rapport à l'instruction HTAB de l'Apple : nous avons donc l'instruction 9933 PRINT TAB (PLXT) + (-2) ;

Surtout ne pas oublier le point virgule !!

Comme nous devons placer le curseur plusieurs fois à cet endroit de l'écran (pour afficher le curseur puis l'ancien caractère et éventuellement le nouveau caractère), nous avons décidé d'utiliser un sous-programme plutôt que de réécrire 3 fois la même série d'instructions. Il a donc fallu mettre une instruction RETURN (ligne 9934).

Maintenant, chaque fois qu'il y a les instructions HTAB P(XT) + C - 1 ; VTAB YT - YR + 1, nous aurons un GOSUB 9930.

Il existe d'autres méthodes pour gêner le curseur écran en faisant des POOKES bien placés mais, nous verrons cela à l'occasion d'un autre programme.

LES CODES CARACTERES

Sur l'Apple IIe, nous avons 4 touches fléchées mais sur Commodore, nous n'avons que 2 touches qui, associées à SHIFT, permettent les déplacements dans les 4 directions.

Pour ne pas avoir à agir sur 2 touches à la fois lorsque nous désirons mouvoir le curseur, nous avons décidé d'utiliser 2 touches supplémentaires : les 2 flèches caractères.

La partie du programme interprétation de la touche actionnée n'est modifiée que pour les codes ASCII.

LA VERSION AMELIOREE DE L'EDITEUR

Cette deuxième version de l'éditeur est plus performante et plus souple à utiliser, elle est dotée de fonctions nouvelles.

Les touches fléchées et la touche RETURN ne servent qu'au déplacement du curseur contrairement à la première version.

La touche RETURN permet de faire revenir le curseur au début de la ligne suivante. Les quatre touches de fonction vont permettre les opérations nouvelles de cet éditeur.

• Action de la touche F₁ ou F₂

Cette fonction permet de rappeler à l'opérateur l'utilisation des différentes touches de commandes. De 9890 à 9899, nous affichons du texte puis lorsque l'opérateur a fini de le lire, il agit sur la touche Return pour revenir à l'éditeur.

• Action de la touche F₃ ou F₄

L'action sur F₃ ou SHIFT F₃ (F₄) permet d'amener directement le curseur au début d'une ligne quelconque.

Ainsi en 9921, nous effaçons l'écran. Le curseur de l'écran se trouve au début de la 2ème ligne car, il n'y a pas de point virgule. Puis nous entrons une chaîne de caractères représentant le n° de la ligne où l'opérateur veut se rendre (ligne 9922).

Nous voulons que le curseur se trouve au début de la ligne indiquée donc, nous faisons XT = 1 pour qu'il soit dans la 1ère colonne du tableau, puis C = 1 pour qu'il soit sur le 1er caractère.

De même, nous vérifions si la valeur du numéro de ligne entrée correspond à une ligne existante pour ne pas planter le système.

En 9925, nous avons $YR = YT - 5$ de manière à avoir le curseur sur la 5ème ligne de l'écran (il est utile de pouvoir connaître les lignes précédentes).

• Action de la touche F_5

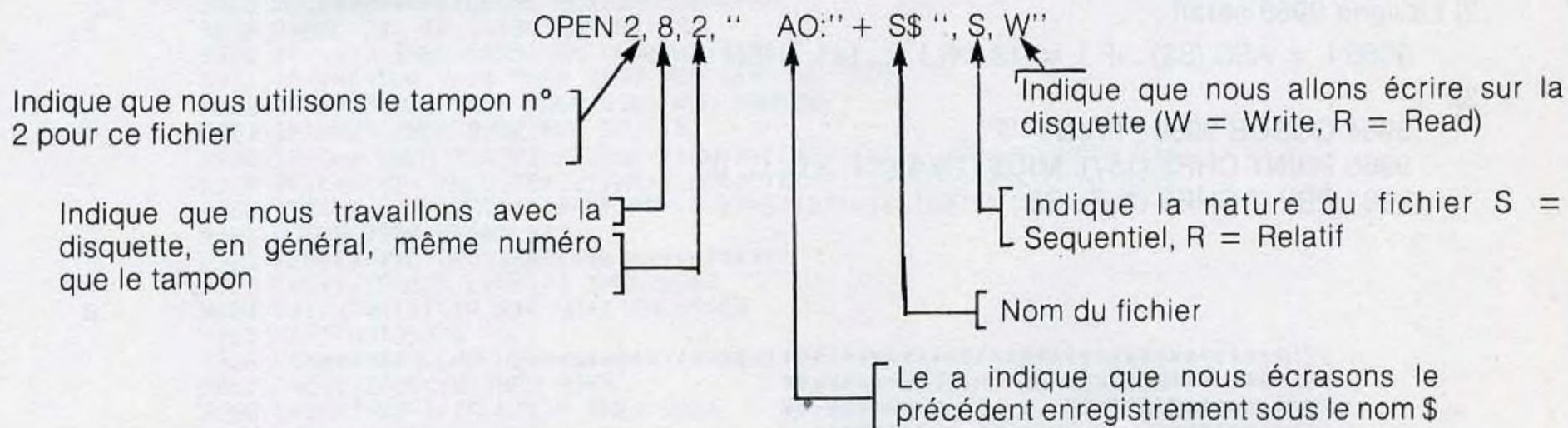
Cette commande permet de charger le tableau sur la disquette.

Nous allons en profiter pour voir quelques commandes de l'Operating System (gestion des disquettes).

En 9911, nous effaçons l'écran et vidons la chaîne de caractère $S\$$ (qui contient F_5).

Si l'opérateur ne tape pas de nom de fichier, nous retournons à l'éditeur.

En 9915, nous indiquons que nous allons faire une entrée/sortie : la syntaxe est la suivante :



De 9916 à 9918, nous chargeons le fichier $S\$$ par les informations présentent dans le tableau.

La boucle I explore toutes les lignes et la boucle J explore toutes les colonnes de la ligne.

Le # 2 après l'instruction PRINT, indique que les informations seront envoyées vers le tampon n° 2 donc vers le fichier $S\$$.

En 9919, l'instruction CLOSE 2 indique que nous avons terminé les opérations sur le tampon n° 2, ce dernier s'est alors transféré sur la disquette sous le nom $S\$$.

• Action de la touche F_6

Cette touche fait l'action contraire de F_5 , c'est-à-dire qu'elle permet de charger le tableau à partir de la disquette.

A la ligne 9905, on ouvre le fichier $S\$$ en écriture (READ). Nous avons choisi le même numéro de tampon mais, nous aurions pu en prendre un autre.

Si le fichier $S\$$ n'existe pas, le système se plantera et répondra «FILE NO FOUND» (fichier non trouvé).

L'instruction INPUT # 2 fait le contraire de l'instruction PRINT # 2.

Il est indispensable de fermer le fichier pour libérer le tampon n° 2 (ligne 9909).

• Action de la touche F_7 ou F_8

Cette touche a la même fonction que la touche RETURN dans la version précédente, c'est-à-dire qu'elle permet d'indiquer que l'opérateur a fini de remplir le tableau.

Remarques diverses

Dans la première version, le programmeur devait afficher en haut de l'écran le nom des colonnes. Dans cette version, le programmeur doit mettre dans la chaîne de caractère $M\$$ le nom des colonnes ou ce qu'il veut faire apparaître en haut de l'écran.

QUELQUES PETITES QUESTIONS

Après l'instruction PRINT de la ligne 9902, nous avons deux virgules. Si nous ne voulions en mettre qu'une seule, où aurions-nous dû la mettre pour avoir la même chose sur l'écran ?

A la ligne 9968, nous avons :

I = ASC(\$) IF I = 13 OR I = 141 THEN C = I : GOTO 9985

qui permet lorsque la touche RETURN ou SHIFT RETURN est actionnée, d'aller au début de la ligne suivante. Qu'aurait-il fallu mettre après THEN pour n'aller qu'à la colonne suivante sur la même ligne ? Aux lignes 9966 et 9981, nous aurions pu remplacer les instructions GOSUB 9930 par des instructions PRINT moyennant quelques modifications. Comment faire ?

Réponses :

1) Nous pouvons remplacer les lignes 9902 et 9903 par :

```
9902 PRINT "CHARGEMENT DU TABLEAU"
9903 PRINT, "A PARTIR DU DISQUE"
```

2) La ligne 9968 serait :

```
9968 I = ASC (S$) : IF I = 13 OR I = 141 THEN 9984
```

3)

```
9964 GOSUB 9930 : PRINT "?";
9966 PRINT CHR$ (157), MID$ (TB $ (YT, XT), C, I);
9981 PRINT CHR$ (157), S$ ;
```

```
1 REM*****
2 REM*****EDITEUR DE TABLEAU*****
3 REM*****ECRIT PAR B.LILAMAND*****
4 REM*****SUR COMMODORE SX64*****
5 REM*****
10 REM PETIT PROGRAMME DE TESTE
15 REM DIMENSIONNEMENT DES
20 DIM TB$(50,4),P(4),L(4)
25 PRINT CHR$(147);
30 REM INITIALISATION DES
31 REM VARIABLES EXTERNES
35 NL=40:NC=4
45 P(1)=4:P(2)=12:P(3)=25:P(4)=34
50 REM NOM DES COLONNES
55 PRINT "  PRODUIT DISTRIBUTEUR QUANTITE PRIX";
70 GOSUB 9940
80 PRINT CHR$(147);
85 PRINT "VOULEZ VOUS FAIRE DES MODIFICATION(O/N)";
90 GET S$:IF S$="" THEN 90
100 IF S$="N" THEN END
105 PRINT CHR$(147);
110 PRINT "  PRODUIT DISTRIBUTEUR QUANTITE PRIX";
120 GOSUB 9952
125 GOTO 80
130 REM*****
131 REM**DANS UNE APPLICATION SEULLES *
132 REM**LES LIGNES 20,35,45,55 ET 70 *
133 REM**SERONT CONSERVEES *
134 REM*****
9930 PRINT CHR$(19);
9931 FOR K=1 TO YT-YR:PRINT
9932 NEXTK
9933 PRINTTAB(P(XT)+C-2);
9934 RETURN
9940 REM**INITIALISATION*****
9941 S=40
9942 FOR J=NC TO 1 STEP -1:L(J)=S-P(J)-1:S=P(J):NEXTJ
9943 FOR I=1 TO NL
9944 FOR J=1 TO NC
9945 TB$(I,J)=LEFT$(TB$(I,J)+".",L(J))
9946 NEXT J:NEXTI
9951 S=23: IF NL<23 THEN S=NL
9952 YR=0:YT=1
```

```

9953 XT=1:C=1
9954 REM****IMPRESSION DU TABLEAU****
9955 PRINT CHR$(19);
9956 FOR I=1 TO S:PRINT
9957 L=I+YR:S#=RIGHT$(STR$(L),2)
9958 S#=LEFT$(S#,2):PRINTS#;
9959 FOR J=1 TO NC
9960 PRINT TAB(P(J)-2);" ";TB$(L,J);
9961 NEXT J:NEXT I
9962 REM **SAISIE D'UN CARACTERE****
9963 GOSUB 9930:PRINT"?"
9964 GET S#:IF S#="" THEN 9965
9965 GOSUB 9930:PRINT MID$(TB$(YT,XT),C,1)
9966 REM****INTERPRETATION DE LA ****
9967 REM****TOUCHE ACTIONNEE****
9968 V=ASC(S#):IF V=13 THEN RETURN
9969 IF V=17 THEN 9995:REM VERS LE BAS
9970 IF V=145OR V=94 THEN 9990:REM VERS LE HAUT
9971 IF V=95 OR V=157 THEN 9986:REM GAUCHE
9972 IF V=29 THEN 9982:REM DROITE
9973 IF C=1 THEN TB$(YT,XT)=S#+RIGHT$(TB$(YT,XT),L(XT)-1):GOTO 9981
9974 IF C=L(XT) THEN TB$(YT,XT)=LEFT$(TB$(YT,XT),C-1)+S#:GOTO 9981
9975 TB$(YT,XT)=LEFT$(TB$(YT,XT),C-1)+S#+RIGHT$(TB$(YT,XT),L(XT)-C)
9976 GOSUB 9930:PRINT S#
9977 REM****A DROITE****
9978 C=C+1:IF C>L(XT)+1 THEN 9964
9979 C=1:XT=XT+1:IF XT>NC+1 THEN 9964
9980 XT=1:GOTO 9996
9981 REM****A GAUCHE****
9982 C=C-1:IF C<0 THEN 9964
9983 C=1:XT=XT-1:IF XT<0 THEN 9964
9984 XT=1
9985 REM****VERS LE HAUT****
9986 IF YT=1 THEN 9964
9987 YT=YT-1:IF YT-YR<6 OR YR=0 THEN 9964
9988 YR=YR-5:IF YR<0 THEN YR=0
9989 GOTO 9954
9990 REM****VERS LE BAS****
9991 IF NL=YT THEN 9964
9992 YT=YT+1:IF YT-YR<17 OR YR=NL-S THEN 9964
9993 YR=YR+5:IF YR>NL-S THEN YR=NL-S
9994 GOTO 9954

```

READY.

```

9890 PRINT CHR$(147)
9891 PRINT:PRINT"F1.....LES COMMANDES"
9892 PRINT:PRINT"F3.....ACCES DIRECTE A UNE LIGNE"
9893 PRINT:PRINT"F5.....SAUVEGARDE SUR LA DISQUETTE"
9894 PRINT:PRINT"F7..CHARGEMENT A PARTIR DE LA DISQUETTE"
9895 PRINT:PRINT"ACTIONNEZ UNE TOUCHE"
9896 GET S#:IF S#="" THEN 9896
9897 REM*CHARGEMENT A PARTIR DU DISQUE**
9898 PRINT CHR$(147);S#=""
9899 PRINT"CHARGEMENT DU TABLEAU",,
9900 PRINT"A PARTIR DU DISQUE"
9901 INPUT"NOM DU FICHIER";S#:IF S#="" THEN 9955
9902 OPEN2,8,2,"0:"+S#+",S,R"
9903 FOR I=1 TO NL
9904 FOR J=1 TO NC:INPUT#2,TB$(I,J)
9905 NEXTJ:NEXTI
9906 CLOSE 2:GOTO 9940
9907 REM****SAUVEGARDE SUR DISQUE****
9908 PRINTCHR$(147);S#=""
9909 PRINT"SAUVEGARDE SUR DISQUE"
9910 INPUT"NOM DU FICHIER";S#
9911 IF S#="" THEN 9955
9912 OPEN2,8,2,"@0:"+S#+",S,W"
9913 FOR I=1 TO NL
9914 FOR J=1 TO NC:PRINT#2,TB$(I,J)
9915 NEXTJ:NEXTI
9916 CLOSE 2:GOTO 9955

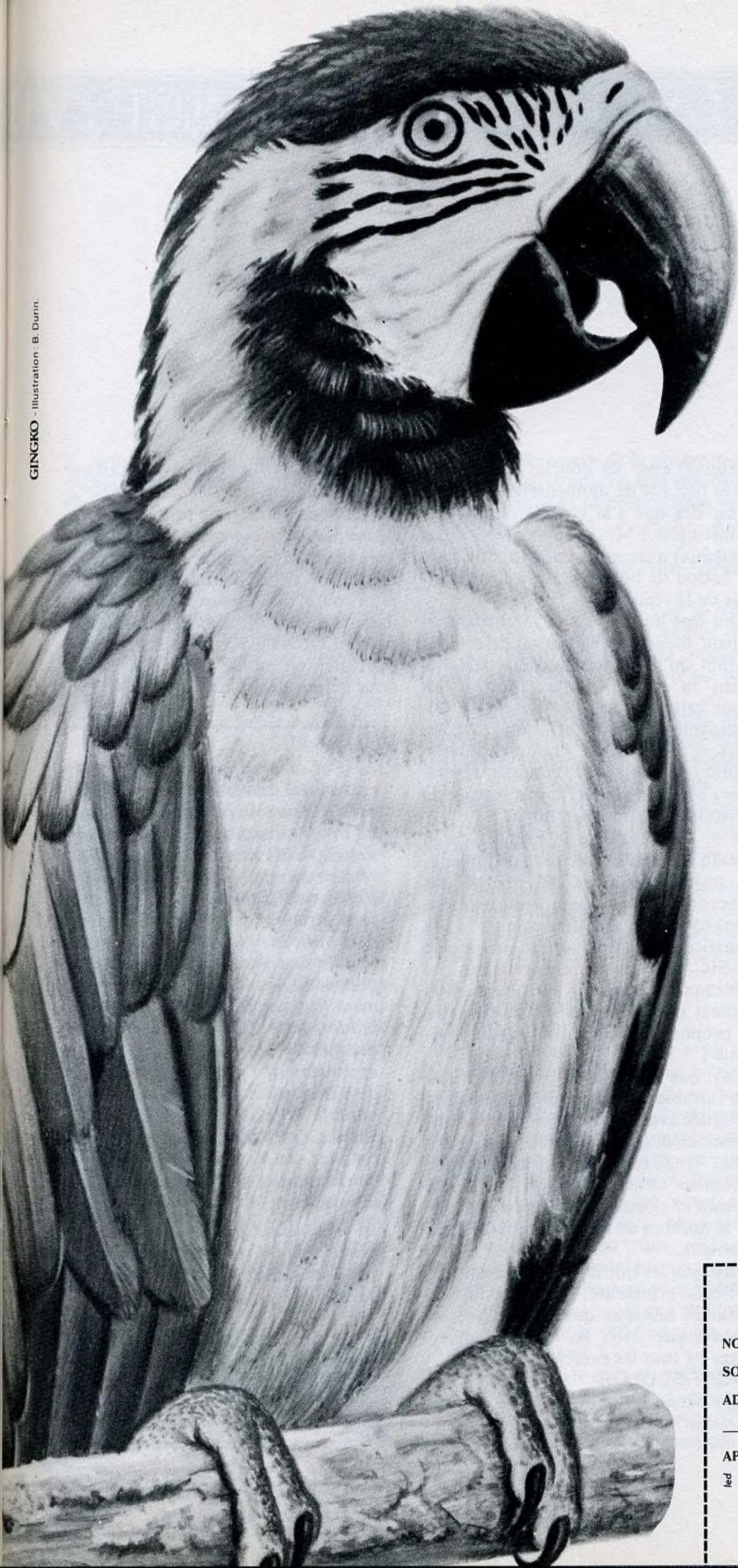
```

LE COIN DES FORTICHES LE COIN D

```
9920 REM***DEMANDE LE NUMERO DE LIGNE***
9921 PRINT CHR$(147)
9922 PRINT:INPUT"NUMERO DE LA LIGNE DESIREE";S#
9923 XT=1:YT=VAL(S#):IF YT<1 THEN YT=1
9924 C=1:IF YT>NL THEN YT=NL
9925 YR =YT-5:IF YR<0 THEN YR=0
9926 IF YR>NL-S THEN YR=NL-S
9927 GOTO9955
9929 REM***POSITIONNE LE CURSEUR ECRAN**
9930 PRINT CHR$(19);
9931 FOR K=1 TO YT-YR:PRINT
9932 NEXTK
9933 PRINTTAB(P(XT)+C-2);
9934 RETURN
9940 REM***INITIALISATION*****
9941 S=40
9942 FOR J=NC TO 1 STEP -1:L(J)=S-P(J)-1:S=P(J):NEXTJ
9943 FOR I=1 TO NL
9944 FOR J=1 TO NC
9945 TB$(I,J)=LEFT$(TB$(I,J)+"......"),L(J)
9946 NEXT J:NEXTI
9951 S=23:IF NL<23 THEN S=NL
9952 YR=0:YT=1
9953 XT=1:C=1
9954 REM***IMPRESSION DU TABLEAU*****
9955 PRINT CHR$(147);M#;
9956 FOR I=1 TO S:PRINT
9957 L=I+YR:S#=RIGHT$(STR$(L),2)
9958 S#=LEFT$(S#,2):PRINTS#;
9959 FOR J=1 TO NC
9960 PRINT TAB(P(J)-2);"I";TB$(L,J);
9961 NEXT J:NEXT I
9962 PRINT:PRINT"F1 POUR CONNAITRE LES COMMANDES";
9963 REM **SAISIE D'UN CARACTERE*****
9964 GOSUB 9930:PRINT"?"
9965 GET S#:IF S#="" THEN 9965
9966 GOSUB 9930:PRINT MID$(TB$(YT,XT),C,1)
9967 REM*****INTERPRETATION DE LA ****
9968 REM*****TOUCHE ACTIONNEE*****
9969 I=ASC(S#):IF I=13 OR I=141 THEN RETURN
9970 IF I=17 THEN 9995:REM VER LE BAS
9971 IF I=145OR I=94 THEN 9990:REM VERS LE HAUT
9972 IF I=95 OR I=157 THEN 9986:REM GAUCHE
9973 IF I=133 OR I=137 THEN 9935
9974 IF I=134 OR I=138 THEN 9920
9975 IF I=135 OR I=139 THEN 9910
9976 IF I=136 OR I=140 THEN 9900
9978 IF C=1 THEN TB$(YT,XT)=S#+RIGHT$(TB$(YT,XT),L(XT)-1):GOTO 9981
9979 IF C=L(XT) THEN TB$(YT,XT)=LEFT$(TB$(YT,XT),C-1)+S#:GOTO 9981
9980 TB$(YT,XT)=LEFT$(TB$(YT,XT),C-1)+S#+RIGHT$(TB$(YT,XT),L(XT)-C)
9981 GOSUB 9930:PRINT S#
9982 REM*****A DROITE*****
9983 C=C+1:IF C>L(XT)+1 THEN9964
9984 C=1:XT=XT+1:IF XT>NC+1 THEN9964
9985 XT=1:GOTO9996
9986 REM*****A GAUCHE*****
9987 C=C-1:IF C<0 THEN 9964
9988 C=1:XT=XT-1:IF XT<0 THEN 9964
9989 XT=1
9990 REM*****VERS LE HAUT*****
9991 IF YT=1 THEN 9964
9992 YT=YT-1:IF YT-YR<6 OR YR=0 THEN 9964
9993 YR=YR-5:IF YR<0 THEN YR=0
9994 GOTO 9954
9995 REM*****VERS LE BAS*****
9996 IF NL=YT THEN 9964
9997 YT=YT+1:IF YT-YR<17 OR YR=NL-S THEN 9964
9998 YR=YR+5:IFYR>NL-S THEN YR=NL-S
9999 GOTO 9954
```

READY.

B. Lilamand



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

Rarement on aura vu aussi doué pour les langues que le Victor S1. Il parle 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 _____

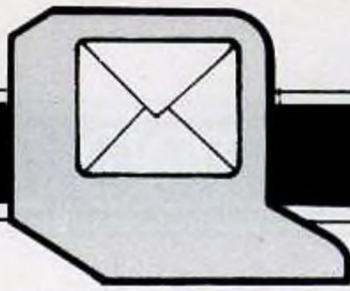
SOCIÉTÉ _____ FONCTION _____

ADRESSE _____

TÉL. _____

APPLICATION _____

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



□ ... Voici non plus quelques exemples mais le cours que j'utilise sur TO7 (MO5) avec mes élèves. Du CM2 à la 3^e. Vous pouvez, bien sûr, en tirer des exemples, mais surtout j'aimerais votre avis (que je crois très sûr) sur ce cours.

Je suis toujours votre cours bien entendu, dans lequel la progression est très intéressante (Je pense cependant que vous allez un peu vite en ce moment. Voir nos 13 et 14 !)

Je remarque (par rapport aux réactions de mes élèves) que la notion de «boucle» est celle qui passe la moins bien, or c'est une structure de base en programmation, non ! (elle est liée directement à la notion de tableau !).

Amicalement vôtre.

Jean Lépine 49120 La Tourlandry

Merci infiniment pour l'envoi de votre cours. Votre lettre tombe à pic pour m'aider à exposer quelques-uns de mes «principes pédagogiques».

Mon cours va trop vite

C'est bien la première fois qu'on me fait cette remarque. J'ai plutôt reçu le reproche inverse. J'espère cependant que la notion d'affectation est bien assimilée (c'est l'essentiel). Les exercices de récapitulation vont permettre à vos élèves de maîtriser peu à peu les fonctions «chaînes de caractères».

La notion de boucle

J'ai éprouvé les mêmes difficultés que vous au début de ma carrière de «professeur de BASIC».

Mes premiers élèves étaient les professeurs de maths et de physique d'un lycée privé. Dès la première leçon, ils étaient capables de rédiger des pro-

grammes avec des boucles : j'étais fier de la rapidité de mon enseignement.

Mais dès que j'ai changé de public : catastrophe ! Mes premiers élèves (des matheux) avaient assimilé tout de suite la notion de boucle grâce à leur pratique de la récurrence.

Pour que les élèves non matheux assimilent facilement la notion de boucle, il faut qu'ils maîtrisent déjà parfaitement la notion d'affectation. C'est pour cette raison que j'ai insisté très lourdement sur l'affectation, avec les dessins des petites «boîte-variables» montrant comment le contenu de chaque «boîte» varie au fur et à mesure du déroulement du programme.

En ce qui concerne le rapport de la notion de boucle avec celle de tableau, je crains que nous ne soyons pas d'accord. Vous avez certainement remarqué que, contrairement à la quasi-totalité des autres cours de BASIC, je n'ai pas encore parlé des tableaux. Et pourtant introduire rapidement des tableaux m'aurait permis de proposer des exercices de jeux amusants !

Il faut que la notion de boucle utilisant des variables simples soit parfaitement assimilée avant que l'élève fasse la connaissance des variables-tableaux. Sans cela, il a tendance à confondre «l'indice» définissant la place dans le tableau et «l'indice» utilisé pour préciser le nombre de fois qu'on passe dans la boucle.

J'enseigne les boucles en «3 passes» de difficulté croissante. Je n'introduis les variables tableaux que lors de la troisième passe avec les boucles imbriquées (et tous les exercices de tri).

LOCATE-COLOR

Vous introduisez ces instructions tout au début. Ce n'est pas ce que je fais.

Mais c'est vous qui avez raison. L'introduction de LOCATE dès le départ vous permet de proposer à vos élèves des exercices très démonstratifs. Très bien. Je les utiliserai. Je procédera ainsi si je devais recommencer mon cours. (J'ai des excuses : les ordinateurs que j'utilisais ne connaissaient pas le LOCATE mais des instructions équivalentes moins faciles d'emploi pour le tout débutant).

□ ... J'ai lu avec beaucoup d'intérêt comme les autres le n° 13 de LED-MICRO surtout les rubriques Cours de Programmation et Coin des Fortiches. Je me suis attaché particulièrement à comprendre le programme Editeur de Tableau ; j'ai une profession commerciale et j'ai trouvé avec vous la réponse à un souci que j'avais concernant un programme pour éditeurs de tableaux. Seulement, j'ai voulu transcrire sur Commodore 64, j'y ai passé des heures et des heures pour avoir un résultat approximatif : j'ai modifié pas mal de lignes pour obtenir une édition potable (90, 100, 105, 110, 9941, 9956) ; des lignes 9963 à 9999 j'ai renoncé. Je n'arrive pas à positionner le curseur et à le déplacer selon le cahier des charges présenté.

Je vous serais gré si vous le voulez bien de me fournir une transcription sur Commodore 64. Par avance, je vous remercie...

B.P. 59121 Prouvy

Votre lettre arrive juste au moment où Bruno Lilamand propose cette traduction dans son article du «Coin des Fortiches».

Je pense que vous avez reçu la lettre d'explications complémentaires de Bruno.

... Disposant d'un budget d'environ 2 000 francs pour l'achat d'un micro-ordinateur, n'ayant aucune notion d'informatique mais possédant tout de même un niveau de mathématiques assez élevé (le DEUG), quel matériel me conseillez-vous d'acheter...

M.C. 30610 Sauve

A l'époque où j'étais le rédacteur en chef de LED-MICRO, je recevais beaucoup de lettres de ce type et je répondais toujours «LED-MICRO vous donnera les moyens de vous faire votre opinion pour que vous puissiez choisir vous-même en analysant vos besoins. Rappelez-vous seulement que l'abondance du logiciel, la fiabilité du matériel, la qualité de la documentation, le sérieux de l'après-vente et la garantie de durée du fabricant sont des facteurs plus importants que les performances pures.»

Les articles de C.-H. Delaleu vont toujours dans ce sens.

... Je suis un de vos nouveaux lecteurs (...). Alors j'ai sauté sur mon crayon, car j'ai très envie de participer, ne serait-ce qu'un tout petit peu à un journal comme le vôtre.

Je vous envoie donc avec cette lettre une utilitaire de tri très rapide et deux sujets d'exercices avec leurs corrections (organigramme et programme)... mon petit programme de bulletin de paye est plus que rudimentaire dans son état actuel, mais

au fur et à mesure de la progression des leçons, il pourra être amélioré aussi bien pour la présentation que jusqu'à l'édition à partir d'un fichier.

M. P.G. (08430 Poix-Terron)

Monsieur Lépine trouve que je vais trop vite mais vous, vous voulez me faire galoper 10 fois plus vite encore ! Le tri (rapide ou lent), ce n'est pas pour tout de suite.

Je vous avoue que j'ai d'abord eu une réaction de rejet en lisant votre programme de bulletin de paie : vraiment très banal ! Mais nos élèves ne préféreront-ils pas des exercices utiles à des exercices originaux ? J'utiliserai vraisemblablement cet exemple le moment venu.

Vous proposez de partir d'un exemple très simple pour le perfectionner petit à petit jusqu'à l'intégrer dans des fichiers : là, nous sommes d'accord à 100 %.

AUX LECTEURS MATHÉMATIENS

Notre samideano René Sipra avait eu l'imprudence de me dire qu'il était professeur de mathématiques. De ce fait, je lui ai envoyé le brouillon de mon texte sur l'équation du 3^e degré avant son impression. Il m'a rédigé une solution meilleure que celle que je vous propose (il finira par me vexer cet homme !) : «l'intérêt de la résolution que je vous propose est triple : elle suit pas à pas la méthode historique de Cardan et Viète. Elle permet d'étendre la résolution à l'ensemble des nombres complexes et elle se réduit à deux cas possibles au lieu de 3.»

Il m'a aussi signalé qu'aujourd'hui on n'utilise plus tg, cotg, arctg, arsin... mais tan, cot, asin, acos et atan. Hé

oui, il y a déjà 40 ans que j'ai quitté la troupe de Monsieur Pougnan (à ce propos, j'aimerais pouvoir retrouver mon ancien professeur et les copains de la taupe de Buffon d'il y a 40 ans !). Je n'ai pas eu le courage ni de saborder entièrement ma rédaction, ni de tout faire recommencer à Chantal (notre claviste), mais si des lecteurs sont à la fois assez matheux pour être intéressés par cet exercice et aussi novices que moi pour ne pas connaître les méthodes de Cardan et de Viète, je leur enverrai une photocopie de la lettre de Monsieur Sipra.

Dites-nous si vous êtes intéressé par les problèmes de programmation pour le niveau mathématiques ★★. Nous n'avons l'intention de poser de tels problèmes qu'assez rarement et pour des motifs d'utilité générale (inversion de matrice, résolution d'équations différentielles par exemple). Si nos lecteurs ne sont pas concernés par ces problèmes, ce «assez rarement» deviendra «jamais».

... 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

Je me rends compte que j'ai répondu de travers à votre lettre. Nous essaierons de tenir compte de votre suggestion, mais nous voulons éviter d'intercaler de la publicité dans la partie «cours» de LED-MICRO.

LIBRE PROPOS

Comme tout secteur économique en très forte progression, la micro-informatique engendre beaucoup d'espoirs mais aussi des illusions et des déceptions. Si les professionnels connaissent les combines, il n'en est pas de même des utilisateurs, dont les mésaventures ne se comptent plus. Certains revendeurs n'hésitent pas à importer en parallèle avec les importateurs des machines qui ne sont pas conformes aux spécificités du marché français. S'il est déplaisant de se retrouver avec une documentation en anglais ou en allemand, cela n'est rien comparé aux difficultés provoquées par une machine non conforme au standard Secam par exemple. Pire encore, les logiciels piratés par de «petits malins» qui les revendent en n'offrant, on s'en doute, aucune sécurité à l'utilisateur quant à «l'intégrité» du logiciel. Et puis, il y a toutes les tricheries sur les caractéristiques techniques où un flou soigneusement voulu permet de confondre la taille du bus interne d'un microprocesseur avec celle du bus des datas. (Combien de faux 16 et 32 bits sont ainsi sur le marché !). Les plus «vicieux» jouent avec l'espace mémoire et mettent dans le même sac Ram et Rom, confondent, volontairement mais parfois involontairement, l'espace mémoire avec le système d'exploitation...

Il existe aussi bien d'autres procédés, plus subtils encore, pour créer le doute dans l'esprit de l'utilisateur. Ainsi, certaines personnes pour vaincre un acheteur hésitant, raconte qu'un constructeur californien est en train de tailler des croupières à Ibm, mettant ainsi en péril les ventes du Pc. Rares sont les acheteurs qui comparent les bilans financiers des entreprises. Il est évident qu'avec un chiffre d'affaires de 300 milliards de francs et un bénéfice de 11 milliards, Ibm considère sûrement ce genre de propos comme une simple blague... Mais l'acheteur le sait-il ? Et sait-il aussi que le deuxième constructeur au monde d'informatique, Digital Equipment, ne réalise que le dixième du chiffre d'affaires d'Ibm. Quant aux constructeurs spécialisés en micro-ordinateurs, aucun d'eux ne fait mieux que le centième du chiffre d'affaires d'Ibm. De là, à mettre en difficulté un tel géant...

Pour vendre, tout est bon. Le plus bel exemple est sans aucun doute l'utilisation du terme «compatibilité». Il n'est pas de micro qui ne soit compatible Ibm-Pc. Mais il y a des degrés dans la compatibilité, ce qui revient à dire qu'acheter un compatible à 70 % est acquérir un «incompatible».

Bref, la micro, c'est la jungle. Il faut y avancer prudemment et prendre un maximum de précautions : vérifier les informations, s'assurer que la documentation est bien en français, que les logiciels sont francisés (un traitement de texte sans accent, quel plaisir), que le service après-vente existe (combien de hot-line qui ne fonctionnent qu'épisodiquement ou avec un personnel incompétent). Attention aussi aux cartes d'extension de provenance douteuse, à la fiabilité aléatoire. Fort heureusement, il y a aussi d'honnêtes gens qui font bien leur métier. Il faut les encourager en refusant de traiter avec les chacals.

C.H. Delaleu

SOMMAIRE

LE TRAITEMENT DE TEXTE EST DE TOUS LES LOGICIELS CELUI DONT L'UTILISATEUR DE MICRO-ORDINATEUR A LE PLUS SOUVENT BESOIN. POUR ÊTRE VRAIMENT UTILE, IL DOIT RÉPONDRE A UN CERTAIN NOMBRE DE CRITÈRES **P. 52**. CE MOIS-CI, EN CONTRE-MESURES, LE VICTOR S1, UNE MACHINE PROFESSIONNELLE TRÈS LARGEMENT RÉPANDUE **P. 56**. LE PÈRE NOËL EST PASSÉ APPORTANT PLEIN DE MICROS ; POUR LES ALIMENTER, UNE SÉLECTION DE LOGICIELS PROPOSÉE DANS NOTRE RUBRIQUE SHOPPING **P. 62**. ET COMME CHAQUE MOIS, PHILIPPE FAUGERAS A LU POUR VOUS LES DERNIERS LIVRES DE MICRO-INFORMATIQUE **P. 64**.



le traitement de texte

Oubliée l'angoisse de la faute de frappe ou de la faute d'orthographe. Avec un traitement de texte, n'importe qui peut rivaliser avec la meilleure des dactylographes. Et que de temps gagné !

Dans la vie quotidienne comme dans la vie professionnelle, nous avons tous besoin d'écrire une lettre, rédiger des notes ou un rapport. Le stylo à bille et la feuille de papier demeurent les moyens les plus simples. Mais voilà, nous ne pratiquons plus l'art «de la belle écriture» comme nos arrière-grands-pères, et nos textes sont généralement illisibles. Quant à savoir utiliser correctement une machine à écrire, rares sont les Français qui le peuvent. Faute d'une bonne âme ou d'un(e) secrétaire qui accepte de dactylographier nos textes, nos lettres, nos rapports sont d'une présentation détestable.

De tous les services rendus via le micro-ordinateur, le traitement de texte est, sans doute, celui qui libère le plus l'utilisateur des contraintes liées à l'utilisation d'une machine. En effet,

avec un traitement de texte, l'utilisateur frappe son texte comme sur une machine à écrire, mais sans se préoccuper des erreurs de frappe, de taper sur la touche «retour à la ligne». Il peut ainsi se concentrer sur la rédaction de son texte. Le logiciel se charge d'aller à la ligne sans couper les mots en deux. Cette frappe donne un brouillon qu'il va être possible de corriger et de modifier tout à loisir. L'utilisateur, en effet, reprend son texte et peut le travailler jusqu'à sa complète satisfaction : il corrige les fautes de frappe ou d'orthographe, change des mots, supprime les répétitions, modifie des paragraphes et les déplace éventuellement. Cet ensemble de possibilités est dévolu à l'éditeur qui est sans doute la fonction la plus importante du traitement de texte. Le texte prend ainsi sa forme définitive sans poser les habituelles difficultés : ratures, manque de place pour ajouter

un mot ou une phrase, corrections illisibles. Il n'y a plus ensuite qu'à imprimer le texte. Et pour ce faire, déterminer la présentation souhaitée : interlignage, largeur de la marge et transférer le texte de l'écran à l'imprimante.

Lorsqu'on a goûté au traitement de texte, on ne peut plus s'en passer, tout simplement, parce qu'il est aisé de mettre en forme ses idées sans contraintes techniques et qu'en outre on dispose d'un texte facile à lire.

En théorie, tous les traitements de texte sont quasiment identiques, puisqu'ils offrent le même service. Dans la pratique, ce n'est pas tout à fait la même chose. Ils offrent plus ou moins de possibilités. Ils sont plus ou moins complexes. Certains sont destinés plutôt à une utilisation personnelle et non pas professionnelle. D'autres demandent un apprentissage de plusieurs jours pour bien les maîtriser. Aussi avant d'acquérir un logiciel, il est judicieux de prendre quelques précautions et de déterminer précisément ce qu'on désire faire.

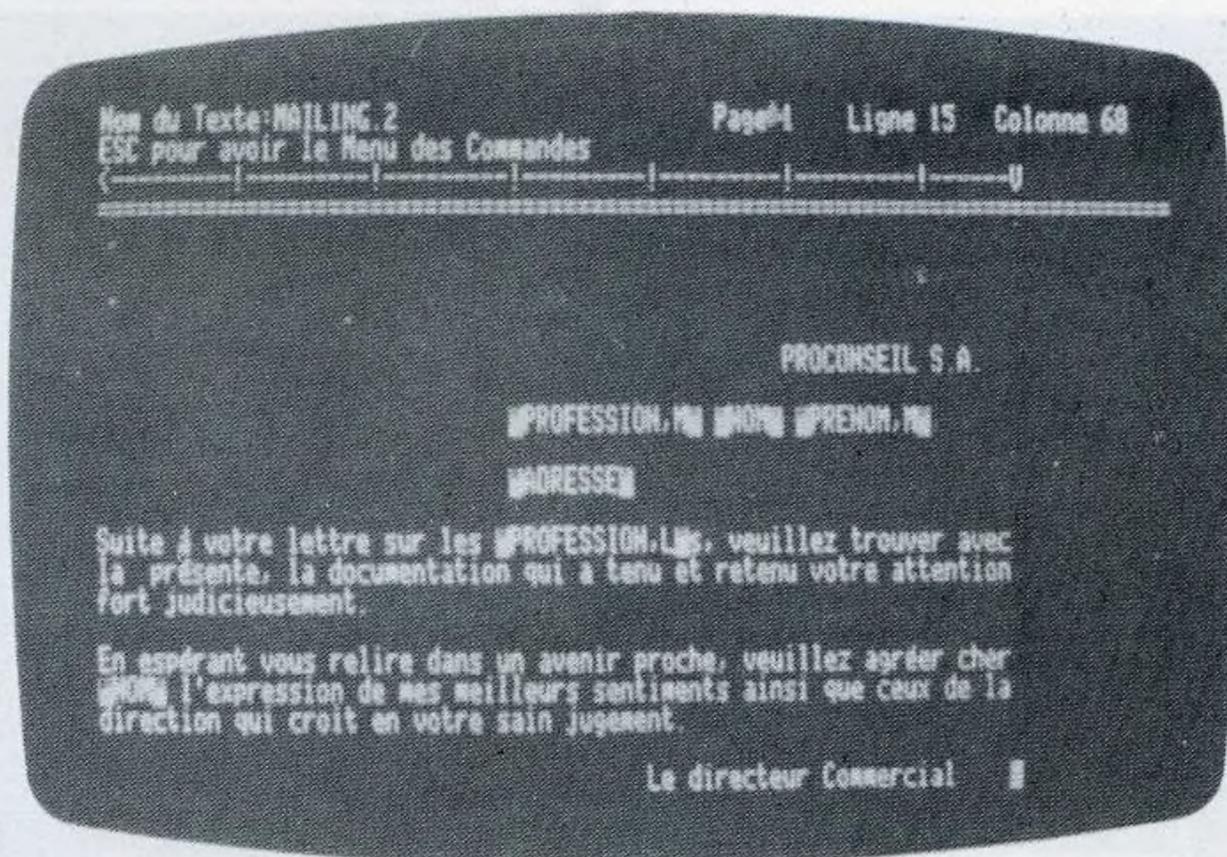
Première chose, à prendre en compte : savoir si le traitement de texte est bien destiné au traitement de textes en français. Bon nombre de logiciels sont d'origine anglo-saxonne et, de ce fait, pas adaptés aux particularités de langue française, comme ces maudits accents aigu, grave et circonflexe sans oublier le tréma qui font le désespoir des écoliers. Rien n'est plus pénible à lire qu'un texte sans accent. En outre, si on peut l'admettre pour un usage personnel, cela n'est pas acceptable pour une utilisation professionnelle où une présentation parfaite est impérative. Comme qui peut le plus peut le moins, mieux vaut choisir un traitement de texte parfaitement « francisé », pourvu d'une documentation en français, claire et aisément compréhensible, ce qui n'est pas toujours le cas. Généralement, lorsque la machine est équipée d'un clavier Azerty, il n'y a

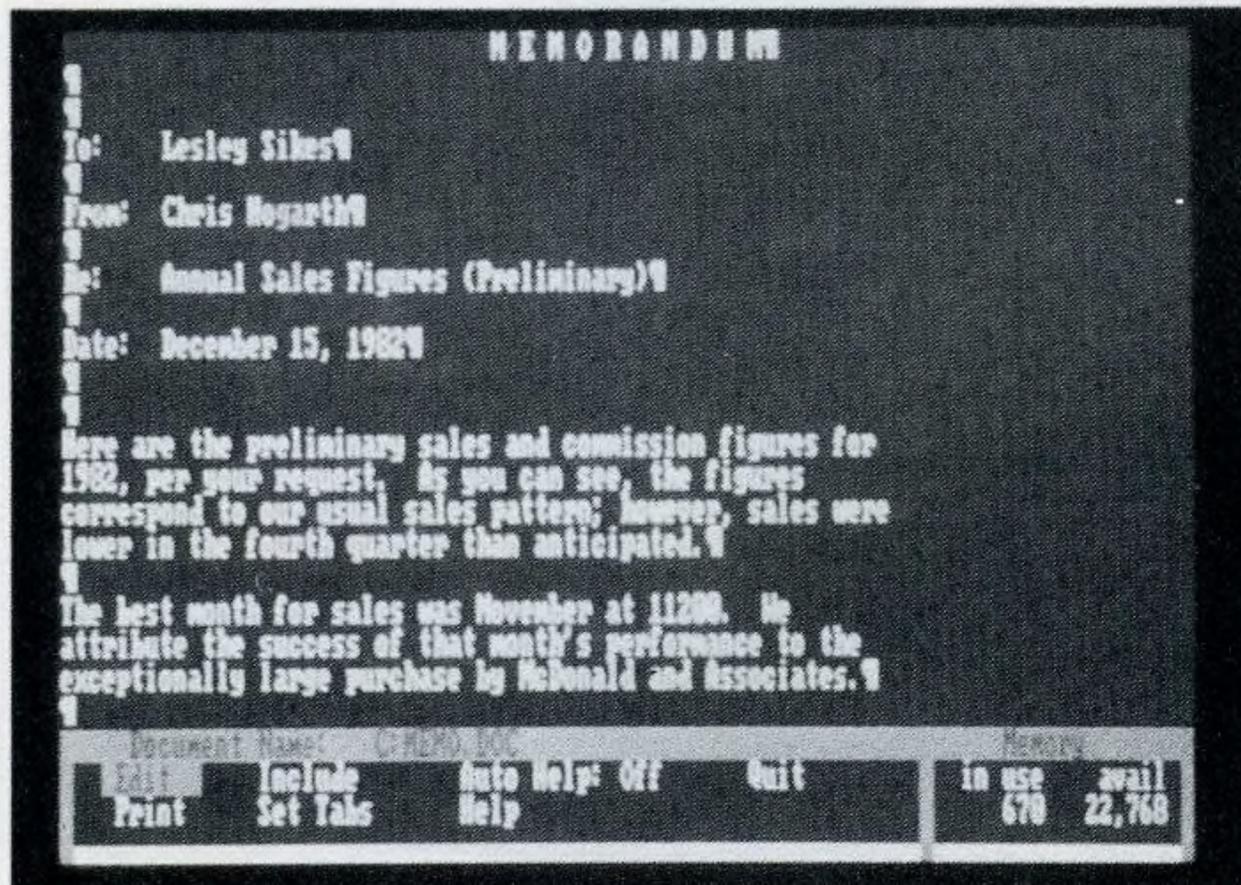
pas de problème pour les accents aigu et grave. En revanche, il peut y en avoir pour l'accent circonflexe.

Certains importateurs, pour vendre leur machine dotée d'un clavier Qwerty, prétendent qu'il est aussi facile de taper sur un clavier Qwerty que sur un clavier Azerty. Cet argument est fallacieux. Un clavier Qwerty pose des problèmes en matière de traitement de texte. Un traitement de texte doit faciliter la vie de son utilisateur pour être utile. Lors du choix, il faut préférer celui qui impose le moins de contraintes. Par exemple, l'éditeur doit utiliser le moins de touches possibles pour chaque fonction, de même le déplacement du curseur doit être le plus simple possible, c'est-à-dire qu'il doit éviter les combinaisons complexes pour se faire par caractère, ligne ou paragraphe. D'autre part, il faut que le texte affiché à l'écran le soit dans sa forme réelle. Autrement dit que l'utilisateur puisse le visualiser à l'écran

comme il sera une fois imprimé. Il est donc souhaitable que les accents soient placés sur les lettres, que les « gras » et « soulignés » apparaissent à l'écran. De cette manière, l'utilisateur sait ce qu'il va obtenir et n'a pas de surprise désagréable. Il faut savoir que certains logiciels de traitement de texte, quelle que soit l'impression choisie, affichent le texte sur 80 colonnes. Dans de très nombreux documents, il est nécessaire d'inclure des tableaux. Cette fonction n'est pas toujours prévue. Lorsqu'elle n'existe pas, l'utilisation devient plus complexe. De même, tous les traitements de texte ne disposent pas des symboles mathématiques, des indices et exposants, des caractères grecs.

Si leur usage est fréquent, il faut choisir un traitement de texte offrant ces possibilités. Il existe dans un grand nombre de traitements de texte une fonction très utile qui évite les répétitions. L'utilisateur indique un mot et la machine le traque partout et effectue





son remplacement par un autre. Cela est également possible pour des groupes de mots. Certains traitements de texte sont dotés d'un dictionnaire de plusieurs milliers de mots qui exécute automatiquement les corrections orthographiques. Ces fonctions sont des «plus» que n'ont pas tous les logiciels et qui sont très utiles lorsqu'on frappe des textes longs comme des rapports où les risques de répétitions sont fréquents.

A signaler également que certains traitements de texte facilitent et simplifient les travaux répétitifs qu'on peut rencontrer lors d'une utilisation professionnelle (lettre commerciale notamment) en disposant d'instructions représentant une suite d'opérations.

Il est évident que plus un traitement est complet, plus il exige un long apprentissage. Le choix est fonction de l'utilisation. C'est une évidence et cependant nombre de vendeurs ont tendance à

conseiller un produit très complet et souvent complexe (parce qu'il peut tout faire !). C'est peut-être satisfaisant pour l'esprit mais pas pour l'utilisateur.

Enfin, il ne faut pas oublier qu'un traitement de texte, c'est un peu comme le carburant d'une automobile ; toutes les voitures ne marchent pas au super. Certains logiciels de traitement de texte demandent une capacité mémoire plus importante que d'autres. Sur Apple 128 k pour Memword, 64 k pour CX Texte sur Ibm-PC : 128 k pour Textor ou Wordstar, 64 k pour Easywriter. Par ailleurs, toutes les imprimantes ne sont pas disponibles en standard. Ce point est à vérifier lorsqu'on possède déjà une imprimante.

Nous vous indiquons quelques traitements de texte disponibles sur les deux machines les plus répandues sur le marché français. Il en existe bien évidemment d'autres et notamment sur les machines dites domestiques. C.R.

QUELQUES TRAITEMENTS DE TEXTE SUR APPLE

- Apple Writer Iie, (Apple) 1 300 F HT
- CX Texte (Contrôle X) ... 1 000 à 3 300 F HT
- Epistole (Version soft) 1 600 F HT
- Papyrus (Ediciel) 850 F HT
- Memword (Memsoft) . 1 900 F HT
- Pie Writer (Gamic)... 1 500 F HT

QUELQUES TRAITEMENTS DE TEXTE SUR IBM-PC

- Easy Writer (Axone)
- Ocean teste (Océan systèmes)
- Spellbinder (Megalpa)
- Textor (Talor Distribution)
- Visiword (Métrologie)
- Volkswriter (Edisoft)
- Word (Microsoft)
- Wordstar (Micropro)
- Word Perfect (Axone).

A NE PAS OUBLIER LORS D'UN ACHAT

- Le type d'utilisation : professionnelle ou personnelle
- Le type et le volume des documents à traiter (lettre, rapport, dossier)
- L'utilisation de caractères spécifiques : indices, exposants, symboles mathématiques
- L'insertion de tableaux
- La visualisation à l'écran du texte imprimé
- La configuration matérielle : taille mémoire, imprimante, système d'exploitation
- La francisation.

Choisissez une carrière d'avenir.

10 métiers informatiques

L'un d'eux peut être demain le vôtre...
quel que soit votre niveau de formation.

Vous pouvez commencer vos études à tout moment, sans interrompre vos activités professionnelles actuelles.

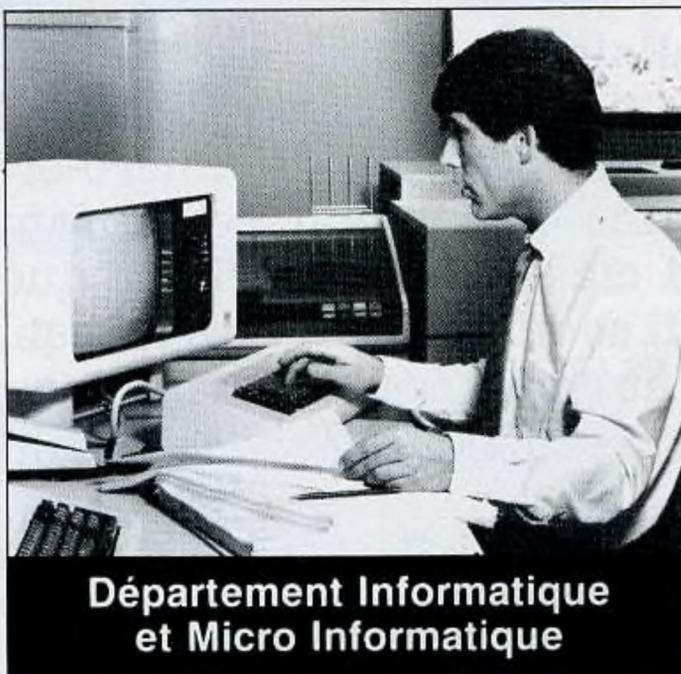
Devenir informaticien en 1985, c'est choisir une carrière d'avenir, avec l'assurance de trouver immédiatement de nombreux débouchés, et des perspectives d'autant plus intéressantes que la place de l'ordinateur ne cesse de s'accroître dans tous les domaines: économique, social, administratif, etc.

Quel que soit votre niveau de formation (et même si vous n'avez pas de diplôme), EDUCATEL se charge de vous apprendre en quelques mois par les moyens les plus modernes, et avec un enseignement personnalisé à votre cas, le métier informatique qui vous convient le mieux.

A la fin de votre formation EDUCATEL, vous recevrez un certificat que savent apprécier les employeurs et nous appuierons votre candidature.

EDUCATEL est la plus grande Ecole privée d'enseignement par correspondance en France: 300 professeurs contrôlés par l'Education Nationale.

Demandez, sans engagement de votre part, notre documentation gratuite en nous renvoyant le bon ci-contre ou en nous téléphonant au (1) 208 50 02.



Choisissez votre futur métier :

PROGRAMMEUR D'APPLICATION

Vous travaillez en collaboration avec l'analyste, testez et mettez au point les programmes. (Niveau d'accès: 3^e - 2^e).

ANALYSTE PROGRAMMEUR

Entre la conception et la réalisation du projet informatique, vous adaptez chaque programme à la demande de l'utilisateur (niveau d'accès: BAC).

PROGRAMMEUR SUR MICRO-ORDINATEUR

En quelques mois, vous pourrez dialoguer avec n'importe quel «micro» et écrire vos propres programmes. (Niveau d'accès: 3^e).

OPERATEUR SUR ORDINATEUR

Vous assurerez principalement les différentes manipulations nécessaires au fonctionnement de l'ordinateur. (Niveau d'accès: 3^e).

INITIATION A L'INFORMATIQUE

L'informatique fait maintenant partie de notre univers quotidien. En quelques mois, apprenez l'essentiel sur cette technique. (Niveau d'accès: 3^e).

OPERATRICE DE SAISIE

Votre travail consiste à saisir des informations en langage compréhensible pour l'ordinateur. (Accessible à tous).

PUPITREUR

Il assure la mise en route, la conduite et la surveillance des opérations effectuées par l'ordinateur. (Niveaux d'accès: 3^e - 2^e).

ANALYSTE

A un niveau intermédiaire entre l'utilisateur et l'application informatique, vous concevez l'application et formalisez la solution qui sera ensuite confiée aux programmeurs. (Niveau d'accès: BAC + 2).

B.P. INFORMATIQUE

Préparation à l'examen officiel (niveau d'accès soit 5 années d'expérience professionnelle dans l'informatique, soit un C.A.P. - B.E.P. - B.P. - B.T.S. ou le Baccalauréat + 2 années d'expérience professionnelle dans l'informatique).

B.T.S INFORMATIQUE

Même débutant, vous pourrez réaliser votre projet d'avenir en préparant ce diplôme officiel qui vous garantit une situation stable. (Niveau d'accès: BAC).

On embauche des milliers d'informaticiens

Les chiffres de l'A.N.P.E. le prouvent: actuellement plus de la moitié des postes proposés par les employeurs à des informaticiens (programmeur, opérateur sur ordinateur, etc.) ne sont pas pourvus, faute de candidats en nombre suffisant. Et les spécialistes du Plan lancent un cri d'alarme: la France a besoin très rapidement de 100.000 nouveaux informaticiens. Les débouchés sont donc nombreux, tant pour les hommes que pour les femmes, et ceci, à tous les échelons de la hiérarchie. Découvrez vite comment devenir réellement l'un de ces «techniciens de l'avenir»!

Educatel

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

Si vous êtes salarié, votre étude peut être prise en charge par votre employeur (loi du 16.7.1971 sur la formation continue).
EDUCATEL
1083, route de Neufchâteau
3000 X - 76025 Rouen Cédex

BON pour une documentation détaillée sur 10 métiers de l'informatique

OUI, je désire recevoir gratuitement (et sans aucun engagement) une documentation détaillée sur la formation EDUCATEL d'enseignement personnalisé des 10 métiers informatiques. J'y trouverai pour chaque métier préparé le plan de formation complet, son niveau d'accès, le programme des travaux pratiques, sa durée et son prix.
Si je le désire, une orientation et des conseils personnels me seront fournis gratuitement.

NOM _____ Prénom _____
Adresse: _____
Code postal _____ Ville _____
Téléphone (facultatif) _____ Age _____
Profession exercée _____ Niveau d'études _____

Précisez le métier qui vous intéresse:

RETOURNEZ DES AUJOURD'HUI CE BON A :
EDUCATEL G.I.E. Unieco Formation,
3000 X - 76025 ROUEN CEDEX
Pour Canada, Suisse, Belgique: 49, rue des Augustins, 4000 Liège
Pour TOM-DOM et Afrique: documentation spéciale par avion.

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

SOSEX

LEM008

ou téléphonez à Paris
(1) 208.50.02

le victor S1



Sorti il y a maintenant quelques années, le Sirius, qui allait vite devenir le Victor S1, s'est tout de suite imposé sur le marché. Après avoir été la référence en matière de micro-ordinateur de table moyenne gamme, il est devenu aujourd'hui un grand classique. Malgré les quelques années passées, il semble qu'il se porte bien. Ses possibilités se sont augmentées aux fils des mois et de fait il demeure un excellent investissement en ce début 85.

Il est présenté en trois parties distinctes : l'unité centrale, l'écran et le clavier séparé. Ce dernier, très agréable à utiliser, est divisé en quatre parties, à droite un pavé numérique, puis six touches de commandes, ensuite le pavé principal alphanumérique AZERTY, pour la version française, enfin à droite cinq touches de commandes pour l'éditeur. La partie supérieure des trois pavés de droite est réservée à dix touches de fonctions programmables par l'utilisateur.

L'écran monochrome vert possède un filtre anti-reflet. La luminosité est réglable, l'ensemble est orientable.

Le Victor S1 est une machine polyvalente qu'on retrouve dans de nombreux domaines. Grâce à sa version disque dur, il pourra gérer la comptabilité d'une entreprise de taille moyenne, sa définition graphique et son interface IEEE-488 (en option) lui autorisent des applications en laboratoire scientifique, enfin sa vitesse de traitement est satisfaisante pour la majorité des problèmes généralement rencontrés. Le Victor peut donc être utilisé en application professionnelle ordinaire. Il occupe une place de choix juste avant la mini-informatique.

Le S1 est équipé d'un microprocesseur

Intel 8088 présenté comme un 16 bits. Cette information, reprise par tous les constructeurs utilisant le 8088, n'est pas totalement exacte et elle rappelle la guerre des watts en haute fidélité. En fait, si le 8088 est architecturé en 16 bits, il fonctionne en entrées-sorties sur 8 bits, ainsi la rigueur voudrait qu'il appartienne à la famille des 8 bits. La mémoire vive est extensible de 128 Koctets à 896 K. Quatre emplacements permettent des extensions telles que :

- Le système audio
- Le réseau local
- CP/M-80
- Le processeur de données numériques.

Les mémoires de masses sont proposées en trois versions, soit deux fois 600 Koctets, soit deux fois 1,2 Moctets, soit la version disque dur de 10 Mégaoctets avec, en plus, un lecteur de disquette 1,2 Moctets.

Il existe plusieurs types d'interfaces possibles, deux ports séries V-24 asynchrone ou synchrone, un port parallèle, une interface IEEE-488 sous contrôle programme, en option les protocoles IBM 3780, 3278, VT 100 et réseau Victor Net.

L'écran 12", phosphore vert, possède



Le Victor S1

différents types de curseurs, bloc fixe, bloc clignotant et trait de soulignement, ainsi que la vidéo inversée. La luminosité se contrôle du clavier.

En standard, l'éditeur possède 25 lignes de 80 colonnes. La définition graphique est de 800×400 points, ce qui est plutôt rare sur une machine de ce prix.

Les langages utilisables sont nombreux ; ils existent dans les systèmes

d'exploitation CP/M 86 et MS/DOS ainsi qu'en Prologue.

Les langages sont le Basic interprété et compilé, le MS Cobol, le MS Pascal, le MS Fortran ainsi que de nombreux outils de programmation.

La programmation graphique est réalisée grâce au logiciel Grafix (en option). Grafix est un programme de gestion de routines graphiques de haute résolution. Grafix est utilisable avec les lan-

gages tournant sur le S1. Il permet de gérer les 320 000 points de l'écran. Il dessine des lignes, des cercles, remplit et colore des surfaces. De même il autorise la sortie sur périphérique.

LA DOCUMENTATION

Dans la configuration du système passé au crible, la documentation était composée de cinq classeurs format moyen (21×15).

- Opérateur reference
- MS-Basic
- MS-Basic Compiler
- Graphics Tool kit
- IEEE-488

Il est à noter que les trois derniers classeurs correspondent à des options (suppléments payants). L'ensemble de la documentation est en anglais. Il convient de préciser qu'une fois de plus la législation en vigueur dans notre pays est complètement ignorée.

OPERATOR REFERENCE

Ce classeur ne sera d'aucune utilité si vous ne comprenez pas l'anglais. Pour les autres, bien que ce document soit copieux, la lecture n'est pas évidente au premier abord. Comme généralement il n'est pas question au départ de lire l'ensemble de la documentation (ce qui est un tort), l'utilisateur classique ne recherche que les informations dont il a besoin. Dans ce cas, le renseignement recherché prendra un peu de temps. Le classeur est divisé en quatre parties principales :

- Les procédures de bases
- Le système d'exploitation MS-DOS
- Le système d'exploitation CP/M-86
- Les utilitaires pour le disque dur.

CONTRE-MESURES

MS-BASIC

Ce document en français est d'une approche bien plus attrayante que «operator reference». Chaque terme du langage Basic est repris en détails, les explications sont claires et les exemples fournis très compréhensibles. Plusieurs chapitres sont ajoutés au Basic afin d'exploiter au mieux le MS-Basic. Trois de ces chapitres sont très utiles : — Conversion de programmes en MS-Basic

— E/S disque en MS-Basic
— Sous-programmes en langage d'assemblage.

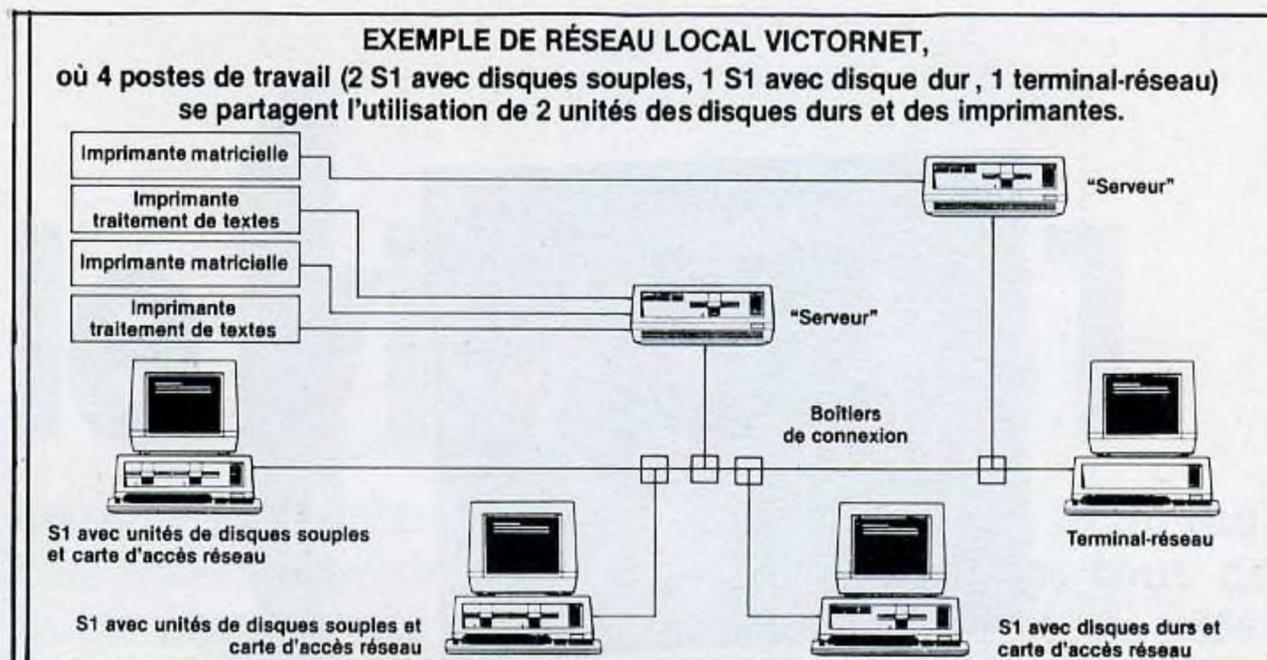
La présence de fichiers à accès de type aléatoire, pourra combler l'inexistence de fichiers à accès indexé. Le système ici est efficace, le temps d'accès à un enregistrement est très intéressant. Le plus grand reproche qu'on peut faire au MS-Basic concerne le traitement des chaînes de caractères. Leur utilisation qui n'est pas très souple est compensée par une série d'ordres qui permettent l'ensemble des traitements.

MS-BASIC COMPILER

Le faible nombre de pages est compensé par une taille de caractères beaucoup plus petite que la moyenne. L'approche de ce classeur n'est pas encourageante. La possibilité de compiler certains programmes Basic permettra de gagner du temps si vos applications comportent beaucoup de boucles et/ou de tris. Les textes sont en anglais, la documentation manque d'exemples.

GRAPHICS TOOL-KIT

Dans ce classeur, lui aussi en anglais, la première chose qui frappe, c'est la pauvreté du graphisme. Pour un docu-



ment qui traite de dessin, c'est un comble. En effet, le Victor S1 possède l'énorme avantage d'avoir dans cette gamme de prix une définition graphique bien supérieure à la moyenne. Or, dans la documentation, les quelques exemples qui existent sont d'une présentation désastreuse, les exemples logiciels sont très rares. Enfin, pour achever le tout, les explications n'accrochent pas au premier coup d'œil. Il est dommage que le constructeur qui propose une si belle définition à ce prix n'ait pas fait plus d'efforts pour cette documentation.

IEEE-488

Écrit en français, ce document est assez clair et sa lecture ne pose pas de problème majeur. Il faut noter que l'interface IEEE-488 n'est pas une interface câblée sur le Victor. En effet, le fonctionnement de la norme IEEE-488 est réalisé de manière «soft». Seul le câble qui se branche sur l'interface parallèle classique possède à une de ses extrémités le brochage adéquat.

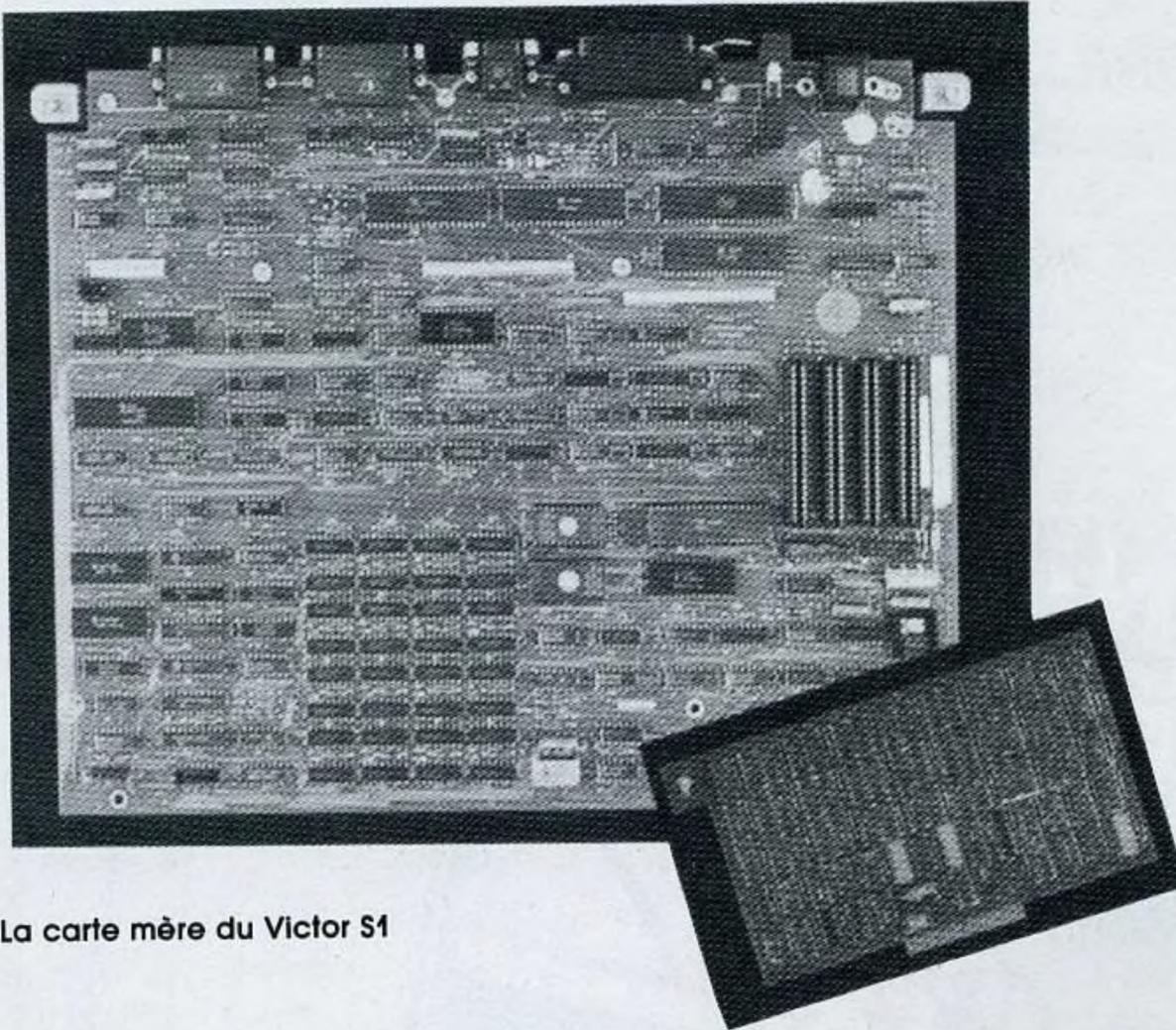
LE RESEAU LOCAL VICTORNET

Le réseau local Victornet permet de connecter jusqu'à 64 postes de travail (Victor S1) et périphériques sur une longueur de 450 mètres. La vitesse de transmission est de 1 mégabit/seconde. Chaque poste est à la fois maître et esclave et peut communiquer avec d'autres unités. Ceci est essentiel :

grâce à un réseau local de ce type, les périphériques, tels que les mémoires de masse (disques durs) et les imprimantes, sont partagés entre plusieurs postes de travail. Ceci autorise une meilleure gestion du parc de machines, et bien sûr d'abaisser le prix. La liaison s'effectue par un câble torsadé, conforme à la norme RS-422. Il ne nécessite pas de répéteur. Le transporteur dans chaque nœud gère le réseau, en tant qu'interface commune à tous les nœuds. Aucun contrôleur maître n'est nécessaire, car les transporteurs individuels assurent tous les contrôles nécessaires du réseau.

Le Victor S1 possède toutes sortes de possibilités en configurations diverses et variées. Ainsi un tel matériel pourra, sans aucun problème, être adapté aux différentes évolutions du service dans lequel il sera intégré. La machine tourne vite et l'accès aux disquettes et disques durs est plutôt rapide.

L'ergonomie du S1 facilite l'utilisation, l'écran ne provoque pas de fatigue prématurée. Nous avons essayé en situation le progiciel Delta avec la version disque dur. La vitesse d'exécution nous a agréablement surpris. Nous avons de même aperçu le S1 en laboratoire de recherches, et là aussi il donnait de bons résultats. L'avantage du S1 concerne ses grandes possibilités d'évolutivité ainsi que la configuration par soft. Inversement, ce dernier atout ralentira la prise en main de la machine. En effet, il est nécessaire de maîtriser les différents utilitaires pour se servir correctement du S1.



La carte mère du Victor S1

LE COUT DU VICTOR S1

S1 version 2x600 Ko, 128 Ko RAM	29 900 F
S1 version 2x1,2 Mo, 256 Ko RAM	37 900 F
S1 version disque dur	52 900 F
Carte réseau	4 900 F
Disque dur externe 10 Méga	22 900 F
Processeur arithmétique 8087	6 900 F
Carte Z80	4 900 F
Carte 3278	8 900 F
Carte d'extension 128 Ko .	4 900 F
256 Ko .	7 500 F
384 Ko	10 000 F
Imprimante (160 cps) 180 s	9 400 F
Compilateur Basic	3 600 F
MS Cobol	6 900 F
MS Pascal	4 900 F
MS Fortran	3 900 F
Prologue	2 500 F
IEEE-488	1 500 F
Package graphique	2 500 F

FICHE SIGNALÉTIQUE

Le micro-ordinateur Victor S1 existe en trois versions.

Version simple face

- Microprocesseur 16-8 bits Intel 8088
- Mémoire RAM de 128 Ko (extensible à 896 Ko)
- Deux disquettes de 2x600 Ko
- Deux ports série V-24 (RS 232-C) synchrone et asynchrone
- Deux ports parallèles qui supportent toutes les imprimantes courantes et interfaces IEEE-488 (soft).
- Deux systèmes d'exploitation : CP/M-86 et MS/DOS
- Basic 86
- Ecran très haute résolution 800x400 points avec affichage de 25 lignes de 80 caractères ou 50 lignes de 132 caractères
- Clavier AZERTY français accentué.

Version double-face

- Capacité disquettes de 2x1,2 Mo
- Mémoire RAM de 256 Ko

Version disque dur intégré

- Disque dur Winchester 5 1/4"

de 10,6 Mo et disquette de 1,2 Mo
— Mémoire RAM de 256 Ko

Progiciels fournis par Victor Technologies

- Victorwriter et Textor : traitement de texte
- Super calc et Multiplan : aide à la décision
- D/Base II et D.M.S. : base de données
- Plusieurs outils d'aide au développement de programmes.

Logiciels proposés par SSCI/Revendeurs

- Comptabilité générale et analytique
- Paie, facturation, gestion de stocks
- C.A.O., D.A.O.
- Gestion de fichiers, traitement de textes
- Gestion de pharmacies, hôtels, chantiers, laboratoires d'analyses médicales, cabinets de médecins, cliniques, etc.

Options et périphériques

- Système d'exploitation: Prologue
- Langages : Fortran, Pascal, Cobol, Basic
- Communications : IBM 3278, DEC VT 100, VIP 7700 de Honeywell, série 4010 de Tektronix...
- Imprimante matricielle 160 cps 132 colonnes
- Imprimante de traitement de texte
- Réseau local type Omninet.

Maintenance

- Le service après-vente et le support technique sont assurés par un réseau national de revendeurs-conseils Victor Technologies.

Garantie : 1 an.

C.-H. Delaleu

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

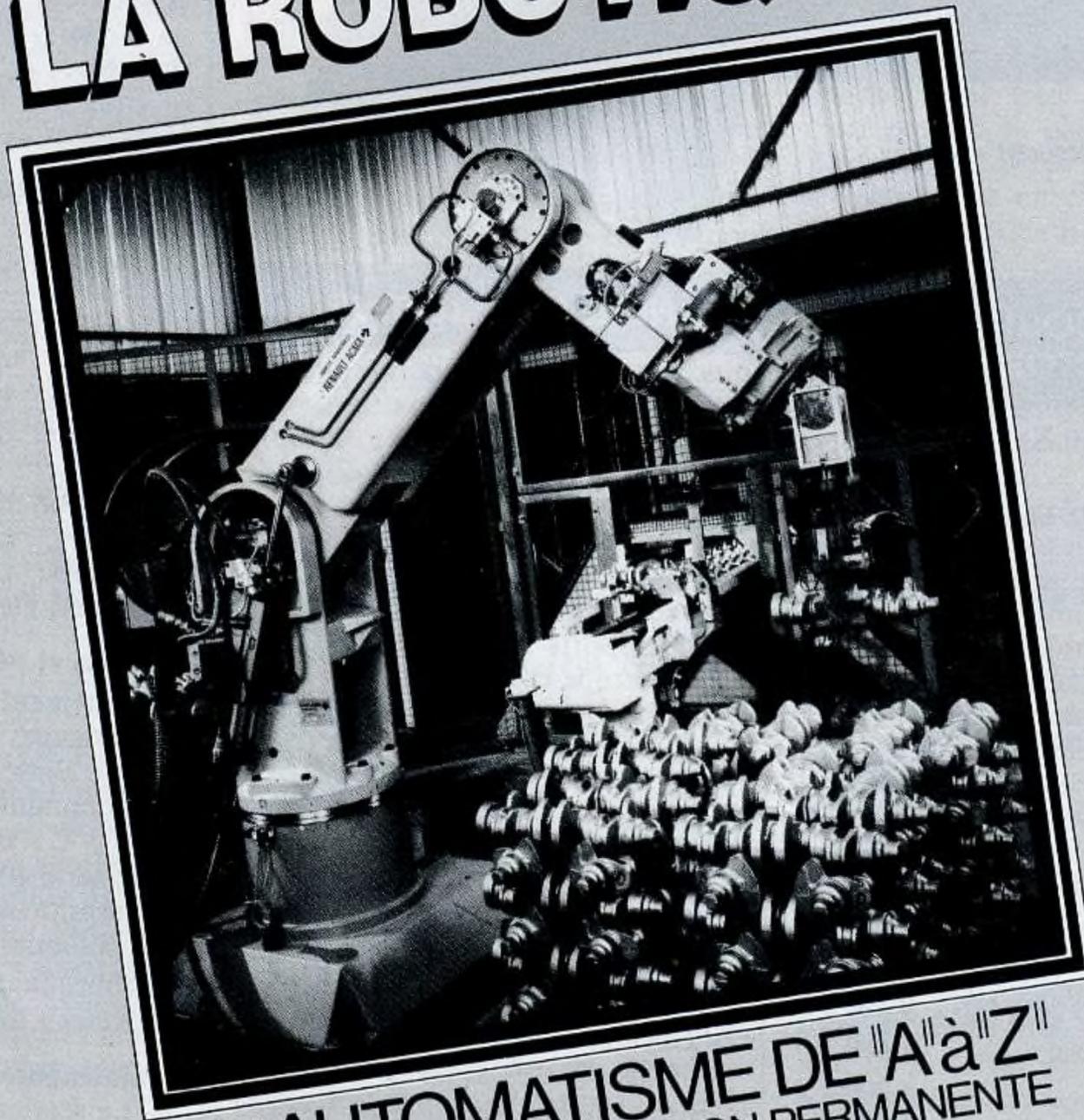
Vous étiez

hors série

L'ed

**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.

• NOUVEAUX PRODUITS •

DES JEUX INTELLIGENTS



Ediciel sort quatre nouveaux logiciels Spinnaker à but éducatif et pédagogique.

- Coccinelle : un jeu de dessin et d'apprentissage de l'informatique. Il permet, tout en réalisant des dessins réutilisables, d'appréhender la logique de la programmation. Ce jeu, déjà sorti sur Commodore 64, existe maintenant sur Apple II, au prix de 500 F TTC.

- Pacific 231 : un jeu de simulation d'une démarche similaire à Profession Détective et à Sur les traces du Deir-

dron. Ce jeu simule des situations et place le joueur en position d'apprendre et de réagir au programme. Comment gérer une entreprise de chemin de fer, utiliser tous les moyens dont on dispose pour la faire grandir, construire de nouvelles voies. Il est disponible sur Apple II, au prix de 400 F TTC.

- Portrait Robot : destiné aux enfants entre 4 et 10 ans, ce jeu fait appel à leur créativité. L'enfant doit imaginer un visage, le réaliser, l'animer, mémoriser les mimiques que le visage fait. Il est commercialisé sur Apple II 350 F TTC et sur Commodore 64 190 F TTC.

- Scenario : pour enfants un peu plus âgés que Portrait Robot (6 à 10 ans), Scenario utilise et développe la créativité et l'imagination des enfants. Construire une histoire en plusieurs tableaux, la réaliser en dessin, raconter l'histoire en texte nécessite de l'enfant d'imaginer et de réaliser successivement son histoire. Scenario existe sur Apple II au prix de 400 F TTC.

DES LOGICIELS POUR TOUS LES GOUTS

Sprites, une jeune société, créée par des jeunes, propose un catalogue de logiciels 100 % français, des jeux mais aussi des logiciels pour la vie quotidienne comme un carnet d'adresses, une gestion de stocks. Ces logiciels sont disponibles sur les micro-ordinateurs domestiques les plus répandus : Spectrum, Oric, Commodore 64. A signaler notamment Gra-

phisto MSV, un utilitaire pour micro au standard MSX, permettant de réaliser toutes sortes de graphismes en 16 couleurs, «tendre poulet» sur Oric 1/Atmos, «Le tour de France», une course de vélo sur Spectrum, Honey-Kong, un jeu d'action sur Oric 1/Atmos. Ces jeux sont commercialisés entre 120 et 150 F environ.

MINI

Peut-être connaissez-vous déjà Mini, mais ce conte pour enfants à partir de deux ans, créé par Anne Bergeron, est si joli qu'il mérite bien qu'on en reparle à nouveau. Ce conte interactif fait vivre aux enfants les aventures de la petite fourmi Mini et de ses amies les lucioles, les escargots, les papillons. A chaque lettre de l'alphabet correspond une scène animée et une pièce musicale issue du répertoire enfantin, folklorique, populaire ou musical. Mini est proposé sur disquette accompagné d'un manuel d'utilisation 390 F TTC. Il est disponible sur Commodore 64.



**+ de
1 500 termes !
Un premier lexique
anglais-français
vraiment pratique
et très complet.**

- Index français-anglais
- Lexique des termes anglais et américains avec explication en français.
- Tableau de conversion

Pour la première fois en électronique, un lexique anglais-français présenté sous forme pratique avec en plus des explications techniques succinctes mais précises.

**En vente chez votre libraire
et aux Editions Fréquences**

■ BON DE COMMANDE ■

Je désire recevoir le livre «le lexique de l'électronique anglais-français» au prix de 72 F (65 F + 7 F de port).
Adresser ce bon aux EDITIONS FREQUENCES 1, bd Ney, 75018 Paris.

Nom

Prénom

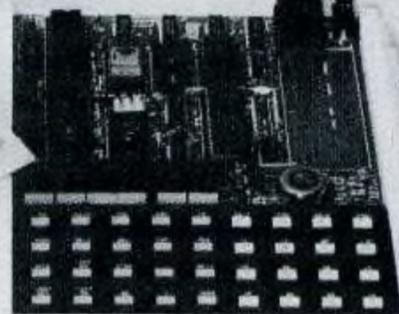
Adresse

Code postal

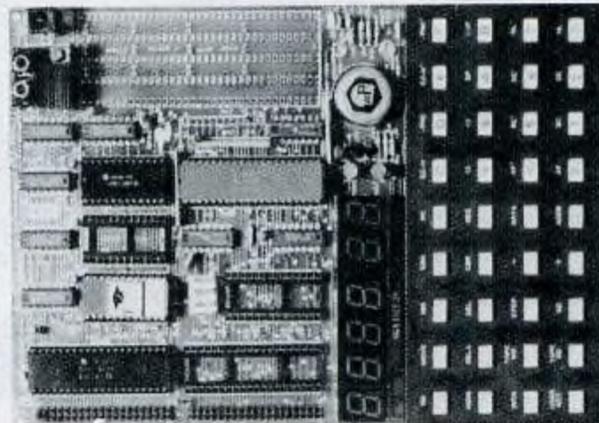
Règlement effectué

par CCP par chèque bancaire par mandat

**nouveau cours
par correspondance
avec micro-ordinateur.**



LES MICROPROCESSEURS



L'architecture du micro-ordinateur MPF 1.

Une formation professionnelle pour préparer l'avenir

Découvrez chez vous les secrets des microprocesseurs.

Ce cours vous permettra d'acquérir toutes les connaissances nécessaires à la compréhension du fonctionnement interne et à l'utilisation d'un micro-ordinateur.

Vous serez capable de rédiger des programmes en langage machine, de concevoir une structure complète de micro-ordinateurs autour d'un microprocesseur (8080 - Z 80).

Un micro-ordinateur chez vous.

Notre cours par correspondance est accompagné en option d'un micro-ordinateur MPF1, équipé d'un microprocesseur Z 80. Un manuel d'utilisation a été spécialement conçu pour vous permettre de réaliser au fur et à mesure de vos études les exerci-

ces pratiques qui viendront concrétiser ce que vous aurez appris.

Votre micro-ordinateur MPF 1 est équipé :

- d'un interface cassette,
- d'un synthétiseur,
- d'extensions mémoires,
- d'un emplacement prévu pour connecter vos circuits de commande,
- d'un transformateur d'alimentation 220 V - 9 V.

Vous n'êtes pas seul chez vous, à tout moment vous pouvez consulter votre professeur.

Notre cours par correspondance avec micro-ordinateur comprend plus de 300 pages illustrées de nombreux schémas, dessins, organigrammes. Elles sont présentées dans trois reliures de qualité, faciles à consulter

Ce cours permet de comprendre tranquillement le fonctionnement des microprocesseurs. Niveau conseillé : BAC.



IPIG

Pour la Suisse : JAFOR
16, avenue Wendt
1203 Genève

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

Envoyez-moi gratuitement et sans engagement de ma part votre documentation n° R3823 sur votre cours de microprocesseurs, micro-ordinateurs et vos cours d'informatique.

Nom _____
Prénom _____
Adresse _____

Code postal _____
Si l'Electronique vous intéresse, veuillez cocher cette case

Micro-informatique : architecture, interfaces et logiciel

J.D. Nicoud - Dunod

La littérature ayant trait à la micro-informatique est vaste et comprend de nombreux livres.

Destinés à un large public, tous ces livres se situent généralement à un niveau accessible à tous et font partie des «ouvrages de vulgarisation». Ce n'est pas du tout le cas de ce nouveau livre édité par Dunod qui est un cours approfondi de la micro-informatique rédigé pour tous les étudiants et techniciens qui désirent acquérir des bases solides en «software et hardware».

Les principales notions utilisées en informatique (matériel et logiciel) sont abordées dans cet ouvrage :

- représentation des nombres et opérations arithmétiques associées
- architecture des ordinateurs (U.A.L., mémoire entrées-sorties)
- langage assembleur (syntaxe et fonction des principales instructions)
- interfaces et périphériques (gestion par interruption, DMA...)
- microprocesseurs et système.

Chaque cours théorique est accompagné d'exercices résolus permettant d'illustrer les différents concepts techniques développés dans cet ouvrage.

Très «traditionnel», ce livre pourra sembler quelque peu abstrait à un lecteur non initié, par contre il devrait ravir tous les ingénieurs dont la fonction

est de concevoir des systèmes informatiques.

Forth pour Micros

Jean-Marie De Geeter - Ed. Eyrolles

Situé entre le Basic et l'assembleur, le langage Forth se fait

peu à peu une place au soleil. Presque tous les micro-ordinateurs à l'heure actuelle disposent de Forth : Jupiter Ace (resident), Apple (disquette), ZX 81 (cassette)...

Les qualités de Forth sont indéniables (rapidité d'exécution, compacité, extensibilité, transportabilité) et ses domaines

d'utilisation très variés (automatisme, jeux...). Enfin, d'un point de vue pédagogique, Forth est un excellent outil pour s'initier à la programmation.

Ce livre de J.M. Degeeter est divisé en deux parties : dans une première partie, l'auteur développe les notions de base de Forth. Les principaux mots du vocabulaire Forth (on parle aussi de dictionnaire) sont analysés : fonction et syntaxe. A partir de ces mots de base l'auteur donne des exemples qui permettent «d'étendre» ce dictionnaire.

Comme les calculatrices Hewlett-Packard, Forth utilise une pile et la notation polonaise inversée pour la manipulation des données, de nombreux exercices permettent de se familiariser avec cette technique. La seconde partie de ce livre contient une bibliothèque de programmes (Othello, Gestion de compte bancaire, biorythme...) qui reprend toutes les notions développées dans la première partie.

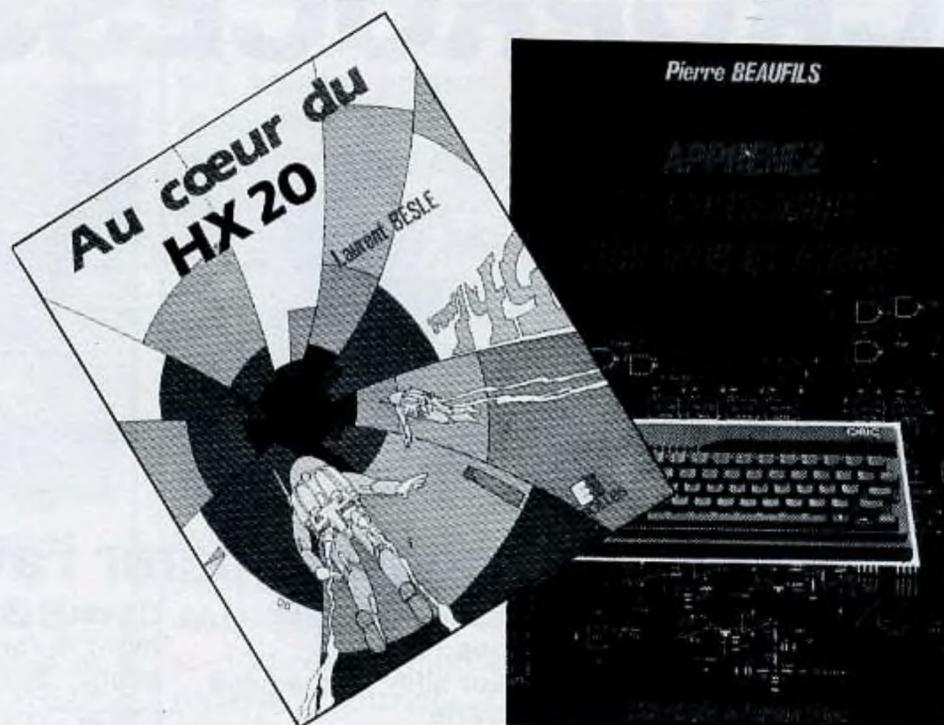
A noter chez le même éditeur un livre traitant du Forth sur ZX 81.

«Introduction au ZX Forth» de M. Petreman et M. Rousseau.

Musique sur Commodore 64

J. Vogel, N.B. Scrimshaw - Cedic/Nathan

Les générateurs de son sont devenus un sous-ensemble indispensable sur les micro-ordinateurs de la nouvelle géné-



ration. Utilisés généralement pour sonoriser les jeux vidéo, ils peuvent aussi transformer votre micro-ordinateur en un véritable instrument de musique. C'est le but que s'est fixé ce nouveau livre Cedic/Nathan avec comme matériel le Commodore 64 et son processeur musical.

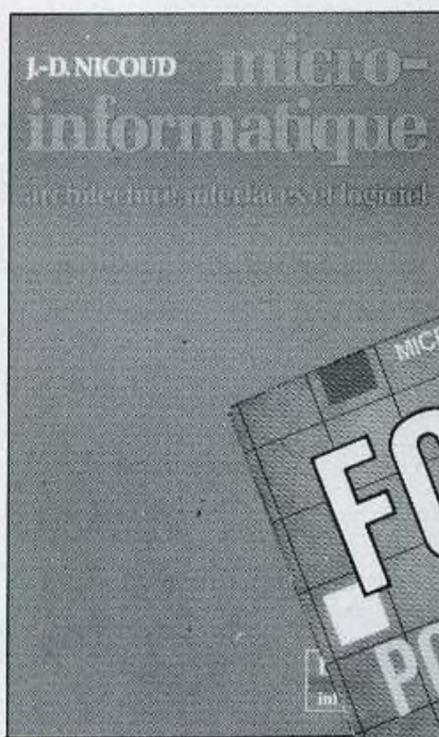
Au niveau son, les principales notions concernant la synthèse de fréquence sont abordées : générateur d'enveloppe, filtrage, effet sonore... En ce qui concerne le Basic, peu d'instructions (READ, DATA, POKE...) sont utilisées et il n'est pas nécessaire d'être un programmeur expérimenté pour devenir un musicien chevronné.

Ce livre contient de nombreux exemples qui devraient permettre à tous les possesseurs de Commodore 64 de débiter ou compléter une bibliothèque sonore.

Apprenez l'électronique sur Oric et Atmos

Pierre Beaufile - Ed. Soracom informatique

La simulation des circuits électroniques sur ordinateur (programme SPICE par exemple) est couramment pratiquée dans l'industrie. Généralement utilisés comme aide à la conception, ces programmes sont un outil indispensable aux électroniciens. Ce nouveau livre de P. Beaufile est une introduction à cette technique de simulation



et devrait combler de nombreux électroniciens amateurs.

Chaque chapitre de cet ouvrage traite d'un sujet d'électronique différent : circuit résonnant, filtre, série de Fourier, régime transitoire, circuit non linéaire, abaque de Smith, boucle à verrouillage de phase. Tous les sujets sont tout d'abord traités d'un point de vue théorique (principe, équations...) et ensuite analysés à l'aide d'un programme écrit en Basic. Chaque programme est accompagné d'un mode d'emploi et les principales instructions utilisées

sont expliquées. Tous les programmes font appel au graphisme (quoi de plus parlant qu'une courbe ?) et aux deux instructions de l'Oric CURSET et DRAW.

Au cœur du HX20

Laurent Besle - Ed. Eyrolles

Le HX20 est un micro-ordinateur proposé par Epson et qu'on range généralement dans la catégorie des portables.

Il intègre dans un même boîtier un affichage à cristaux liquides et une mini-imprimante à aiguilles. Conçu suivant une architecture originale (2 micro-processeurs travaillent en parallèle suivant le principe «Maître-Esclave»), ce micro-ordinateur utilise des composants «maison» qui ont été développés pour cette application.

C'est le cas des microprocesseurs (référence 6301 technologie CMOS : consommation oblige) qui sont des dérivés du 6800 de Motorola. La première partie de ce livre traite donc du jeu d'instructions du 6301. Les techniques de programmation en langage machine sont décrites accompagnées par de nombreux exemples. Afin de simplifier la programmation en langage machine (code hexadécimal !) on utilise généralement le langage assembleur qui associe à chaque code machine un mnémonique. Le listing complet, accompagné de nombreux commentaires, d'un assembleur désassembleur suit la description du 6301.

Toute la seconde partie de ce livre est consacrée à l'utilisation du HX20. Des exemples de programmes utilisant des routines contenues en ROM sont décrits. Enfin de nombreux utilitaires (chronomètre, utilisation de la mémoire, protection du logiciel) sont présentés à la fin de cet ouvrage. Très bien conçu, ce livre est un excellent complément aux notices techniques données par le constructeur.

P.F.

P.A. GRATUITES

Vds Canon X-07 12 K, cordon magnéto cassette, livres jeux sur Canon et faites vos jeux sur Canon : 2 000 F et à vendre interface couleur Péritel Canon + alim : 2 000 F ou le tout à 4 000 F. Moniteur couleur Thomson : 2 500 F. Mr Guillemot Daniel 15 rue la Bruyère 93800 Epinay s/seine. Tél : 412.16.33.

Vds cause chômage Oric 1 64 K + alim + câbles + nombreux livres + cassettes + doc : 2 200 F. Tél : (40) 20.56.35.

Vds TRS-80 modèle I + minuscules + écran vert (fin 81) + extension Tandy 48 Ko + lecteur de disquettes 40 pistes ss Newdos/80 (4/83), le tout 7 000 F avec nbreux programmes. Imprimante 4 couleurs CGP-115 : 1 200 F. Lebreton 33360 Cénac. Tél : (56) 20.68.01.

Vds Lynk 96 Kp, prise Péritel, 3 cassettes jeux + programmes interface Joystick. Prix entre 3 000 et 3 500 F. Tél : 868.70.65 après 20 h 30.

Vds Oric 1 48 K + alim + CGV + magnéto Aquarius + 5 K7 jeux + 1 K7 assembleur + 10 livres Oric. Valeur 4 000 F. Tél : 063.77.20.

Vds Atari 800 2-82 + drive + interface RS 232 + GP 100 : 7 650 F, touch tablet + 2 sticks : 800 F, log. cartouche + disk : 1 500 F, 7 livres + doc : 1 000 F. Gallaire JP 16 rue A. Renoir 78250 Meulan. Tél : 474.12.29.

Club Ronchin (Nord) recherche micros ZX81 et accessoires. Télé portable n & b même en panne. Tél : (20) 88.06.62 après 18 h.

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.

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. 2-40-60-61-62-67 - EMIA, p. 4 - HBN, p. 1 - IPIG, p. 63 - UNIECO, p. 55 - Victor Technologie, p. 47-68

**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

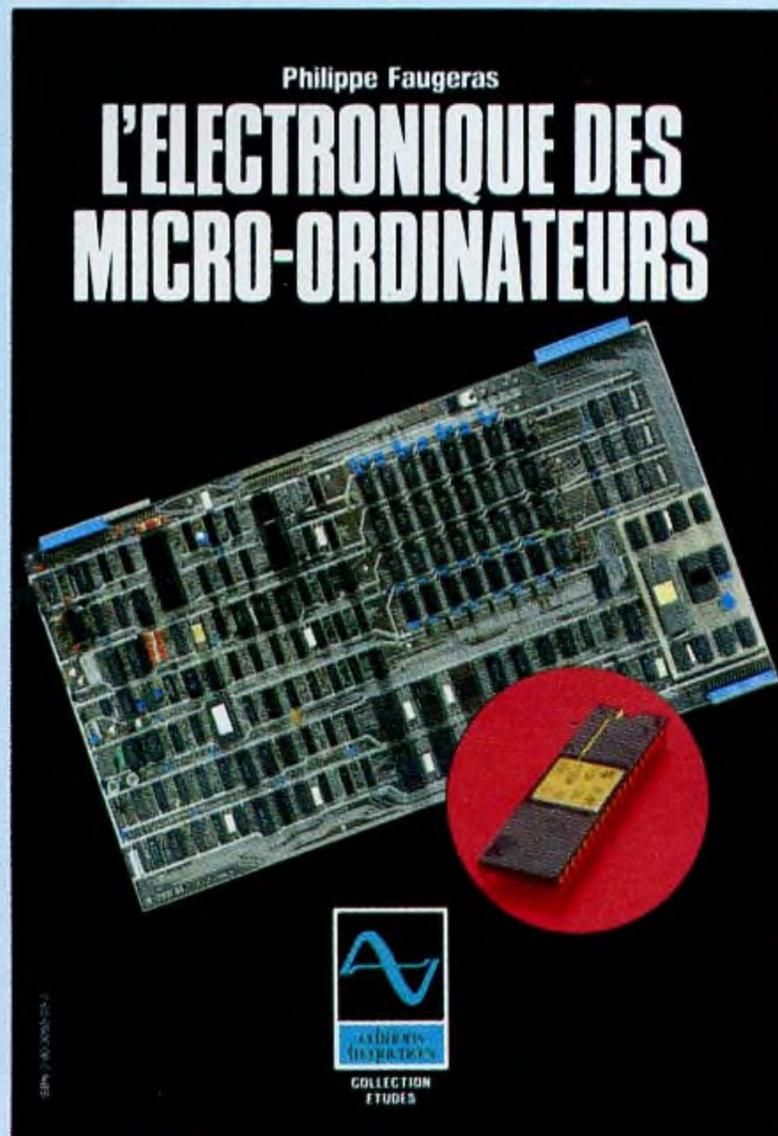
Ville Code Postal

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

VOYAGE AU CŒUR DES MICRO-ORDINATEURS

dans la
COLLECTION
«ETUDES»
aux
éditions
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

En vente chez votre libraire et aux Editions Fréquences

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**» au prix de 165 F (port compris).

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.

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