

LO... JOURN...
hors série

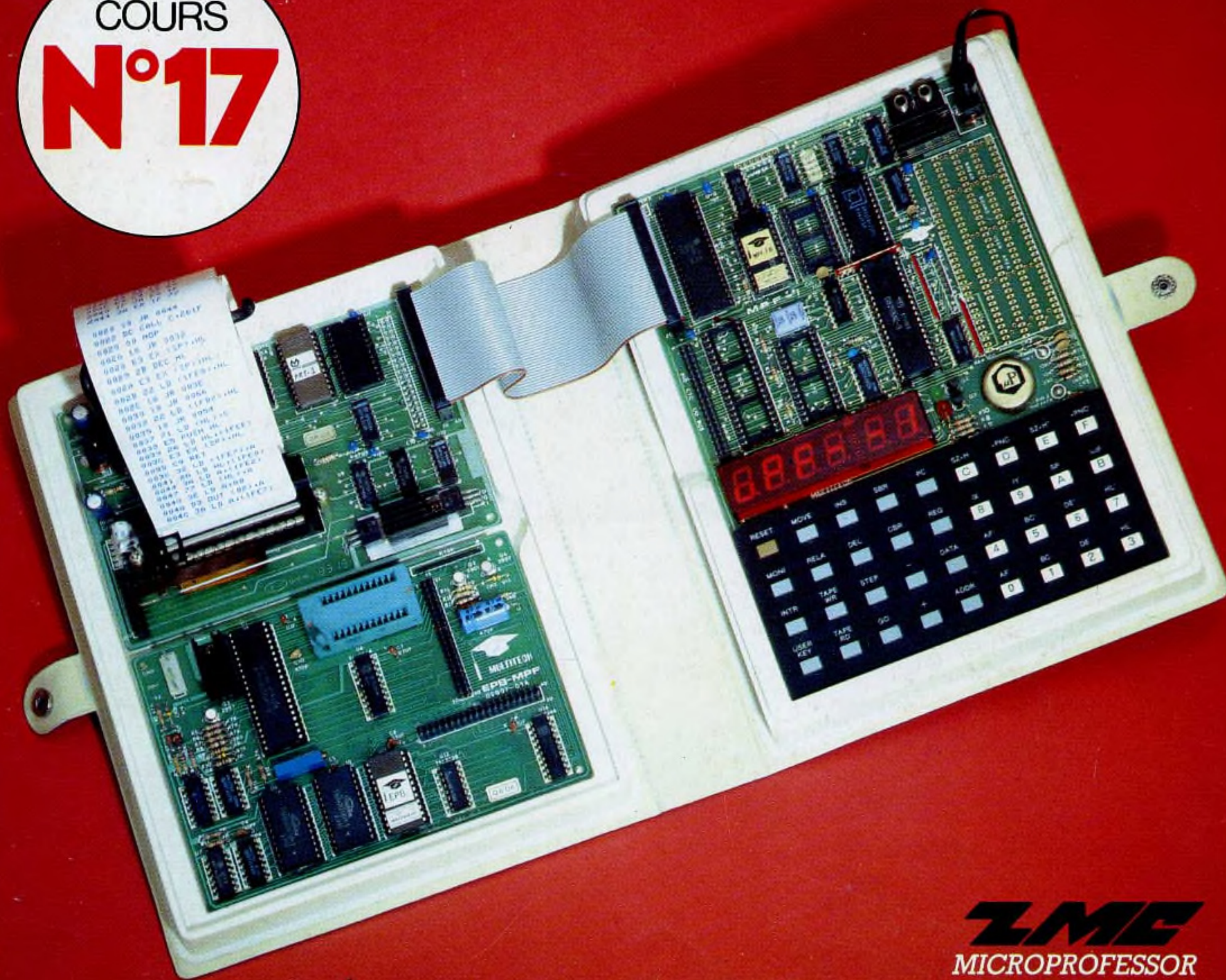
Led

MICRO

APPRENDRE • La programmation, cours de Claude Polgar • Le Microprofessor MPF 1B, cours pratique de microprocesseur par Philippe Duquesne. • Les fortiches : gérants de formulaire. MAGAZINE • L'avenir aux portables • La contre-mesure du Guépard • Le progiciel Context MBA à l'essai.

DE L'INITIATION A LA PRATIQUE DE L'INFORMATIQUE

COURS
N°17

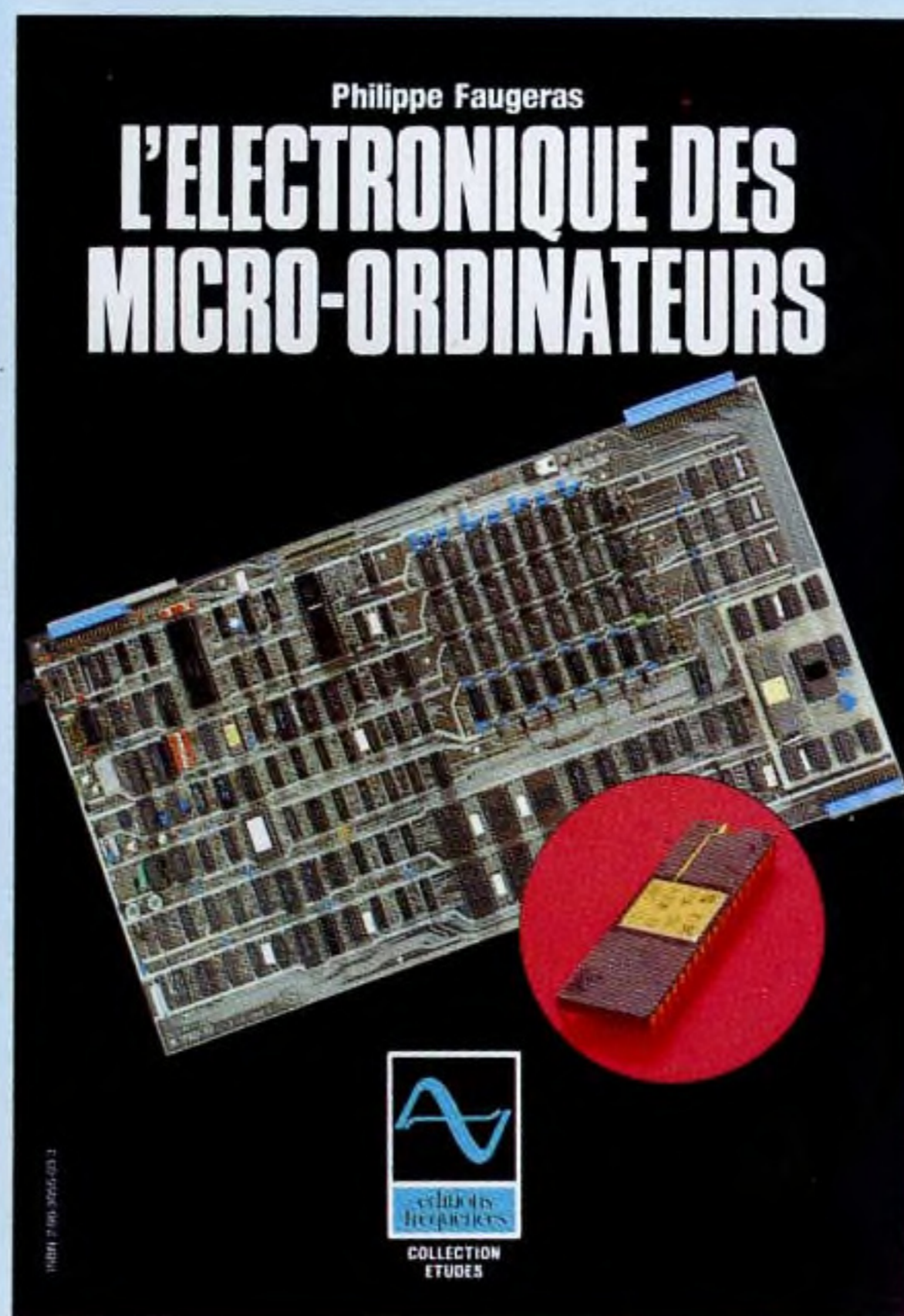


ISSN 0757-6889

ZMC
MICROPROFESSOR

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.

hors série

LED

MICRO

FEVRIER 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

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, Jean Yrytov
Rubriques Magazine
Chef de rubrique :
 Claude-Hélène Roze
 Ont collaboré à ce numéro :
 Charles-Henry Delaleu, Philippe
 Faugeras

Secrétaire de Rédaction
 Chantal Cauchois

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
 Impression
 Berger-Levrault - Nancy

RUBRIQUES PEDAGOGIQUES



5

COURS DE PROGRAMMATION EN BASIC

Initiation progressive à l'informatique

par **Claude Polgar**

26

LE COURRIER DES LECTEURS

Claude Polgar et Bruno Lilamand répondent

28

COURS PRATIQUE DE MICRO-PROCESSEUR

avec le Microprofessor MPF 1B

par **Philippe Duquesne**

38

LE COIN DES FORTICHES

Aide à la conception des graphiques

par **Jean Yrytov**

Gérant de formulaire sur Goupil

par **Bruno Lilamand**

48

LIBRES PROPOS

Réflexions sur la micro-informatique

RUBRIQUES MAGAZINE

50

L'avenir aux portables

54

LES CONTRE-MESURES

Le Guépard



58

PROGICIELS A L'ESSAI

Le Context MBA

62

LES INFOS ET LES PRODUITS

64

BIBLIOGRAPHIE

A lire

NOTRE COUVERTURE : Le MPF 1B de ZMC, de l'initiation à la pratique du micro-processeur.

Vous découvrez Led-Micro avec ce n°17
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 et la reliure
180 F
PORT COMPRIS

SOMMAIRE DES COURS

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

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

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

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

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

N°6 Premier programme en Basic - Ponctuation dans le Print • 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

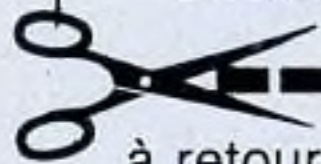
N°13 L'affectation (fin) : INPUT - La sélection (1^{re} partie : Sélection simple).

N°14 La sélection - Le langage Z80^R (suite).

N°15 Complément sur les chaînes de caractères - Le langage du Z80^R (suite).

N°16 La sélection multiple - Les modes d'adressage.

Vous désirez un ou plusieurs numéros qui vous manquent (de 1 à 16) : 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 **et la reliure**

Je désire le n°

(cocher le ou les n°s désirés)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

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

Nom : prénom :

Adresse :

Ville Code postal

COURS DE PROGRAMMATION(17)

COURS ET EXERCICES

Vous vous souvenez de la distinction que nous avons faite entre :

- les exercices d'application immédiate, que vous devez effectuer en lisant le cours et dont nous vous donnons aussitôt la solution ;
- les exercices de récapitulation qui sont parfois plus complets et dont nous vous demandons de bien vouloir nous envoyer votre solution.

Depuis le numéro 15, vous avez atteint un niveau de connaissances vous permettant d'aborder des exercices de récapitulation intéressants. De ce fait, la partie «Corrigé des exercices de récapitulation» commence à prendre de l'ampleur. J'ai pensé qu'il serait plus clair de séparer désormais mon cours en deux parties :

- une partie «cours proprement dit», comportant le cours et ses «exercices d'application immédiate» ;
- une partie «exercices et traductions» dans laquelle seraient réunis les exercices de récapitulation et leurs corrigés.

QUAND LES ELÈVES SONT TROP FORTS

Lorsque j'ai demandé aux lecteurs de m'envoyer leurs solutions à mes exercices de récapitulation, j'avais pour seul objectif de vérifier que mon cours était bien assimilé (afin, éventuellement, de revenir sur tel ou tel sujet).

Il me semble que j'ai «mis à côté de la plaque» :

La quasi-totalité des solutions que je reçois sont des «sans-faute» alors que dans mes cours oraux, j'obtenais des résultats beaucoup moins satisfaisants. Les lecteurs de LED MICRO seraient-ils tous des génies ? Je crois que, en fait, seuls m'écrivent les «élèves» qui sont sûrs d'eux... et en particulier ceux qui ont déjà acquis une bonne teinture de BASIC.

D'autre part, ces élèves «en avance» m'envoient quelquefois des solutions meilleures que celles auxquelles j'avais pensé. Bien sûr, en publiant ces solutions, je perdrais mon prestige (?) mais nous gagnerons tous grâce à cette discussion.

EXERCICES ET TRADUCTIONS

Il faut s'adapter aux circonstances. Je prévois maintenant que la deuxième partie de ce cours (que je vais appeler «Exercices et Traductions») ne sera pas consacrée uniquement à proposer des exercices et à rédiger leurs solutions. Elle s'efforcera d'être un complément au cours sous tous les aspects qui s'avèreront utiles :

- d'abord, comme prévu, les exercices de récapitulation (en principe le mois N : le sujet de l'exercice ; le mois N + 1, le corrigé que je propose ; le mois N + 2, les éventuelles solutions différentes proposées par les lecteurs) ;
- ensuite des traductions de certains exercices d'application sur des micros ayant un BASIC différent du BASIC Microsoft de l'IBM-PC (Sinclair, Commodore, Apple...);
- puis des échanges d'idées diverses à propos du cours ou des exercices.

SOMMAIRE DES N^{OS} PASSES

Chapitres		LM n ^{os}
1	Introduction à l'informatique	1
2.1 à 2.10	Notion de matériel - U.C. et périphériques	2 et 3
2.11 à 2.14	Notions générales sur le logiciel Le choix d'un ordinateur	4
3.1 à 3.5	De l'installation au caractère d'attente	5
3.6 à 3.10	Structure du BASIC - Le PRINT	6

Chapitres		LM n ^{os}
3.11	Nombres et calculs	7 et 8
3.12 3.13	Notions élémentaires sur les fichiers et les SED	10 à 12
3.14 3.15	L'affectation - Notion de variables	12 à 14
3.16	La sélection	14 à 16
3.17	Les boucles	17 à...

3.17. La structure "boucle"

COURS

3.17.1. Note au lecteur

La structure de boucle est à la fois l'une des plus importantes et l'une des plus difficiles à maîtriser par les programmeurs débutants. Pour que vous assimiliez parfaitement cette notion, nous procéderons très lentement en vous faisant faire de nombreux exercices d'application. Même si ces exercices vous semblent trop faciles, rédigez chacun des programmes que nous vous proposons et comparez votre solution à la nôtre.

Donnons un exemple. Un grand nombre de cours de BASIC présentent au lecteur le calcul du nombre d'or. Et le lecteur comprend tout de suite. Mais prenons le problème à l'envers : demandez à un débutant de rédiger lui-même ce programme. Ce n'est plus du tout aussi facile ! Nous considérerons que vous aurez compris un programme lorsque vous serez capable de le rédiger vous-même.

Donc : ne vous contentez pas de lire les solutions. Faites réellement les exercices. Même s'ils vous semblent trop faciles.

3.17.2. L'instruction IF... THEN... dans les boucles

Chaque fois qu'un programme fait recommencer un travail jusqu'à ce qu'on ait obtenu un certain résultat, il construit une boucle.

Nous avons déjà rencontré des boucles dans nos exercices :

- les contrôles de vraisemblance (on demande à l'opérateur de recommencer jusqu'à ce que sa réponse soit acceptable) ;
- le cheval errant (on fait faire un autre saut au cheval jusqu'à ce qu'il ait atteint la case d'arrivée).

Dans le présent chapitre 3.17, nous allons dépasser les exemples de création «instinctive» de boucle pour aborder l'étude **systematique** de cette notion. «Etude» est d'ailleurs un bien grand mot : nous ne ferons aucune découverte théorique ; mais nous nous entraînerons à faire beaucoup d'exercices pratiques. Les classifications que nous vous proposerons n'ont pas d'autre but. Elles n'ont aucune prétention «scientifique».

3.17.3. Un programme pas très astucieux

Regardez le programme SUITE 18 (figure 4 page ci-contre). Il a pour effet d'afficher la suite des nombres impairs compris entre 9 et 23 dans des colonnes successives : voir son exécution figure 5.

Ce n'est pas une solution très astucieuse. Nous allons obtenir le même résultat de différentes façons (plus élégantes).

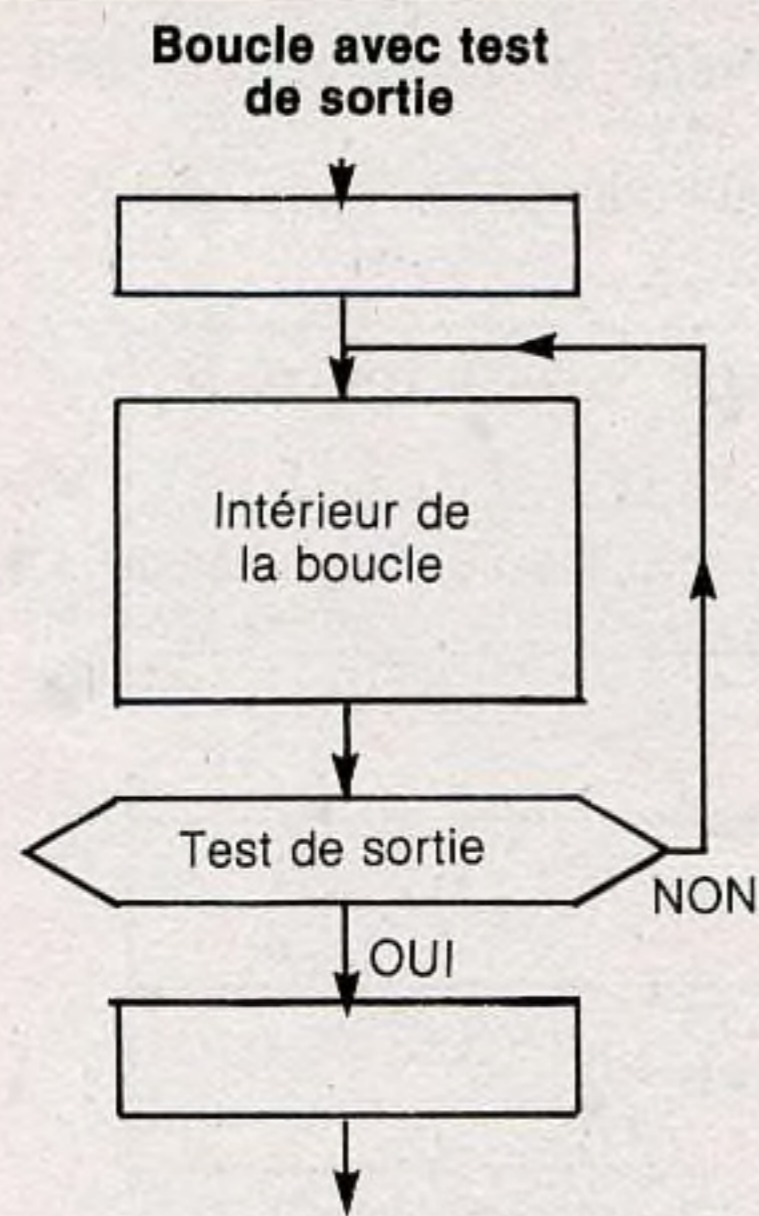


Figure 1

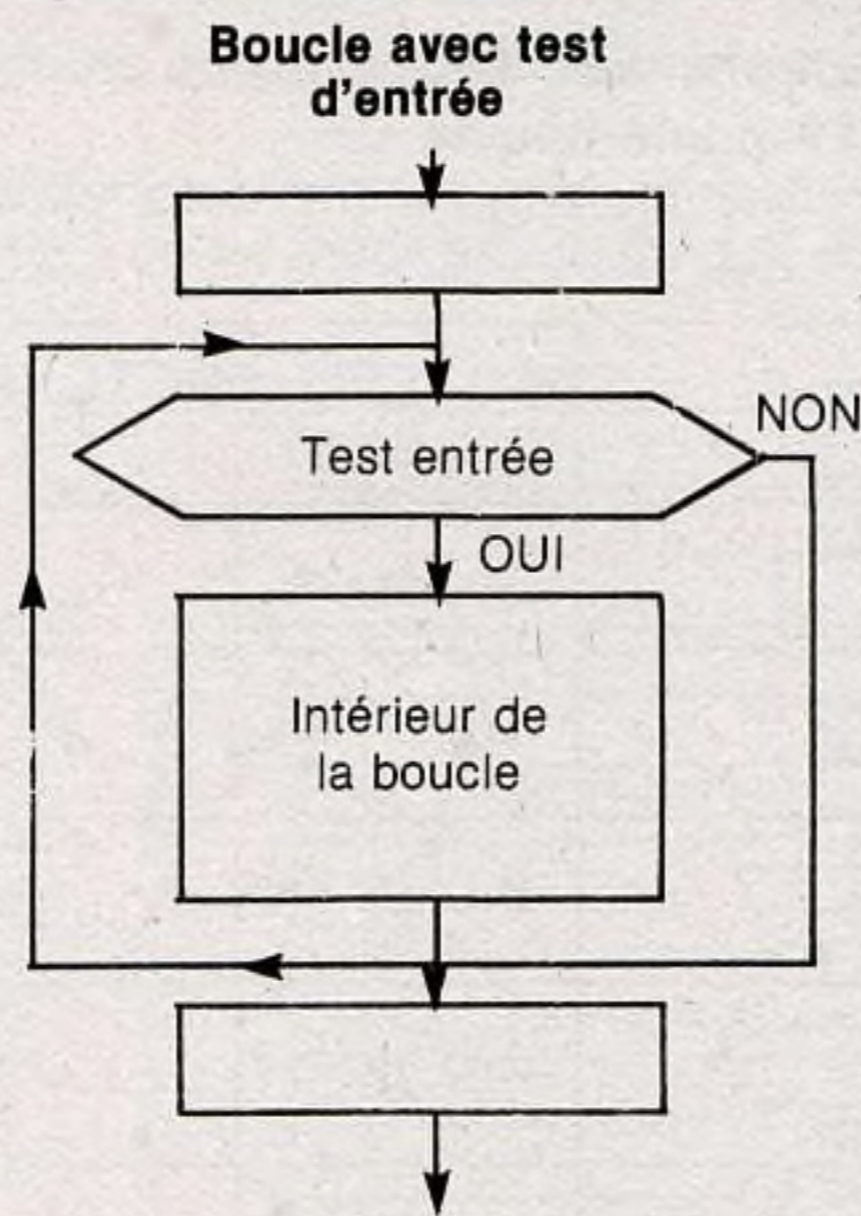


Figure 2

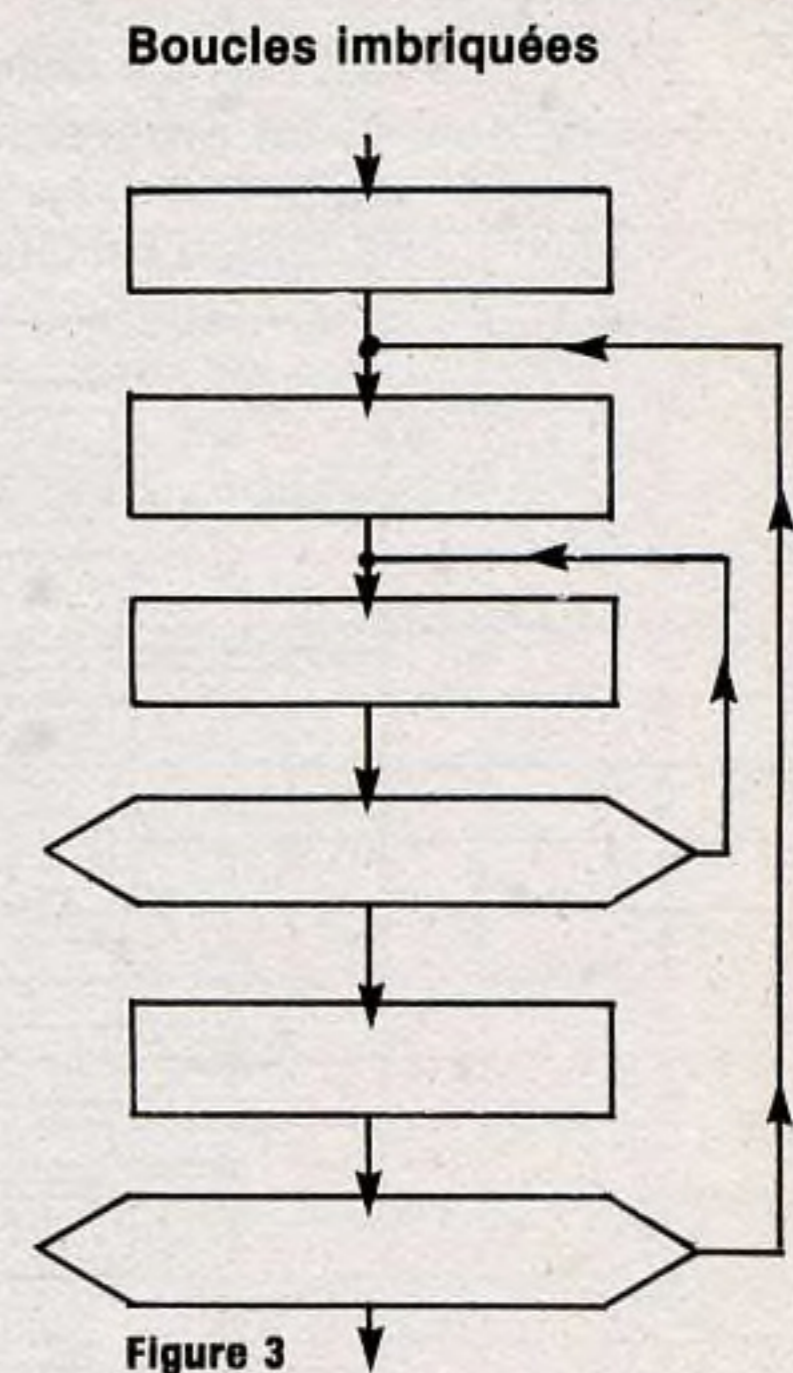


Figure 3

```

LIST
10 REM SUITE18 SUR CP-DK05-PAP
20 CLS
30 PRINT 9,
40 PRINT 11,
50 PRINT 13,
60 PRINT 15,
70 PRINT 17,
80 PRINT 19,
90 PRINT 21,
100 PRINT 23,
110 END
OK ⌘

```

Figure 4

```

9          11          13          15          17
19
OK ⌘

```

Figure 5

Bien sûr, on aurait pu écrire :
 30 PRINT "9",
 40 PRINT "11",
 en considérant ces nombres comme des chaînes de caractères.

Note aux enseignants

Nous nous permettons de conseiller aux enseignants qui utilisent LED-MICRO comme base de leur cours :

- 1°) de ne pas modifier l'ordre des exercices que nous proposons, en particulier :
 - de **ne pas** commencer par donner comme exercice de départ l'affichage de la suite des nombres entiers : cet exercice, apparemment plus simple que notre programme SUITE 19, évite les confusions commises par certains débutants entre la valeur de la variable interne à la boucle et la valeur du «compteur de boucle».
 - de ne pas introduire les instructions FOR... TO et NEXT avant que les élèves aient acquis une certaine aisance dans l'emploi des IF... THEN ;
- 2°) de faire faire de nombreux travaux pratiques en faisant suivre l'exécution avec des TRON (ou des TRACE) en laissant aux élèves le soin de rechercher leurs erreurs.

3.17.4. Le déroulement d'un programme

A. Un programme avec une boucle

Le programme ci-dessous et l'organigramme correspondant donnent à l'exécution le même résultat que le programme «bêbête» du §3.17.3.

	<pre> 10 REM SUITE19 SUR CP-DK05-PAP 20 CLS 30 A = 7 40 A = A + 2 50 PRINT A, 60 IF A < 23 THEN 40 70 END </pre>
--	---

B. Qu'est-ce que «dérouler» un programme ?

«Dérouler un programme», c'est suivre son exécution pas à pas «à la main» en se mettant à la place de l'ordinateur.

Nous allons commencer par «dérouler le programme» ci-dessus.

		Ce que l'ordinateur lit :	Ce qu'il fait :
	1 ^{er} temps	10 REM	Une REM ? Rien à faire
	2 ^e temps	20 CLS	Effacer l'écran
	3 ^e temps	30 A = 7	Affecte la valeur 7 à la variable A
1 ^{er} passage dans la boucle	4 ^e temps	40 A = A + 2	Le contenu de la variable A devient 7 + 2 soit 9
	5 ^e temps	50 PRINT A	L'ordinateur affiche sur l'écran la valeur actuelle de A soit 9
	6 ^e temps	60 IF A < 23 THEN 40	L'ordinateur compare la valeur de A au nombre 23. Comme la valeur de A est 9, il va se brancher à la ligne 40
2 ^e passage dans la boucle	7 ^e temps	40 A = A + 2	Le contenu de la variable A devient 9 + 2 soit 11
	8 ^e temps	50 PRINT A	L'ordinateur affiche la valeur actuelle de A, qui est maintenant 11
	9 ^e temps	60 IF A < 23 THEN 40	L'ordinateur compare la valeur de A au nombre 23. Comme 11 est inférieur à 23, il va se brancher à la ligne 40
3 ^e passage dans la boucle	10 ^e temps	40 A = A + 2	Le contenu de la variable A devient 13
	etc...	etc...
Dernier passage dans la boucle	25 ^e temps	40 A = A + 2	A qui valait 21 devient égal à 23
	26 ^e temps	50 PRINT A	L'ordinateur affiche le nombre 23
	27 ^e temps	60 IF A < 23 THEN 40	Maintenant A = 23, donc la condition A < 23 est fautive et l'ordinateur, au lieu d'aller se brancher en 40 ira à l'instruction suivante.
	28 ^e temps	70 END	Fini ! Ouf ! Repos.

Deux représentations un peu moins «bavardes» du même traitement

Le tableau ci-contre représente le déroulement du programme d'une façon un peu moins «bavarde».

Instruction en cours d'exécution :	Valeur à ce moment :	
	du nombre de passages dans la boucle	de la variable A
10 REM		0
20 CLS		0
30 A = 7	1	7
40 A = A + 2	1	9
50 PRINT A	1	9
60 IF A 23 THEN 40	1	9
40 A = A + 2	2	11
50 PRINT A	2	11
60 IF A 23 THEN 40	2	11
40 A = A + 2	3	13
50 PRINT A	3	13
60 IF A 23 THEN 40	3	13
40 A = A + 2	4	15
etc.		

Fig. 1

La figure 2 (ci-dessous) est une autre représentation du déroulement du même programme

	Valeurs successives de A							
10 REM	0							
20 CLS	0							
30 A = 7	7							
40 A = A + 2	9	11	13	15	17	19	21	23
50 PRINT A	9	11	13	15	17	19	21	23
60 IF A 23 THEN 40	9	11	13	15	17	19	21	23
70 END								23

Fig. 2

D. TRON et TROFF - TRACE ET NOTRACE

Nous étudierons bientôt (4^e partie de ce cours) divers procédés permettant d'une part de «structurer» et de «mettre au point» les programmes.

L'un des procédés de mise au point consiste à utiliser la commande TRON (abréviation de TRACE ON) pour vérifier qu'un programme se déroule bien comme on l'avait prévu. Lorsqu'on frappe successivement les deux commandes TRON puis RUN, l'ordinateur lance l'exécution du programme (c'est l'effet du RUN) mais simultanément affiche le numéro des lignes par lesquelles il passe.

Nous reviendrons sur l'emploi du TRON tout au long de la quatrième partie de ce cours. Disons simplement que — TRON est efficace jusqu'à ce qu'on frappe la commande TROFF (= TRACE OFF = supprimer TRACE)

— TRON et TROFF peuvent s'utiliser aussi bien à l'intérieur d'un programme (comme instruction) que comme commande

— l'Apple II utilise une syntaxe un peu différente : au lieu de TRON il utilise TRACE, au lieu de TROFF il utilise NO TRACE et au lieu des crochets, il utilise le #.

3.17.5. Exercice d'application A51

Enoncé

En vous inspirant directement du programme SUITE 19, rédigez un programme qui affiche la suite des nombres entiers de 1 à 10.

Solution

	10 REM SUITE22 SUR CP-DK05-PAP
	20 CLS
	30 I = 0
	40 I = I + 1
	50 PRINT I,
	60 IF I < 10 THEN 40
	70 END

Remarque

A la ligne 10 de ce programme, on met la variable I à 0. On pourrait très bien se passer de cette ligne puisque, chaque fois qu'on fait RUN, toutes les variables sont remises à 0.

C'est cependant une bonne habitude à prendre : le BASIC ne connaît pas des variables « locales ». Si, dans un long programme on a déjà donné une certaine valeur à cette variable, elle garde cette valeur.

3.17.6. Exercice d'application A52

Enoncé

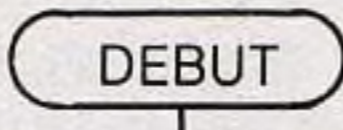
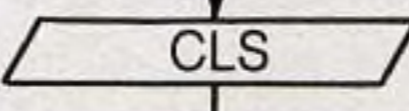
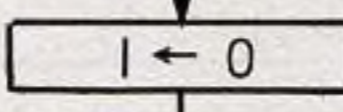
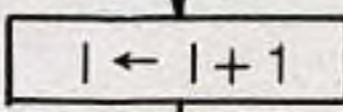
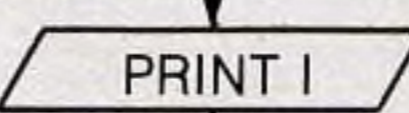
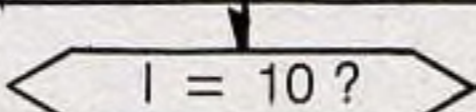
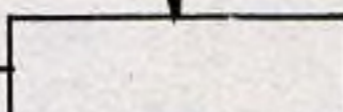
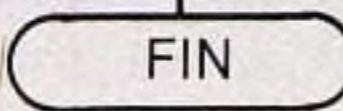
Rédigez un programme qui :

- 1°) fournit à l'exécution exactement le même résultat que le programme précédent
- 2°) conserve telles quelles les instructions 10, 20, 30, 40, 50 et 70 de ce programme
- 3°) remplace l'instruction 60 par un IF... THEN (sans ELSE) mais en utilisant le symbole =

Solution

Voir page ci-contre.

Solution de A52

	10 REM SUITE25 SUR CP-DK05-PAP
	20 CLS
	30 I = 0
	40 I = I + 1
	50 PRINT I,
	60 IF I = 10 THEN 70
	65 GOTO 40
	70 END

3.17.7. Exercice d'application A53

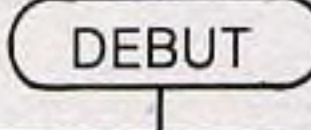
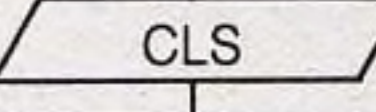
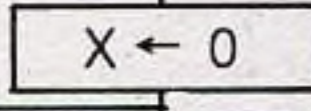
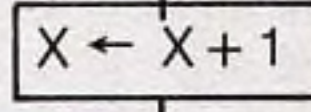
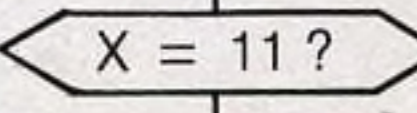
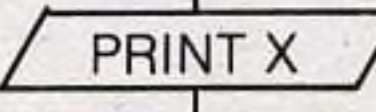
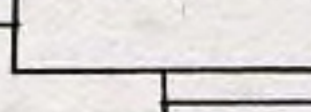
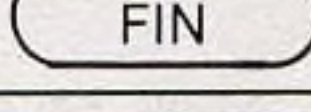
Enoncé

Nous supposons que dans votre réponse à l'exercice A51, vous avez utilisé une boucle avec un «test en queue» (organigramme figure 1 §3.17.2).

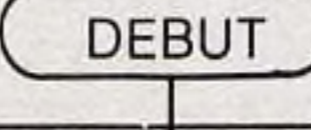
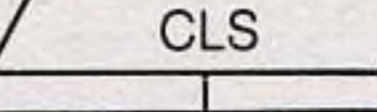
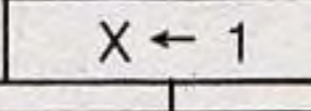

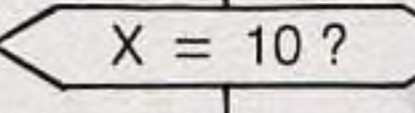
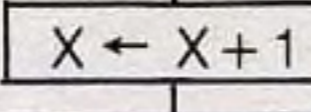
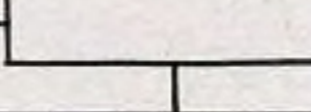
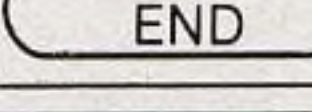
Rédigez maintenant un programme qui produit le même effet (afficher la suite des dix premiers nombres dans des colonnes) mais en utilisant une boucle avec un «test en tête» (organigramme figure 2 §3.17.2).

Pour donner un peu de piquant, nous vous interdisons d'utiliser les symboles $<$ ou $>$. (N'utilisez que l'égalité). Nous vous interdisons également d'utiliser le ELSE. Et (en plus !) nous vous demandons de trouver deux solutions.

Une solution

	10 REM SUITE28
	20 CLS
	30 X = 0
	40 X = X + 1
	50 IF X = 11 THEN 80
	60 PRINT X,
	70 GOTO 40
	80 END

Une autre solution

	10 REM SUITE29
	20 CLS
	30 X = 1
	40 PRINT X,
	50 IF X = 10 THEN 80
	60 X = X + 1
	70 GOTO 40
	80 END

3.17.8. Exercice d'application A54 : l'alphabet

Enoncé

Ecrivez un programme qui affiche la suite des 26 lettres majuscules sur l'écran. Cette suite de lettres devra apparaître «en un seul bloc» (et non disposées dans des colonnes). Vous devrez utiliser l'instruction CHR\$(X) une seule fois (en la plaçant dans une boucle).

Nous vous rappelons que :

Lettre	A	B	C	X	Y	Z
Code ASCII (en décimal)	65	66	67	89	90	91

Mais oui, c'est très facile ! Rédigez quand même votre programme avant de regarder les solutions que nous vous proposons ci-dessous.

Une solution

```
10 REM ALPHAB21
20 CLS
30 I = 0
40 I = I + 1
50 PRINT CHR$(64+I);
60 IF I < 26 THEN 40
70 END
80 END
```

ABCDEFGHIJKLMN OPQRSTUVWXYZ
OK ☞

Une autre solution

```
10 REM ALPHAB22
20 CLS
30 I = 64
40 I = I + 1
50 PRINT CHR$(I);
60 IF I = 90 THEN 70 ELSE 40
70 END
80 END
```

ABCDEFGHIJKLMN OPQRSTUVWXYZ
OK ☞

3.17.9. Exercice d'application A55 : carrés et racines

Enoncé

Ecrivez un programme qui affiche la table des nombres de 1 à 20, leur carré et leur racine carrée en présentant le résultat de la façon suivante :

Nombre	Carré	Racine
1	1	1
2	4	1.414214
3	9	1.732051
4	16	2
5	25	2.236068
6	36	2.44949
7	49	2.645751
8	64	2.828427

Une solution

```
10 REM TABLE21
20 CLS
30 PRINT "Nombre", "Carré", "Racine"
40 K = 0
50 K = K + 1
60 PRINT K, K*K, SQR(K)
70 IF K < 20 THEN 50
80 END
```

3.17.10. Exercice d'application A56 : dessin d'un triangle d'étoiles

Énoncé

Rédigez le programme qui affiche sur l'écran le dessin suivant :

```

★
★ ★
★ ★ ★
★ ★ ★ ★
★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★
    
```

Pour ce faire, vous **ne** devez **pas** utiliser une suite de :

```

30 PRINT "★"
40 PRINT "★★"
50 PRINT "★★★"
    
```

etc... (comme dans l'exercice SUITE 18),

mais une boucle (comme dans les exercices SUITE 22 et suivants).

Vous ne devrez utiliser qu'une seule fois le symbole "★". Autrement dit : rédigez une boucle sans fin qui affiche à chaque fois une chaîne A\$ qui augmente d'une étoile (par concaténation) à chaque passage.

Vous «sortirez» de la boucle lorsque la variable A\$ aura atteint la valeur

A\$ = "★★★★★★"

Une solution

	Valeurs successives de A\$ aux passages-numéros					
	1	2	3	4	5	6
10 REM ETOILE 21	vide					
20 CLS	vide					
30 A\$ = ""	vide					
40 A\$ = A\$ + "★"	★	★★	★★★	★★★★	★★★★★	★★★★★★
50 PRINT A\$	★	★★	★★★	★★★★	★★★★★	★★★★★★
60 IF A\$ = "★★★★★★" THEN 80	★	★★	★★★	★★★★	★★★★★	★★★★★★
70 GOTO 40						★★★★★★
80 END						★★★★★★

Commentaires

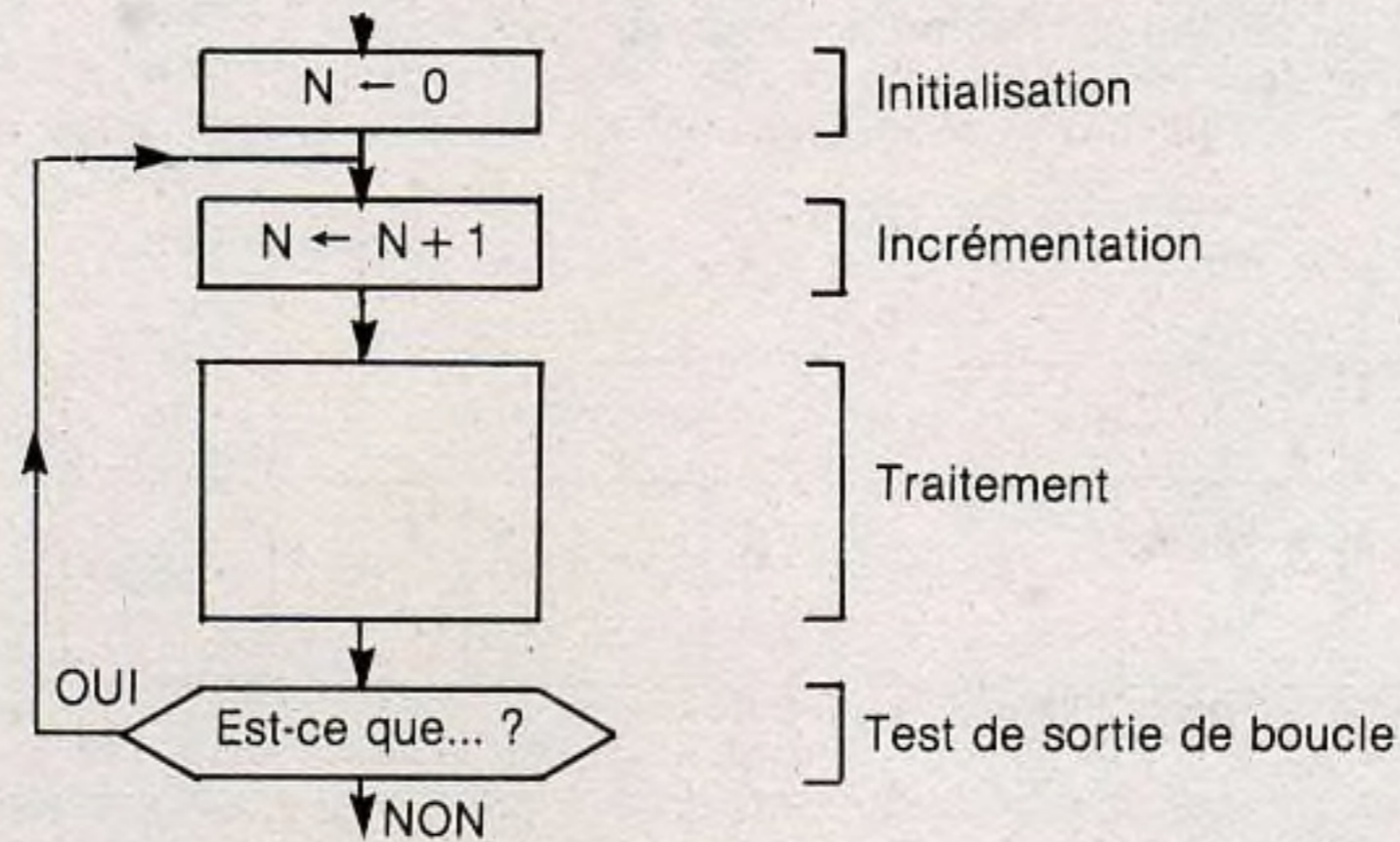
ligne 30 La «chaîne vide» est l'équivalent pour les chaînes de caractères de la valeur 0 pour les valeurs numériques.

ligne 40 C'est la ligne qui fait croître la valeur de A\$ d'une étoile à chaque passage.

ligne 60 C'est le «test» de sortie de boucle. Dans le cas présent, on utilise l'égalité comme pour l'exercice A52. Il aurait été possible d'utiliser le signe d'inégalité dans le sens «classement alphabétique» (voir LED MICRO n° 13 §3.15.6.A page 18)

3.17.11. Compteur de passages

Les organigrammes de plusieurs des programmes que nous venons de rédiger peuvent être représentés de façon générale par le schéma suivant :



La variable N utilisée dans ces programmes est appelée «compteur de boucles» car :

- au premier passage dans la boucle, on a $N = 1$
- au deuxième passage dans la boucle, on a $N = 2$
- etc.

3.17.12. Exercice d'application A57

Enoncé

Résolvez le même problème que dans l'exercice A56 (triangle d'étoiles) mais utilisez une variable compteur N pour sortir de la boucle.

Une solution et son déroulement

	1 ^{er} passage		2 ^e passage		3 ^e passage		6 ^e passage	
	N	A\$	N	A\$	N	A\$	N	A\$
10 REM ETOILE 22	0	vide						
20 CLS	0	vide						
30 A\$ = " " : N = 0	0	vide						
40 N = N + 1	1	vide	2	★	3	★★	6	★★★★★★
50 A\$ = A\$ + "★"	1	★	2	★★	3	★★★	6	★★★★★★
60 PRINT A\$	1	★	2	★★	3	★★★	6	★★★★★★
70 IF N 6 THEN 40	1	★	2	★★	3	★★★	6	★★★★★★
80 END							3	★★★★★★

3.17.13. Une nouvelle instruction — Exercice de récapitulation R19

Entraînez-vous à lire les textes des notices techniques.

Voici ce qu'on trouve dans la notice BASIC de l'IBM-PC (qui est parfaite... sauf en ce qui concerne son prix).

STRING\$ (N, M) : renvoie de caractère dont la valeur ASCII est M répété N fois.

STRING\$ (N, X\$) : renvoie le premier caractère de X\$ répété N fois.

Nous vous demandons :

- 1°) de résoudre le même problème que ci-dessus en utilisant l'instruction STRING\$, sous chacune de ses deux formes.
- 2°) si votre BASIC ne connaît pas cette instruction STRING\$, mais une instruction jouant un rôle analogue (ou presque) — de l'utiliser dans un programme — de nous envoyer le listing correspondant (en indiquant sur quel système il a été exécuté).

3.17.14. Passons-nous du RND grâce à une boucle

Préambule

Nous avons découvert dans LED MICRO n° 15 (pages 8, 9 et 23) l'emploi du RND et du RANDOMIZE dans le BASIC du PC d'IBM. Si on omet d'ajouter une fonction RANDOMIZE, chaque fois qu'on fera RUN pour recommencer l'exécution d'un programme, le RND fournira toujours la même série de nombres : donc aucun intérêt pour des programmes de jeu.

Mais si on utilise la fonction RANDOMIZE — par exemple avec TIME\$ — on obtient une valeur au hasard (le nombre de secondes par exemple)... et on n'a plus besoin de RND.

Exercice d'application A58

Ecrivez un programme qui fournit au hasard un chiffre compris entre 0 et 7 — en n'utilisant **pas** le RND. Cherchez un peu avant de lire la solution (ci-dessous).

Une solution de A58

```
10 REM RANDOM21
20 A$ = RIGHT$(TIME$,1)
30 A = VAL(A$)
40 IF A > 8 THEN 20
50 PRINT A
```

3.17.15. Encore un peu d'anglais informatique

loop : boucle
step : pas, incrément
for : depuis
next : suivant

In the last exemple, the loop executes ten times

until the NEXT statement is encountered

if you do any branching out of nested loops

When you omit the step clause, BASIC assumes a step of 1

the statements between FOR and NEXT are indented. The computer ignores this indentation.

... are executed five times each...

... Flowcharts are intended to be read by human beings, not by computers

counter : compteur
nexted loops : boucles imbriquées

Dans le dernier exemple, la boucle s'exécute 10 fois

jusqu'à ce que l'on rencontre l'instruction NEXT

si vous faites des branchements en dehors des boucles imbriquées

Si vous oubliez la clause «STEP», le BASIC supposera que le pas d'incrément est de 1.

les instructions entre FOR et NEXT sont décalées. L'ordinateur ne tient pas compte de ce décalage.

sont exécutées 5 fois chacune

... les organigrammes sont destinés à être lus par des êtres humains et non par des ordinateurs.

3.17.16. Un test

Ne trichez pas !

Dans le prochain numéro 18, nous exposerons un procédé qui a pour ambition de vous faire programmer des boucles automatiquement. J'ai appelé ce procédé «méthode béquille» parce qu'il a pour but de vous apprendre à marcher (avec des béquilles) : vous devrez l'abandonner dès que vous n'en aurez plus besoin.

Ceux d'entre vous qui ont l'habitude du raisonnement par récurrence peuvent ne pas utiliser ces béquilles, puisqu'ils savent déjà marcher tout seuls.

Pour savoir si vous êtes dans ce cas, je vous propose de faire sans tricher l'exercice suivant.

Exercice d'application A58 : le nombre d'or

Je considère la suite de nombres U_n

$$U_n = 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89...$$

C'est une «suite de Fibonacci», chaque terme de cette suite est obtenu par l'addition des deux termes précédents.

Point de départ : $U_1 = 1$ $U_2 = 1$

termes suivants

$$U_3 = U_1 + U_2 = 1 + 1 = 2$$

$$U_4 = U_2 + U_3 = 1 + 2 = 3$$

$$U_5 = U_3 + U_4 = 2 + 3 = 5$$

.....

$$U_n = U_{n-2} + U_{n-1}$$

A partir de cette suite U_n , je vais calculer une suite V_n telle que :

$$V_1 = \frac{U_2}{U_1} = \frac{1}{1} = 1$$

.....

$$V_2 = \frac{U_3}{U_2} = \frac{2}{1} = 2$$

$$V_n = \frac{U_{n+1}}{U_n}$$

$$V_3 = \frac{U_4}{U_3} = \frac{3}{2} = 1,5$$

Au fur et à mesure que n grandit, le nombre V_n se rapproche de plus en plus du «nombre d'or» qui a pour valeur $\frac{1+\sqrt{5}}{2} = 1,62$.

Nous ne vous demandons pas de retrouver la démonstration mathématique qui détermine la valeur de ce nombre d'or mais de faire effectuer ce calcul par votre ordinateur, de sorte qu'il calcule (dans une boucle) les valeurs successives de U et de V , et arrête son calcul lorsqu'il aura trouvé deux valeurs de V successives différent entre elles de moins de 0,000001.

Attention ! La majorité des élèves auxquels je fais passer ce petit test pour connaître leur niveau de pratique du BASIC utilisent pour ce faire des «variables tableau». Ceci est totalement inutile et démontre qu'ils n'ont pas bien assimilé la notion d'affectation. N'utilisez pas de «variable tableau» pour résoudre cet exercice — d'ailleurs, nous ne les avons pas encore étudiées.

Résultats des courses

Si vous avez résolu cet exercice en moins de dix minutes, c'est parfait. Vous n'avez pas besoin de béquilles. Vous pouvez vous dispenser de lire le n° 18 de mon cours de programmation.

Si vous n'avez pas réussi — ou si vous avez mis trop longtemps (ceci veut dire que ces notions ne sont pas instinctives pour vous), alors de deux choses l'une :

— ou bien vous n'avez pas envie de vous casser la tête. Vous pouvez également vous dispenser de lire le n° 18 du présent cours. Vous pourrez continuer à avancer en programmation «simple», réaliser la majorité des programmes de jeu et/ou de gestion... mais vous serez gênés pour résoudre des exercices «pointus» (problèmes de tri, de C.A.O., etc.)

— ou bien vous acceptez d'avancer lentement pour pouvoir arriver assez haut (ne pas monter bien vite mais très haut — disait à peu près Cyrano), alors suivez-nous. (J'aimerais d'ailleurs connaître votre avis sur le résultat : est-ce compréhensible sans trop d'efforts ?).

3.17.17. Résultat (sans commentaires) de A58

	1 ^{er} passage					2 ^e passage					3 ^e passage				
	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
10 REM NOMDUR 21	0	0	0	0	0										
20 A = 1 : B = 1	1	1	0	0	0										
30 C = A + B	1	1	2	0	0										
40 D = C/B	1	1	2	2	0										
50 E = D	1	1	2	2	2	1	2	3	1,5	1,5	2	3	5	1,67	1,67
60 A = B	1	1	2	2	2	2	2	3	1,5	1,5	3	3	5	1,67	1,67
70 B = C	1	2	2	2	2	2	3	3	1,5	1,5	3	5	5	1,67	1,67
80 C = A + B	1	2	3	2	2	2	3	5	1,5	1,5	3	5	8	1,67	1,67
90 D = C/B	1	2	3	1,5	2	2	3	5	1,67	1,5	3	5	8	1,60	1,67
100 IF ABS(D - E) < 0,0001 THEN 50	1	2	3	1,5	2	2	3	5	1,67	1,5	3	5	8	1,60	1,67
110 PRINT "NOMBRE D'OR=" ; D															
120 END															

3.17.18. La notation «crochet»

Au fur et à mesure du déroulement du programme ETOILE22 (§3.17.13), la valeur des variables A\$ et N change :

— au premier passage dans la boucle A\$ = "★" et N = 1
Nous écrivons ceci

$$A\$\{1\} = \text{"★"} \text{ et } N\{1\} = 1$$

— au deuxième passage dans la boucle A\$ = "★★" et N = 2
Nous écrivons ceci

$$A\$\{2\} = \text{"★★"} \text{ et } N\{2\} = 2$$

etc.

Ne confondez pas cette notation «entre crochets» avec la représentation des variables tableaux (que nous étudierons ultérieurement). A\$\{1\}, A\$\{2\}, A\$\{3\}... etc... représentent les contenus d'une seule «boîte» A\$ à différents instants.

Les notations N\{1\}, N\{2\}, N\{3\}... sont équivalentes aux notations N_1, N_2, N_3 utilisées par les mathématiciens dans les démonstrations par récurrence — avec un (petit) avantage de composition typographique (ou de dactylographie).

Cette représentation n'est nullement standardisée. Nous ne l'emploierons que dans deux numéros de LED-MICRO, le temps que vous acquériez l'«instinct de la boucle».

Lorsque nous écrivons :

$$A \leftarrow A + 3$$

ceci signifie :

« prendre l'ancienne valeur de A, y ajouter 3 et mettre cette nouvelle valeur numérique dans la «petite boîte» A »

Si cette expression est à l'intérieur d'une boucle et que l'on en est au N^o passage dans la boucle, on pourra l'écrire :

$$A\{N + 1\} = A\{N\} + 3$$

Ici le signe = représente une égalité mathématique (opérateur de comparaison) et non l'affectation.

3.17.19. Exemple SUITE33 : somme d'une suite arithmétique de N nombres

Remarque préalable

Les «matheux» trouvent certainement que les explications qui vont suivre sont trop longues et pénibles parce que trop détaillées. Mais leur habitude du raisonnement par récurrence n'est (vraisemblablement) pas partagée par tous nos lecteurs.

Pour savoir si vous pouvez aller plus vite, essayez d'écrire un programme permettant de calculer $n!$ à l'aide d'une boucle. Si vous réussissez en moins de 2 minutes, d'accord — ne perdez pas votre temps et contentez-vous de feuilleter les exemples qui suivent jusqu'aux paragraphes consacrés aux FOR... TO... NEXT.

Enoncé du problème

Je me donne un nombre N quelconque. Par exemple $N = 23$. Je vais demander à l'ordinateur d'effectuer la somme :

$$S = 5 + 8 + 11 + 14 + 17 + 20 + 23 + 26 + 29 + 32$$

c'est-à-dire d'effectuer la somme d'une série de nombres entiers «de 3 en 3» (série arithmétique) compris entre 5 et N. Bien sûr, tous les matheux connaissent la formule qui donne directement le résultat, mais ce n'est pas cela qui nous intéresse : nous voulons que l'ordinateur obtienne le résultat en effectuant la suite d'additions :

$$\text{d'abord } S = 5 + 8 = 13$$

$$\text{puis } S = 13 + 11 = 24$$

$$\text{puis } S = 24 + 14 = 38$$

etc.

Expression du terme général A

Dans la suite des nombres à additionner, on passe d'un terme au suivant en lui ajoutant 3.

$$\begin{array}{ll} A[1] = 5 & A \leftarrow 5 \\ A[2] = A[1] + 3 & A \leftarrow A + 3 \\ A[3] = A[2] + 3 & A \leftarrow A + 3 \\ A[4] = A[3] + 3 & A \leftarrow A + 3 \\ \dots\dots & \\ A[N] = A[N-1] + 3 & A \leftarrow A + 3 \end{array}$$

Expression des sommes partielles successives

$$\begin{array}{ll} S[1] = & 5 & S \leftarrow 5 \\ S[2] = S[1] + 8 = & 5 + 8 = 13 & S \leftarrow S + 8 \\ S[3] = S[2] + 11 = & 13 + 11 = 24 & S \leftarrow S + 11 \\ S[4] = S[3] + 14 = & 24 + 14 = 38 & S \leftarrow S + 14 \\ S[5] = S[4] + 17 = & 38 + 17 = 55 & S \leftarrow S + 17 \\ \dots\dots\dots & & \\ S[10] = S[9] + 32 = & 150 + 32 = 182 & \end{array}$$

$$S[N] = S[N-1] + A[N] \qquad S \leftarrow S + A$$

D'où le programme BASIC

et son «déroulement» :

	1 ^e passage		2 ^e passage		3 ^e passage	
	A	S	A	S	A	S
10 REM	0	0				
20 S=0	0	0				
30 A=5	5	0				
40 S=S+A	5	5	8	13	11	24
50 A=A+3	8	5	11	13	13	24
60 IF A 32 THEN 40	8	5	11	13	13	24
70 PRINT "S=" ; S						
80 END						

3.17.20. Exercice d'application A58 : calcul de factorielle

Rappel d'une définition

Les mathématiciens représentent par le symbole $5!$ le produit des 5 premiers nombres :

$$5! = 1 \star 2 \star 3 \star 4 \star 5$$

$5!$ se prononce «factorielle 5»

De même

$$9! = 1 \star 2 \star 3 \star 4 \star 5 \star 6 \star 7 \star 8 \star 9$$

Énoncé du problème

L'ordinateur demande à l'opérateur de lui donner un nombre entier inférieur à 50. Soit N ce nombre. Et l'ordinateur calcule $N!$

Rédigez le programme correspondant en vous inspirant de l'exemple ci-dessus. Mais cherchez avant de comparer votre solution avec celle que nous vous proposons ci-dessus. Si vous ne trouvez pas comme nous, essayez de «dérouler» votre programme pour voir s'il «tourne» bien. Ne méprisez pas cet exercice !

Analyse

Utilisons notre notation «crochet» pour représenter les valeurs des factorielles :

$$1! = F[1] = 1$$

$$2! = F[2] = 1 \star 2 = 2$$

$$3! = F[3] = 1 \star 2 \star 3 = 6$$

$$4! = F[4] = 1 \star 2 \star 3 \star 4 = 24$$

etc.

On peut écrire

$$F[1] = 1$$

$$F[2] = F[1] \star 2$$

$$F[3] = F[2] \star 3$$

$$F[4] = F[3] \star 4$$

.....

$$F[N] = F[N-1] \star N$$

L'expression générale «de récurrence»

$$F[N] = F[N-1] \star N$$

peut s'écrire avec l'opérateur d'affectation

$$F \leftarrow F \star N$$

c'est-à-dire par la ligne BASIC

$$F = F \star N$$

Le programme et son déroulement

	1 ^e passage			2 ^e passage			3 ^e passage			4 ^e passage		
	N	F	Z	N	F	Z	N	F	Z	N	F	Z
10 REM	0	0	0									
20 INPUT "NOMBRE" ; Z	0	0	6									
30 N=Q : F=1	0	1	6									
40 N=N+1	1	1	6	2	1	6	3	2	6	4	6	6
50 F=F★N	1	1	6	2	2	6	3	6	6	4	24	6
60 IF N<Z THEN 40	1	1	6	2	2	6	3	6	6	4	24	6
70 PRINT "FACTORIELLE DE N=" ; F												
80 END												

Exercices et traductions

Les solutions des lecteurs à R12 (le cheval errant)

1. Les 5 premières réponses

M. G.D. de Tessy-sur-Vire (50420)

M. G.D. traduit la position en langage échiquéen d'une façon assez élégante : il introduit la chaîne de caractères :

```
A$ = "A B C D E F G H"
```

et en extrait la lettre désignant la colonne par une instruction MID\$. (Plus exactement par l'instruction L\$ spécifique du ZX81 de Sinclair). Ceci permet de ne pas employer le CHR\$: en effet le ZX81 n'utilise pas le code ASCII.

M. E.C. (de Montpellier) et M. C.D. (de Grenoble)

proposent également une variante sur ZX81. (Le ZX81 serait-il l'ordinateur le plus utilisé par nos lecteurs ?).

M. J.T. de Saint-Martin du Bec (76133)

ne résoud pas exactement le problème que nous avons posé : au lieu de choisir une case au hasard, il choisit systématiquement le saut AJ = + 10. S'il aboutit à une impossibilité, il prend le saut suivant.

Après chaque saut, il va chercher un sous-programme pour vérifier que la position est acceptable, et un sous-programme pour traduire la position en langage échiquéen. C'est interdit, monsieur : vous n'êtes pas censé connaître le GOSUB.

Le docteur G.L. de Cannes (06400)

propose une solution sur Apple II. A la fin de chaque partie, il repart pour un autre tour grâce à un GOTO (donc ne remet pas à zéro les variables, donc son RANDOM donne une **nouvelle partie** différente de la précédente).

2. La solution du samideano

M. Sipra nous a envoyé une solution que j'ai recopiée sur mon ordinateur, testée et enregistrée sous le n° R12LEC21 :

```
5 REM R12LEC21
7 REM SOLUTION SIPRA A R12
10 DP$ = "0103121636404951" : AC = 27 : I = 0
20 PRINT "DEPART : A1"
30 V$ = RIGHT$(TIME$,2)
40 RANDOMIZE VAL(V$)
50 H = INT(RND*8)
60 DP = VAL(MID$(DP$,2*H+1,2))
70 N = AC + DP - 26
80 Q = INT(N/12)
90 R = N - 12 * Q
100 IF (Q<2) OR (Q>9) OR (R<3) OR (R>10) THEN 30
120 A$ = CHR$(R+62) + STR$(Q-1)
130 AC = N : I = I+1
140 PRINT "SAUT " ; I ; " EN " ; A$
150 IF AC<> 118 THEN 30
155 PRINT
160 PRINT "TERMINE EN " ; I ; " SAUTS"
180 END
```

Il part de la même idée de base que M. G.P. de Tessy-sur-Vire : grouper une suite de données dans une chaîne de caractères puis l'extraire à l'aide d'une instruction MID\$. Mais alors que M. G.P. entrait dans sa chaîne A\$ la suite des noms des colonnes, M. Sipra entre dans DP\$ la suite des 8 sauts possibles. Pour ne pas avoir à entrer une valeur négative, il ajoute systématiquement 26 à ce saut (valeur qu'il retire ligne 70). Il obtient ainsi n'importe quelle valeur du saut DP par une seule instruction (ligne 60) : solution plus élégante qu'une collection de IF... THEN...

Exercice de récapitulation R20 : le cheval qui erre moins

Pourquoi ce silence !

Je n'ai reçu aucune réponse à ma demande (LED MICRO n° 16 page 13) de recherche de solutions optimisées : au lieu de laisser le cheval errer entre une case de départ et une case d'arrivée complètement «au hasard», on lui donne quelques directives pour atteindre ce résultat en un nombre minimum de coups.

Pourquoi ce silence ? Parce que vous vouliez trouver «la» solution parfaite ? Ce n'est pas ce que je demandais : le but n'est pas de maîtriser le jeu du cheval, mais de s'entraîner à programmer.

— Vous pouvez, pour commencer, interdire à votre cheval de faire un pas en avant suivi immédiatement du pas en arrière rigoureusement inverse. Le gain en nombre de coups n'est pas grand, mais il n'est pas nul.

— Un peu plus élaboré : distinguer dans l'échiquier deux zones : la zone «proche» (les cases entourant la case «but». Dans le cas présent les cases F8, F7, F6, G8, G7, G6, H7, H6) et le reste de l'échiquier. Lorsque le cheval n'est pas dans la zone «proche», on peut imaginer qu'il «vise» la case but. Lorsque le cheval entre dans la zone proche, il change de stratégie et (par exemple) se déplace au hasard (mais sans sortir de cette zone). Ce n'est pas l'idéal, mais c'est déjà mieux.

— Encore un peu mieux : éliminer le hasard dans la zone proche : le cheval fait des essais systématiques pour «apprendre» quel est le chemin à adopter à partir de n'importe quelle case. (Mais ceci nécessite la connaissance des tableaux. A ne tenter que par les «fortiches»).

Pour tester la «qualité» de chaque «stratégie», faites faire 100 parcours à votre cheval (grâce à une boucle) en lui faisant calculer la moyenne des coups nécessaires pour un parcours donné entre deux cases (cette fois-ci différentes de E1-H8).

Cet exercice est peut-être trop difficile. Dites-le moi : je ne reçois que des lettres de fortiches. Comment voulez-vous que je réajuste mon tir ?

Pour me faciliter le travail, ne rédigez pas plusieurs exercices sur la même feuille et indiquez le numéro de l'exercice. Ce «cheval pas tellement errant» portera le n° R20.

Dernière heure

Si ! Je viens de recevoir non seulement une réponse mais presque un bouquin sur ce thème accompagné d'une cassette pour ZX81 de la part de M. L.T. (de Marseille). Merci pour cette œuvre : je vais l'étudier. Bravo et merci.

Exercice de récapitulation R21 : conjugaison

Notre lecteur M. P.P. (de Guyancourt 78280) propose un exercice utilisant des chaînes de caractères.

Voici comment on peut poser son problème :

«L'ordinateur demande

''INSCRIS UN VERBE DU PREMIER GROUPE''

L'ordinateur teste si ce verbe se termine en GER (ou en YER), si c'est le verbe ALLER, si c'est bien un verbe du premier groupe. Puis il le conjugue à l'indicatif présent.

Exercice de récapitulation R22 : le code Morse

Vous connaissez certainement l'antique code Morse qui, à chaque caractère (lettre ou chiffre) fait correspondre une suite de points et de traits.

Rédigez un programme qui, lorsque l'opérateur tape la lettre A puis appuie sur la touche ENTER, imprime . - (qui est le code Morse de la lettre A). Si, ensuite, il tape sur la lettre K, imprime la succession de . et de - correspondant au code de la lettre K, etc. Chacun des codes des lettres étant séparé par deux espaces.

Si vous disposez d'un ordinateur capable d'émettre des sons, remplacez l'impression des points et des traits par une succession de "TIT" et "TAT".

Solution de R16 : le mot le plus long

Pas de succès !

Cet exercice ne semble pas avoir inspiré beaucoup nos lecteurs. Il était vraisemblablement trop long. Je vais d'une part le simplifier et d'autre part vous mettre sur la voie.

Simplifications : ce jeu se jouera avec seulement deux personnages : l'arbitre et le joueur. A la fin de la partie, arbitre et joueur changeront de rôle. Chaque partie comporte la recherche de 5 mots. L'ordinateur «cachera» le mot en plaçant les lettres dans le désordre (les placer dans l'ordre alphabétique, c'est pour plus tard). Nous vous proposons l'organigramme de la figure 1.

Pour vous mettre sur la voie

Commencez par résoudre un élément de cet organigramme : «faire l'anagramme d'un mot, c'est-à-dire écrire ses lettres dans le désordre».

Essayez de résoudre ce problème sans regarder ni l'organigramme (figure 2 page ci-contre), ni la solution que nous vous présentons ci-dessous. Pendant que vous y êtes : cherchez à faire l'anagramme d'un mot comportant un nombre quelconque de lettres.

```
10 REM ANAGRA22
20 INPUT A$
30 L = LEN(A$)
40 X = 1 + INT(RND*L)
50 PRINT "X=" ; X ; " A$=" ; A$ , "MID$(A$,X,1)=" ; MID$(A$,X,1)
60 A$ = LEFT$(A$,X-1) + RIGHT$(A$,L-X)
70 L = L-1
80 IF L>0 THEN 40
90 END
```

```
CLAUDE
X= 1  A$= CLAUDE
X= 4  A$= LAUDE
X= 4  A$= LAUE
X= 3  A$= LAU MID$(A$,X,1)=U
X= 2  A$= LA  MID$(A$,X,1)=A
X= 1  A$= L   MID$(A$,X,1)=L
Ok ☺
```

```
MID$(A$,X,1)=C
MID$(A$,X,1)=D
MID$(A$,X,1)=E
```

le programme d'étude

et son exécution

Vous remarquerez que la virgule place les textes dans des colonnes... de façon pas toujours idéale. Nous verrons bientôt comment améliorer la présentation.

Commentaires

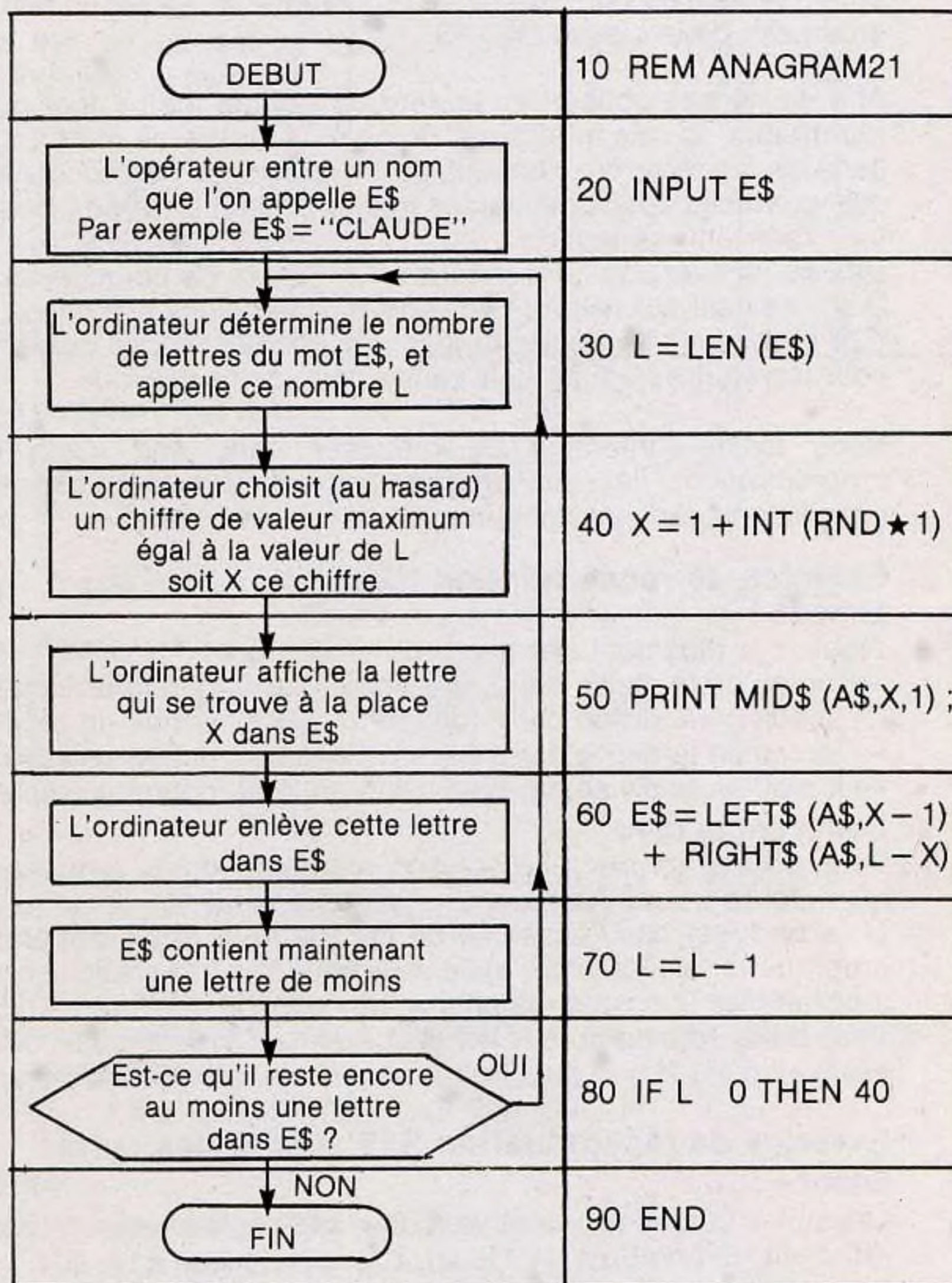
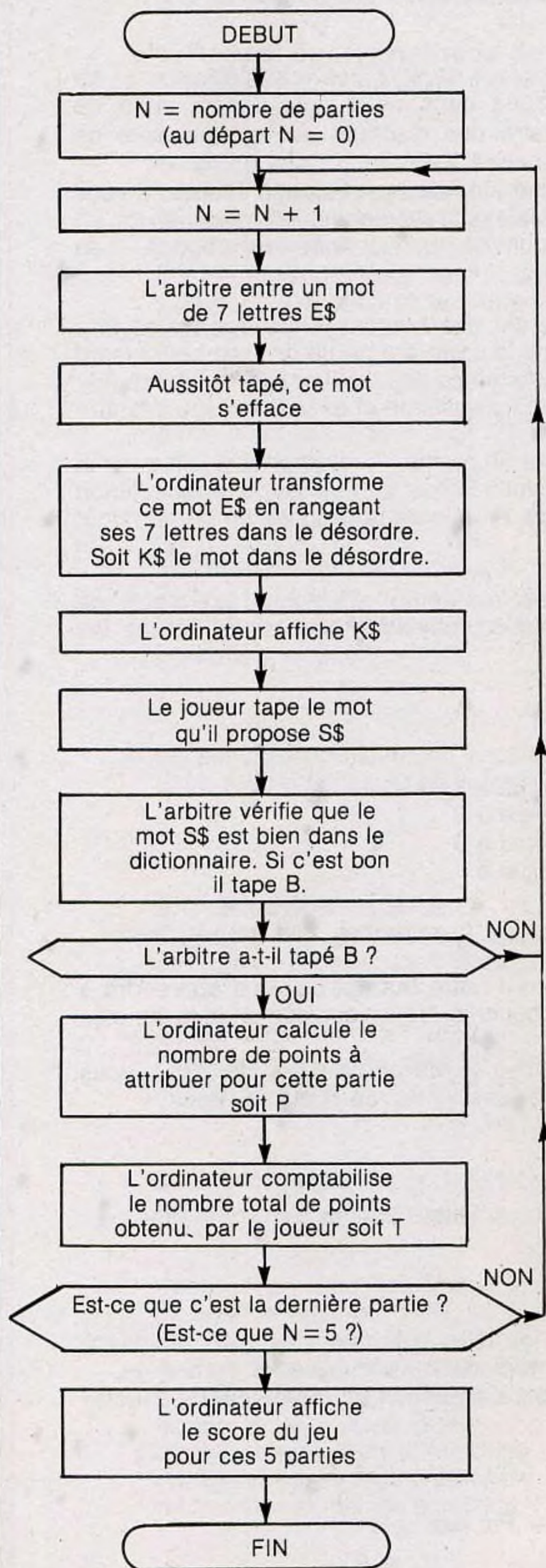
ligne 50 : Au lieu d'entrer les lettres de E\$ dans la chaîne de caractères K\$ (par concaténation), le programme se contente de faire afficher les lettres à la suite l'une de l'autre. Il faudra procéder autrement pour votre solution de R16.

Remarquez que dans le programme d'étude ci-dessous on a ajouté des instructions complémentaires (X=, A\$=) permettant de suivre l'évolution des variables dans la boucle. Bien entendu, il faudra les enlever quand on sera sûr de soi. Nous venons de rencontrer un des procédés de mise au point que nous développerons dans notre quatrième partie.

ligne 60 : il n'existe pas d'instruction permettant d'enlever directement la lettre de rang X de la chaîne de caractères E\$. Ce n'est pas grave : on se fabrique l'équivalent à l'aide de la concaténation d'un LEFT\$ et d'un RIGHT\$.

ligne 70

On pourrait obtenir la nouvelle valeur de L à l'aide de l'instruction 30 portant sur la nouvelle valeur de E\$. Essayez en n'utilisant qu'une seule instruction LEN dans ce programme : il vous faudra modifier la structure de la boucle.



Le coin des matheux - Niveau 1^{re}S

Pourquoi un coin des matheux ?

Il semble que d'assez nombreux lecteurs de LED MICRO veulent devenir des informaticiens et ne jamais avoir à faire aux mathématiques. Ils sont encouragés dans cette voie par quantité de «théoriciens de la pédagogie» qui passent leur temps à faire des discours sur la nécessité de «démythifier l'informatique». Je ne vous dirai pas ce que j'en pense.

Certes vous pourrez rédiger des programmes divers (de gestion, de jeu...) en langage évolué en vous contentant de connaissances mathématiques très modestes. Mais pour devenir un informaticien (ou un automaticien) de bon niveau, il vous faudra posséder un minimum de notions de mathématiques — en gros celui de la classe de 1^{re}S.

Afin de ne pas obliger les lecteurs à qui les maths font pousser des boutons à lire des textes trop rébarbatifs, je vais m'efforcer de ne plus mettre de maths dans la première partie de mon cours (sauf dans les numéros qui viennent, parce que je n'ai pas encore eu le temps de «rectifier le tir» et de trouver des exercices absolument sans maths, et sauf en 1986 (début du graphisme et de la C.A.O. où il faudra bien redevenir sérieux).

Dès qu'un exercice nécessitera un minimum de connaissances en maths, je le reporterai soit dans le Coin des matheux niveau 1^{re} S, soit dans le Coin des matheux niveau Prépa. L'exercice de récapitulation R20 ci-dessous ne nécessite pas des connaissances du niveau 1^{re} S, mais il suppose un certain goût pour les chiffres : alors je le range dans cette rubrique.

Nous avons l'intention de proposer dans nos «Coin des matheux» différents exercices de programmation liés aux programmes officiels en mathématiques, physique ou chimie. Toutes les suggestions dans ce domaine seront les bienvenues.

Exercice de récapitulation R22

Énoncé :

Trouvez le plus petit des nombres entiers qui :

- lorsqu'on le divise par 2 (division entière), donne un reste égal à 1
- lorsqu'on le divise par 3 (division entière), donne un reste égal à 2
- lorsqu'on le divise par 4 (division entière), donne un reste égal à 3
- lorsqu'on le divise par 5 (division entière), donne un reste égal à 4.

Comment le faire

Vous pouvez trouver une solution mathématiques. Amusez-vous à la chercher. Elle est «évidente»... quand on a trouvé l'astuce. Envoyez-moi votre solution.

Mais ce n'est pas l'essentiel de ce que nous vous demandons : notre but est de vous apprendre à programmer et (aujourd'hui) de vous entraîner à la pratique des boucles. Nous vous demandons donc de programmer la solution la plus «bête» qui soit :

Vous essayez le nombre $N = 1$ et cherchez s'il répond à la fois à ces 4 critères. Ce n'est pas le cas. Vous essayez donc $N = 2$, puis $N = 3$, puis $N = 4$, etc., jusqu'à trouver une valeur de N qui convient.

Exercice de récapitulation R23 : limite des suites

Énoncé :

Les suites ci-après tendent vers une limite finie lorsque n tend vers l'infini. Ecrivez les programmes qui affichent les nombres $U_1, U_2, U_3, U_4...$ etc... jusqu'à ce que

$$(U_n - U_{n-1}) < 0,000001$$

Bien sûr, utilisez une boucle pour ce faire.

Si, au lieu d'afficher les nombres sur l'écran, vous préférez les faire imprimer, changez de ligne à chaque valeur (pas de , ou de ;) : nous vous expliquerons pourquoi lors de la publication du corrigé.

Commencez par des suites, dont le terme général est exprimable en fonction du «compteur de boucle» N :

$$U_n = \frac{2n+1}{n+3} \quad U_n = \frac{1}{(n+1)(n+2)} \quad U_n + \frac{2^n+3^n}{3^n}$$

Puis attaquez des suites définies par une relation de récurrence. Par exemple :

$$U_0 = 7 \quad 5U_{n+1} - 2U_n = 6$$

Ne cherchez pas pour le moment à étudier sur votre ordinateur des suites définies par une relation de récurrence $f(U_n, U_{n-1}, U_{n-2}) = 0$ — genre «nombre d'or». C'est encore un peu trop difficile.

Le coin des matheux - Niveau Prépa

1. Rappel sur la méthode de Newton

Soit à résoudre l'équation $f(x) = 0$.

Nous considérons la courbe $y = f(x)$ et la fonction dérivée de $f(x)$.

Cette courbe coupe l'axe des x en un point M_s d'abscisse X_s qui est la solution cherchée.

Partons d'une valeur de X égale à $X[1]$, qui ne soit «pas trop éloignée» de X_s .

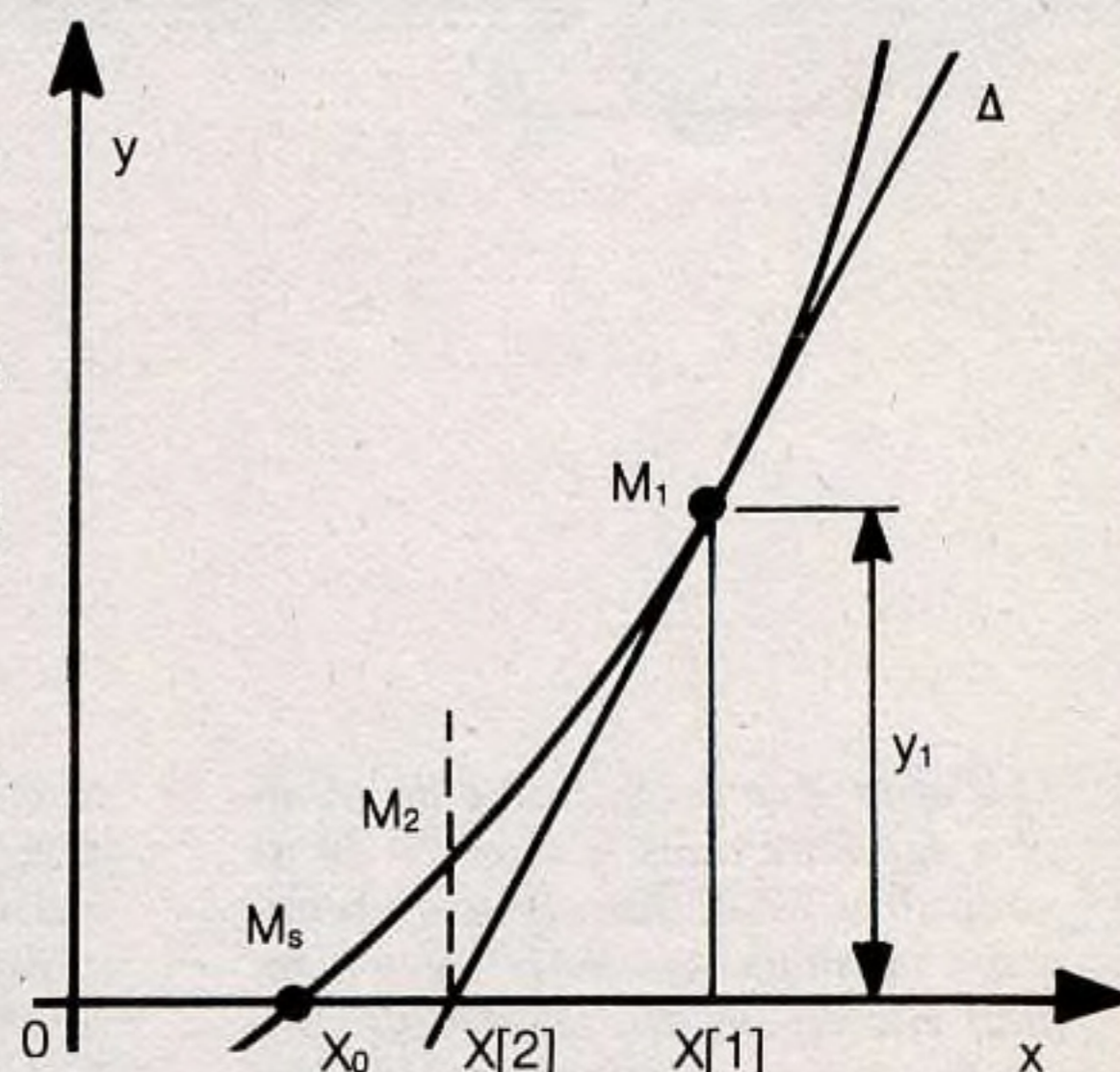
En introduisant cette valeur dans $f(x)$, on obtient la valeur y_1 , l'équation de la droite Δ tangente à y au point M_1 est

$$f'(X[1]) = \frac{f(X[1])}{X[2] - X[1]}$$

d'où l'on déduit la valeur de $X[2]$ (abscisse du point où Δ coupe l'axe des x).

$$X[2] = X[1] - \frac{f(X[1])}{f'(X[1])}$$

$X[2]$ est une approximation de X_s déjà meilleure que $X[1]$.



Il suffit de recommencer le calcul en partant de $X[2]$ au lieu de $X[1]$.

On obtient ainsi une valeur :

$$X[3] = X[2] - \frac{f(X[2])}{f'(X[2])}$$

qui est une approximation encore meilleure.

Et on recommence. D'où la formule de récurrence :

$$X[N] = X[N-1] - \frac{f(X[N-1])}{f'(X[N-1])}$$

2. Traduction en BASIC

Appelons X , Y et D les valeurs successives de la variable $X[N]$, la fonction $Y[N]$ et la dérivée $Y'[N]$. La formule de récurrence ci-dessus peut se traduire par l'instruction BASIC.

$$100 X = X - (Y/D)$$

ce qui constituera le traitement interne de votre boucle.

Pour sortir de cette boucle, testez la différence $X[N] - X[N-1]$. Lorsque sa **valeur absolue** devient inférieure à (par exemple) 0.00001, vous affichez

```
500 PRINT "X = " ; X ; "CONSTITUE UNE SOLUTION"
```

3. Exercice de récapitulation R24

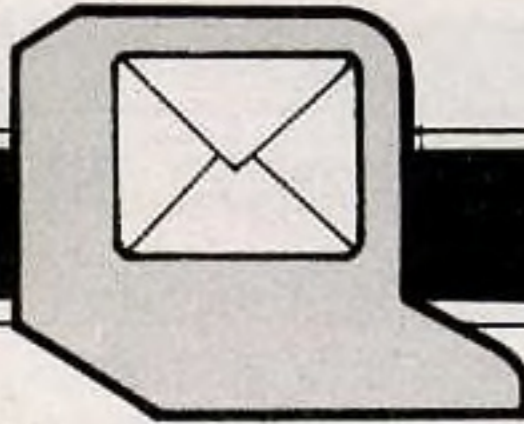
Ecrivez un programme qui permet de résoudre l'équation du troisième degré

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

Chercher dans quelles limites on peut prendre la première approximation de X , pour éviter la divergence de la série $X[N]$.

4. Exercice de récapitulation R25

Essayez de généraliser cette méthode à la résolution générale des équations algébriques. Il vous faudra examiner les cas de divergence. Comptez les approximations successives et affichez un message d'erreur demandant d'introduire une autre première approximation. Notre samideano mathématicien (M. Sipra) va peut-être plus loin en faisant balayer toutes les valeurs de X (de $-\infty$ à $+\infty$) pour rechercher toutes les solutions. Ce n'est plus de la programmation, c'est de la mathématique.



NOS LECTEURS NOUS ECRIVENT

□ ... J'ai tapé le programme du plongeur explicité dans le numéro 14 de novembre 1984. La version «plongeur normal» marche très bien, mais elle est un peu lente. Par contre, la version «plongeur rapide» est nettement mieux, malheureusement à la ligne 25, vous voulez charger le dessin 2 à l'adresse AS 9000. Une petite procédure est expliquée dans cette revue, mais la table de forme à charger en octets par groupe de 16, je ne vois pas où la trouver.

J.C.D. 75011 Paris

□ ... Dans le LED MICRO n° 14, vous proposez un programme pour Apple : plongeur normal ; plongeur rapide. Après un travail long de programmation, j'ai eu la désagréable surprise de constater le non-fonctionnement de ces deux programmes. Etant un débutant sur Apple, imaginez ma déroute !

Pour le plongeur normal : le programme indique qu'il y a une erreur à la ligne 9005 et il se bloque. De plus, ligne 1003 il y a FOR I=0 TO 3000, malheureusement 3000 n'existe pas. Puis, il manque une partie du programme lignes 9100 à 9110 !

Pour le plongeur rapide : je n'ai pas les dessins des requins et du plongeur mais simplement des points, et la ligne 1 a dû être supprimée pour que le programme puisse se lancer. A la ligne 1003, même problème que pour le plongeur normal et à la ligne 8270 FOR I=1 TO 5000 GOTO 105, 5000 n'existe pas. Je vous en prie, aidez-moi. (...) Peut-être d'autres lecteurs vous ont-ils fait part de ces problèmes.

J'espère vous lire bientôt à ce sujet

sinon je resterai plongé dans le désespoir car 2 h 1/2 de programmation pour rien, avouez que n'importe quel débutant tel que moi serait découragé.

Nota : Je n'ai commis aucune erreur de transcription, j'ai vérifié 5 fois moi-même et j'ai fait vérifier par ma femme qui n'a pas trouvé d'erreur de frappe.

J.R.C. 75019 Paris

Je suis désolé : un certain nombre de lignes de programme ont disparu.

Veillez rajouter :

Pour le plongeur normal, les lignes suivantes :

```
9100 REM TRESOR(DEPOT)
9105 HPLLOT XT,YT TO XT - 5,YT TO
      XT - 5,YT - 5 TO XT,YT - 5 TO
      XT,YT
9110 RETURN
```

Pour le plongeur rapide, les lignes suivantes :

```
9000 - 05 00 0C 00 38 00 68 00
9008 - A8 00 B0 00 07 38 38 38
9010 - 3F 3F 17 17 17 07 38 38
9018 - 38 36 36 36 2E 2D 2D 2D
9020 - 2D 2D 2D 2D 2D 15 2D 38
9028 - 38 24 05 38 17 3F D8 D8
9030 - 18 18 3C 3C 34 36 00 00
9038 - 3C 3C 3F 35 36 35 37 3E
9040 - 3D 2C 2D 2D 2D 2D 2D 2D
9048 - 2D 2D 25 24 24 24 3C 3E
9050 - 3E 3E 3E 3C 3C 3C 24 3E
9058 - 3E 3E 2E 3F 37 07 00 00
9060 - FF FF 00 00 FF FF 00 00
9068 - 35 36 36 36 36 36 2D 25
9070 - 3F 24 24 24 24 24 2D 36
9078 - 16 25 24 24 24 37 3E 27
9080 - 25 27 24 24 3F 3F 36 36
9088 - 3E 3F 36 36 36 2E 04 20
```

```
9090 - 24 2C 36 35 36 36 36 36
9098 - 3E 37 2D 25 24 24 24 2C
90A0 - 3C 2C 3C 27 24 2D 2D 00
90A8 - 35 3F 2D 2D 00 00 00 00
90B0 - 37 36 2E 25 24 04 00 00
90B8 - FF FF 00 00 FF FF 00 00
90C0 - FF FF 00 00 FF FF 00 00
```

Ceci étant dit (et fait !) vous ne devriez plus rencontrer de problèmes. Si ce n'est pas le cas, écrivez-moi en me donnant votre numéro de téléphone. Je vous contacterai.

B. Lilamand

□ ... Je désirerais savoir s'il est possible de construire un programme en assembleur, langage machine ou basic pour obtenir la double précision sur un ordinateur individuel qui ne la possède pas. L'ordinateur qui m'intéresse est l'Amstrad CPC 464. Ni l'ordinateur individuel (voir lexique des thèmes traités dans son n° 66), ni HEBDOGICIEL par exemple n'ont publié un utilitaire de ce genre après information. Ayant aussi lu tous les numéros de votre revue LED MICRO, je n'ai rien vu non plus sur cette question pourtant utile. J'ai beaucoup apprécié votre utilitaire de copie d'écran dans votre n° 15, bien utile aussi.

J.P.F. 77610 Fontenay-Trésigny

Je prends note de votre suggestion. Si plusieurs lecteurs nous signalent qu'ils ont le même besoin, je rédigerais (et publierais) un programme (en Basic) résolvant ce problème. A titre d'information, lorsque j'étais en hypotaube, je me suis amusé à faire calculer par ma calculette programmable le nombre

PI avec 200 décimales. Chacun prend son plaisir comme il peut.

B. Lilamand

... *Je pense que vous n'avez pas compris ce que souhaitait ce lecteur : ne pas trouver la fin d'un article côté recto et le début d'un autre côté verso, ce qui gêne quand, comme lui et moi, on sépare les articles pour d'une part se débarrasser de la pub, etc., d'autre part, mieux et plus utilement classer ces articles et les retrouver plus facilement en cas de besoin... à ce moment-là, une pub, etc., au verso, ne dérange pas...*

J'avais signalé le fait à M. Aisberg qui avait eu la bonté d'en tenir compte pour ses revues...

Abbé C. 06000 Nice

Plusieurs autres lecteurs me font la même remarque et formulent le même souhait.

Nous allons nous efforcer de commencer à vous donner satisfaction : les deux pages du courrier des lecteurs sépareront le cours de Philippe Duquesne du mien.

C. Polgar

... *Je me suis aperçu que les Basics avaient différentes dénominations du genre MBASIC, CBASIC, SBASIC, GWBASIC... Que signifient toutes ces dénominations ? Y trouve-t-on toutes les fonctions de l'IBM-PC...*

R.R. 03600 Commeny

Et il y en a d'autres : Toshiba utilise un TBASIC (T comme Toshiba) bien qu'il

soit d'origine Microsoft (comme le MBASIC). Le GWBASIC comporte des instructions graphiques. Le CBASIC est particulièrement adapté à la gestion des fichiers... Je vous avoue que je ne connais pas tous ces Basics. Certains de nos confrères ont publié des tableaux de correspondance entre les Basics. C'est utile, mais insuffisant : des Basics différents utilisent les mêmes instructions de façon légèrement différentes (compter les colonnes à partir de 0 ou à partir de 1 par exemple), on a des surprises si on se contente de consulter un simple tableau de correspondance.

C. Polgar

... *Vous ne faites aucune allusion au LOGO. Pourtant si mes informations sont exactes, c'est le langage qui va être adopté par l'Éducation Nationale (...).*

M.G. Marie-Anne

77230 Saint-Souplets

D'autres lecteurs réclament un cours de PASCAL, ou de FORTH. Mon cours n'est pas un cours de BASIC, mais un cours de programmation en langage évolué prenant le BASIC comme «support». Le plus difficile n'est pas d'apprendre la syntaxe de tel ou tel langage mais d'assimiler les notions essentielles : affectation, sélection, boucle, sous-programme. Ces notions étant assimilées, il est facile de les traduire soit en BASIC, soit en PASCAL, soit en... (en FORTH... c'est quand même un peu différent). Depuis mes promesses inconsidérées concernant la date de sortie de LED-ROBOT, je n'ose rien affirmer sur ce que sera le déroulement de nos articles.

Ces réserves étant faites, j'envisage de faire démarrer une rubrique LOGO de six pages en septembre, lorsque nos lecteurs auront étudié toutes les structures fondamentales.

VITE OU BIEN ?

J'ai reçu une lettre d'un lecteur qui n'est pas content. Il me reproche la lenteur de mon cours, le développement excessif du corrigé des exercices et la part trop importante prise par les mathématiques. Il pense que «notre objectif de vendre du papier prend le dessus sur l'intérêt du lecteur».

Voici pourquoi mon cours est «lent» : «Apprendre à écrire un programme en BASIC ou en PASCAL ou en n'importe quoi n'est pas difficile, l'essentiel est d'apprendre à programmer, c'est-à-dire à analyser un problème indépendamment du langage utilisé.

S'il s'agit d'un problème classique de gestion, les méthodes d'analyse ne manquent pas : mais nous n'en sommes pas là. Comment apprendre à résoudre un problème quelconque autrement qu'en donnant de nombreux exemples de raisonnement — en cherchant, en se trompant, en confrontant des idées ? C'est forcément long, mais je ne sais pas faire autrement.

Contrairement à ce que souhaite ce lecteur, j'envisage de donner de l'importance à la partie «Exercices de récapitulation» : de 6 à 8 pages par numéro. Qu'en pensent les autres lecteurs ?

P.S. Ce lecteur nous demande de ne pas publier sa lettre. Dommage : elle posait d'autres problèmes de fond importants.

COURS PRATIQUE AVEC LE MICROPROFESSOR MPF-IB

NEUVIEME PARTIE

Contrôle du CPU et E/S

SOMMAIRE

I. INTRODUCTION

II. INSTRUCTIONS DE «DECALAGE» ET «ROTATION»

- II.1. Introduction
- II.2. Définitions des «OPERATIONS»

III. MANIPULATION ET TEST DE BITS

- III.1. Exemple
- III.2. Manipulation de bits
- III.3. Test de bit
- III.4. Exercices

IV. INSTRUCTIONS «CONTROLE DU CPU»

- IV.1. Instruction "NOP"
- IV.2. Instruction "HALT"
- IV.3. Instructions d'autorisation et d'interdiction des interruptions
- IV.4. Modes d'interruptions

V. INSTRUCTIONS ENTREE-SORTIE

- V.1. Introduction
- V.2. Instructions d'ENTREE
- V.3. Instructions de SORTIE
- V.4. Exemples d'application
- V.5. Instructions de transfert par bloc
- V.6. Instructions de transfert par bloc, ENTREE-SORTIE
- V.7. Instructions de transfert par bloc dans la mémoire
- V.8. Instructions de recherche de caractères

I. INTRODUCTION

Avec la neuvième partie, la phase théorique du cours «Microprocesseurs» se termine. Notre objectif est de balayer l'ensemble du répertoire des instructions du Z-80 et de montrer à l'aide de quelques exemples leur mise en œuvre.

Dans les prochains numéros, nous nous consacrerons essentiellement à des applications, que chacun d'entre vous pourra exécuter sur son matériel, le Microprofessor MPF-IB. Chacune d'elles vous permettra de bien comprendre le déroulement d'un programme, d'en suivre son évolution et peu à peu de concevoir vos propres applications.

II. INSTRUCTIONS DE «DECALAGE» ET «ROTATION»

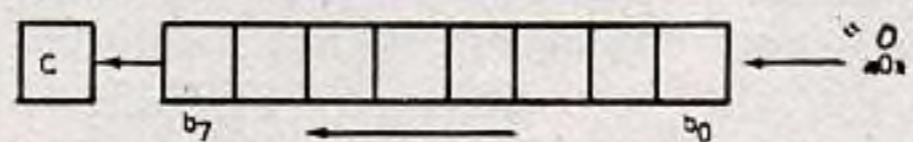
1. Introduction

Nous allons étudier un ensemble d'instructions qui s'apparentent aux instructions logiques puisqu'elles portent leur effet sur les «bits» d'un registre donné. Leurs emplois sont multiples, mais principalement dans les opérations arithmétiques comme la multiplication, la division, l'extraction d'une racine carrée, etc.

Il faut avant d'entreprendre l'étude de cette partie avoir bien compris le mécanisme des registres à décalage.

2. Définitions des «OPERATIONS»

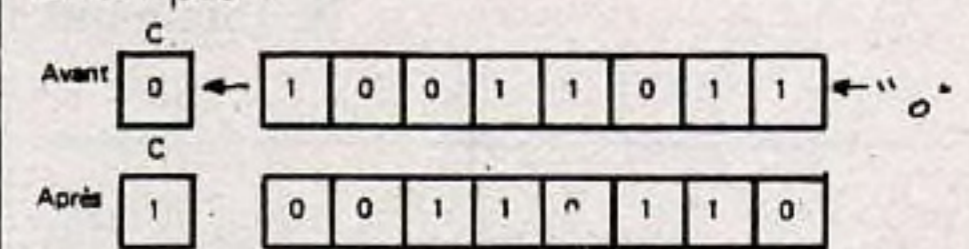
a) Décalage arithmétique à gauche



Description :
L'exécution de cette instruction décale tous les bits du registre «s» ou de l'octet «m» d'une position **vers la gauche**. L'indicateur C (du registre

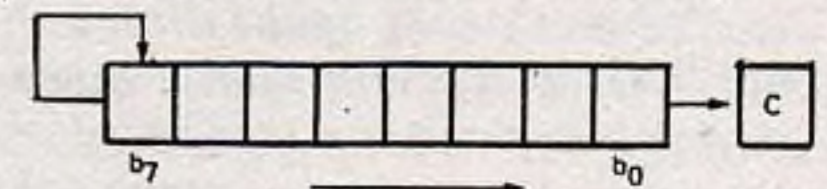
F) est chargé avec le bit de poids fort du registre «s» ou de l'octet «m». Le bit «b» est chargé avec un zéro.

Exemple :



b) Décalage arithmétique à droite

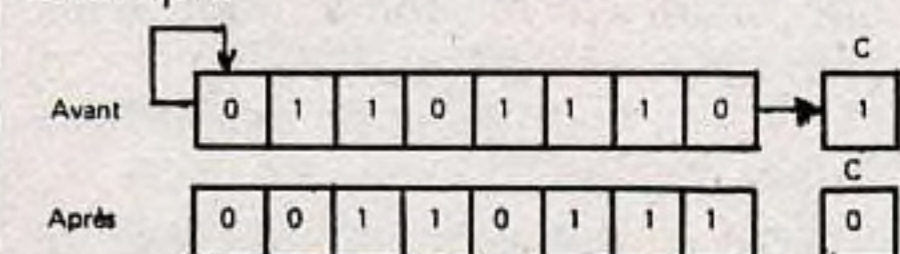
Opération :



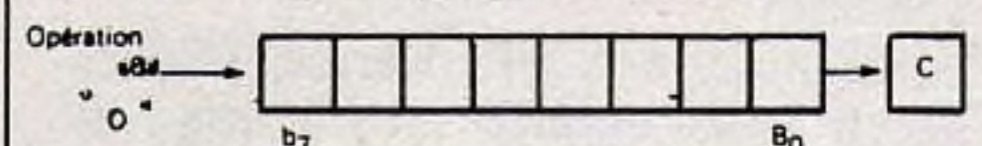
Description :

L'exécution de cette instruction décale tous les bits du registre «s» ou de l'octet «m» d'une position **vers la droite**. Le bit 7 conserve son ancienne valeur. Le bit 0 est chargé dans l'indicateur C (registre F).

Exemple



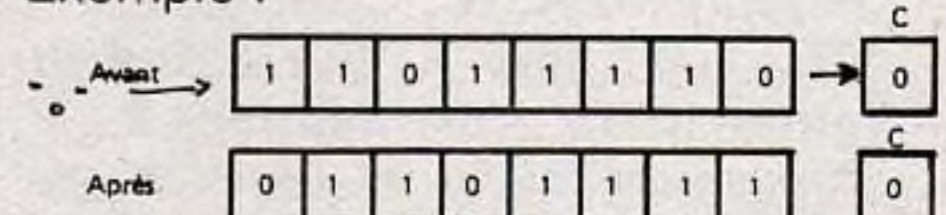
c) Décalage logique à droite



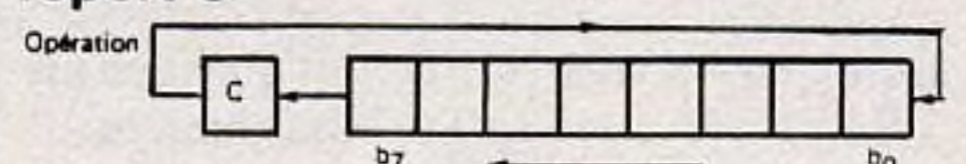
Description :

L'exécution de cette instruction décale tous les bits du registre «s» ou de l'octet «m» d'une position vers la droite. Le bit 7 est chargé avec «0». L'indicateur C (registre F) est chargé avec le bit de poids faible du registre «s» ou de l'octet «m».

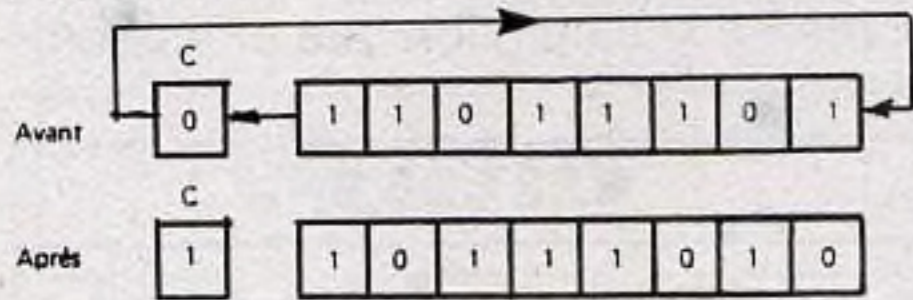
Exemple :



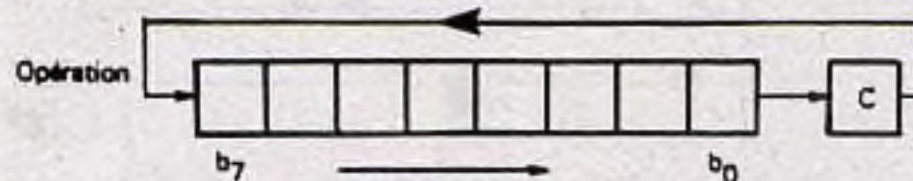
d) Rotation à gauche à travers le report C



L'exécution de cette instruction décale tous les bits du registre «s» ou de l'octet «m» d'une position vers la gauche. Le bit 0 est chargé avec le bit C. Le bit C est chargé avec le bit de poids fort du registre «s» ou de l'octet «m».



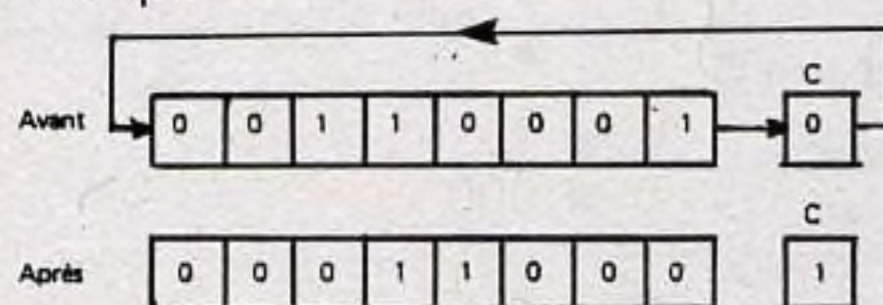
e) Rotation à droite à travers le report



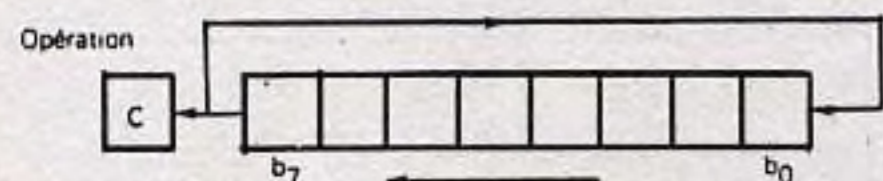
Description :

L'exécution de cette instruction décale tous les bits du registre «s» ou de l'octet «m» d'une position vers la droite. Le bit 7 est chargé avec le bit C. Le bit C est chargé avec le bit de poids faible du registre «s» ou de l'octet «m».

Exemple



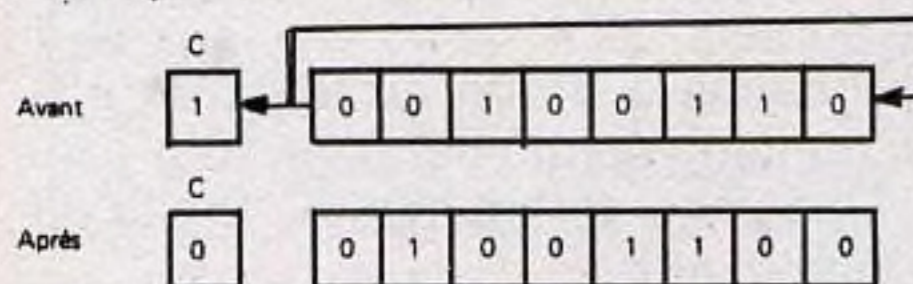
f) Rotation à gauche sans le report C



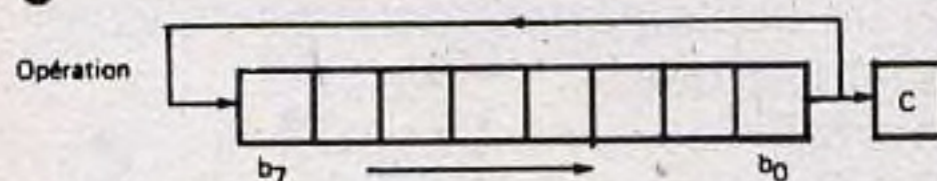
Description :

L'exécution de l'instruction décale circulairement sur lui-même vers la gauche tous les bits du registre «s» ou de l'octet «m». Le bit 7 est chargé dans le bit C ainsi que dans le bit 0 du registre «s» ou de l'octet «m».

Exemple :



g) Rotation à droite sans le report C

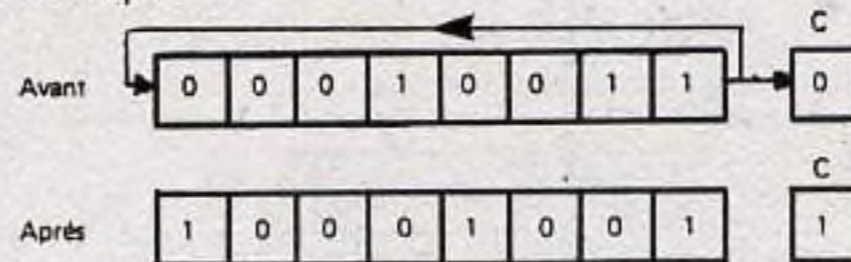


Description :

L'exécution de l'instruction décale circulairement sur lui-même vers la droite tous les bits du registre «s» ou de l'octet «m». Le bit b_0 est chargé

dans le bit C ainsi que dans le bit 7 du registre «s» ou de l'octet «m».

Exemple :



h) Codes mnémoniques

Dans la définition des opérations de décalage et de rotation, nous n'avons pas précisé ce qu'était le registre «s» ou l'octet «m».

Comme pour les opérations logiques et arithmétiques, l'opérande :

«s» désigne l'un des registres A, B, C, D, E, H ou L.

«m» désigne un octet (une case mémoire) pointée par le contenu de la paire de registres HL ou par le contenu de l'un des registres index IX ou IY auquel s'ajoute le déplacement d. L'ensemble du répertoire des instructions de décalage et de rotation, se trouve ainsi résumé dans les quatre tableaux suivants (I à IV).

III. MANIPULATION ET TEST DE BITS

1. Exemple

Supposons que le contenu d'une case mémoire doit être pour la suite du déroulement du programme une quantité paire (donc le bit de poids faible à «0»). Pour cela il suffit d'effec-

tuer un ET logique avec le contenu de cette case et le contenu de A précédemment chargé avec FE (1111 1110), puis de replacer le tout dans la case d'origine.

Il peut être aussi nécessaire de tester un bit particulier d'un registre ou d'une case mémoire. Et pour ce faire, il suffit d'employer les instructions logiques adéquates.

En réalité, dans le Z 80, des instructions dites «manipulation de bits» et «test de bits» existent «toutes faites» et permettent ainsi de «soulager» d'autant les programmes.

2. Manipulations de bits

L'instruction RES b_i, s ou RES b_i, m permet de mettre le bit désigné ($b_0 \leq b_i \leq b_7$) à l'état «0» ou effacement. Dans cette instruction l'opérande :

«s» désigne l'un des registres A, B, C, D, E, H ou L

«m» désigne un octet (une case mémoire) pointée par le contenu de la paire de registres HL ou par le contenu de l'un des registres index IX ou IY auquel s'ajoute le déplacement d. L'instruction SET b_i, s ou SET b_i, m permet de mettre le bit désigné ($b_0 \leq b_i \leq b_7$) à l'état «1». Dans cette instruction l'opérande a la même signification que précédemment.

L'ensemble du répertoire des instructions de manipulations de bits, se trouve résumé dans le tableau V.

DECALAGE ARITHMETIQUE							
SL A, «s» ou SL A, «m»			SR A, «s» ou SR A, «m»				
	C ← $b_7 \leftarrow b_0 \leftarrow$ «0»			$b_7 \rightarrow b_0 \rightarrow$ C			
r	1 1 0 0 0 0 1 0	1 0 1 1 0 r	CB 2-	r	1 1 0 0 0 0 1 0	1 0 1 1 1 r	CB 2-
(HL)	1 1 0 0 0 0 1 0	1 0 1 1 0 1 1 0	CB 26	(HL)	1 1 0 0 0 0 1 0	1 0 1 1 1 1 1 0	CB 2E
(IX + d)	1 1 0 1 1 1 0 0	1 1 0 1 1 0 1 1	DD CB	(IX + d)	1 1 0 1 1 1 0 0	1 1 0 1 1 0 1 1	DD CB
	$\leftarrow d \rightarrow$			$\leftarrow d \rightarrow$			
	0 0 1 0	0 1 1 0	26		0 0 1 0	1 1 1 0	2E
(IY + d)	1 1 1 1 1 1 0 0	1 1 0 1 1 0 1 1	FD CB	(IY + d)	1 1 1 1 1 1 0 0	1 1 0 1 1 0 1 1	FD CB
	$\leftarrow d \rightarrow$			$\leftarrow d \rightarrow$			
	0 0 1 0	0 1 1 0	26		0 0 1 0	1 1 1 0	2E

avec r : A = 111 D = 010 L = 101
 B = 000 E = 011
 C = 001 H = 100

DECALAGE LOGIQUE			
SRL «s» ou SRL «m»			
«0» → b_7 b_0 → C			
r	1 1 0 0	1 0 1 1	CB 3-
	0 0 1 1	1 r	
(HL)	1 1 0 0	1 0 1 1	CB 3E
	0 0 1 1	1 1 1 0	
(IX + d)	1 1 0 1	1 1 0 1	DD
	1 1 0 0	1 0 1 1	CB
	0 0 1 1	1 1 1 0	3E
(IY + d)	1 1 1 1	1 1 0 1	FD
	1 1 0 0	1 0 1 1	CB
	0 0 1 1	1 1 1 0	3E

Tableau II

Avec r : A = 111 D = 010 L = 101
 B = 000 E = 011
 C = 001 H = 100

ROTATION A TRAVERS LE REPORT C							
RL «s» ou RL «m»				RR «s» ou RR «m»			
r	1 1 0 0	1 0 1 1	CB 1-	r	1 1 0 0	1 0 1 1	CB 1-
	0 0 0 1	0 r			0 0 0 1	1 r	
(HL)	1 1 0 0	1 0 1 1	CB 16	(HL)	1 1 0 0	1 0 1 1	CB 1E
	0 0 0 1	0 1 1 0			0 0 0 1	1 1 1 0	
(IX + d)	1 1 0 1	1 1 0 1	DD	(IX + d)	1 1 0 1	1 1 0 1	DD
	1 1 0 0	1 0 1 1	CB			1 1 0 0	1 0 1 1
	0 0 0 1	0 1 1 0	16		0 0 0 1	1 1 1 0	1E
(IY + d)	1 1 1 1	1 1 0 1	FD	(IY + d)	1 1 1 1	1 1 0 1	FD
	1 1 0 0	1 0 1 1	CB			1 1 0 0	1 0 1 1
	0 0 0 1	0 1 1 0	16		0 0 0 1	1 1 1 0	1E

Tableau III

avec r : A = 111 D = 010 L = 101
 B = 000 E = 011
 C = 001 H = 100

3. Test de bit

L'instruction Bit b_i , s ou Bit b_i , m teste le bit désigné ($b_0 \leq b_i \leq b_7$). L'indicateur Z du registre F contiendra le complément (ou l'INVERSE) du bit ainsi désigné.

Dans cette instruction, l'opérande a la même signification que précédemment.

TEST DE BIT			
Bit b_i «s» ou bit b_i «m»			
r	1 1 0 0	1 0 1 1	CB
	0 1	← b → ← r →	
(HL)	1 1 0 0	1 0 1 1	CB
	0 1	← b → 1 1 0	
(IX + d)	1 1 0 1	1 1 0 1	DD
	1 1 0 0	1 0 1 1	CB
	0 1	← b → 1 1 0	
		← d →	
(IY + d)	1 1 1 1	1 1 0 1	FD
	1 1 0 0	1 0 1 1	CB
	0 1	← b → 1 1 0	
		← d →	

Tableau VI

Avec r : avec b :
 A = 111 0 = 000
 B = 000 1 = 001
 C = 001 2 = 010
 D = 010 3 = 011
 E = 011 4 = 100
 H = 100 5 = 101
 L = 101 6 = 110
 7 = 111

4. Exercices

1. Ecrire les instructions permettant de réaliser les opérations suivantes :

- Rotation à gauche sans report du contenu de D
- Décalage logique du contenu pointé par HL
- Décalage arithmétique à gauche du contenu de A
- Rotation à droite au travers de C du contenu pointé par (IX + 10d)
- Décalage arithmétique à droite du contenu de E
- Rotation à gauche sans report du contenu pointé par HL.

2. Le contenu des registres suivants sont tels que :

A = 8B D = EC
 B = C1 E = DD
 C = 9C F = 19

HL = 1A 3E et le contenu de la case d'adresse mémoire 1A3E : 5A.

Quels sont les contenus des registres et le contenu de l'adresse 1A3E,

ROTATION SANS LE REPORT C							
RLC «s» ou RLC «m»			RRC «s» ou RRC «m»				
r	1 1 0 0 0 0 0 0	1 0 1 1 0 r	CB 0-	r	1 1 0 0 0 0 0 0	1 0 1 1 1 r	CB 0-
(HL)	1 1 0 0 0 0 0 0	1 0 1 1 0 1 1 0	CB 06	(HL)	1 1 0 0 0 0 0 0	1 0 1 1 1 1 1 0	CB 0E
(IX + d)	1 1 0 1 1 1 0 0 0 0 0 0	1 1 0 1 1 0 1 1 0 1 1 0	DD CB 06	(IX + d)	1 1 0 1 1 1 0 0 0 0 0 0	1 1 0 1 1 0 1 1 1 1 1 0	DD CB 0E
(IY + d)	1 1 1 1 1 1 0 0 0 0 0 0	1 1 0 1 1 0 1 1 0 1 1 0	FD CB 06	(IY + d)	1 1 1 1 1 1 0 0 0 0 0 0	1 1 0 1 1 0 1 1 1 1 1 0	FD CB 0E

Tableau IV

Avec r : A = 111 D = 010 L = 101
 B = 000 E = 011
 C = 001 H = 100

MANIPULATION DE BIT							
Res b _i , s ou Res b _i , m			Set b _i , s ou Set b _i , m				
r	1 1 0 0 1 0	1 0 1 1 ← b → ← r →	CB	r	1 1 0 0 1 1	1 0 1 1 ← b → ← r →	CB
(HL)	1 1 0 0 1 0	1 0 1 1 ← b → 1 1 0	CB	(HL)	1 1 0 1 1	1 0 1 1 ← b → 1 1 0	CB
(IX + d)	1 1 0 1 1 1 0 0 1 0	1 1 0 1 1 0 1 1 ← d → ← b → 1 1 0	DD CB	(IX + d)	1 1 0 1 1 1 0 0 1 1	1 1 0 1 1 0 1 1 ← d → ← b → 1 1 0	DD CB
(IY + d)	1 1 1 1 1 1 0 0 1 0	1 1 0 1 1 0 1 1 ← d → ← b → 1 1 0	FD CB	(IY + d)	1 1 1 1 1 1 0 0 1 1	1 1 0 1 1 0 1 1 ← d → ← b → 1 1 0	FD CB

Tableau V

avec r : A = 111 E = 011
 B = 000 H = 100
 C = 001 L = 101
 D = 010

b : 0-000 4-100
 1-001 5-101
 2-010 6-110
 3-011 7-111

après l'exécution des instructions suivantes :

- CB 1A
- CB 07
- CB 2E
- CB 38
- CB 14

Notez le contenu de C après chaque instruction.

3. Indiquez les codes opératoires qui réalisent les instructions suivantes :

- a) Mettre à 1 le bit 3 de E
- b) Mettre à «0» le bit 7 de A
- c) Mettre à «0» le bit 2 de F
- d) Tester le bit 5 de C
- e) Mettre à «0» le bit 7 de l'emplacement pointé par (IX + 09)
- f) Mettre à «1» le bit 4 de B
- g) Mettre à «1» le bit 5 de D
- h) Tester le bit 0 de H

IV. INSTRUCTIONS «CONTROLE DU CPU»

Les instructions qui vont suivre agissent directement sur l'unité centrale et contrôlent son fonctionnement.

1. Instruction NOP

La première est l'instruction NOP qui signifie «pas d'opération» (NO OPERATION) dont le code hexadécimal est 00.

Pendant l'exécution de cette instruction, seul le déroulement du cycle du microprocesseur a lieu. Le microprocesseur se trouve dans une position de «Stand-by». Il génère les signaux normalement, notamment ceux indispensables au rafraîchissement des mémoires dynamiques RAM. Le compteur ordinal est incrémenté. **Aucune «opération» n'est effectuée.**

Une telle instruction, qui apparemment «ne fait rien» est sans objet. Eh bien non ! Nous allons donner deux exemples qui montrent l'intérêt de cette instruction.

Cas 1

Au cours de la mise au point d'un programme, il arrive bien souvent qu'il soit nécessaire de supprimer un ou plusieurs octets, sans pour autant réécrire tout le programme. Il peut s'agir d'instructions en trop ou d'instructions que l'on veut momentanément supprimer pour effectuer des essais de «mise au point».

Dans ce cas, il suffit de remplacer les octets concernés par des instructions «NOP» sans autre changement.

C'est une manière de «gommer» provisoirement ou définitivement une ou plusieurs instructions.

Cas 2

Une deuxième application dans laquelle on trouve fréquemment les instructions NOP sont les boucles de temporisation.

Le registre B ayant été chargé avec N (figure 146), par exemple 64H (soit 100d), le programme sera temporairement suspendu pendant l'exécution de $2 \times N$ fois l'instruction «NOP». Si l'horloge est de 4 MHz, le temps d'exécution de NOP est de 1 μ s, ce qui va donner lieu à une temporisation d'environ 525 μ s (le T.E. de DJNZ étant de 3,25 μ s avec l'horloge à 4 MHz).

Pour augmenter ou diminuer le «temps de suspension» du déroulement du programme, il suffit d'agir sur le nombre N ou la quantité de NOP introduite dans la boucle.

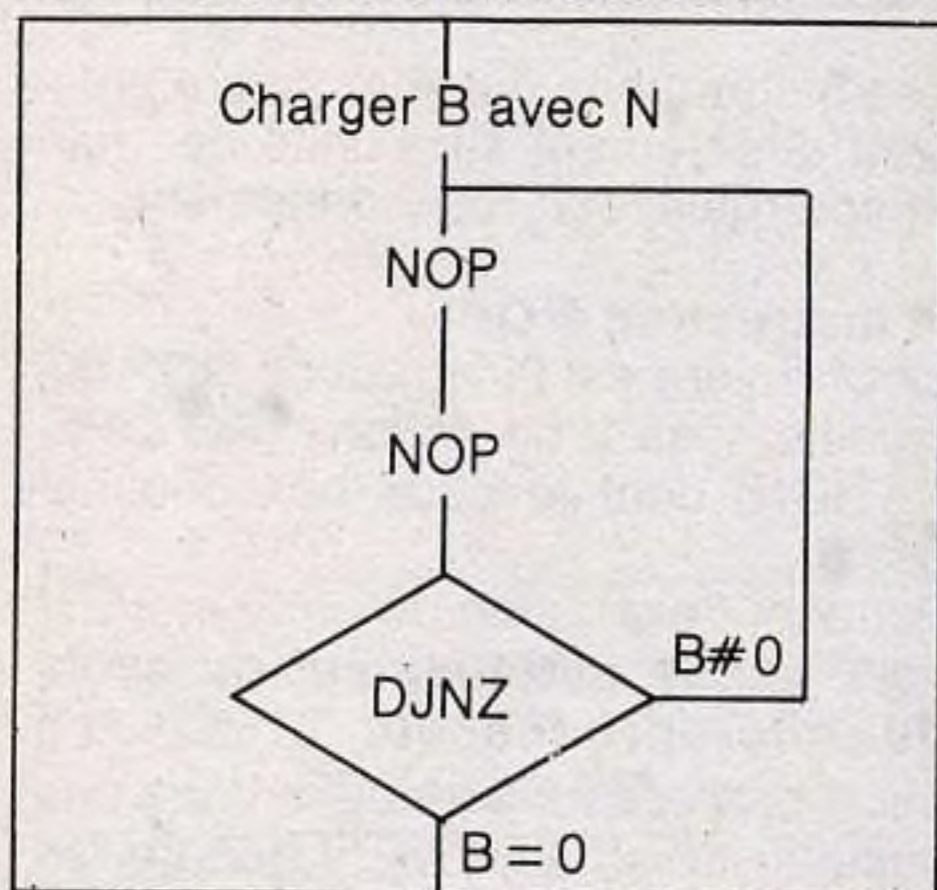


Figure 146

2. Instruction «HALT»

Une deuxième méthode pour suspendre le déroulement d'un programme est l'instruction HALT dont le code hexadécimal est 76.

L'exécution de l'instruction HALT amène le CPU à **exécuter continuellement les instructions NOP** : c'est une **boucle de temporisation sans fin**. Le CPU ne peut sortir de lui-même de cette boucle ; il faut un **stimuli provenant d'une commande extérieure**.

En reprenant la configuration hardware, telle qu'elle fut présentée figure 44 (LM 11, p. 49), le nombre d'entrées qui peuvent fournir l'impulsion qui permettra au CPU **de continuer est en réalité limité à trois**. Ce ne peut être que : Reset, NMI ou INT. Une commande RESET (active sur le flanc descendant) force le compteur ordinal à 00 H. Le CPU démarre

comme s'il s'agissait d'une première initialisation.

Une commande NMI (active sur le flanc descendant) force le compteur ordinal à se rendre en 0066 H (saut en première page). Le contenu du compteur ordinal (qui pointe l'instruction qui suit HALT) est sauvegardé dans la pile.

Après exécution de la routine, placée en 0066 H, le déroulement du programme reprend là où il avait été interrompu par l'exécution de l'instruction HALT.

Une commande INT (active sur le flanc descendant) permet la poursuite du fonctionnement du CPU, si et seulement si les interruptions ont été préalablement autorisées par le logiciel (donc inclus dans le programme). Ce signal INT est souvent généré par le ou les circuits d'entrée/sortie comme nous le montrerons dans l'exemple ci-après.

Lorsque le CPU honore une demande d'interruption, il peut y répondre selon l'un des trois modes qui seront décrits en détails dans le paragraphe suivant l'exemple.

Exemple :

Une imprimante de vingt caractères de type thermique par exemple, est connectée à un système par l'intermédiaire d'un circuit périphérique d'entrée/sortie (figure 147).

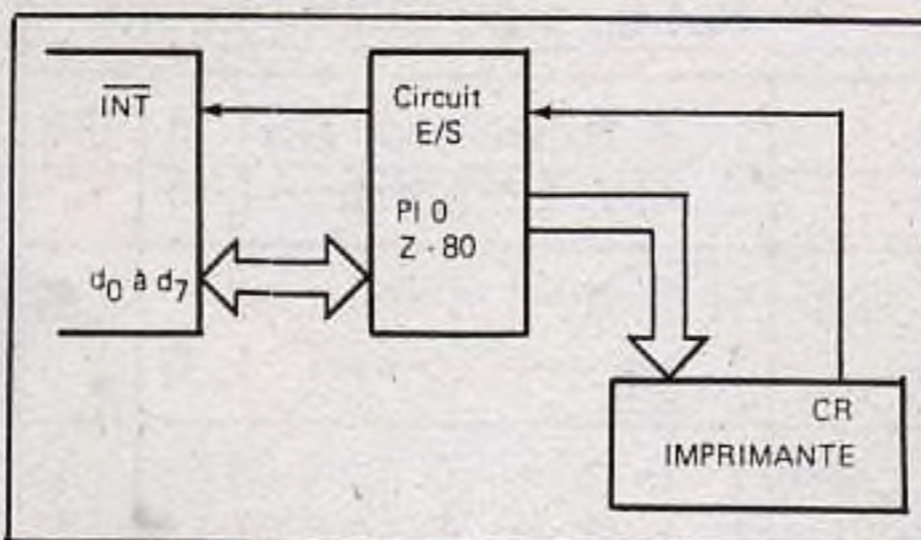


Figure 147

La vitesse de l'imprimante est de 4 lignes/seconde, ce qui correspond à un temps d'impression de 250 ms par ligne.

L'imprimante possède un registre d'entrée (ou buffer) qui stocke la ligne à imprimer.

Le temps de chargement du registre BUFFER qui est en fait un transfert de 20 octets est de l'ordre de 3 à 400 μ s.

Dans le programme, après l'envoi des 20 caractères qui représentent une ligne, une instruction HALT sera placée : le CPU attendra le message «fin de ligne» ou CR (Carriage Return ou retour chariot) pour envoyer le bloc de caractères suivant. Et ainsi

de suite jusqu'à ce que tout le texte soit transmis.

Pourquoi ne pas remplacer l'instruction HALT par une boucle de temporisation ?

Le temps d'impression de 250 millisecondes correspond à un régime continu. Pour un fonctionnement ligne par ligne, le temps est plus long. D'autre part, il s'agit là d'une valeur typique, la valeur réelle variant d'une imprimante à l'autre, ainsi que tout au long de sa durée de vie. Si bien que si on utilisait une boucle de temporisation, il faudrait utiliser le temps maximal, et par conséquent au détriment des performances de rapidité de l'imprimante.

Par contre, en utilisant l'instruction HALT, **notre système s'adapte automatiquement** au temps d'impression de l'imprimante ; ce qui permet d'obtenir dans tous les cas la vitesse optimale, quelle que soit l'imprimante, son mode de fonctionnement ou son évolution dans le temps.

3. Instructions d'autorisation et d'interdiction des interruptions

Pour qu'une demande d'interruption masquable soit honorée par le CPU, il faut au préalable que le logiciel l'autorise. Cette autorisation s'effectue par l'exécution de l'instruction EI (Enable Interrupt) dont le code hexadécimal est : FB.

Inversement, le programmeur peut à tout moment interdire l'autorisation des interruptions (par exemple pendant une boucle de temporisation). Pour cela, il place dans son programme l'instruction DI (Disable Interrupt) dont le code hexadécimal est : F3.

A noter qu'à la mise sous tension (et après une action RESET) les interruptions ne sont pas autorisées.

4. Modes d'interruptions

Le Z80 possède trois modes différents d'interruption qui peuvent être sélectionnés par 3 instructions :

IM0 ou Mode 0 (ED 46)

Dans ce mode d'exploitation, le circuit périphérique générant l'interruption place en même temps un octet sur le bus de donnée correspondant à l'un des codes des instructions Restart.

Le CPU autorise le compteur ordinal à se charger avec cette adresse après avoir effectué une sauvegarde dans la pile.

IM1 ou Mode 1 (ED 56)

Ce mode est une version simplifiée du précédent. Le CPU exécute automatiquement un saut à l'adresse 0038 H (Rest 56), en ayant au préalable sauvegardé le contenu du PC dans la pile.

IM2 ou Mode 2 (ED 5E)

Ce mode d'exploitation est, par contre, plus sophistiqué et beaucoup plus puissant que les deux autres. **Il permet au CPU de se rendre à n'importe quelle adresse paire du champ d'adresse.**

Initialement, le registre I (le registre Interruptions) a été chargé avec les 8 bits de poids fort de l'adresse à atteindre. Tandis que les 8 bits de poids faible (en réalité 7, le plus faible étant 0) **sont fournis pas le circuit interrompant.**

Le CPU forme ainsi une adresse de 16 bits qui sera chargée dans le compteur ordinal après que son contenu soit sauvegardé dans la pile.

Le tableau suivant (figure 148) résume les différentes instructions qui permettent le contrôle du CPU.

NOP	00
HALT	76
EI	F3
DI	FB
IMO	ED-46
IMI	ED-56
IM2	ED-5E

Figure 148

V. INSTRUCTIONS ENTREE-SORTIE

1. Introduction

Tous les échanges que nous avons étudiés jusqu'à présent s'effectuaient soit à l'intérieur même du microprocesseur, échanges entre registres, soit avec les mémoires ROM's ou RAM's.

Pour ces derniers, le canal utilisé est le «Bus de données». Dans cette partie nous allons étudier les échanges avec les unités extérieures.

Le transfert des informations qu'elles soient des données d'«entrée» ou de «sortie» transitent toujours par le même canal «le bus de données». Il est inconcevable qu'une unité périphérique (surtout si elle est lente) immobilise cette ligne

de communication pendant tout le temps nécessaire pour accomplir sa tâche. C'est pourquoi un circuit d'interface est généralement placé entre le système et le périphérique de manière à ne pas nuire à l'efficacité du CPU. Le tout restant bien entendu **sous le contrôle du CPU par des échanges de signaux via le bus de commandes.**

Un boîtier d'interface (40 broches) peut contenir 2 (PIO-Z80) ou 3 (8255, Intel) circuits d'entrées-sorties parallèles (figure 12, LM 9).

La configuration fonctionnelle d'un circuit d'interface est programmée lors de l'initialisation du système. Chaque circuit doit être programmé en «entrée» ou en «sortie» ou bien encore en «entrée-sortie» avec les lignes de commandes correspondantes.

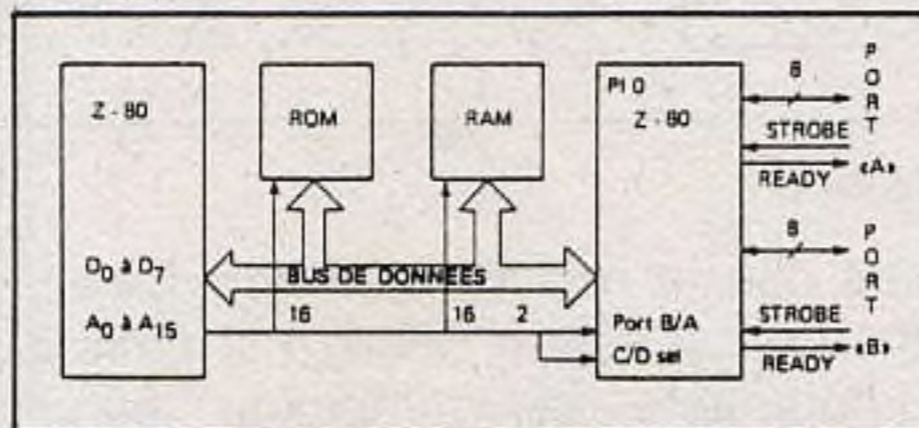


Figure 149

Pour une bonne compréhension des instructions qui vont suivre, nous décrivons succinctement le circuit PIO-Z80. Ce circuit d'interface est essentiellement constitué de deux circuits (deux voies) indépendants appelés «PORT A» et «PORT B».

Chaque «PORT» est avant tout un registre 8 bits, bidirectionnel. Il peut être mis en liaison avec le «bus de données» donc avec le CPU au moyen de signaux d'échange générés par le Z80 (donc sans contrôle du programme).

La sélection du port s'effectue au moyen d'un fil d'adresse, noté «Port B/A sel». Lorsque le niveau est bas, le port B est sélectionné ; lorsque le niveau est haut, le port A est sélectionné.

La configuration fonctionnelle que l'on veut donner à chaque port s'effectue **à l'aide du mot de contrôle**, placé dans le registre de commande du circuit d'interface. Une deuxième entrée, notée : «C/D sel» permet de charger ce registre.

Un niveau bas sur cette broche signifie que le bus de données est utilisé **pour transférer une donnée entre le CPU et l'un des «ports»**, tandis qu'un niveau haut signifie que **la donnée est un mot de commande.**

Cette fonction de contrôle est souvent assurée par le bit A_1 du bus d'adresse, tandis que le choix du port est souvent assuré par le bit A_0 (figure 150).

	A_1	A_0	«n»
Port A	0	1	1
Port B	0	0	0
Mot de commande	1	X	

Figure 150

Le mot de commande définit entre autres, si un port donné travaille en mode :

- ENTREE
- SORTIE
- BIDIRECTIONNEL

Deux signaux d'échanges (READY et STROBE) supplémentaires sont disponibles au niveau de chaque port pour synchroniser les échanges avec l'extérieur.

Vis-à-vis du CPU, les «ports» se comportent comme l'une quelconque des cases mémoires (ainsi que le registre de commande). Cependant, comme les signaux de contrôle sont spécifiques par rapport à un emplacement mémoire, les instructions seront : OUT (ou sortie) pour l'écriture d'un mot

IN (ou entrée) pour la lecture d'une donnée.

Généralement, un microprocesseur peut sélectionner jusqu'à 256 registres (entrée-sortie ou commande), ce qui revient à dire que seuls les 8 fils du bus d'adresse (A_0 à A_7) sont utilisables.

L'adresse du registre (figure 150) sera noté «n» avec $0 \leq n \leq 255$.

2. Instructions d'ENTREE

a) Chargement de l'accumulateur à partir d'un port (n)

Opération :

Le contenu du port (n) est chargé dans l'accumulateur A.

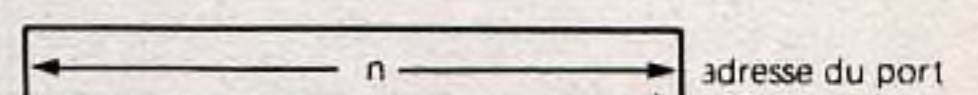
Le contenu de A apparaît sur les bits d'adresses A_8 à A_{15} .

Code mnémorique :

$A \leftarrow (n)$

1	1	0	1	1	0	1	1	DB
---	---	---	---	---	---	---	---	----

Code hexadécimal :



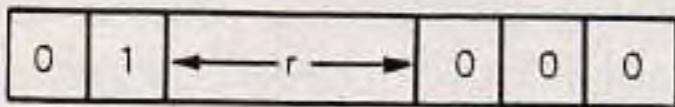
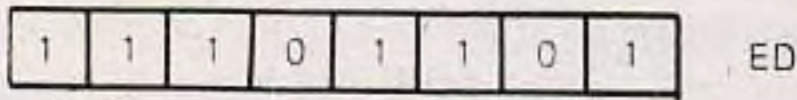
b) Chargement d'un registre r à partir d'un port désigné par le contenu de C

Opération :

Le contenu du port (n) est chargé dans l'un des registres A, B, C, D, E, H ou L. L'adresse (n) du port est l'octet contenu dans le registre C.

Code mnémorique : $r \leftarrow (C)$

Codes machines :



avec r : A = 111 E = 011
 B = 000 H = 100
 C = 001 L = 101
 D = 010

3. Instructions de SORTIE

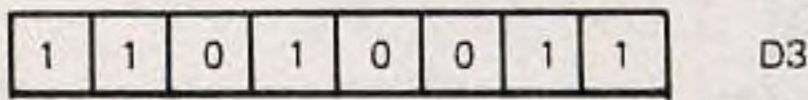
a) Transfert du contenu de A dans un port (n)

Opération :

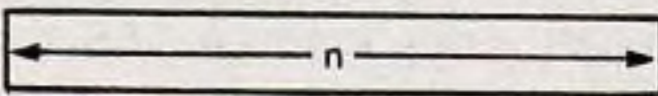
Le contenu de l'accumulateur A est transféré dans le port «n».

Code mnémorique :

$(n) \leftarrow A$



Code hexadécimal :



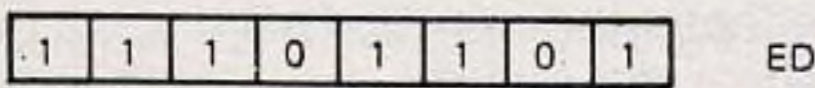
b) Transfert du contenu de r dans un port désigné par le contenu de C

Opération :

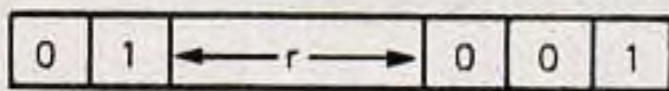
Le contenu du registre r est transféré dans le port d'adresse «n». L'adresse n étant l'octet contenu dans le registre C.

Code mnémorique :

$(C) \leftarrow r$



Code hexadécimal :



avec r : A = 111 D = 010
 B = 000 E = 011
 C = 001 H = 100
 L = 101

4. Exemples d'application

a) Exemple 1

Problème :

Une imprimante de 40 caractères est connectée au port B d'un circuit d'interface PI0-Z80. Chaque caractère est représenté en code ASCII par 1 octet. La ligne à imprimer est stockée à partir de l'adresse 1800 H à 1827 H (40 emplacements) (figure 151).

Ecrire le programme qui réalise le transfert.

1.a. Aspect hardware

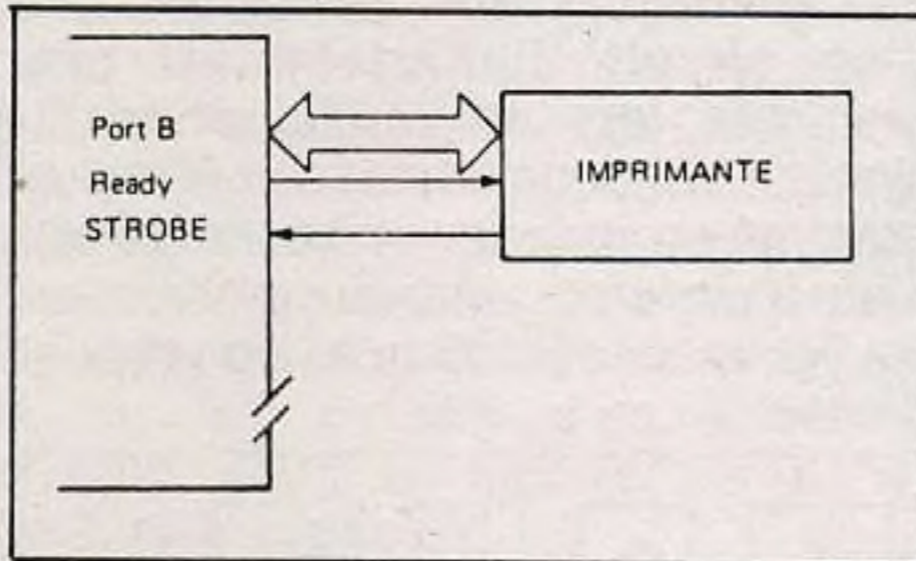


Figure 151

Dès que le registre du «Port B» est chargé, un signal Ready est émis à l'imprimante pour indiquer qu'un octet est prêt. Quand celle-ci après avoir prélevé le mot, l'aura placé dans le registre, elle envoie un signal «STROBE» pour signifier qu'elle est prête à accepter le caractère suivant.

Ainsi, les 40 caractères sont transférés dans le registre imprimante, la ligne est alors imprimée, l'imprimante est prête pour la suivante.

1.b. Aspect software

Le programme est présenté en mnémorique par la figure 152.

Les 3 premières instructions constituent la partie «initialisation».

Quand le transfert de la case mémoire pointée par HL dans l'accumulateur a eu lieu, la paire de registres HL est incrémentée. Le contenu de A, c'est-à-dire 1 caractère, est transféré dans le port de sortie.

Etant donné que le temps nécessaire à l'imprimante pour stocker et ranger l'octet est de quelques dizaines de microsecondes (40 environ), il va falloir suspendre le déroulement du programme par une instruction HALT. Cependant, il faut prendre soin auparavant de **valider les interruptions**.

Lorsque l'imprimante renvoie le message «STROBE» (message d'acquiescement), une «demande d'interruption» est émise par le circuit d'interface. Les interruptions étant validées, le

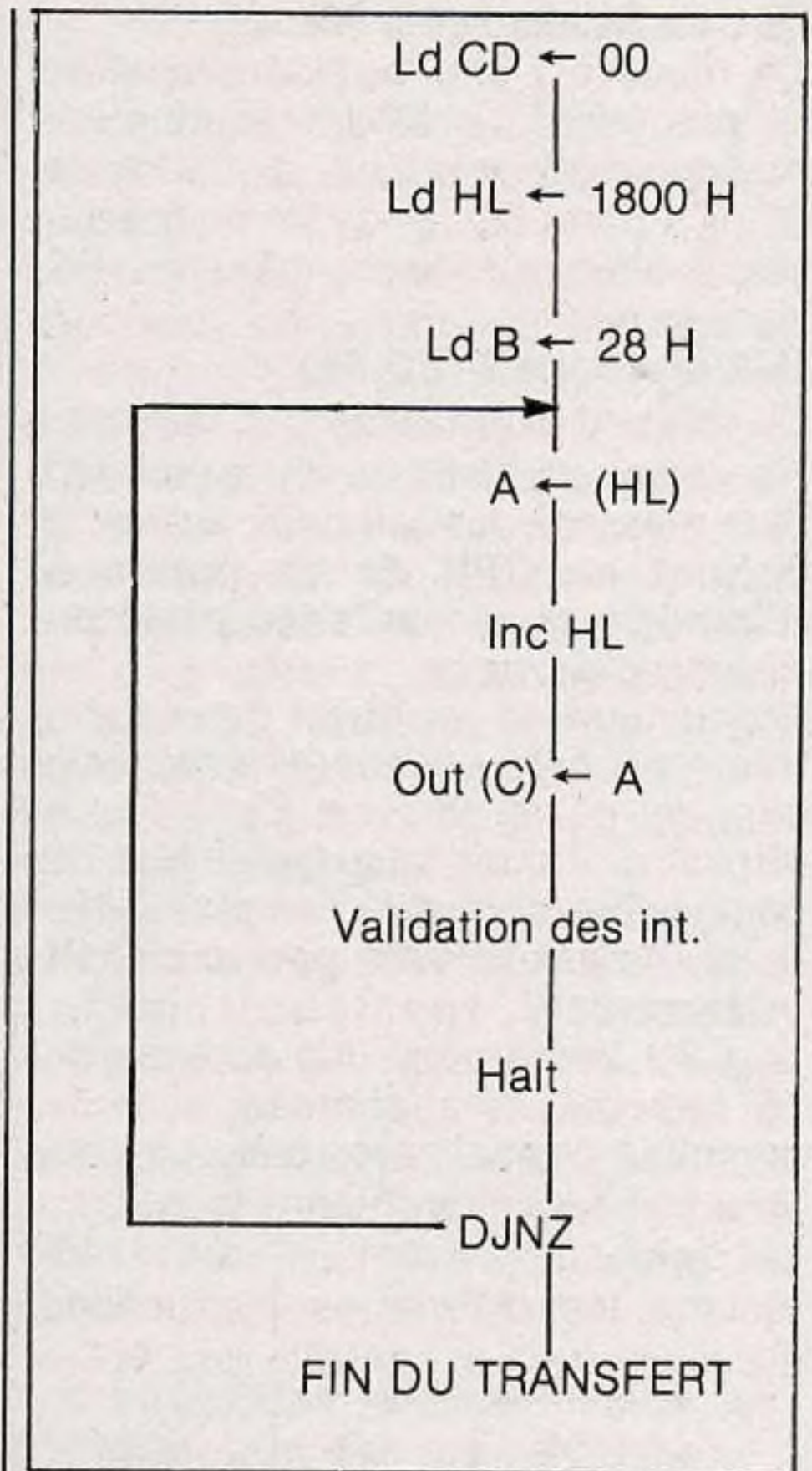


Figure 152

CPU peut l'honorer. Dans le cas présent, il suffit de poursuivre le programme (exécution de DJNZ).

Donc nous choisissons le mode d'interruption le plus simple, le mode 1, qui charge le compteur ordinal avec 0038 H. Dans cette case mémoire nous aurons placé au préalable l'instruction RET (C9).

Le PC sera chargé avec les 2 octets de la pile, c'est-à-dire l'adresse de l'instruction DJNZ.

Pour une parfaite compréhension de la boucle, nous aurions pu écrire celle-ci sous la forme de la figure 153.

Avant d'étudier les instructions de «transfert par bloc» qui peuvent simplifier notre problème, étudions un second exemple dérivé du précédent.

2. Exemple 2

Problème

Le système microprocesseur est le cœur d'un banc de test, qui reçoit des paramètres, les traite et périodiquement les envoie ensuite par bloc de 100 octets à un mini-ordinateur connecté à ce banc de test (figure 154). (Par exemple, le mini-ordinateur peut être connecté à 10

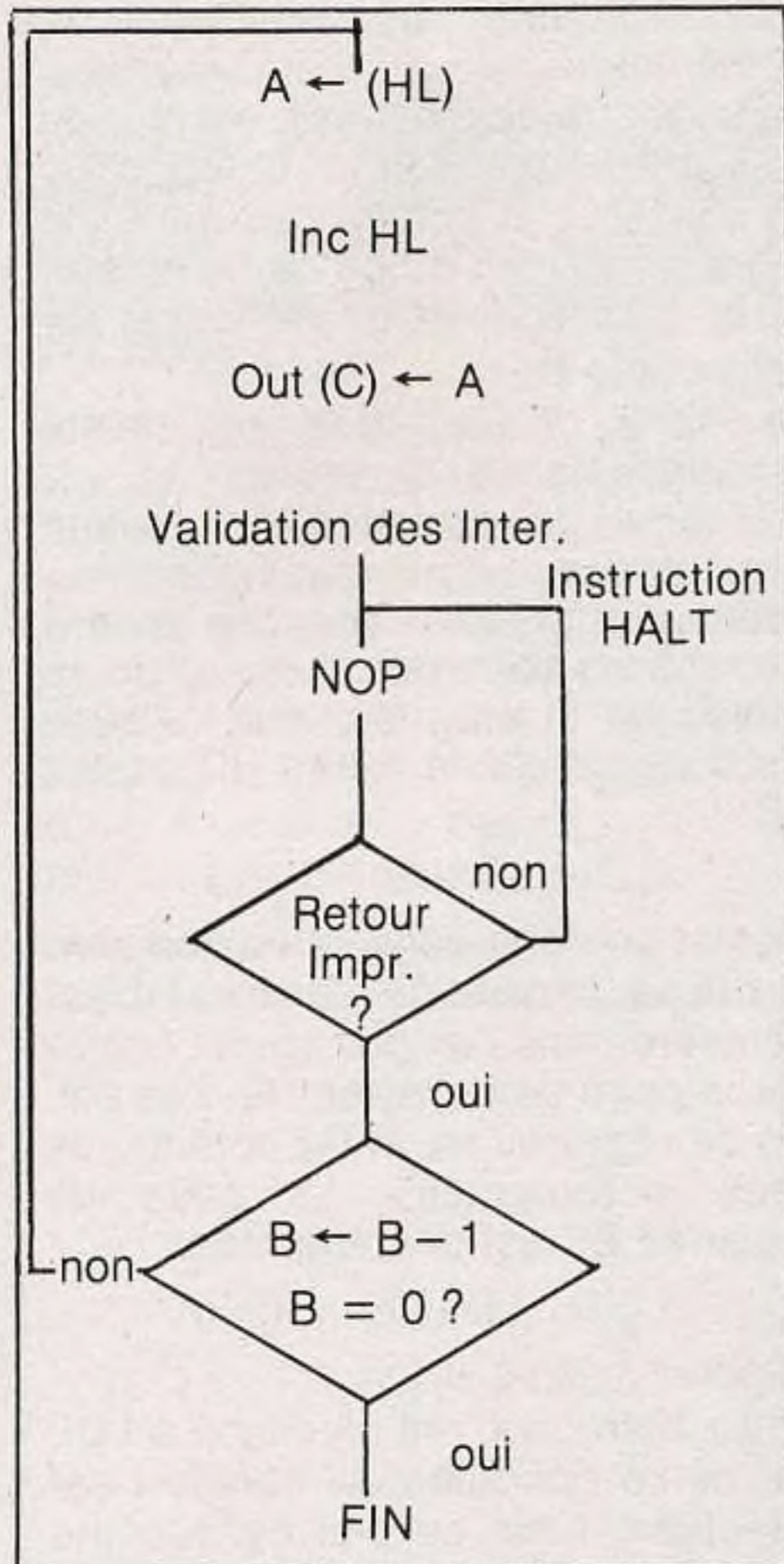


Figure 153

bancs de test : il centralise les informations déjà traitées par le système à microprocesseur).

Le temps de saisie d'un octet par le mini-ordinateur étant de 1 microseconde, **il n'est pas nécessaire d'inclure dans le programme une boucle de temporisation (Halt).**

Ecrire le programme correspondant pour ce problème.

2.a. Aspect hardware

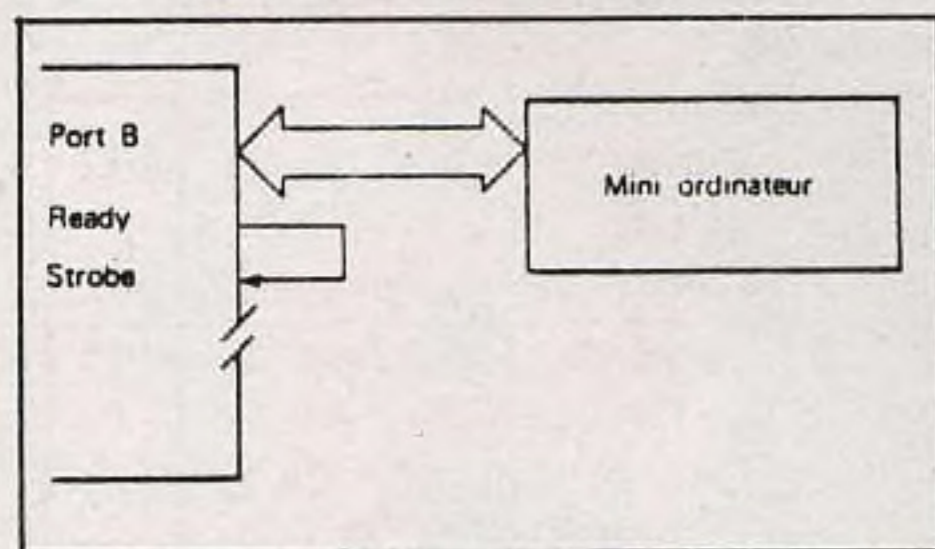


Figure 154

Dès que le registre du «Port B» est chargé, l'octet est saisi par le mini-ordinateur sans qu'il soit nécessaire d'attendre. (En réalité, le temps nécessaire pour placer un autre octet de l'ordre de 10 μs est largement suffisant au temps de saisie par le mini 1 μs).

On notera que les signaux Ready (sortie) et Strobe (entrée) sont connectés entre eux.

2.b. Aspect software

Le programme est présenté en mnémoniques par la figure 155. A noter que pendant la boucle de transfert, les interruptions sont interdites (DI). Lorsque la boucle est terminée, c'est-à-dire que le bloc de 100 octets (64 H) a été transféré, les interruptions sont à nouveau autorisées.

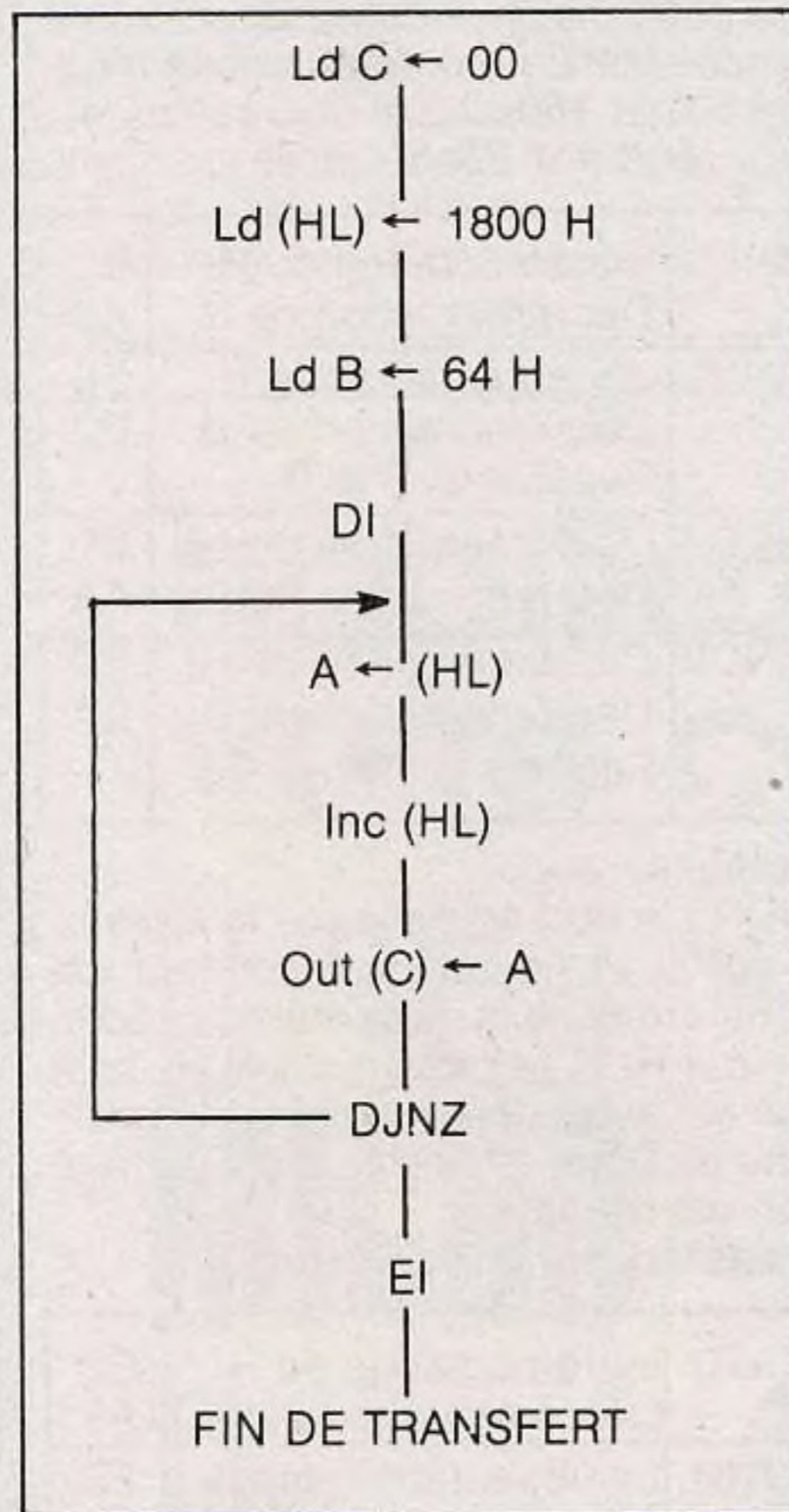


Figure 155

5. Instructions de transfert par bloc

L'ensemble des 4 instructions qui constituent la boucle de transfert peut être remplacé par une instruction unique notée OTIR dont le code hexadécimal est ED 83.

Description

Le contenu de l'emplacement mémoire pointé par la paire de registres HL est transféré dans le Port de sortie adressé par le contenu du registre C. Le registre B est décrémenté, et le registre HL incrémenté. Si B ≠ 0, l'instruction est répétée (compteur ordinal décrémenté de 2).

Si B = 0, l'instruction suivante est exécutée.

Le programme est ainsi réduit à celui de la figure 156.

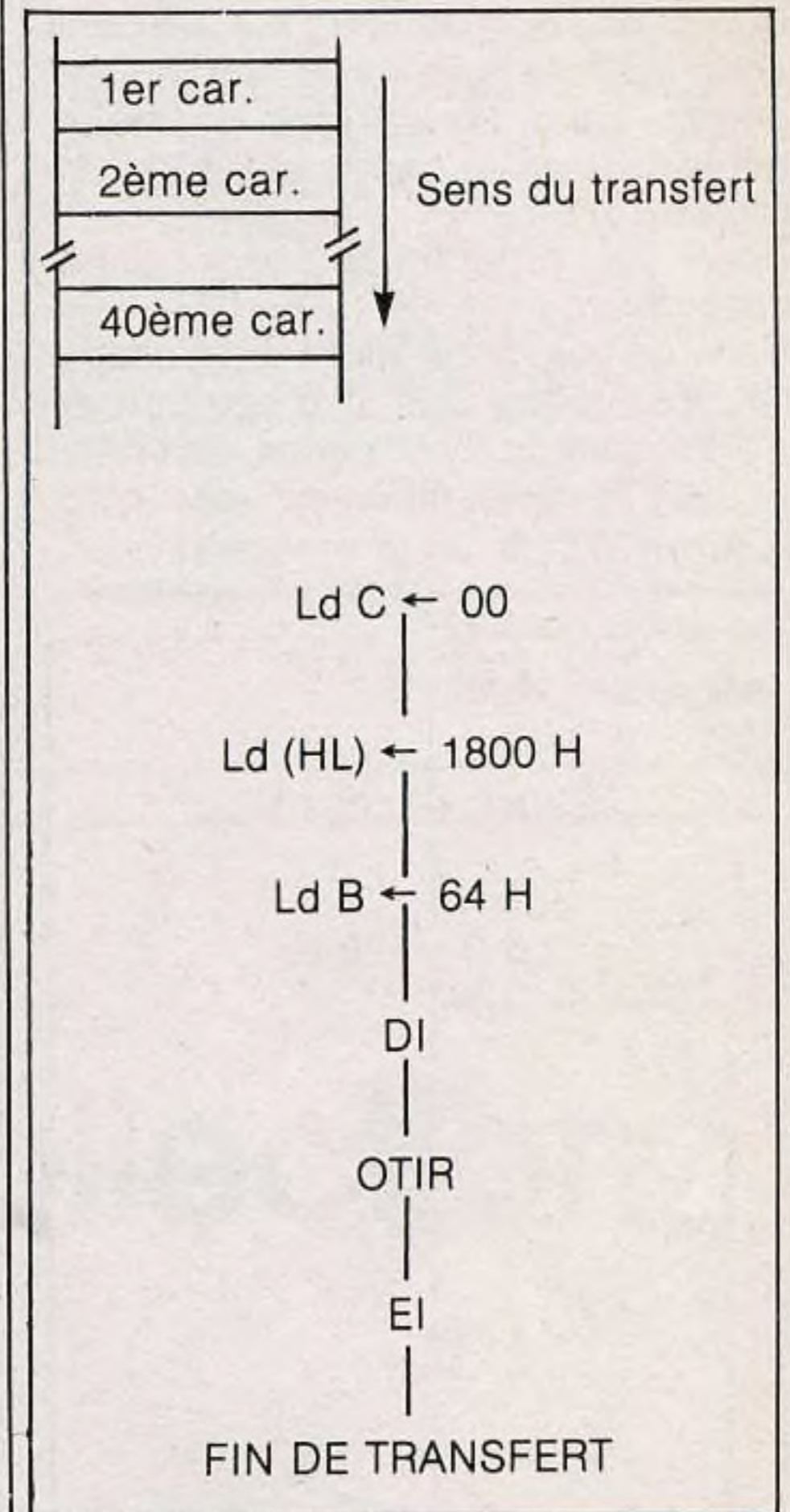


Figure 156

Une seconde instruction analogue à «OTIR», mais dans laquelle **au lieu d'incrémenter (I), la paire de registres HL, celle-ci est décrémentée (D)** est disponible, et se note «OTDR». Le code hexadécimal correspondant est : ED BB.

Dans le premier exemple (figure 133), la boucle ne peut pas être remplacée par l'instruction OTIR. Il existe cependant une instruction similaire **mais sans la fonction saut intégré** : elle se note OUTI, dont le code hexadécimal est : ED A3.

Description :

Le contenu de l'emplacement mémoire pointée par la paire de registres HL est transféré dans le Port de sortie adressé par le contenu du registre C. Le registre B est décrémenté, et le registre HL est incrémenté.

Une seconde instruction analogue à «OUTI», mais dans laquelle la paire de registres HL est décrémentée (D) au lieu d'être incrémentée (I) est disponible. Elle se note «OUT D». Le

code hexadécimal correspondant est ED AB.

Exemple :

Supposons que les 40 caractères qui constituent la ligne à imprimer soient placés dans l'ordre des adresses décroissantes à partir de 1827 H (figure 157).

Ecrire le programme dans cette application.

Dans ce cas, nous allons faire usage de l'instruction OUT D d'une part et d'autre part de l'instruction saut relatif si B est différent de zéro (figure 157).

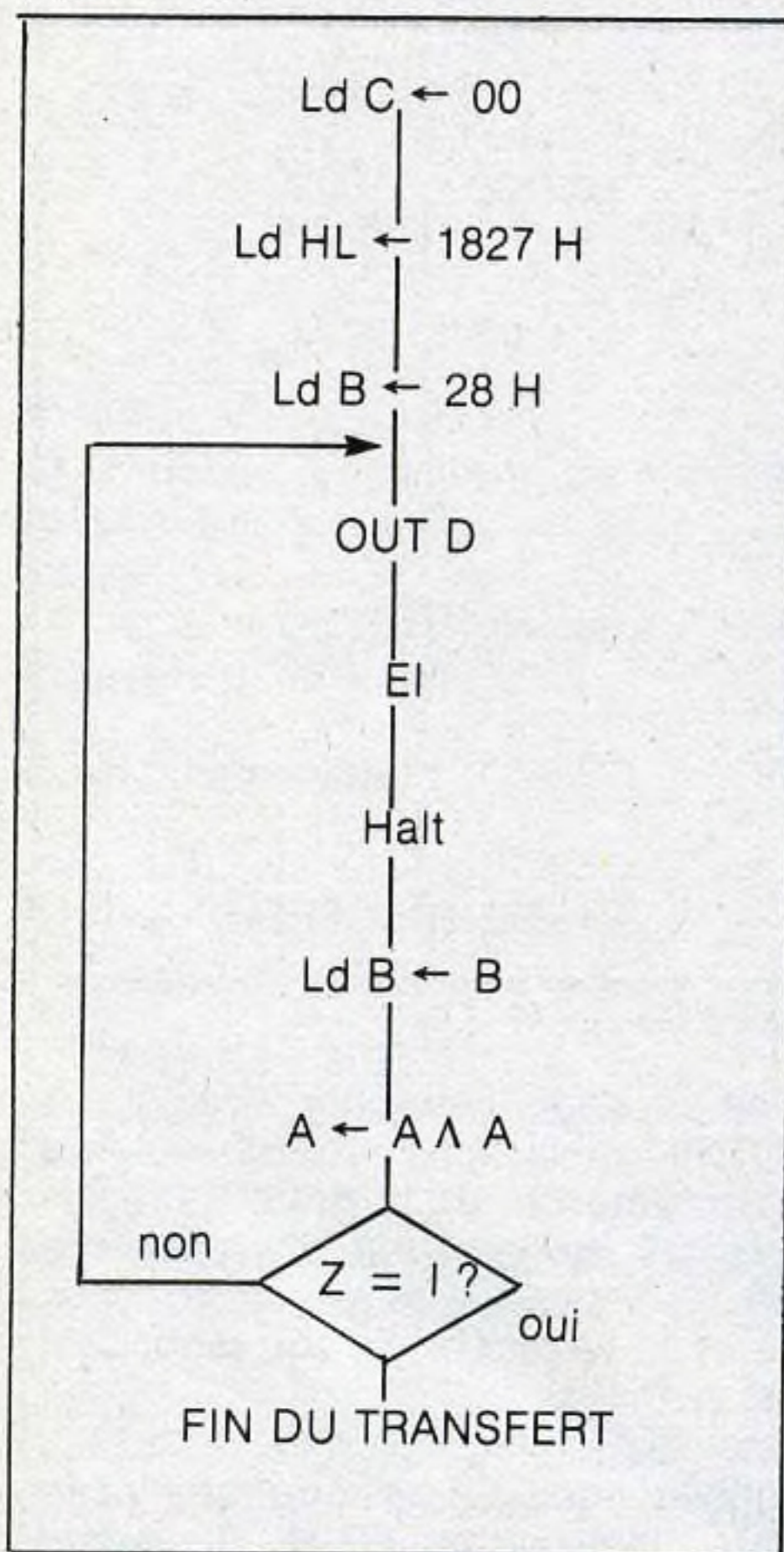


Figure 157

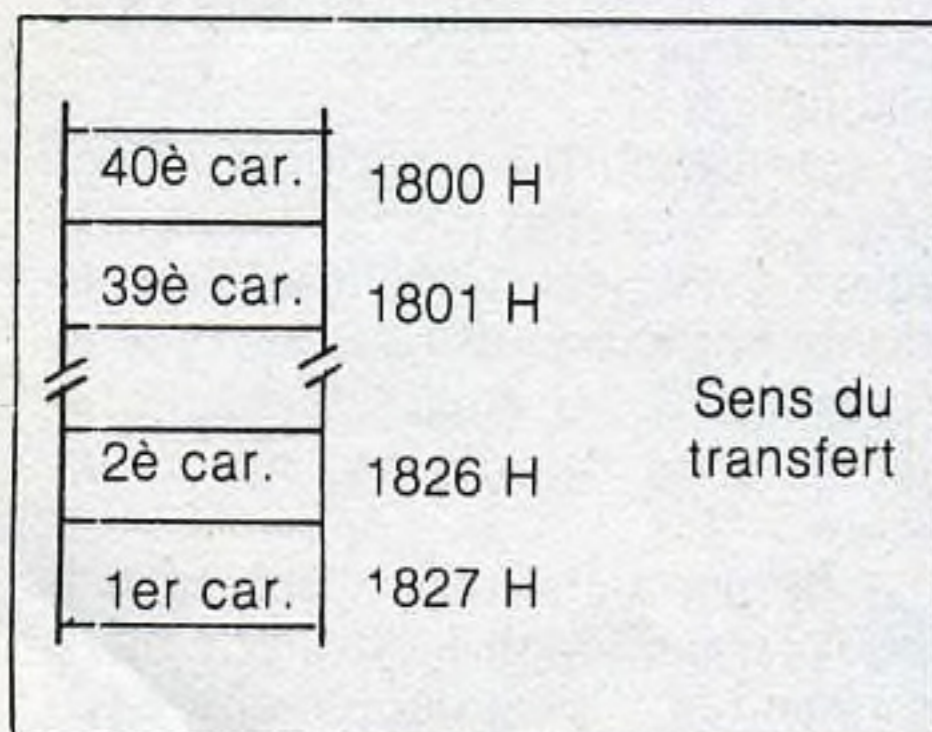


Figure 158

Essayez de trouver d'autres solutions plus simples pour effectuer le test B = 0. En employant par exemple l'instruction DJNZ.

6. Instructions de transfert par BLOC, ENTREE-SORTIE

Nous venons de voir en détail les 4 instructions de transfert par bloc pour la «sortie» d'un paquet d'octets. Il existe 4 autres instructions analogues pour l'«entrée» d'un bloc d'un maximum de 256 octets.

Les deux tableaux suivants résument l'ensemble de ces instructions (figures 159 et 160).

Transfert par Bloc-Entrée

INI	Incrémentation de HL Décrémentation de B	ED A2
INIR	Incrémentation de HL Décrémentation de B Répétée si B ≠ 0	ED B2
IND	Décrémentation de HL Décrémentation de B	ED AA
INDR	Décrémentation de HL Décrémentation de B Répétée si B ≠ 0	ED BA

Exemple

INDR : le port adressé par le registre C est lu et l'octet qu'il contient est transféré dans la case mémoire pointée par HL. Les registres B et HL sont l'un et l'autre décrémentés. Si le contenu de B est différent de 0, l'instruction est répétée.

Transfert par Bloc-Sortie

OUTI	Incrémentation de HL Décrémentation de B	ED A3
OTIR	Incrémentation de HL Décrémentation de B Répétée si B ≠ 0	ED B3
OUTD	Décrémentation de HL Décrémentation de B	ED AB
OTDR	Décrémentation de HL Décrémentation de B Répétée si B ≠ 0	ED BB

Figure 159

7. Instructions de transfert par bloc dans la mémoire

Nous venons d'étudier les 8 instructions de transfert par bloc qui mettent en jeu l'un des ports et un emplacement mémoire.

Nous allons présenter un autre ensemble de 4 instructions bâties sur le même principe, mais qui permettent d'effectuer des transferts de

blocs d'octets à l'intérieur même de la mémoire.

La taille du bloc d'octets n'est plus de 256 maximum (soit 1 octet) mais 65 536 (soit 2 octets). La paire de registres BC fait office de compteur, et la détection de 0 s'effectue sur l'ensemble BC.

La paire de registres HL pointe l'emplacement de la source.

La paire de registres DE pointe l'emplacement de la destination.

Avant d'effectuer l'une des quatre instructions suivantes, il est indispensable que le programmeur initialise les 3 paires de registres BC, HL et DE.

LDI : LD (DE) ← (HL)

L'octet contenu dans l'emplacement pointé par la paire de registres HL est transféré dans l'emplacement pointé par la paire de registres DE. Les paires de registres HL et DE sont toutes deux incrémentées. La paire de registres BC est décrémentée.

LDIR : LD (DE) ← (HL)

Répéter jusqu'à BC = 0.

Cette instruction est identique à LDI. La même opération de transfert est effectuée, mais celle-ci est répétée jusqu'à ce que la paire de registres BC soit nulle.

LDD et LDDR

Ces deux instructions sont respectivement identiques à LDI et LDIR, avec la seule différence que les paires de registres HL et DE au lieu d'être incrémentées sont dans ce cas décrémentées.

Le tableau suivant résume les instructions de transfert par bloc à l'intérieur de la mémoire (figure 159).

Transfert par bloc dans la mémoire

LDI	Incrémentation de HL et DE Décrémentation de BC	ED A0
LDIR	Incrémentation de HL et DE Décrémentation de BC Répétée si BC ≠ 0	ED B0
LDD	Décrémentation de HL et DE Décrémentation de BC	ED A8
LDDR	Décrémentation de HL et DE Décrémentation de BC Répétée si BC ≠ 0	ED B8

Figure 160

8. Instructions de recherche de caractères

En plus des instructions de transfert par bloc, le Z 80 possède 4 autres instructions bâties sur le même principe qui réalisent la recherche des caractères.

Le principe est le suivant. Le caractère de «référence» est placé dans l'accumulateur A. Le contenu de l'emplacement mémoire pointé par le contenu de la paire de registres HL est comparé au contenu de l'accumulateur A. (L'opération $A - (HL)$ est réalisée, le résultat n'est pas conservé, A ne change pas). Deux cas se présentent :

— si la comparaison est vraie (c'est-à-dire $A - (HL) = 0$) l'indicateur Z du registre F est positionné en 1.

— si la comparaison est fautive (c'est-à-dire $A - (HL) \neq 0$), l'indicateur Z du registre F est à «0».

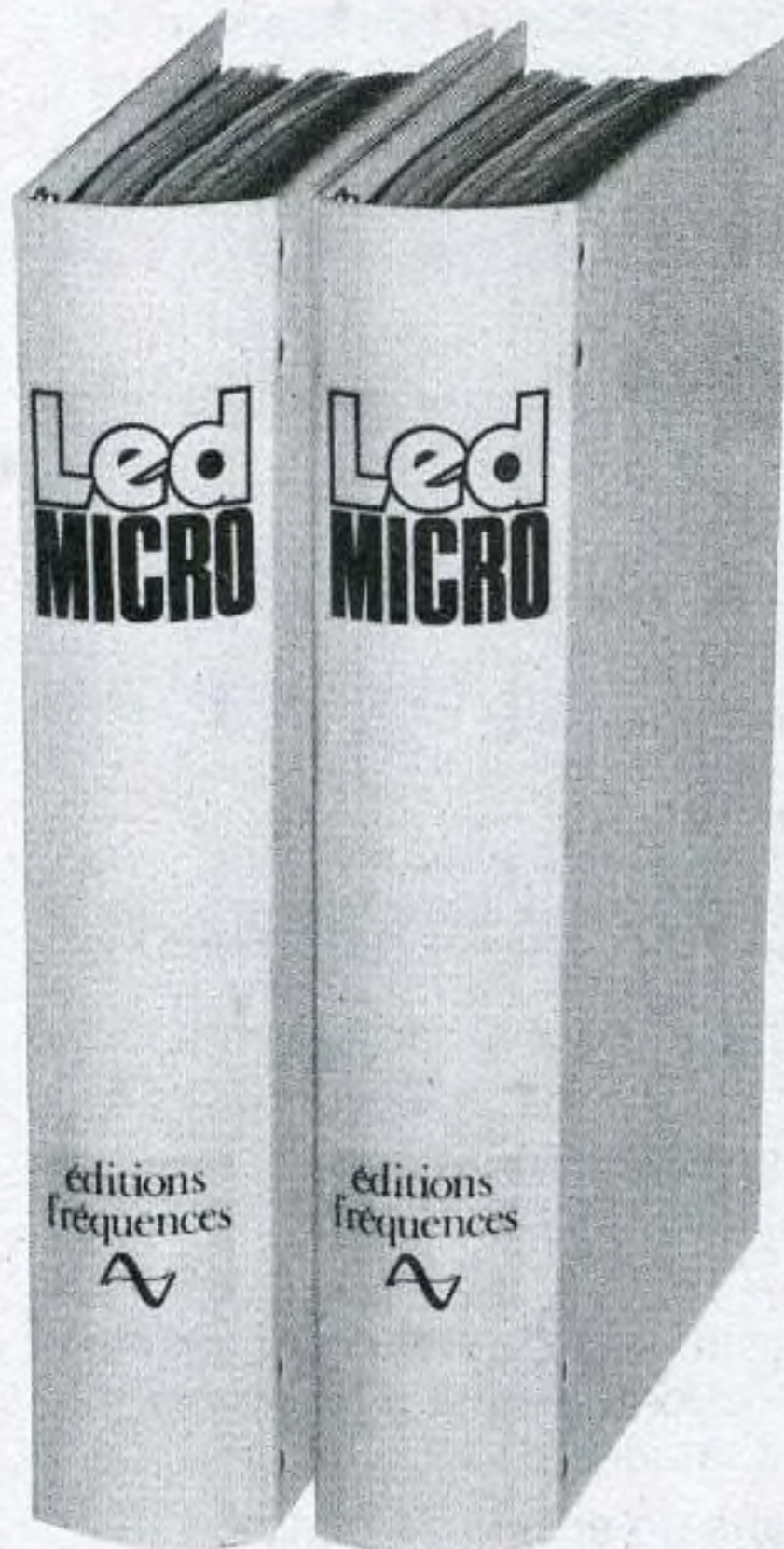
Au préalable, la taille du bloc (le nombre de caractères) est placée dans la paire de registres BC. A l'issue de chaque comparaison, le registre BC est décrémenté. Le test suivant est alors effectué. Si le contenu de la paire de registres BC est différent de 0, l'indicateur P/V est positionné à 1. Si le contenu de la paire de registres BC est nul, l'indicateur P/V du registre F est positionné à «0».

Les quatre instructions de recherche

de caractères sont résumées dans le tableau suivant (figure 160).

CPI	Incrémentation de HL Décrément de BC	ED A1
CPIR	Incrémentation de HL Décrément de BC Répétée si BC # 0 ou A # (HL)	ED B1
CPD	Décrément de HL Décrément de BC	ED A9
CPDR	Décrément de HL Décrément de BC Répétée si BC # 0 ou A # (HL)	ED B9

Philippe Duquesne



habilitez votre collection

Led MICRO

avec
une
superbe
reliure
toilée
jaune



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, accompa-
gné de votre règlement à :
EDITIONS FREQUENCES
1, boulevard Ney, 75018 Paris

Nom

Adresse

Ci-joint le montant de

CCP Chèque bancaire Mandat

AIDE A LA CONCEPTION DES GRAPHIQUES

GRAPHIQUES ET GRAPHISME

Les progiciels les plus répandus en micro-informatique professionnelle sont :

- les traitements de texte (Easy Writer, Wordstar, Applewriter... C.-H. Delaleu vous en a parlé dans LED MICRO n° 16 page 52 et suivantes) ;
- les tableurs (Visicalc, Multiplan... P. Pallu vous en a parlé dans LED MICRO n° 8, pages 34 et suivantes) ;
- les gestionnaires de fichiers (DBASE II et consorts) ;
- et divers progiciels permettant d'illustrer des résultats de calculs, d'expérience, de précision... d'une façon moins austère que par des tableaux de chiffres.

Ce sont toutes ces représentations appelées **graphiques** dont nous allons parler dans la série d'articles qui commence avec le présent numéro.

Conformément à l'esprit LED MICRO, nous ne vous fournirons pas de listings à recopier tels quels, mais nous construirons ensemble les divers modules des divers graphiques formant notre utilitaire. Si vous le voulez bien, nous l'appellerons CGRAPH (C = conceptin ; GRAPH = graphique).

QUELS GRAPHIQUES ?

Précisons les types de graphiques que nous vous aiderons à dessiner et le domaine privilégié d'applications de chacun d'eux.

GRAPHIQUE SIMPLE (figure 1)	permet l'étude de la variation d'une donnée en fonction d'un paramètre.
GRAPHIQUE CUMULE (figure 2)	comparaison des courbes de deux variables, l'une étant l'expression cumulée de l'autre.
GRAPHIQUE MULTICOURBES (fig. 3)	les différentes courbes ou droites sont associées à des variables qui évoluent en fonction des deux paramètres d'abscisse et d'ordonnée.
HISTOGRAMME (figure 4)	utilisé principalement en économie et pour la représentation de données statistiques.
BICOLONNE (figure 5)	comparaison de deux ensembles de valeurs et de leur somme (très utilisé dans la comptabilité et l'analyse financière).
NUAGE DE POINTS (figure 6)	(très particulier) il permet de constater la dispersion de points représentant des variables en fonction de deux paramètres (statistiques, analyse de données, études de marché...)
GRAPHIQUE CIRCULAIRE (fig. 7)	proportion de divers éléments dans un ensemble donné.
GRAPHIQUE POLAIRE (figure 8)	utilisé pour représenter des phénomènes «pseudo-cycliques».

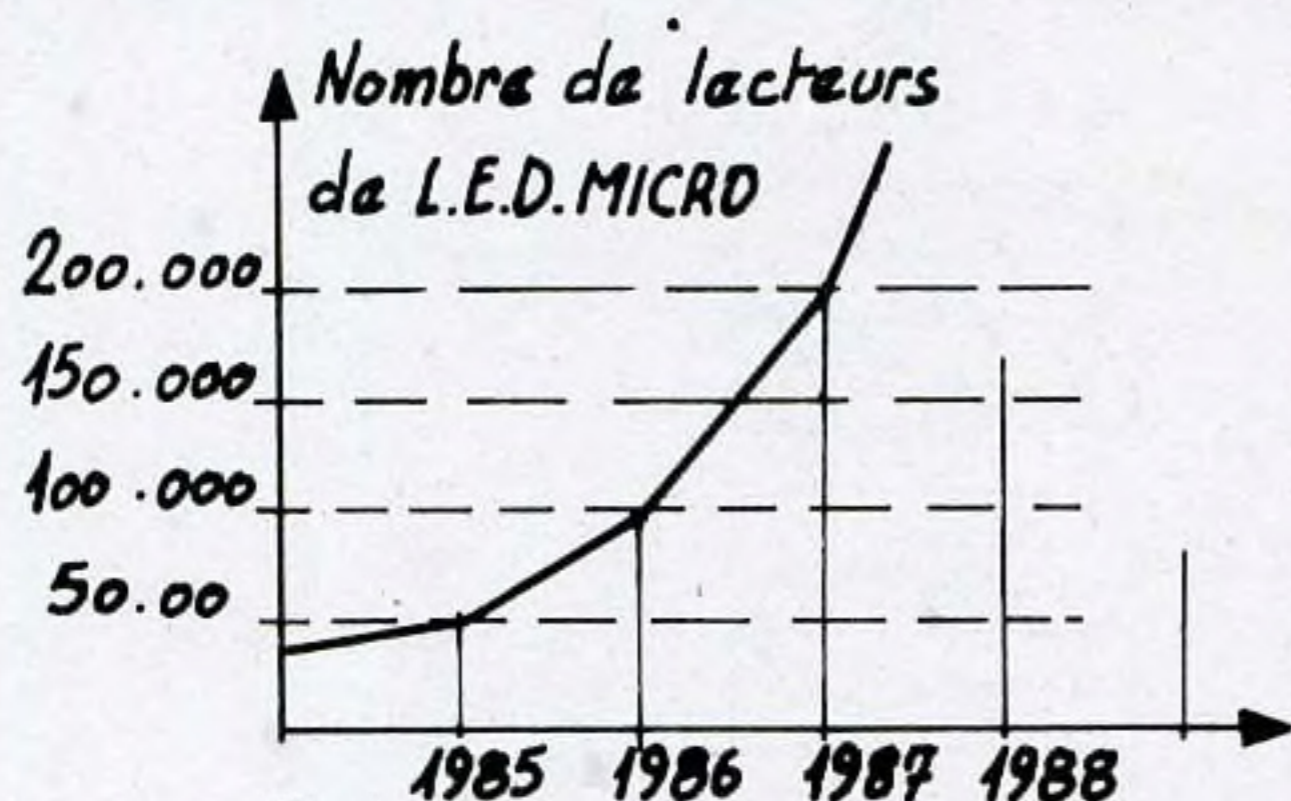


Fig. 1

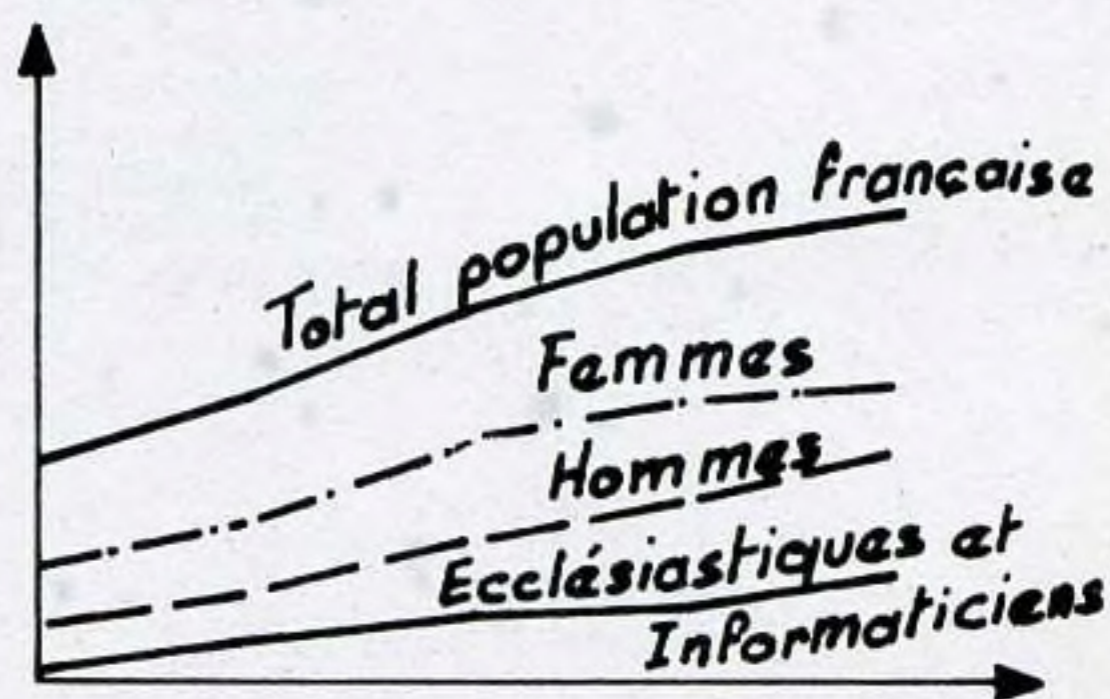


Fig. 2

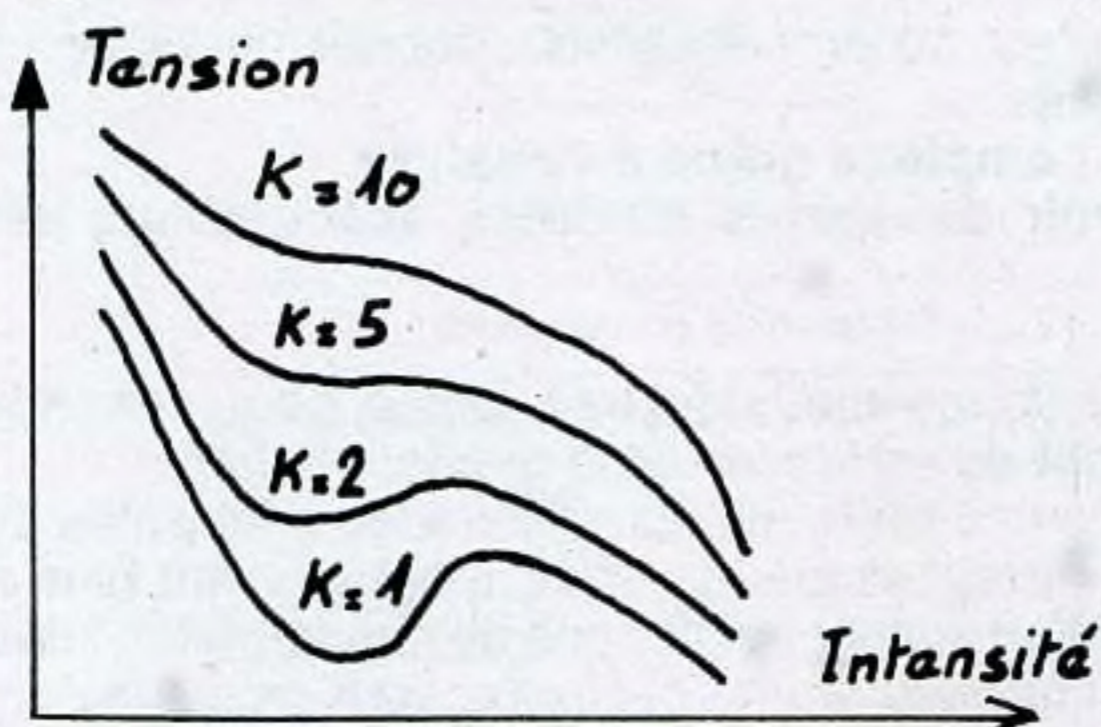


Fig. 3



Fig. 4

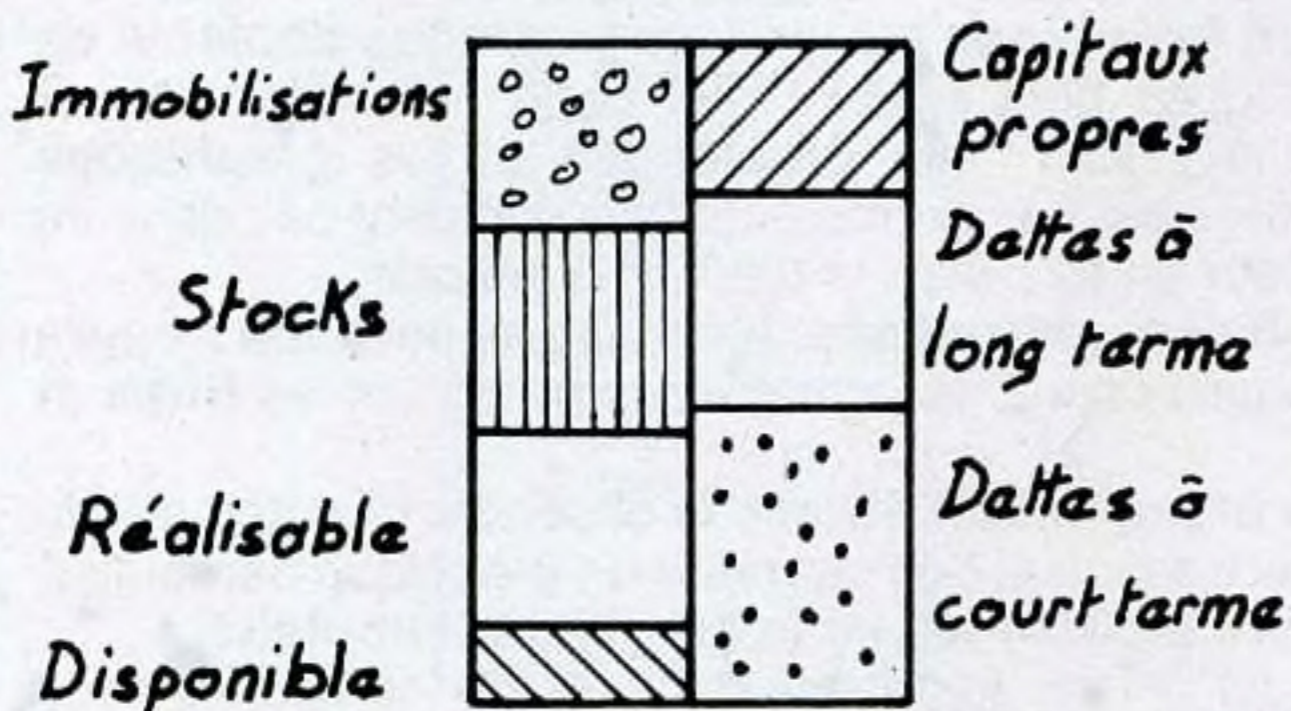


Fig. 5

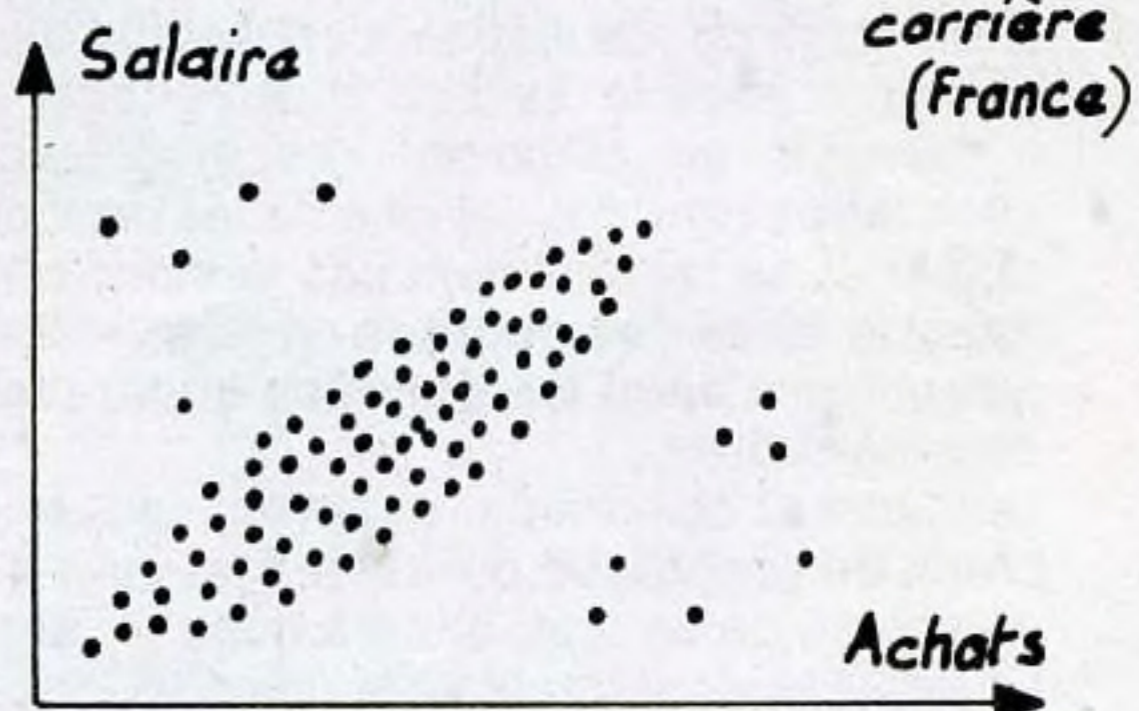


Fig. 6

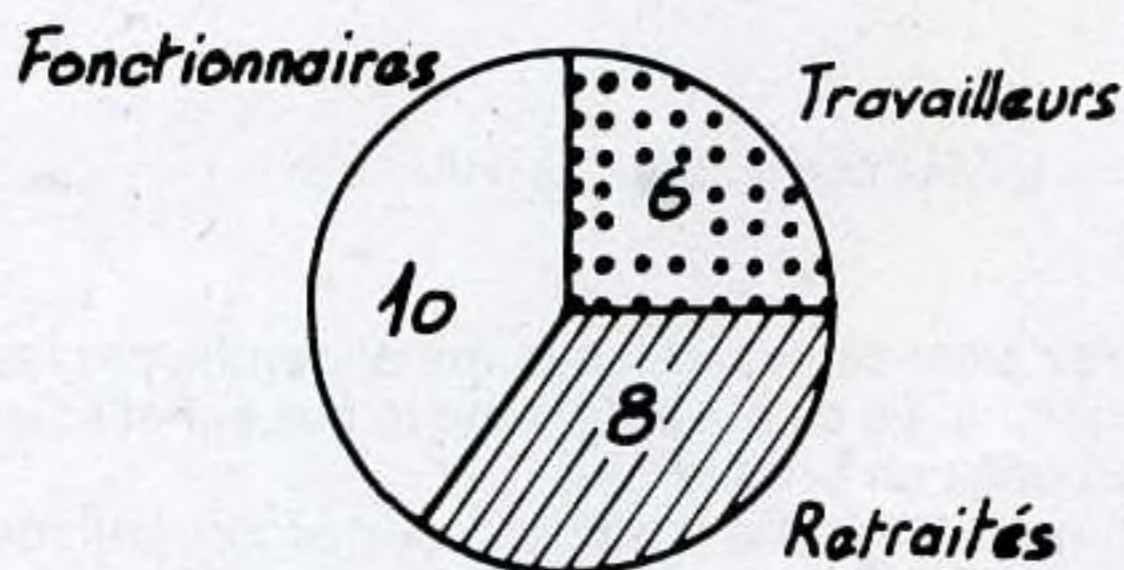


Fig. 7

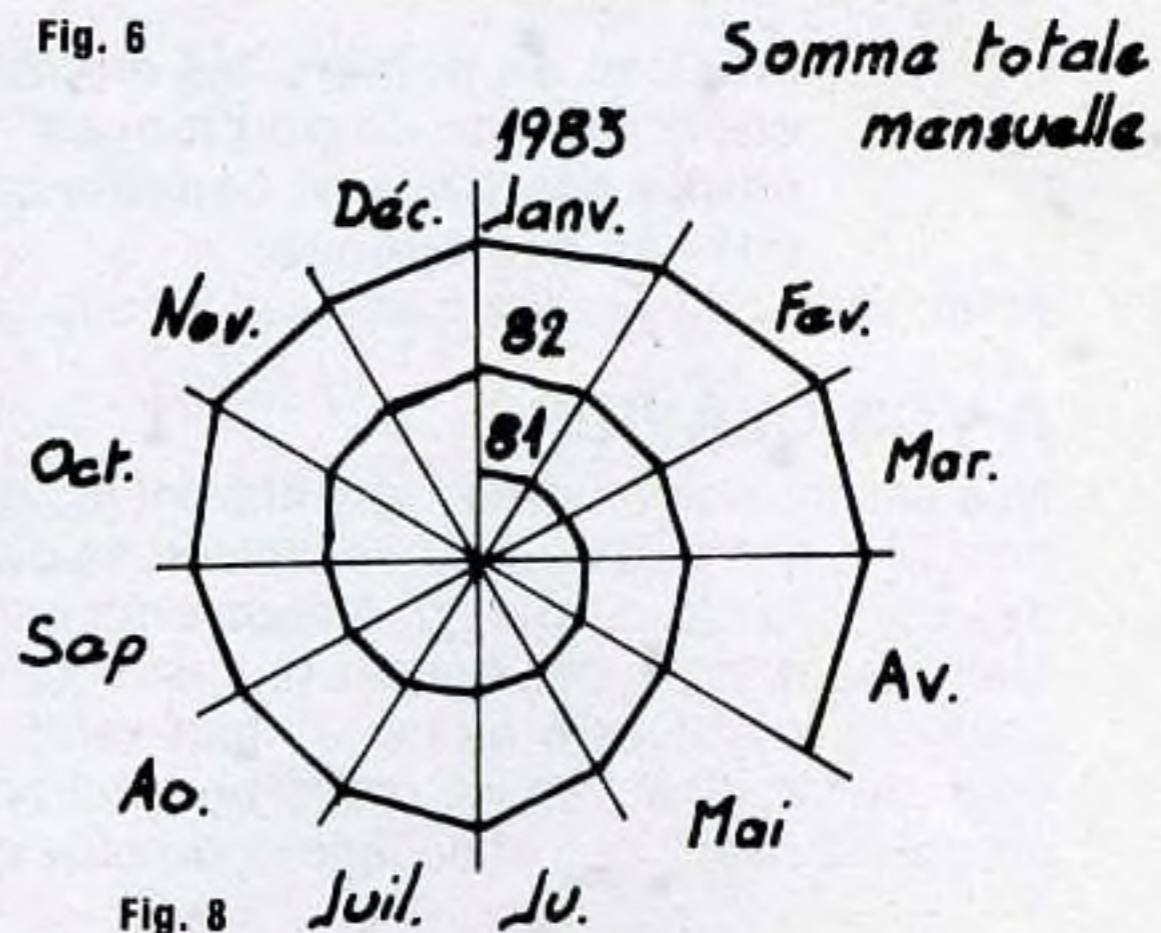


Fig. 8

MICRO-ANALYSE DE CGRAPH

Avant d'aller plus loin, précisons que notre but n'est nullement de vous fournir un listing qu'il vous resterait à recopier pour faire vos études graphiques.

Nous allons apprendre à **résoudre un problème complexe grâce à l'analyse.**

Dans un second temps, nous pourrons **concevoir des petits modules exécutables**, définis par l'analyse qui formera notre utilitaire.

L'organigramme (appelé aussi ordinogramme) est une méthode très usitée pour concrétiser une analyse. C'est le plan que tout architecte fera avant de se lancer dans la construction.

Pour construire cet organigramme, nous utiliserons une méthode qui définira les différentes étapes du traitement en partant du résultat attendu de notre programme. En effet, il est souvent plus facile de savoir ce que l'on désire obtenir de notre machine à un instant t plutôt que de connaître les informations dont nous disposons à l'instant $t - 1$. Donc, dans l'organigramme ci-contre, vous trouverez les étapes ainsi définies du haut vers le bas.

Nous désirons l'**AFFICHAGE SUR L'ECRAN** ou l'**IMPRESSION** d'un graphique de notre choix.

L'affichage du résultat sera consécutif au **traitement spécifique du type de graphique choisi**. A ce niveau, il semble évident que certains des graphiques dont nous avons parlé ont de nombreux traitements en commun. Les graphiques particuliers sont : **les bicolonnes** et les **graphiques circulaires** (une des raisons de les dissocier des autres est par exemple qu'ils n'évoluent pas dans un repère et ne représentent pas la variation d'une valeur en fonction de deux paramètres).

Avant le **tracé des courbes** on fera le traitement **échelle automatisée**. L'échelle ne peut être trouvée que si l'on a **saisi les données auparavant** il faut aussi connaître la place que prendront les **titres et commentaires**.

Les titres et commentaires seront sous une certaine **présentation** et celle-ci dépendra directement du **choix du graphique** qui devra être suivi d'un **test** pour savoir si l'on va traiter un graphique particulier.

Au niveau de ce test, on fera donc un **aguillage vers bicolonnes** ou le **graphique circulaire**.

Le début du traitement sera des plus classiques ; pour rentrer toutes les données ou les options nous utiliserons des menus...

MENU de **création de fichiers** liés au traitement

choix du type de graphiques

entrée des titres et commentaires

entrées des options

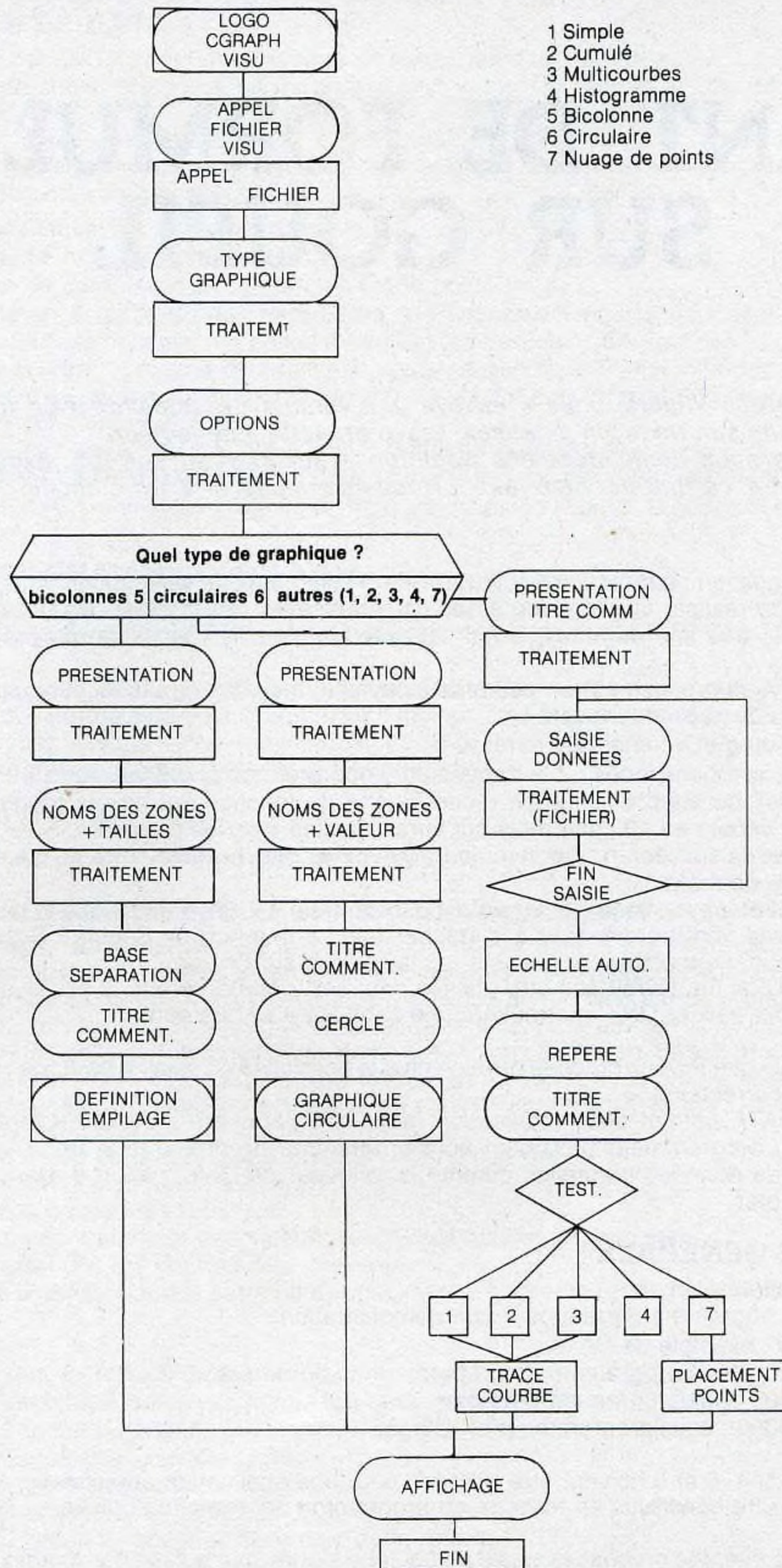
Enfin, vous pourrez vous amuser à nous proposer le PLUS BEAU LOGO pour CGRAPH.

A VOS CLAVIERS

Nos prochains numéros présenteront les différents modules ainsi définis. Nous commencerons par les principes généraux que nous utiliserons dans chaque graphique. Le calcul automatique des échelles et des portions de cercle (graphique circulaire) seront développés en premier lieu.

Maintenant, nous désirons aussi que vous participiez activement à cette étude en nous proposant vos solutions adaptées à tel ou tel matériel. Si, à la vue de l'organigramme, des simplifications évidentes vous venaient à l'esprit, nous ne désirons qu'une chose : que nos échanges nous permettent de progresser vers des applications de plus en plus passionnantes.

Jean Yrytov



GERANT DE FORMULAIRE SUR GOUPIL

Monsieur F.D. (02600 Villiers) nous a envoyé une version très améliorée de mon éditeur de tableau. Voici donc son texte «in extenso». Bravo et merci à ce lecteur.

D'autres lecteurs nous demandent des traductions sur ZX81 et sur T05. Excusez-nous : les formulaires d'écran, ça finit par bien faire... Nous allons passer à autre chose.

B. Lilamand

Le programme de gestion de formulaire écrit par Bruno Lilamand est, sans aucun doute, un modèle de clarté qui permet de réaliser une fonction assez complexe avec des instructions simples.

La structuration est très intéressante, ce qui est précieux pour la translation du programme et ses performances.

La version que je vous propose est un peu plus complexe, mais les atouts qu'elle apporte répondent bien aux faiblesses de la première version :

- Pas d'insertion ou d'effacement de caractères.
- Risque de voir une touche indésirable frappée par l'opérateur de saisie (ex. : une lettre dans le n° de Sécurité Sociale), ou encore la frappe d'une touche de fonction (au pire la touche d'effacement d'écran) qui ne serait pas affichée mais qui aurait modifié tous les pointeurs.

Un principe pour les saisies écran : il doit toujours y avoir identité parfaite entre ce qui est affiché et ce qui est en mémoire centrale.

- Le débordement d'une rubrique à l'autre n'est pas contrôlé. Ex. : le prénom que je tape est trop long, les lettres en trop vont commencer à s'afficher dans le numéro de Sécurité Sociale. Surtout si, comme beaucoup, mes yeux sont plus sur le clavier que sur l'écran.
- La touche RETURN ne devrait pas être utilisée pour sortir du bordereau. Il est plus usuel d'utiliser cette touche pour valider UNE information (une habitude à perdre donc).

Le masque de saisie que je vous propose offre en plus la possibilité de saisir à partir de zones vierges ou déjà remplies (en correction).

L'utilisation des DATA permet une modification facile des paramètres de saisie et offre un confort supplémentaire au programmeur puisqu'un sous-programme de préparation trace le bordereau en calculant à partir de données naturelles comme la longueur de l'information à saisir, les bornes de caractères à accepter.

DESCRIPTION GENERALE

Lignes 10 à 20 : initialisation

Lignes 90 à 310 : programme d'essai pour une démonstration

Lignes 999 à 1070 : exemple de DATA

Lignes 8000 à 8120 : sous-programme de préparation du bordereau

Lignes 8200 à 9350 : sous-programme de saisie

Lignes 10000 à 10050 : sous-programme d'affichage.

Seules les parties 1, 4, 5 et 6 doivent être utilisées pour une application personnelle.
Les DATA doivent être construits en fonction du programme qui exploite l'utilitaire.

La démonstration présentée propose la saisie d'une fiche vierge puis la réédition indéfinie de cette fiche en correction.

LES LIGNES DE DATA

En 999 un premier DATA contient le nombre de rubriques d'un bordereau.

Ce DATA lu une seule fois en début de programme (ligne 15) permet le bon dimensionnement des tableaux (ligne 20).

Notre exemple est de huit rubriques, il est modifiable à loisir. Chaque DATA suivant contient les spécifications d'une seule rubrique. (Classer ces datas par numéro de ligne croissant).

Cinq renseignements sont nécessaires :

LI : la ligne sur laquelle s'affiche la rubrique.

CO : la colonne (sur Goupil écran 24 x 80).

CT : le nombre de caractères de la rubrique (code postal = 5).

CA : l'intervalle en code ASCII des caractères que l'utilisateur pourra frapper à l'intérieur de cette rubrique. Par exemple le code postal s'écrit en chiffres, donc CA vaut «48-57», ce qui signifie que si je tape la lettre A, elle ne sera pas prise puisque son code 65 est supérieur à 57. Si je n'ai pas bloqué les majuscules, aucun risque de voir un point d'exclamation à la place d'un 8 (le code de «!» est 35 donc inférieur à 48).

TT : indique le titre de la rubrique qui doit être édité sur le bordereau.

La rubrique centimes du montant a pour titre une virgule (dans mon exemple).

Note : A propos de CA, remarquez à la ligne 1050 que seuls un D ou un C pourront être frappés (Crédit/ Débit).

LE SOUS-PROGRAMME D'AFFICHAGE (appelons-le AFFICHE)

Ce sous-programme devrait faciliter la transposition du programme.

Admirez la syntaxe de la ligne 10030, cette dissertation veut dire : LOCATE C, L : PRINT M\$; ... c'est tout !

Un appel en 10000 ou 10010 provoque un affichage en vidéo inverse. Un appel en 10020 provoque un affichage en vidéo normale. Si C (la colonne) vaut moins 1, le SP recalcule cette colonne pour centrer le message à afficher. Ecrans 40 col. changez 80 par 40.

La ligne 10040 remet systématiquement la vidéo normale.

LA VARIABLE BL\$ (blancs)

Contient... des blancs, une réminiscence de COBOL !

LE SOUS-PROGRAMME DE PREPARATION (appelons-le PREPA)

A partir de la ligne 8000 commence le programme qui va utiliser les informations des DATA pour dessiner l'écran et calculer divers paramètres nécessaires à l'utilitaire de saisie.

Il est appelé autant de fois qu'il y a de rubriques, puisque sa fonction est de traiter une rubrique.

Vous voyez donc aux lignes 120 et 260 que l'appel de ce sous-programme est à l'intérieur de la boucle I qui va de 1 à NR (nombre de rubriques défini par le DATA ligne 999). Lors de son premier passage (c'est-à-dire ici la rubrique Nom), grâce au drapeau PA qui est à zéro, le sous-programme PREPA appelle un autre sous-programme INIT (appelons-le ainsi) qui commence en 8100 et qui :

- efface l'écran en 8115
- initialise le numéro de la rubrique à zéro (S)
- initialise la ligne la plus haute à 24 (LH)
- la ligne la plus basse à 61 (LB)

Ces deux valeurs sont bien sûr des valeurs exagérées qui seront modifiées.

— place le drapeau PA à 1 pour ne pas y revenir.

Lors d'un appel ultérieur pour un autre bordereau, ne pas oublier la remise à zéro de ce drapeau (voir la figure 220).

Ensuite pour chaque rubrique (numéro S) :

- Le tableau T reçoit les indices de lignes (LI) et de colonne (CO).
- Le tableau S\$ reçoit le contenu initial de la rubrique (ED\$)
 - des blancs pour une nouvelle fiche
 - l'ancien contenu pour une correction.

Comparez à cet effet le chargement de ED\$ en 115 et en 250.

- Le tableau LD reçoit la longueur maximale de la rubrique (CT).
- Le tableau CA reçoit le code ASCII des caractères acceptables. CA(S,0) est le code inférieur et CA(S,1) le code supérieur. Voyez le test de ces valeurs à la ligne 8260.

LE COIN DES FORTICHES LE COIN DES

Si la ligne d'affichage est la plus haute jusqu'ici, sa valeur est conservée par LH, si c'est la plus basse, sa valeur est conservée par LB.

En 8030, on commande l'affichage du titre en vidéo normale. (Attention, la longueur du titre ne doit pas être supérieure à la colonne d'affichage CO).

En 8040, on commande l'affichage du contenu de la rubrique, des blancs dans le cas d'une création, en vidéo inversée.

LE SOUS-PROGRAMME DE SAISIE (à partir de 8200)

L'initialisation (8210-8220) se fait sur le premier article du bordereau.

La saisie du caractère au clavier se fait à l'aide de l'instruction $T\$ = INCH\(0) qui est équivalente à GET T\$ (avec attente). Le code ASCII de cette touche est rangé dans la variable A. Puis A2 est calculé par la valeur logique pondérée (ligne 8240). C'est-à-dire, si A vaut 11 alors l'expression $A = 11$ est vraie, sa valeur logique est 61, dont $(A = 11) \star 5$ vaut 65.

La valeur absolue (fonction ABS) donne la valeur 5 à A2, et le branchement du ON A2 GOTO se fait à la cinquième adresse, c'est-à-dire 8700.

Si A2 vaut 0 cela signifie que la touche frappée n'est pas une fonction (test en 8245).

Dans ce cas, il s'agit peut-être d'un caractère à saisir, il est testé (8260) afin de savoir s'il est acceptable ou non. Son affichage est alors effectué (8295) après une mise en mémoire grâce à une opération de chaîne de caractères (8285).

L'appel des fonctions entraîne l'exécution de l'un des onze modules, tous les retours (excepté la sortie par EXECUTER ou BREAK) se font sur la saisie d'un nouveau caractère ou touche.

Le pointeur qui permet de savoir sur quel caractère on se trouve pour une insertion, un effacement ou un placement de caractère est calculé à partir de la colonne courante du curseur :

La formule $NC = C - T(AR, 1)$ indique ce pointeur.

AR est la rubrique en cours de saisie.

$T(AR, 1)$ est l'adresse colonne du début de rubrique.

C est la colonne courante du curseur.

Donc NC est le nième caractère de la rubrique (en partant de 0).

Le transformateur du contenu de la rubrique se fait par une fonction de chaîne de caractères.

Elle est nécessaire dans trois cas : insertion, effacement, placement.

Les autres cas entraînent seulement une modification du pointeur.

CONTROLE DES TOUCHES DE FONCTIONS

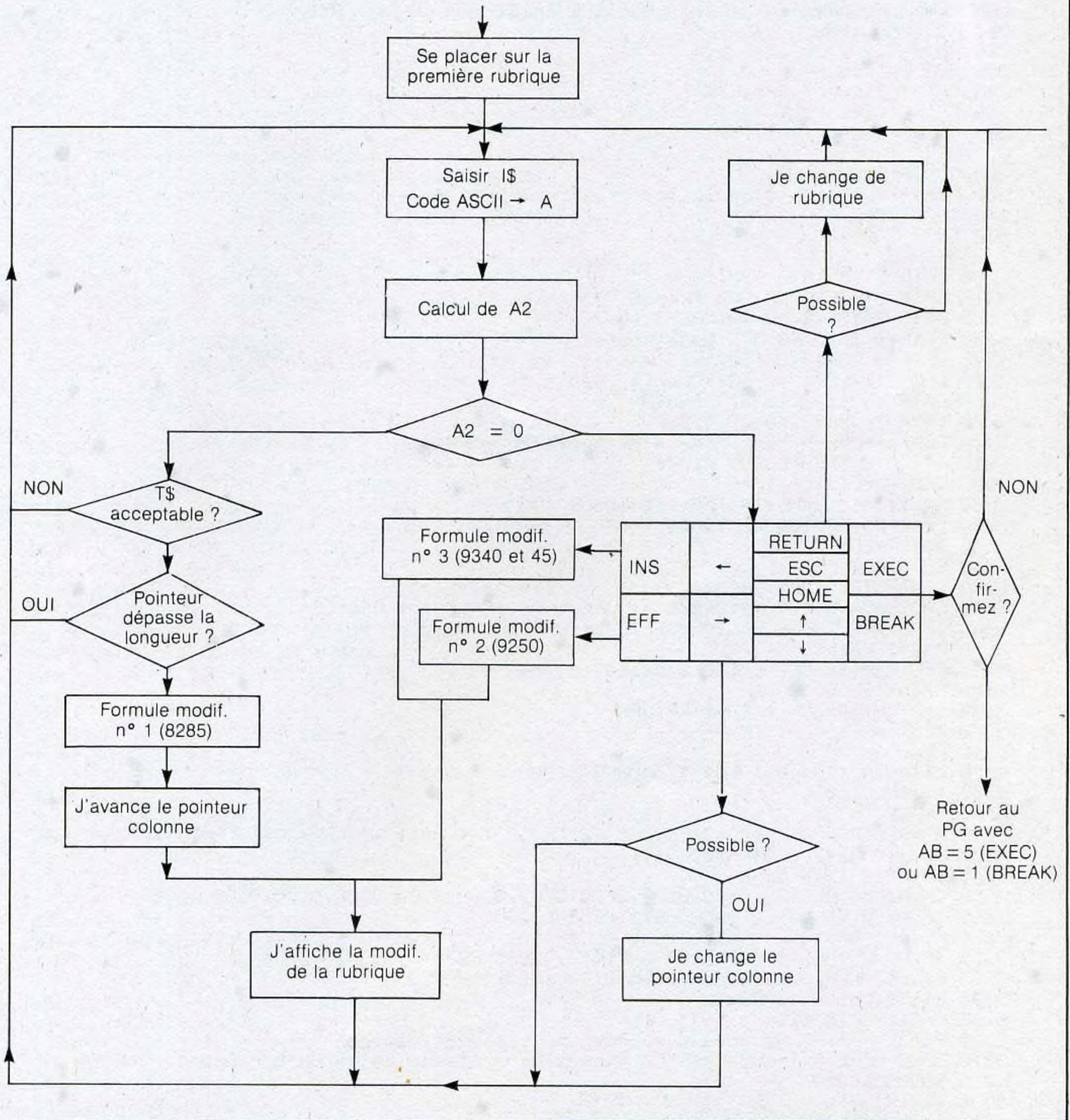
Touche	Code ASCII Goupil	EFFET	Ligne traitement
→	29	Dans une rubrique, déplacement du curseur vers la droite.	9100
←	8	Dans une rubrique, déplacement du curseur vers la gauche.	8400
↑	11	Retour au début de la première rubrique de la ligne précédente.	8700
↓	10	Passage au début de la première rubrique de la ligne suivante.	8600
RETURN	13	Passage au début de la rubrique suivante.	8900
ESC	27	Retour au début de la rubrique précédente.	9000
↶ (HOME)	12	Retour au début de la première rubrique du bordereau.	8800
INS	246	Insertion d'un blanc après le curseur, le reste de la rubrique est d'un cran vers la droite.	9300
EFF	247	Suppression du caractère pointé par le curseur, le reste de la rubrique est déplacé d'un cran vers la gauche et le dernier caractère devient un blanc.	9200
EXECUTER	245	Les infos contenues dans le bordereau sont validées et le programme appelant reçoit le drapeau AB à 5.	8500
BREAK	0	Toutes les infos sont annulées, le programme le sait grâce au drapeau AB qui est à 1.	8300

Les huit premières fonctions ne sont effectuées qu'après un test de probité.

Si votre machine ne possède pas certaines touches, vous pouvez adapter avec des doubles touches (CNTR avec une autre).

Tenez compte des codes ASCII de votre machine et modifiez la ligne 8240 qui calcule le numéro de la fonction suivant le code ASCII de la touche frappée.

ORGANIGRAMME DU GESTIONNAIRE DE SAISIE



```

10 BL$=CHR$(32):FOR I%=1 TO 6:BL$=BL$+BL$:NEXT I%
15 READ NR
20 DIM T(NR,1),L(NR),C(NR,1),S$(NR)
90 REM ----- Premier appel FICHE VIERGE -----
100 FOR I=1 TO NR
110 READ LI,CO,CT,CA$,TT$
115 ED$=LEFT$(BL$,CT)
120 GOSUB 8000
130 NEXT I
140 GOSUB 8200
200 REM ----- Deuxième appel LA FICHE SAISIE EN CORRECTION -----
210 RESTORE 1000
220 PA=0
230 FOR I=1 TO NR
240 READ LI,CO,CT,CA$,TT$
250 ED$=S$(I-1)
260 GOSUB 8000
270 NEXT I
280 GOSUB 8200
300 REM --- et ainsi de suite, c'est une DEMO ! ---
310 GOTO 200
500 END
999 DATA 8
1000 DATA 5,15,18,32-90,"Nom :"
1010 DATA 5,42,18,32-90,"Prénom :"
1020 DATA 7,15,42,32-90,"Adresse :"
1030 DATA 9,15,5,48-57,"Code Postal :"
1040 DATA 9,30,18,32-90,"Ville :"
1050 DATA 12,25,1,67-68,"Crédit /Débit "
1060 DATA 14,15,5,48-57,"MONTANT ="
1070 DATA 14,21,2,48-57,""
8000 REM ----- s.P. Trace et Prépare -----
8010 IF PA=0 THEN GOSUB 8100
8020 T(S,0)=LI:T(S,1)=CO
8030 L=LI:C=CO-LEN(TT$):M$=TT$:GOSUB 10020
8040 C=CO:M$=ED$:GOSUB 10000
8045 S$(S)=ED$
8050 LO(S)=CT
8060 CS=INSTR(1,CA$,"-")
8070 CA$(S,0)=VAL(LEFT$(CA$,CS-1)):CA$(S,1)=VAL(MID$(CA$,CS+1))
8080 IF LI>LB THEN LB=LI ELSE IF LI<LH THEN LH=LI
8090 S=S+1:RETURN
8100 REM ----- Premier Passage -----
8115 PRINT CHR$(12)
8120 S=0:LH=24:LB=-1:PA=1:RETURN
8200 REM ----- SAISIE -----
8210 AR=0
8220 L=T(0,0):C=T(0,1):M$="":GOSUB 10000
8230 T$=INCH$(0)
8237 A=ASC(T$)
8240 A2=ABS((A=0)+(A=8)*2+(A=245)*3+(A=10)*4+(A=11)*5+(A=12)*6+(A=13)*7+(A=27)*8
+(A=29)*9+(A=247)*10+(A=246)*11)
8245 IF A2=0 THEN 8260
8250 ON A2 GOTO 8300,8400,8500,8600,8700,8800,8900,9000,9100,9200,9300
8260 IF A<CA(AR,0) OR A>CA(AR,1) THEN 8230
8270 NC=C-T(AR,1)
8275 IF NC=LO(AR) THEN PRINT CHR$(7):GOTO 8230
8285 S$(AR)=LEFT$(S$(AR),NC)+T$+MID$(S$(AR),NC+2)
8295 M$=T$:GOSUB 10000
8296 C=C+1:GOTO 8230
8300 REM ----- Abandon du bordereau sans validation -----
8310 CC=C:LL=L:C=1:L=23:M$="Confirmez-vous l'annulation du bordereau (O/N)?"
8315 GOSUB 10020
8320 T$=INCH$(0)

```

FORTICHES LE COIN DES FORTICHES

```

8330 IF INSTR(1,"Oo",T#)>0 THEN 8350
8340 IF INSTR(1,"Nn",T#)=0 THEN 8320
8345 M#=CHR$(27)+"L":GOSUB 10020:L=LL:C=CC:M#="" :GOSUB 10000:GOTO 8230
8350 AB=1:RETURN
8400 REM ----- Caractère précédent -----
8405 IF C<=T(AR,1) THEN PRINT CHR$(7):GOTO 8230
8410 C=C-1
8415 M#="" :GOSUB 10000
8420 GOTO 8230
8500 REM ----- Bordereau valide -----
8510 CC=C:LL=L:C=1:L=23:M#="Confirmez-vous la validation du bordereau (O/N)?"
8515 GOSUB 10020
8520 T#=INCH$(0)
8530 IF INSTR(1,"Oo",T#)>0 THEN 8350
8540 IF INSTR(1,"Nn",T#)=0 THEN 8320
8545 M#=CHR$(27)+"L":GOSUB 10020:L=LL:C=CC:M#="" :GOSUB 10000:GOTO 8230
8550 AB=5:RETURN
8600 REM ----- Début de ligne suivante -----
8610 IF L=LB THEN PRINT CHR$(7):GOTO 8230
8620 AR=AR+1:IF T(AR,0)=L THEN 8620
8630 L=T(AR,0):C=T(AR,1):M#="" :GOSUB 10000
8640 GOTO 8230
8700 REM ----- Début de ligne précédente -----
8710 IF L=LH THEN PRINT CHR$(7):GOTO 8230
8720 AR=AR-1:IF T(AR,0)=L THEN 8720
8724 A=T(AR,0)
8725 IF AR=0 THEN 8730
8726 AR=AR-1:IF T(AR,0)=A THEN 8725 ELSE AR=AR+1
8730 L=T(AR,0):C=T(AR,1):M#="" :GOSUB 10000
8740 GOTO 8230
8800 REM ----- Premier article du bordereau ----
8810 AR=0:L=T(AR,0):C=T(AR,1):M#="" :GOSUB 10000:GOTO 8230
8900 REM ----- Article suivant -----
8910 IF AR=S-1 THEN PRINT CHR$(7):GOTO 8230
8920 AR=AR+1:GOTO 8730
9000 REM ----- Retour à l'article précédent -----
9010 IF AR=0 THEN PRINT CHR$(7):GOTO 8230
9020 AR=AR-1:GOTO 8730
9100 REM ----- Passage au caractère suivant mme article -----
9105 NC=C-T(AR,1)
9110 IF NC=LO(AR) THEN PRINT CHR$(7):GOTO 8230
9120 C=C+1:M#="" :GOSUB 10000
9130 IF MID$(S$(AR),NC+1,1)="" THEN S$(AR)=LEFT$(S$(AR),NC)+CHR$(32)+MID$(S$(AR),NC+2)
9140 GOTO 8230
9200 REM ----- efface caractère Pointé -----
9220 NC=C-T(AR,1)
9230 S$(AR)=LEFT$(S$(AR),NC)+MID$(S$(AR),NC+2)+CHR$(32)
9240 CC=C
9250 M#=S$(AR)+LEFT$(BL$,LO(AR)-LEN(S$(AR))):C=T(AR,1)
9260 GOSUB 10000
9280 C=CC:M#="" :GOSUB 10000:GOTO 8230
9300 REM ----- Insertion -----
9320 NC=C-T(AR,1)
9330 IF NC=LO(AR) OR RIGHT$(S$(AR),1)<>CHR$(32) THEN PRINT CHR$(7):GOTO 8230
9340 S$(AR)=LEFT$(S$(AR),NC)+CHR$(32)+MID$(S$(AR),NC+1)
9345 S$(AR)=LEFT$(S$(AR),LO(AR))
9350 GOTO 9240
10000 REM ----- SF affichage -----
10010 PRINT CHR$(27);"I"
10020 IF C=-1 THEN C=INT((80-LEN(M#))/2)
10030 PRINT CHR$(27);"=";CHR$(L);CHR$(C);M#;
10040 PRINT CHR$(27);"J"
10050 RETURN

```

LIBRE PROPOS

En ce début de 1985, il est intéressant de faire le point sur la situation actuelle dans ce domaine si à la mode qu'est l'informatique. Comme tout phénomène suivi de près par le public ainsi que par les médias, les superlatifs et les exagérations sont de bon ton. En effet, à croire tout le monde, l'informatique serait actuellement une science où les évolutions sont très rapides, au point de porter le cycle de vie d'une machine à six mois, voire un an au maximum. Cela va tellement vite que les constructeurs ne doivent pas pouvoir être à la page.

En fait, que se passe-t-il exactement ? Beaucoup d'entre-vous seront peut-être déçus, mais à la vérité, les concepts avancent très doucement. Du côté des microprocesseurs, au risque de choquer certains, il est possible d'affirmer que pour les micro-ordinateurs domestiques, la vitesse des évolutions correspond à celle de la tortue. Le premier microprocesseur à être fabriqué de façon industrielle est sans doute le Z 80. Ce composant reste aujourd'hui la référence pour le petit matériel, il a même été retenu pour le nouveau standard MSX. Pour les mémoires, des progrès importants ont été réalisés en laboratoire, mais les retombées grand public ne se profilent pas encore à l'horizon. Il faudra attendre encore longtemps avant de disposer à des prix raisonnables, des mémoires de masse optique à grande intégration.

En direction des logiciels et progiciels, on observe la même constatation que pour le hardware. Pas de révolution, pas de logiciel franchement nouveau. La seule évolution sensible ne concerne pas de nouveaux algorithmes révolutionnaires, ni les progiciels miracles, mais plus simplement l'intégration sur le même support de plusieurs programmes pouvant utiliser les mêmes données : les logiciels intégrés (Framework, Symphony). Si la tâche de l'utilisateur est simplifiée, il ne faut pas oublier qu'il s'agit d'un montage qui n'est pas forcément très réussi car la capacité mémoire nécessaire pour faire tourner ces logiciels est très importante pour de petites machines (environ 256 Ko). Il ne faut pas se tromper, la banalisation de l'intelligence artificielle, des circuits à l'arsenure de gallium, ne sont pas pour tout de suite. Les deux points importants que l'on peut retenir concernent un glissement des logiciels professionnels vers des applications plus larges, ainsi qu'une baisse des prix relative à l'intégration des circuits.

Non ! Le cycle de vie d'une machine n'est pas passé à quelques mois, mais de dix ans, il approche les cinq ans.

C.-H. Delaleu

SOMMAIRE

EMPORTER PARTOUT AVEC SOI SON MICRO, CE N'EST PLUS UN RÊVE, MAIS PARFOIS IL PEUT CÔTER TRÈS CHER A RÉALISER. HEUREUSEMENT, IL Y A DES PORTABLES POUR TOUTES LES BOURSES **P. 50**. CONTRE-MESURES : UN MICRO FRANÇAIS QUI NE MANQUE PAS D'INTÉRÊT : LE GUÉPARD **P. 54**. PROGICIELS A L'ESSAI : UNE NOUVELLE RUBRIQUE QUI FAIT LE POINT, CE MOIS-CI, SUR CONTEXT MBA, UN LOGICIEL INTÉGRÉ **P. 58**. SHOPPING : UNE SÉLECTION DE PRODUITS MATÉRIELS ET LOGICIELS **P. 62**. COMME CHAQUE MOIS, PHILIPPE FAUGERAS A LU POUR VOUS **P. 64**.

l'avenir aux portables



Portable, transportable, portatif. Autant d'adjectifs qui qualifient des machines différentes les unes des autres tant par leurs caractéristiques que leur utilisation. Ils sont bien séduisants ces micros qu'on peut amener avec soi, en particulier les portatifs à écran à cristaux liquides. Seul inconvénient : le prix.



Le TRS-80, modèle 100 de Tandy

u fil des années, les nouvelles technologies ont permis de développer des machines de plus en plus puissantes, de plus en plus petites, et tout récemment «mobiles». C'est Charles Osborne qui, le premier, a inventé le concept du micro portable en 1982, avec l'Osborne 1 qui se présentait sous la forme d'une valise d'une vingtaine de kilos. Cette valise contenait l'unité centrale, les unités de disquettes, un écran cathodique de 12 cm de diagonale et le clavier qui servait de couvercle et protégeait l'écran pendant le transport.

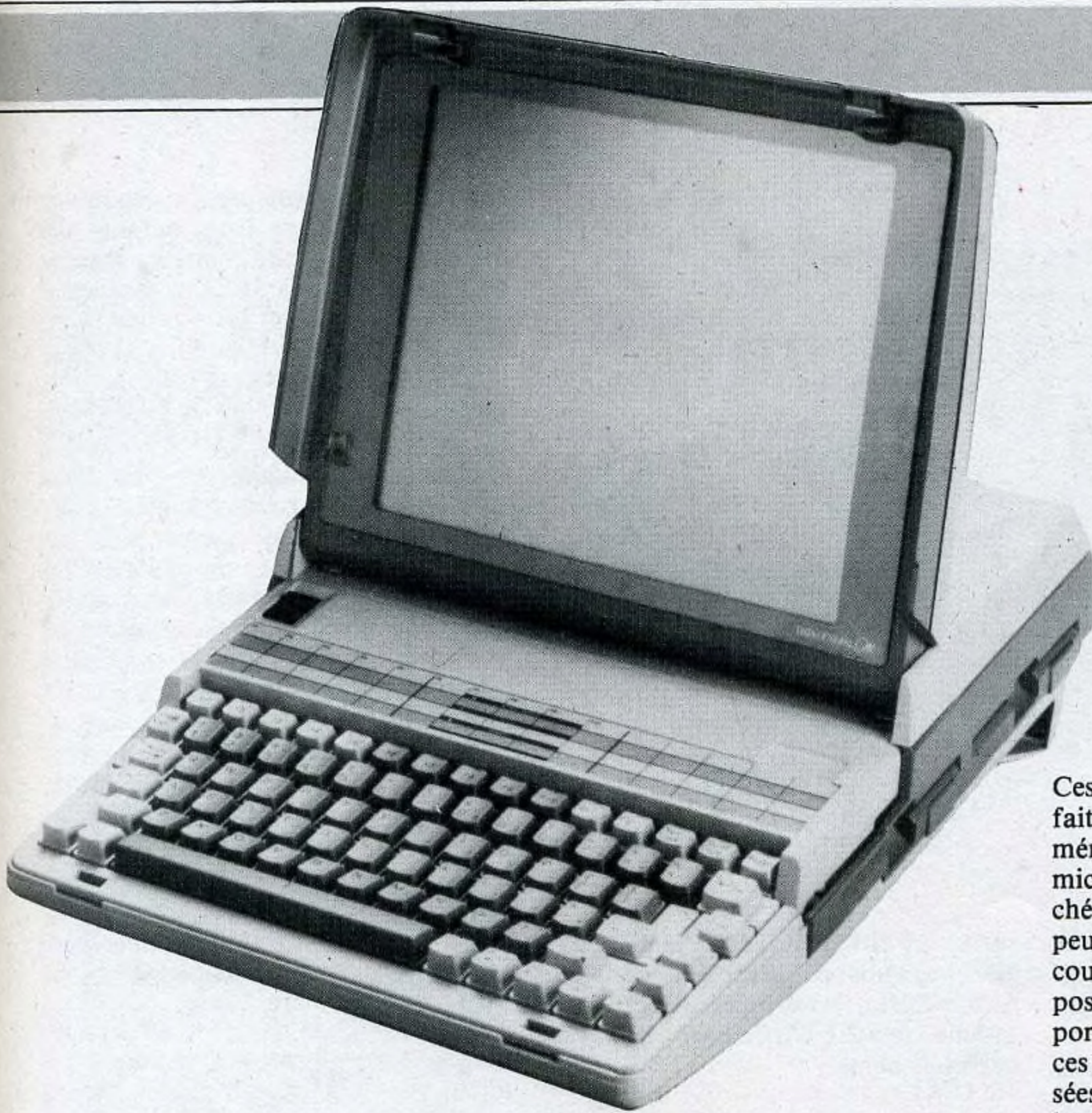
Cette machine avait tout pour séduire puisque, du point de vue de ses caractéristiques, elle n'avait rien à envier aux machines de bureau alors sur le marché : un vrai clavier professionnel, 64 Ko de mémoire centrale et pour système d'exploitation CP/M. Malgré cela, l'Osborne 1 ne fut pas un succès commercial. Les utilisateurs de l'époque n'étaient sans doute pas convain-

cus de l'intérêt de la portabilité. Il faut bien convenir qu'en raison de son poids, il n'était pas facile à transporter. De plus, l'écran était petit ce qui était une gêne pour un travail de longue durée.

Dans le sillage de l'Osborne, sont venus d'autres micros utilisant le même principe : le Kaypro basé sur un microprocesseur Z80, puis les lancements de nouvelles machines se sont accélérés. Aujourd'hui, on distingue trois types de machines : les transportables du style de l'Osborne, les portables et les portatifs.

LES TRANSPORTABLES

Pour ce type de micro, comme pour les deux autres — portable et portatif — il n'existe pas de normes très précises. Disons qu'un micro transportable ou micro-valise pèse entre 10 et 15 kg. Il comporte, dans une valise d'un encom-



Ces machines sont désormais tout à fait comparables au plan de la capacité mémoire, du stockage des données aux micro-ordinateurs de bureau du marché. Au niveau du graphisme, certains peuvent recevoir en option un écran couleur, ce qui leur donne les mêmes possibilités que les machines non transportables. Jusqu'à présent néanmoins, ces machines sont encore peu généralisées ; sans doute toujours en raison de leur poids.

brement moyen, un clavier de type professionnel, un écran cathodique, une mémoire centrale, une ou deux unités de disquettes, et peut même comme le Compaq, intégrer un disque dur. Son prix TTC évolue entre 23 et 30 000 F. Dans cette catégorie, on trouve donc le Compaq, le Zenith 160, version portable du Zenith 160, un micro compatible IBM-PC, l'Olivetti M21, version portable du M24, l'Apri-

cot, le Corona, l'Eagle. De plus en plus, les constructeurs lancent, parallèlement à leur machine de bureau, un modèle portable. IBM a dans sa gamme PC, un portable qui a été annoncé en France, mais qui n'est pas encore commercialisé, le micro pèse 13,6 kg, utilise le même microprocesseur 8088 que l'XT, un écran monochrome ambre de 23 cm, un lecteur demi-hauteur pour disquette de 360 Ko, une mémoire utilisateur de 256 Ko. Il sera commercialisé environ 23 000 F HT. Dans la même catégorie, le Compaq qui, aux Etats-Unis, est en tête du hit-parade des transportables. Ses caractéristiques : 128 Ko de mémoire vive extensible à 256 Ko, une unité de disquette 5 1/4 de 360 Ko. Dans sa version «Compaq Plus», il est équipé d'un disque dur de 10 Mo, d'une unité de disquette de 360 Ko et d'un écran haute résolution de 23 cm. Il dispose également d'un branchement pour un co-processeur Intel 8087.

LES PORTABLES

Pour répondre à cet handicap du poids, certains constructeurs ont développé des machines dites portables. Pesant moins de 2 kg, ces machines se transportent aisément dans un attaché-case. Elles disposent d'un affichage à cristaux liquides, ce qui les différencie des transportables. L'un des tout premiers modèles de ce type a été l'Epson HX 20. Dans la foulée sont apparus l'Olivetti M10, le Tandy 100, le Nec qui sont en fait la même machine mais habillée différemment (écran de 8 lignes de 40 caractères, fixe sur le Tandy, orientable sur le M10). Le plus souvent, ces machines intègrent dans la mémoire morte, outre un langage de programmation (généralement le Basic) plusieurs logiciels d'application comme un traitement de texte, une gestion de fichier d'adresses, un agenda.



Le Compaq Plus

MATERIEL



Le Grid

Grâce à leur poids et à leur encombrement réduit, une capacité mémoire non volatile, ils s'adressent à tous ceux qui sont appelés à se déplacer fréquemment : représentants, géomètres, cadres commerciaux.

Parmi les micros de ce groupe, l'IS-11 de Sord qui se distingue par le fait qu'il ne demande à l'utilisateur de connaître ni la programmation, ni un langage spécifique. Il comporte en effet, des logiciels câblés permettant d'effectuer diverses fonctions : tableau, traitement de texte, calculatrice, agenda, fichier, répertoire téléphonique, graphique, communications. Pips, le tableur développé précédemment par Sord, est intégré dans les 64 Ko de mémoire Ram. Il est possible de lui adjoindre un clavier numérique, une Rom 32 Ko supplémentaire et une imprimante. Dans sa version de base, il est commercialisé environ 8 000 F HT, ce qui est le prix courant pour ce type de machines. Les portables, pour un grand nombre d'entre eux, ne peuvent répondre à des applications professionnelles comme les tableurs à cause de leur capacité mémoire de l'ordre de 32 Ko. En outre, se pose le problème du stockage des données. Plusieurs procédés ont été utilisés, comme la cas-

sette audio-standard, la micro-cassette, sans donner pleine satisfaction. Enfin, ces micros n'offrent qu'une capacité d'affichage réduit, de quelques lignes.

LES PORTATIFS

Cette dernière catégorie, la plus récente, reprend les avantages des deux

autres, sans leurs défauts, ou du moins sans la plupart de leurs défauts. Ils offrent une capacité mémoire interne de 128 Ko jusqu'à 512 Ko, disquettes de plus de 300 Ko et surtout une capacité d'affichage sur écran à cristaux liquides bien supérieure à celle des portables (jusqu'à 25 lignes de 80 caractères pour DG One de Data General). Comme les portables, ces machines sont d'un faible encombrement. Elles pèsent généralement moins de 5 kg. Les constructeurs ont repris l'idée qui est le point fort des portables, à savoir l'intégration des logiciels résidents en mémoire morte : Wordstar sur Epson HX-8, Lotus 1.2.3 sur HP-110, Easywriter sur Sharp PC-5000. Le seul inconvénient de ces machines remarquables par leurs performances, leur facilité de transport, le prix qui dépasse les 25 000 F.

Dans cette catégorie, le nec plus ultra est sans aucun doute, le DG One de Data General. Cette machine pèse 3,6 kg. Elle est compatible IBM-PC. Elle est la première à être dotée d'un écran LCD de 25 lignes de 80 caractères. Elle est construite autour d'un microprocesseur Intel C-Mos, 50 C 88; associé à une mémoire de 128 Ko extensible à 512 Ko. Elle est équipée d'un ou deux lecteurs de disquettes 3" 1/2 de 737 Ko en mode formaté. Là encore, c'est une nouveauté.



L'Epson HX-20

En option, le DG One peut recevoir une unité de disquette 5" 1/4, un disque dur 5" 1/4, un modem intégré permettant la connexion aux ordinateurs centraux, une batterie donnant une autonomie de 10 heures, une petite imprimante pesant 2,2 kg ayant sa propre batterie intégrée. Le DG One fonctionne sous MS/Dos, CP/M86 et Venix (version sous licence AT & T d'Unix). La version de base (un lecteur de disquettes) est commercialisée 29 000 F HT environ.

Dans cette catégorie, deux autres machines sont particulièrement intéressantes : le Sharp PC 5000 équipé d'une mémoire à bulle tout comme la Grid, autre Rolls Royce du créneau, et le HP-110 de Hewlett-Packard qui



L'ordinateur portable Apricot

6 kg. Ses «plus» : la reconnaissance sociale et la transmission infra-rouge pour la souris et le clavier. Il comporte une mémoire Ram de 256 Ko extensible à 760 Ko, une unité de disquette 3" 1/2 de 720 Ko, un écran LCD de 25 lignes de 80 caractères.

POUR QUELS BESOINS ?

Face à ces trois catégories de machines, on peut s'interroger sur l'intérêt qu'elles présentent. En fait, il faut tout

d'abord s'interroger sur les besoins à satisfaire. Si on veut jouer la carte du rapport prix/performance/possibilités, il est évident que les transportables constituent la réponse. Un transportable compatible IBM-PC comme le Zenith 160 par exemple a les avantages du modèle de table compatible, en particulier l'accès à une immense bibliothèque de programmes, plus la portabilité. Par ailleurs, il est évident qu'on ne peut comparer des micros comme le DG One, le HP-110, le Grid, le Sharp PC-5000, aux portables, notamment en raison de leur puissance et de leurs possibilités. Dans bien des cas, un portable entre 8 et 10 000 F résoudra parfaitement les problèmes des cadres amenés à se déplacer fréquemment.

Les micros de la dernière génération, les portatifs, sont en train de révolutionner la micro-informatique, parce qu'ils suppriment toutes ces distinctions subtilement complexes des informaticiens, pour être des machines pour l'utilisateur, c'est-à-dire faciles à utiliser partout. Demain, elles seront à la portée du plus grand nombre. En attendant, rêvons un peu ou bien achetons un portable qui permette de faire bien des choses.

C.R.



Le Sharp PC-5000

comporte un écran de 16 lignes de 80 caractères. Cette machine aussi puissante qu'un ordinateur de table est réellement autonome (16 heures). Elle possède en mémoire morte quatre logiciels dont le célèbre Lotus 1.2.3.

L'accent est mis sur la facilité d'utilisation. L'ordinateur est réellement portable, sans fil qui traîne. Il fonctionne sans programmation. C'est la machine idéale pour le cadre.

Dans ce groupe des portatifs, on peut inclure l'Apricot, bien qu'il n'offre pas les mêmes facilités de transport que les autres. Il est beaucoup plus encombrant, mais ne pèse pas tout à fait



Le Hewlett Packard 110

le guépard

Les micro-ordinateurs français n'ont pas toujours une bonne réputation. HBN vient de sortir le Guépard qui, de par son nom veut s'attaquer à un marché très convoité et déjà bien servi. Les critiques sur le «made in France» sont-elles justifiées ? Le Guépard a été pour la première fois présenté au public au dernier SICOB. Il est depuis peu disponible. LED-MICRO a voulu en savoir plus à son sujet...

Le Guépard est un ordinateur qui, lorsque nous l'avons vu pour la première fois, ne nous a pas précisément attiré. Le second contact n'a pas été meilleur, nous avons été surpris par le poids de l'unité centrale. Imposant, lourd, esthétique moyenne, cela commençait mal. Un premier coup d'œil sur la documentation, première surprise, cette dernière débute par «Micro-ordinateur autonome avec sauvegarde totale (y compris vidéo et disque) d'une heure environ». Dès lors, le poids important trouvait toute sa signification, et ce n'est pas sans étonnement que nous ouvrîmes le boîtier principal. L'alimentation très généreuse est placée à côté d'une batterie de taille respectable.

Un coup d'œil rapide permet d'observer un fond de panier très respectable. Il est possible d'insérer sept cartes de taille confortable. Dans la version testée, il restait quatre emplacements disponibles, qui permettent de connecter soit des extensions mémoire, soit une carte graphique haute résolution, soit

des cartes entrée-sortie pour piloter des processus externes. Les lecteurs de disquettes fixées verticalement, sont placés devant les cartes extensions (taille maximum des cartes extension 20 cm de large, 23 cm de haut), de ce fait, l'encombrement est non négligeable. Ceci explique la taille de l'appareil. Un évidement à l'arrière de l'appareil permet d'implanter une extension du bus. Le micro-processeur est un grand classique, puisqu'il s'agit du Z80 tournant à 4 MHz. En version de base, la mémoire vive est de 64 K, la mémoire ROM à 2 K octets, peut être augmentée jusqu'à 16 K octets.

L'écran est un 12 pouce haute résolution anti-reflet.

L'affichage est de 24 × 80 colonnes ou de 16 × 64. Le Guépard possède en vidéo 32 couleurs, il est possible de monter un moniteur interne en option, ou d'utiliser une prise Péritel placée sur la face arrière. En version de base, la machine est équipée de deux lecteurs de disquettes 40 pistes double densité - double face permettant 360 k octets



formatés. Le clavier est séparé et son architecture est bien réalisée. Son utilisation est agréable. Il possède un pavé numérique séparé, 15 touches de fonction programmables.

Les différentes interfaces disponibles sont : Interface parallèle type centronics, Interface série RS 232 C, une entrée light pen, une entrée - sortie cas-

sette, une sortie péritélévision RVB. De plus, on trouvera sur la machine un générateur de son programmable trois canaux, une horloge temps réel gardant l'heure même l'appareil hors tension.

Les extensions sont nombreuses. Notons des lecteurs disque souple de 720 koctets, des extensions mémoire

permettant de porter la mémoire vive à 3×256 koctets, un processus arithmétique, une interface IEEE 488, un convertisseur A/D, un convertisseur D/A, un moniteur couleur haute résolution. Mais ceci n'est pas fini car les amateurs d'électronique pourront équiper le Guépard d'un programmeur d'eprom. Les fans de jeux trouveront des

CONTRE-MESURES

manettes de jeux et une souris.

Actuellement, deux systèmes d'exploitation sont disponibles, mais le constructeur annonce que le MP/M est actuellement à l'étude pour l'adaptation multipostes. Il est à noter que les deux systèmes d'exploitation disponibles, sont ici en version française (rare et très intéressant).

LA DOCUMENTATION

La machine mise à notre disposition était accompagnée de trois documents : deux classeurs, un document photocopié.

— Manuel d'utilisation et systèmes d'exploitation

— Manuel SBASIC sous NEW DOS 80

— Résumé des commandes CP/M PLUS (R - 3).

L'utilisation de classeurs possède un double avantage. Pour le constructeur, il facilite les mises à jour. Il n'est pas nécessaire pour chaque modification de réimprimer tout l'ensemble de la documentation. Pour l'utilisateur, la formule permet d'insérer entre les pages sa propre documentation, par exemple le listing de petite routine, ou sous-programme utilisant les ordres décrits. Il n'est pas rare qu'un programmeur utilise un nombre réduit des ordres disponibles sur une machine.

Une ou deux fois par an, pour une application spécifique, notre analyste chevronné voudra insérer un ordre peu connu. Si dès la première mise au point, il insère dans son classeur son application, six mois plus tard, ayant oublié les astuces, il pourra, grâce à son archivage refaire la même application sans perdre de temps.

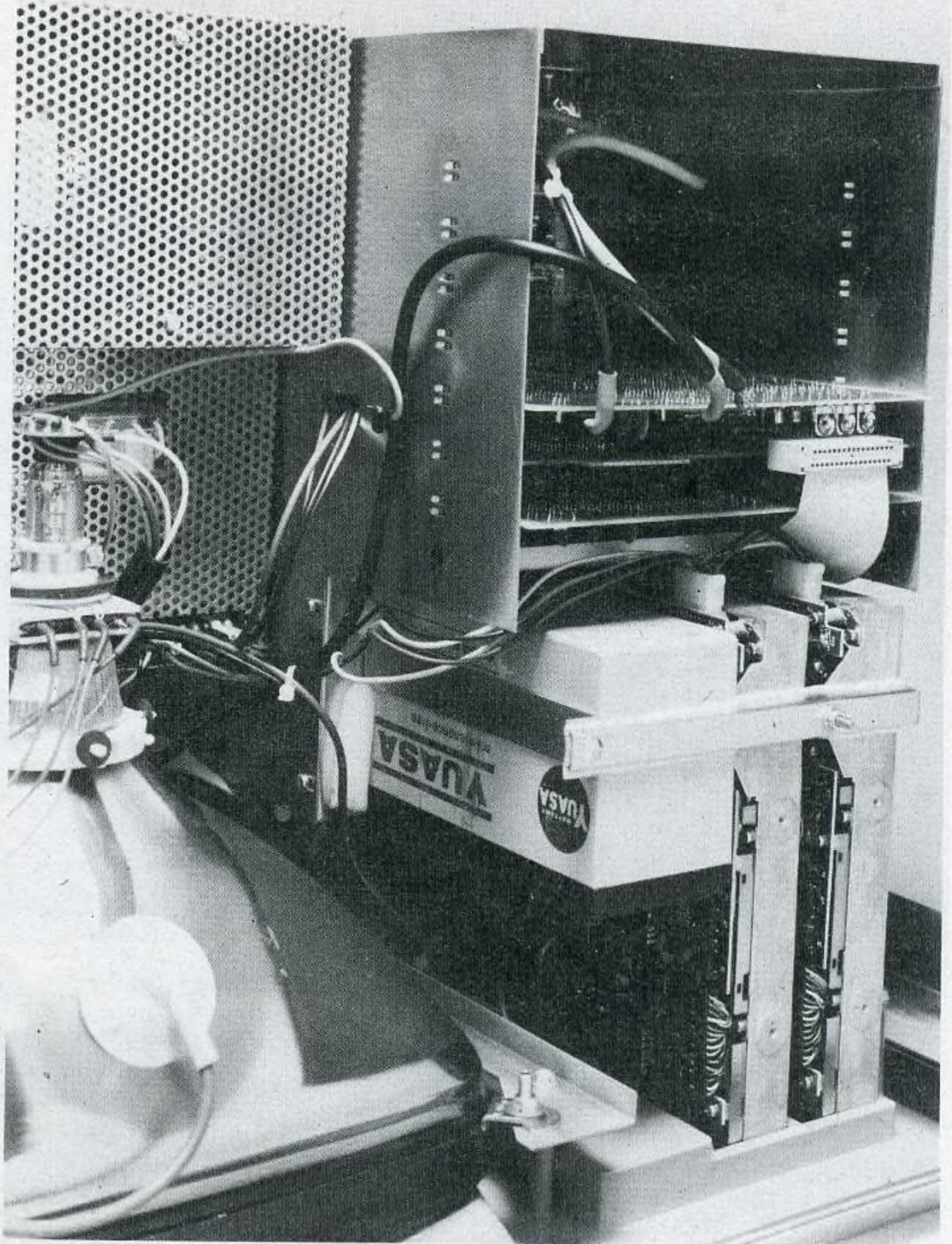
MANUEL D'UTILISATION ET SYSTEMES D'EXPLOITATION

Ce manuel est divisé en trois parties :

- Manuel d'utilisation
- Manuel New Dos 80 - 2
- Manuel CP/M Plus 3

Manuel d'utilisation

En préambule, le manuel vous annonce que vous êtes abonné gratuitement pour six mois au club Guépard.



Ce club permet de se tenir au courant des évolutions du produit, des extensions, des logiciels, ainsi qu'une réponse aux problèmes posés par les adhérents.

Le contenu de la première partie concerne un descriptif général de la machine, la procédure de mise en route, l'utilisation du clavier, les commandes de façade, les entrées sorties, la génération des sons, les lecteurs de disques et l'organisation mémoire.

Ces vingt deux pages sont très agréables à lire. Le contenu est très bien présenté. En appendice, on trouvera les diagnostics de première urgence.

Manuel New Dos 80 - 2

Ces trente neuf pages comprennent l'introduction au New Dos, la bibliothèque des commandes du Dos, les facilités du New Dos, les modules, les routines, les utilitaires et enfin un chapitre sur le Directory.

Moins facile à lire que la première partie, cet ensemble est complet, mais un peu trop dense.

Manuel CP/M Plus 3

Cette troisième partie n'est pas incluse dans le classeur, elle est livrée en papier 21 x 29,7 photocopie, que nous verrons plus tard.

MANUEL SBASIC

Ce manuel rappelle le deuxième chapitre du précédent document, complet mais très dense. Il serait souhaitable d'augmenter le nombre de pages et d'aérer un peu l'ensemble.

Chaque ordre du SBASIC est décrit, souvent accompagné de petits programmes (trop courts à notre avis).

Interrogé sur ce problème, HBN nous a répondu que le Guépard n'était pas destiné à l'apprentissage du Basic, mais plutôt aux personnes déjà initiées à l'informatique. Il est vrai que commencer avec une telle machine serait sans doute une erreur. En effet, mieux vaut entrer dans le monde de l'informatique avec de petites machines à moins de 2 000 francs.

Il est à noter que cette procédure ne pose aucun problème, car les petites machines se vendent très bien d'occasion.

Ce classeur très riche en informations est divisé en onze chapitres et six appendices, qui ne laissent aucune application de côté.

RESUME DES COMMANDES

CP/M PLUS 3

Comme nous l'avons déjà écrit, cette documentation est non conforme aux deux documents présentés ci-dessus. La première page vous rassure tout de suite, en mentionnant que la mise à jour définitive est chez l'imprimeur, et y sera jointe prochainement.

Les trente sept pages reprennent l'ensemble des commandes de ce système d'exploitation.

DESCRIPTIF GENERAL

Microprocesseur Z80 A, 4 MHz
Mémoire vive 64 koctets

Extension mémoire 3 x 256 koctets
Clavier QWERTY ou AZERTY

Entrées sorties :

- parallèle : centronics
- série : RS 232 C
- Péritel : RVB (32 couleurs)
- Crayon optique
- Cassette

Système d'exploitation (en version française) :

- NEWDOS 80. 2. 0 G

- CP/M + R3

Générateur de sons

Horloge temps réel

Lecteur de disque (deux lecteurs)

- 40 pistes (360 K)

- 80 pistes (720 K)

Ecran

12 pouces monochrome vert

- 12 pouces monochrome ambre

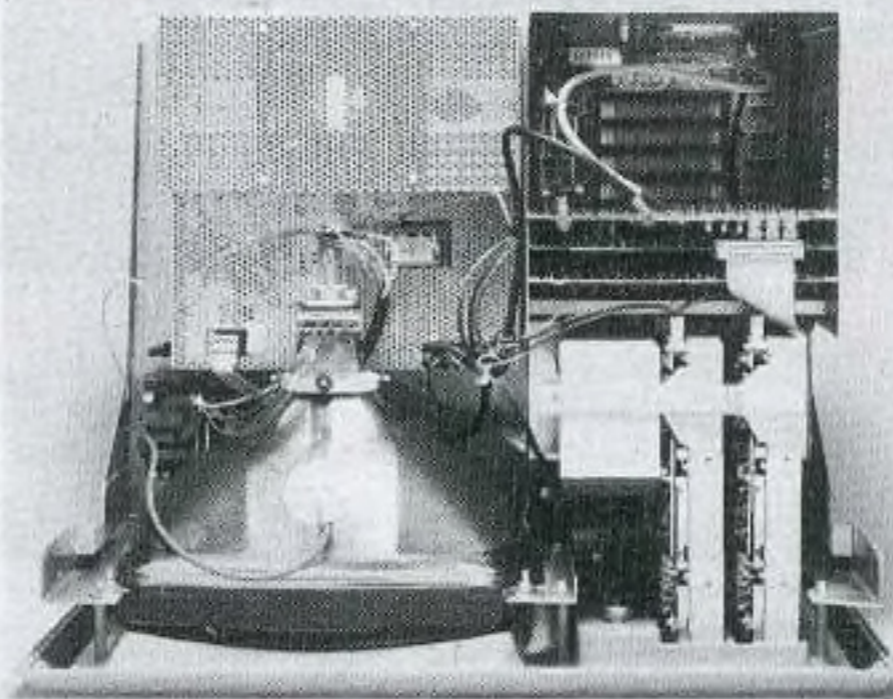
- 12 pouces couleur

Couleur

- 32 couleurs (option haute résolution possible)

Dimensions : 45 x 55 x 34 cm

Poids : 25 kg



Que penser du Guépard ? Il semble que cette machine n'ait pas encore su attirer le public (méconnaissance du produit, esthétique moyenne, méconnaissance des possibilités). Toujours est-il qu'après avoir fait tourner quelques programmes dont deux progiciels (Supercalc 2, dBase 2), le Guépard semble être rapide. Nous avons essayé quelques boucles, une boucle FOR-NEXT de 10 000 passages prend 12

secondes : l'interpréteur travaille vite. Le rapport qualité/prix est plus que favorable, en effet les machines possédant une sauvegarde totale dans cette gamme de prix sont inexistantes. Rappelons que ce genre d'option coûte très cher. Les différentes extensions permettent une utilisation du Guépard dans de très nombreux domaines, gestion, acquisition de données, etc. Le seul problème que nous ayons à mentionner concerne, en mode édition, la touche retour en arrière du chariot. En mode minuscule, il y a effacement de la ligne. En mode commande, ce phénomène peut être inversé sur les majuscules. Ce petit problème sera vite amélioré par le constructeur. Cette machine, d'un prix très raisonnable, n'aura aucun problème à s'adapter à des applications professionnelles. Une distribution directe permet au Guépard d'avoir un rapport qualité/prix actuellement sans comparaison avec la majorité de ses concurrents. Avant tout achat d'une machine de moins de 15 000 F, le Guépard devra toujours être essayé.

C.-H. Delaleu

PRIX

Version de base

le bloc unité centrale

le clavier 102 touches

le câble secteur

1 manuel d'utilisation, système d'exploitation

1 manuel SBASIC

1 manuel technique

1 disquette NEWDOS 80-2-OG-80 colonnes

1 disquette NEWDOS 80-2-OG-64 colonnes

1 disquette CP/M + 3 13 900 F

Carte moyenne résolution

graphique 2 000 F

Carte haute résolution

graphique 5 000 F

Carte extension

mémoire 64 K 2 400 F

Carte entrées-sorties IEEE

+ processeur arithmétique . 4 500 F

Moniteur interne

couleur 4 500 F

Disquettes 720 K + 2 000 F

Context MBA

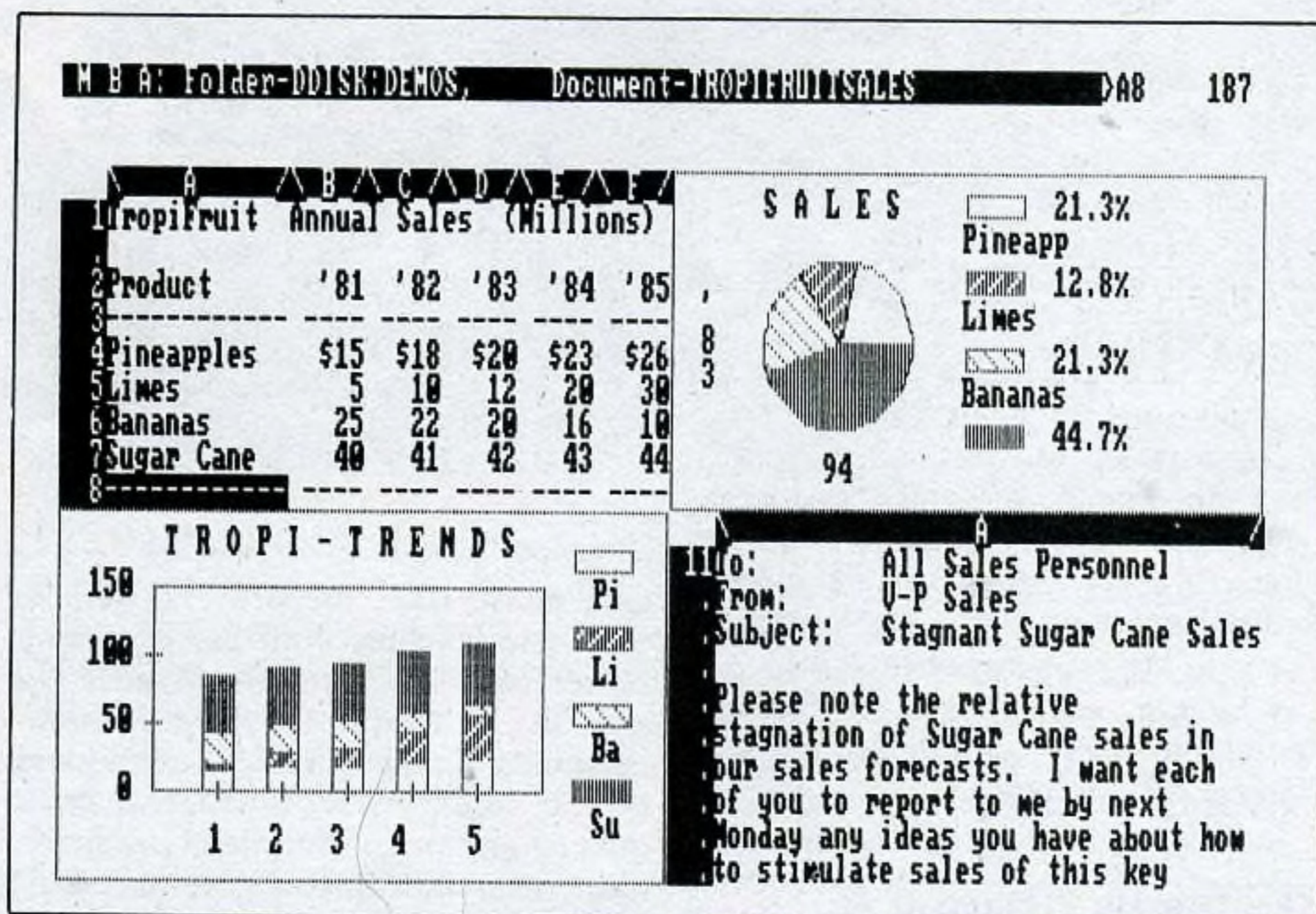
Nous avons annoncé il y a quelques mois la naissance d'une nouvelle rubrique dans LED MICRO. Après les contre-mesures « machine », voici les « progiciels à l'essai ». Rappelons qu'un progiciel est un logiciel destiné à une vente en grande diffusion.

C

Context MBA a été mis au point par Context Management Systems, une société établie en Californie. C'est un progiciel dit « intégré ». En effet, autour d'un noyau de commandes s'articulent 6 modules :

- un tableur
- une base de données
- un traitement de texte
- un module graphique
- un module de masquage
- un module communications.

L'ensemble se présente sous la forme d'une abondante documentation accompagnée de cinq disquettes. C'est la version Hewlett-Packard éditée pour le 9816 S que nous avons testée. Les cinq disquettes comprennent les 6 modules ainsi que le système d'exploitation. Le gros avantage de Context MBA concerne la possibilité de diviser l'écran en plusieurs fenêtres au choix et de parachever ainsi différents modules (ex. : une lettre, un graphisme et un tableau).



Un tableau, une lettre et deux graphismes sont affichés ensemble sur l'écran : c'est un document de synthèse très réussi.

I C	Current (ma)	Total Current	Prop Delay (ns)	Path	Number
74LS175	12	36.0	25	A	3
74S00	57	171.0	7.5	C	3
74LS00	4.4	13.2	15	C	3
74LS367	24	144.0	40	D	6
TOTAL	189	648			33

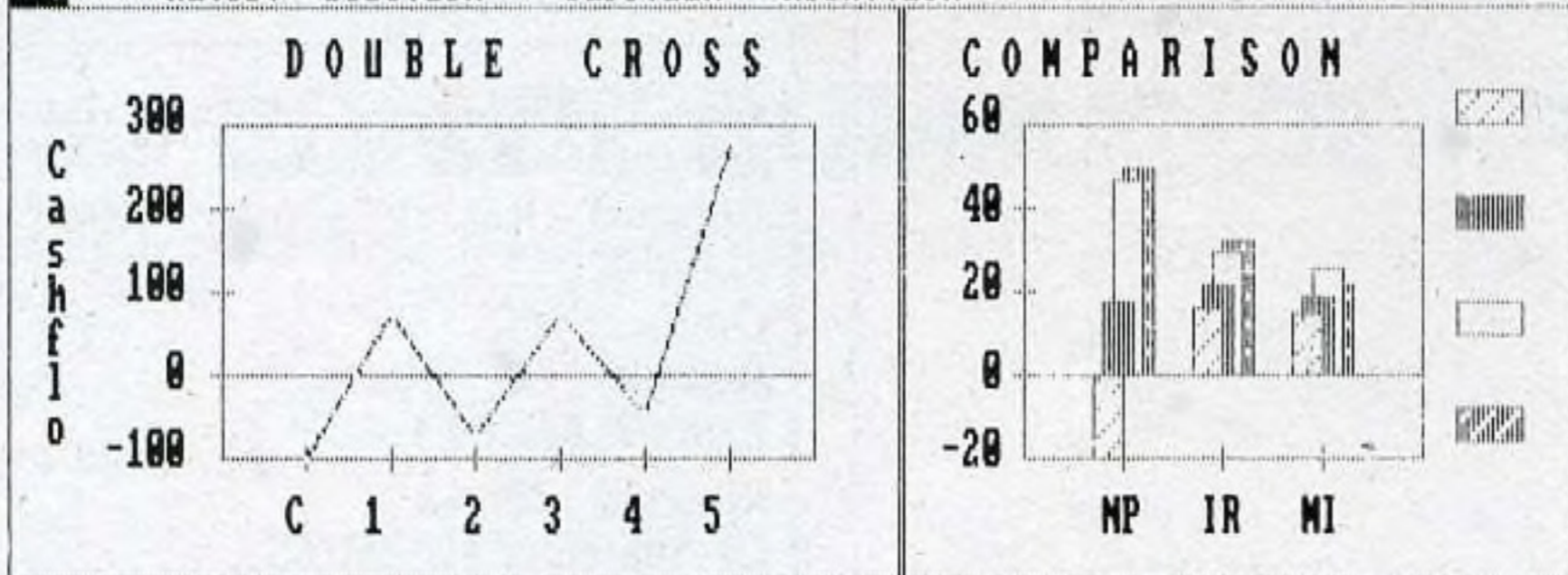
New screen
Sam-

As you can see, since we are putting 5 of our new memory boards into the new box, power for the memory is going to draw almost 3.5 additional Amps from the power supply. I don't think that the design team is going to be especially happy to see that!!

On the positive side, the propagation delay specs are well within the goals we outlined three years ago

Un tableau issu de la base de données reprenant les caractéristiques principales des composants logiques TTL est accompagné d'une lettre.

	A	B	C	D	E	F	G	H	Total	NPV	IRR	MIRR
1 Financing	Cap.											
2 Alternatives												
3 Up Front	-175	-125	50	100	150	200	\$200	\$-19	15.6%	15.2%		
4 Mid load	-75	-100	-25	50	150	200	200	\$18	21.1%	18.9%		
5 Gradual	-50	-75	0	50	125	150	200	\$47	29.7%	25.2%		
6 Dble cross	-100	75	-75	75	-50	275	200	\$50	31.4%	21.4%		
7 Rates:	Disc:18%	Safe:12%	Reinv:15%									



Représentation de chiffres : deux graphismes permettent de saisir immédiatement le contenu du tableau.

LE TABLEUR

Ce tableur possède 999 lignes, les différentes cellules sont notées de A à CQ, ce qui autorise la manipulation d'un nombre très important de variables. Les connaissances de Visicalc et Multiplan seront en terrain connu avec ce tableur. Il est doté des fonctions mathématiques les plus utilisées ainsi que des fonctions logiques.

LA BASE DE DONNEES

Cette base de données permet de créer toutes sortes de fichiers : carnet d'adresses, fichier employés, stocks, etc. C'est, en fait, une collection d'enregistrements regroupant des articles d'origines diverses. Cette collection peut être divisée en série de documents spécialisés dans des applications précises. Il est possible de réaliser toutes les fonctions généralement rencontrées sur les bases de données.

LE TRAITEMENT DE TEXTE

Le traitement de texte de Context MBA est très performant, les modifications se font instantanément. Le seul problème que nous avons rencontré concerne l'application aux textes français. Les «à» et «ç» ne sont pas acceptés, ceci posera des problèmes. Il est dommage qu'une adaptation n'ait pas été prévue.

LE MODULE GRAPHIQUE

Le module permet de présenter de manière graphique des données numériques et offre ainsi une lecture plus rapide. Différentes possibilités sont offertes, cela va du «camembert» aux lignes, en passant par les barres. Les fonds de barres peuvent avoir des teintes diverses. Il est possible de remplir une surface comprise entre deux courbes, etc.

PROGICIELS A L'ESSAI

LE MODULE DE MASQUAGE

Il est courant dans l'emploi de bases de données d'utiliser des masques de saisie. On appelle un masque, une délimitation précise de l'écran qui permet une présentation claire des données à saisir, ou transmises à l'opérateur. L'écran peut ainsi être divisé en une série de cellules présentant une donnée. La position et la taille des cellules est programmable par l'opérateur.

LE MODULE COMMUNICATION

Ce programme permet d'échanger des données par modem via le téléphone avec d'autres systèmes. Le protocole d'échange des données est programmable. Il est possible de travailler en

ASCII ou EBCDIC (format IBM), la vitesse de transmission est de 300 ou 1 200 bauds.

LA DOCUMENTATION

Chez Hewlett-Packard, on ne plaisante pas avec la documentation. Une fois de plus, la règle est respectée, et c'est plus de 400 pages qu'il faudra lire pour faire le tour de Context MBA.

Chaque application est largement détaillée. Il est à noter qu'une disquette comprend des exemples d'application de ces progiciels. La documentation est très explicite, il n'est pas nécessaire de relire. Le seul inconvénient de cette documentation concerne sa langue. Une fois de plus c'est en anglais, décidément les constructeurs ne veulent pas investir dans les traductions.

CONCLUSION

Context MBA n'est pas un petit progiciel, car c'est 512 Koctets en mémoire qu'il vous faudra pour travailler en toute sécurité.

Les différents programmes sont livrés en code, c'est-à-dire compilés. Il convient de noter que le 9816 peut travailler en BASIC, HPL, PASCAL ou ASSEMBLEUR. Ici, c'est Pascal qui a été choisi. La vitesse d'exécution est donc multipliée par 7 par rapport à une utilisation BASIC. Lequel, sur cette machine, est déjà 8 fois plus rapide que MSBASIC sur MSDOS par exemple. Sur 9816 Context MBA tourne donc très vite, et offre une grande souplesse de manœuvre.

Le gros avantage de ce progiciel concerne l'utilisation de plusieurs documents affichés ou dessinés à l'écran en même temps. Pour le business-man c'est un outil très intéressant.

Les logiciels intégrés sont aujourd'hui très performants, leur coût reste encore important par rapport aux progiciels classiques.

Avec un moniteur couleur, Context MBA vous propose 16 teintes. De même, afin de profiter de toutes les possibilités, il sera nécessaire de connecter une imprimante (HP 82906 A), une table traçante (HP 9872 C/T ou HP 7470 A).

C.-H. Delaleu

Employer	Industry	Name	Phone
Siegal & Siegal	Law	Brian Davis	(212) 467-9854
Statewide	Banking	Richard Martin	(415) 678-3321
CompuSoft	Computers	Mary Nelson	(213) 435-9121
Kelaan & Co.	Accounting	Daniel Smith	(213) 543-3909
Datacorp	Computers	Sally Winter	(415) 722-0900
General Kitchens	Food	Barney Bates	(701) 655-2341
Overall	Insurance	Robert Reeve	(212) 455-8877
SafeTech	Electronics	Claire Taylor	(415) 344-1565
NewsWorld	Newspaper	Thomas Sheer	(201) 877-4500
Bates Bros.	Retail Clothing	Michael Jordan	(415) 677-8721
Berrill Corp.	Finance	Terry Tobias	(212) 344-4400
Custom Built	Construction	Melissa Morton	(203) 988-5566
Foodarama	Supermarket	Sheldon Fink	(415) 633-3009
Datanews	Electronics	Nancy Bolton	(213) 545-2233
Backup	Electronics	Sandy Altman	(212) 644-0200
Dustin Books	Publishing	Cheryl Brooks	(415) 766-5454
Holidays	Travel	Nathan Levy	(301) 777-9008
Undercover	Private Inves.	Laura Woodman	(415) 722-3344

Une autre application de Context MBA : Ici la base de données est utilisée pour enregistrer des renseignements généraux (nom, téléphone, etc.)

LE COUT DU CONTEXT MBA

CONTEXT-MBA : Context Management Systems - Edition : Hewlett-Packard - Réf. : 92196 A, 92196 Q, 92196 X.

Option : 630

Release : 2.3

Prix : 9 882,90 F HT

Matériel utilisé :

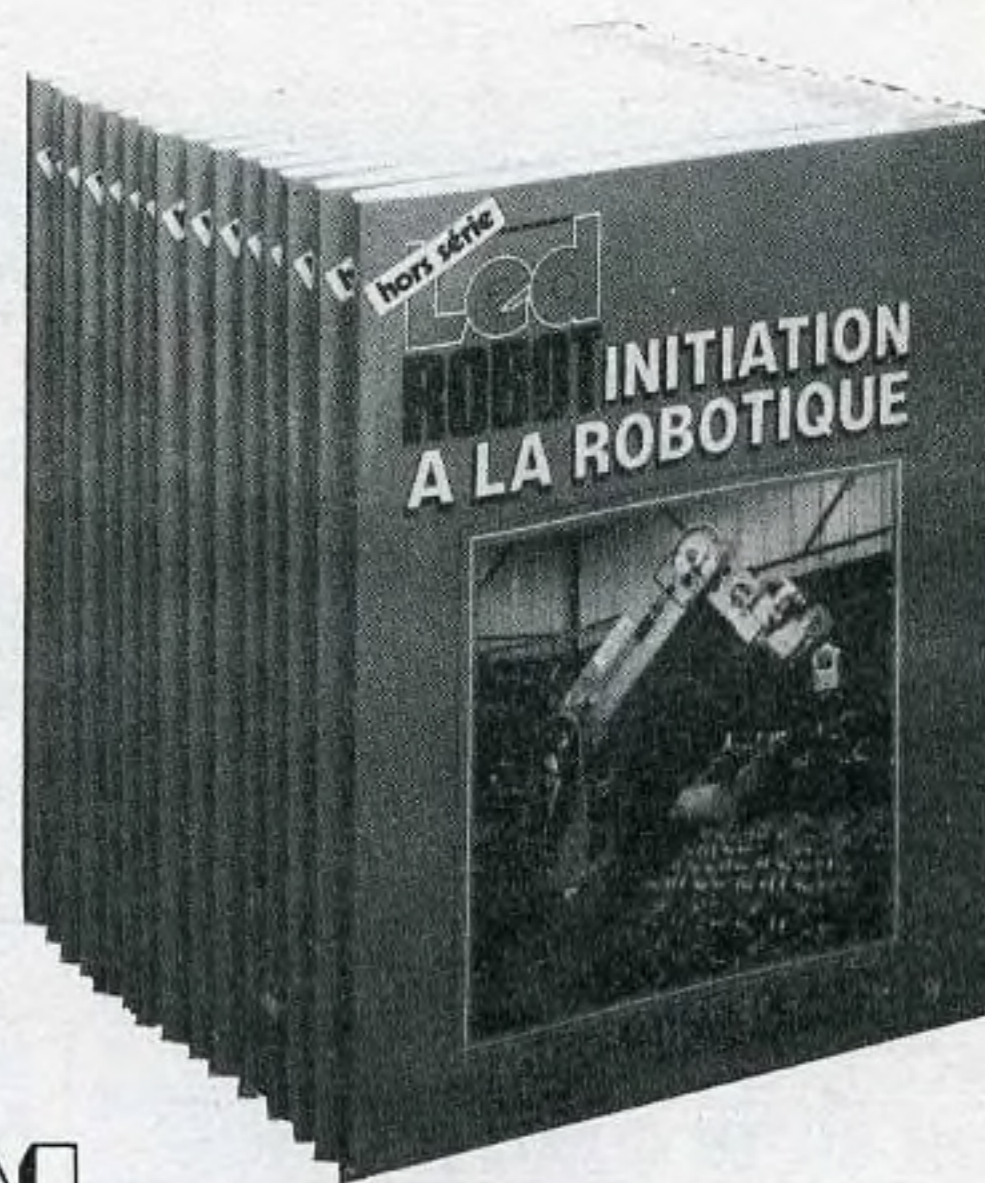
Ordinateur : HP 9816S

Imprimante : HP 82906 A

Traceur : HP 7470 A

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

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



PRIX TTC 115 F

hors série

...QUES D'AUJOURD'HUI

**Led
ROBOT**

INITIATION A LA ROBOTIQUE

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 paierai que **100 F** (port compris).

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

Ci-joint un chèque bancaire chèque postal mandat .

Adressez votre commande et votre règlement aux **EDITIONS FRÉQUENCES 1**, boulevard Ney, 75018 Paris.

Pas de nouveauté retentissante en ce début d'année. Les constructeurs font une pause. C'est dans le domaine des progiciels qu'on trouve le plus de nouveautés. Nous vous en proposons quelques exemples.

WORKSLATE : UNE SUPER CALCULATRICE



Calculer, communiquer, gérer son emploi du temps, tout ceci est possible grâce à cette petite machine pesant 1,5 kg. Grâce à sa calculatrice multifonction, à l'enregistrement des données sur micro-cassette, elle permet de résoudre tous les problèmes quotidiens.

Elle est munie d'un affichage à cristaux liquides de 16 lignes de 46 caractères, d'un clavier de 60 touches, d'un modem. Un logiciel intégré traite les tableaux numériques, le calendrier, les communications. Tout récemment Workslate vient d'être dotée de Translate PC/XT, un logiciel pour l'interfacer au PC ou à l'XT d'IBM. Workslate dispose également d'un catalogue de logiciels sur micro-cassettes : Time Project Management (gestion de projets), Electronic Mail (connection aux banques de données et messagères), Marketing management (analyse de prix). Il est également possible de connecter Workslate à une mini-imprimante. C'est d'évidence la machine du cadre facile à emporter partout avec soi, car elle sert également de réveil-matin, de bloc-notes, d'agenda, de calendrier...

MACTERMINAL : LOGICIEL DE TELECOMMUNICATION POUR MACINTOSH

Apple propose la version française de MacTerminal, le logiciel de télécommunication qui permet à Macintosh de communiquer avec d'autres Macintosh, avec des ordinateurs centraux ou des services de banques de données. MacTerminal profite totalement des caractéristiques de puissance, de facilité d'utilisation et de graphique/souris de Macintosh. La souris permet de choisir toutes les options du terminal, de même qu'elle permet le «Couper/Coller» entre les diverses applications

de Macintosh. On peut ainsi prendre de l'information de MacTerminal pour la passer dans MacWrite ou dans Multiplan, et ceci dans les deux sens. L'option de transfert de fichiers (réservée en général au transfert de textes ou de nombres) a été agrémentée d'une fonction de transfert de documents spécifiquement Macintosh. Un document graphique, un document de Multiplan et même des applications peuvent être transmises directement par voie téléphonique. MacTerminal profite également des avantages internationaux de Macintosh comme les jeux de caractères et les claviers. MacTerminal est disponible au prix public indicatif de 1 035 F HT.

LES MUPPETS ET L'INFORMATIQUE

Le clavier Muppets rend aux jeunes enfants les touches du clavier familières. L'ordre des lettres de A à Z facilite l'apprentissage de l'alphabet. Des images simples que les enfants comprennent, regroupent différents types de touches, des chiffres sur une règle, des couleurs sur une palette de peinture, des touches de direction sur une boussole. Chaque touche est très large, donc plus facile à presser pour les petits. Le clavier a une surface lavable, il est équipé d'un long cordon et de poignées de transport.

Le programme éducatif inclus comprend les caractères de Jim Henson. Kermit, Miss Piggy et ses amis, introduit des images animées qui stimulent l'apprentissage de l'alphabet et des chiffres —avec une bonne dose d'humour. Toute touche a une réponse visuelle et il n'y a pas vraiment de mauvaises réponses.

Le logiciel est conçu avec les Educateurs d'enfants Henson Associates (le «Muppets Show») et développé par Sunburst Communications, experts de matériel éducatif pour les moins de 12 ans.



UN SYSTEME DE BUREAUTIQUE INTEGRE POUR NON VOYANTS



Air France vient de présenter un système de bureautique intégré pour non voyants qui permet l'accès en braille à l'information textuelle, principale difficulté pour l'insertion d'un non-voyant dans le monde du travail. C'est un jeune ingénieur, M. Philippe Balin, lui-même non voyant, qui a mis au point ce système.

La partie matériel se compose d'un micro-ordinateur, d'un terminal braille à affichage éphémère, de deux imprimantes (une standard et une en braille) et d'un simulateur écran.

La partie dite logiciel permet d'accéder

aux quelques 30 000 programmes du système d'exploitation universel CP/M (traitement automatique des textes, consultation de l'annuaire électronique...), sans aucune modification lors de leur utilisation par un non voyant.

L'intérêt d'un tel système réside dans le fait qu'un non voyant peut l'utiliser sans plus de difficultés qu'une personne non handicapée. Il n'est pas nécessaire que l'opérateur voyant connaisse le braille, ni l'aveugle l'écriture courante, pour accéder à l'information stockée (en braille ou non).

Le grand mérite de Philippe Balin a été de réaliser, à partir de différents modules vendus séparément dans le commerce, un système dont le coût, loin d'être excessif (environ 125 000 F) est comparable à celui d'un poste de travail de bureautique normal.

Un tel poste de travail devrait se révéler très utile dans l'enseignement, notamment pour la formation de jeunes aveugles.

Cette expérience pilote, menée à Air France avec succès, a été soutenue par le ministère de la Solidarité. Ce système devrait prochainement être utilisé par d'autres grandes entreprises.

Utilisable par des non voyants mais également par des voyants, il se compose de plusieurs éléments :

- le «simubraille» fabriqué par la société Aides,

- un micro-ordinateur Leanord, qui comprend, outre un écran et un clavier standard, une unité centrale, une double unité de disquettes et une interface avec le réseau général de télécommunication,

- une imprimante de traitement de textes standard,

- un terminal braille à affichage éphémère, muni d'un clavier en braille (7 touches) et d'une plage tactile de 20 caractères, sorte de «traducteur inter-sensoriel» se substituant à la vue pour la saisie des informations,

- une imprimante braille, conçue par la société Sagem.

Ce système permet d'utiliser toute la bibliothèque de logiciels CP/M dont est doté le micro-ordinateur Leanord, donnant accès à plus de 30 000 programmes, sans la moindre modification. Les principaux, nécessaires au travail de bureau quotidien, sont : le traitement de texte, le répertoire électronique, l'annuaire électronique, l'agenda électronique, le logiciel calcul.

PENCIL II : POUR JOUER ET S'INITIER

Le Pencil II a été conçu par Hanimex pour répondre à trois demandes des utilisateurs. Pour s'initier, il possède un basic en cartouche, simple, complet et évolutif, des touches de clavier pré-programmées : instructions basic et graphiques, 4 générateurs de sons indépendants dont un de bruits, 1 basic graphique très complet, 6 touches de fonction, le câble de liaison magnéto-cassette est fourni avec la console et permet l'usage d'un magnéto standard.

Pour jouer : il reçoit ses propres cartouches et grâce à un adaptateur, toute la bibliothèque de programmes au standard Coleco.

Pour progresser : il permet un usage semi-professionnel avec : une sortie imprimante parallèle en standard, et le câble de liaison (option), une mémoire

additionnelle de 64 K, une carte 80 colonnes pour branchement avec un moniteur spécialisé, une sortie série RS 232 C, un contrôleur de disquettes, 2 lecteurs de disquettes de 163 Koctets chacune, le système d'exploitation CP/M et le Basic Microsoft.

Il est construit autour d'un Z80A et possède 20 Ko de Rom et 18 Ko de Ram, ainsi qu'une prise Péritel. La console est vendue environ 2 000 F.



J'APPRENDS LA C.A.O.

Si vous vous sentez l'âme d'un dessinateur ou d'un architecte, voici un programme qui permet tout simplement de créer des dessins en trois dimensions. A l'aide de nœuds, de vecteurs, de cercles : créez des cellules, composez des dessins. L'ordinateur vous donnera non seulement les vues normalisées mais aussi les vues en perspective sous tous les angles, de près comme de loin. De nombreuses autres options : rotation et translation automatiques, duplication de l'objet en perspective, sauvegarde du dessin sur cassette, etc. Utilisant les ordres de la même façon que sur des gros ordinateurs, voilà une très bonne initiation à la Conception Assistée par Ordinateur.

Ce logiciel est disponible sur Oric 1 et Atmos. Il est au catalogue de Loriciel qui propose un vaste choix de logiciels, sur Oric, Spectrum, Thomson, Commodore, ZX81, Segre-Yeno. Celui-ci est vendu entre 150 et 200 FF.

A LIRE



Les ordinateurs

J.M. Johnston - Editions Lonreys

Tome 1 Qu'est-ce qu'un ordinateur ?

Tome 2 Comment fonctionnent-ils ?

Tome 3 Comment s'en servir ?

A quand la bibliothèque rose pour les mordus de la micro-informatique ? Cela ne saurait tarder avec cette nouvelle série, proposée par l'éditeur Lonreys qui s'adresse aux enfants à partir de huit ans.

Cette initiation à l'informatique et aux ordinateurs est divisée en trois tomes, chaque livre traitant d'un sujet particulier.

Le premier tome est une présentation générale d'un ordinateur. A partir de nombreux exemples pris dans la vie courante (automatisation d'une usine, fichier d'un commissariat de police, gestion de stock d'une épicerie...), tous les éléments (unité de traitement, entrées sorties, mémoire) constituant un ordinateur sont décrits.

Le second tome analyse en détail les composantes de l'ordinateur et son fonctionnement. C'est ainsi que des notions de calcul binaire sont données et que les mécanismes fondamentaux des langages informatiques sont présentés.

Le troisième tome est consacré aux utilisations d'un ordinateur dans un système de chauffage solaire. Toutes les étapes de la conception du programme : analyse, programmation, mise au point sont développées.

Ces trois livres sont présentés sous la forme d'un dialogue entre un adulte et un enfant. De nombreux dessins viennent agrémenter un texte toujours clair et précis. A recommander fortement aux parents qui éprouvent quelques difficultés à expliquer l'informatique à leurs enfants.

L'assembleur facile du 6809

François Bernard - Ed. Eyrolles
Le langage assembleur est un peu la bête noire des programmeurs. On le considère comme fastidieux, illisible, ou encore sujet à de nombreuses erreurs, pourtant au niveau performance (rapidité, compacité), il est imbattable. Rappelons que l'assembleur est un langage qui substitue aux codes numériques du langage machine des expressions ou mnémoniques plus

compréhensibles par l'utilisateur, par exemple 10001011 en assembleur 6809 est remplacé par ADDA (addition sans retenu en adressage immédiat). L'inconvénient d'un tel langage est qu'il existe un assembleur différent pour chaque microprocesseur, en particulier pour le microprocesseur 6809 de Motorola qui est pris comme exemple dans ce livre de François Bernard. Il est bien évident donc que cet ouvrage s'adresse avant tout à tous les possesseurs de micro-ordinateurs équipés d'un 6809 (Thomson T07, M05, Dragon, TRS 80 couleur, Vegas, Tavernier...). Après quelques rappels sur les systèmes numériques (binaire, hexadécimal) et les codes (ASCII, BCD), l'auteur fait une description détaillée du 6809 et de ses registres.



En langage assembleur, l'aspect machine est toujours lié à la rédaction d'un programme et un bon programmeur se doit de connaître en «profondeur» son microprocesseur.

Une grande partie de ce livre est ensuite consacrée au jeu d'instructions du 6809. Chaque instruction est décrite (fonction et syntaxe) à l'aide de petits exemples rédigés en assembleur.

Le dernier chapitre de ce livre traite de la mise au point d'un programme écrit en assembleur et de l'utilisation de différentes méthodes (pas à pas, points d'arrêt...) qui permettent de détecter des erreurs.

La concurrence est toujours d'actualité ! «Manuel de l'assembleur 6809» Michel Weissgerber - Ed. Cédic Nathan.

La micro en 100 questions

Bruno de Latour. Collection Micro Onde - Cédic Nathan

Une initiative intéressante de Cédic Nathan «toutes les questions que vous vous êtes toujours posées sur la micro-informatique sans jamais les demander».



Quelques exemples :
 Qu'est-ce qu'un microproces-
 seur ?
 Qu'est-ce qu'un réseau ?
 Un micro peut-il dessiner ?
 Avez-vous la sex-compurite ?
 Toutes ces questions et beau-
 coup d'autres trouvent des
 réponses dans ce livre de Bruno
 de Latour. Un livre à conseiller
 à tous ceux qui désirent, avec
 beaucoup d'humour (les BD de
 IZOARD sont très réussies), se
 cultiver en micro-informatique.

Comment choisir votre micro-ordinateur

Ilya Virgatchik - Ed. Marabout
 Un micro-ordinateur est devenu
 un bien de consommation au
 même titre qu'une chaîne hifi,
 ou un appareil photographique.
 Il n'est plus réservé à quelques
 initiés mais destiné à tous les
 consommateurs qui doivent
 être conseillés et même défen-
 dus. Alors que choisir ?
 C'est le thème de ce nouveau
 livre de la collection Marabout
 service, dont le but est d'aider
 chaque utilisateur à acquérir la
 machine qui conviendra le
 mieux à son usage.

En premier lieu un futur acqué-
 reur doit posséder quelques
 notions sur la machine qu'il
 désire acheter. Quelques expli-
 cations sont donc nécessaires,
 en particulier un petit cours sur
 le vocabulaire informatique est
 indispensable pour dialoguer
 avec votre vendeur. C'est le
 thème de la première partie où
 l'auteur définit les principaux
 éléments logiciel et matériel
 constituant un micro-
 ordinateur.
 La seconde partie de ce livre est
 consacrée au cahier des charges
 d'un micro-ordinateur. En
 effet, la question primordiale
 que doit se poser tout utilisa-

teur est que faire avec un micro-
 ordinateur (des jeux, de la ges-
 tion, des automatismes...) et
 quels périphériques sont néces-
 saires pour une application
 donnée.
 La dernière partie est un vérita-
 ble guide d'achat. Tous les
 micro-ordinateurs les plus
 répandus y sont présentés,
 accompagnés d'une fiche tech-
 nique. Des listes regroupent les
 différents micro-ordinateurs
 suivant leur utilisation (fami-
 liale, professionnelle) ou leur
 microprocesseur. Une seule
 information manque à cette
 analyse : les prix.



Macintosh votre micro-ordinateur

J.B. Touchard - Ed. Cédic Nathan

Le Macintosh (ou le «MAC»
 pour les initiés) est le nouveau
 cheval de bataille de la
 «Pomme». Destiné aux appli-
 cations professionnelles (direc-
 tement concurrent de l'IBM
 PC ; dur combat !) il présente
 de nombreuses innovations très
 intéressantes qui révolutionnent
 le dialogue homme-machine.
 Dans un premier chapitre,
 l'auteur décrit les premières
 manipulations à effectuer pour
 mettre en route un MAC.
 Le second chapitre est consacré
 aux différents outils logiciels
 (horloge, calepin, tableau de
 bord...) disponibles avec le
 système d'exploitation.
 Une fois muni de ces différentes
 notions, l'utilisateur peut ren-
 trer dans le domaine des
 fichiers et de ses nombreuses
 applications en gestion.
 Un des principaux avantages du
 Mac est son graphisme qui per-
 met de réaliser des dessins
 d'une grande définition.
 Deux programmes, utilisant le
 graphisme du Mac, sont analy-
 sés dans ce livre : Mac Paint qui
 est un logiciel de D.A.O. (Des-
 sin Assisté par Ordinateur) et
 Mac Write qui est un logiciel de
 traitement de texte.
 A noter deux nouveaux livres
 traitant du Mac :
 — Le guide Marabout du
 Macintosh. Robert Van Loo -
 Ed. Marabout Informatique
 — Macintosh Multiplan Mac
 Paint. Eddic Adamis - Ed.
 Cédic Nathan.

P.F.

P.A. GRATUITES

A vendre Laser 200 clavier, Péritel, ext. mémoire 16 Ko, magnéto K7, 2 poignées jeux, livres d'instruction. Le tout : 2 500 F (possible achat séparément). 25 K7. Jeu pour Laser 200 : 45 F la K7 (moitié prix). Tél. : 934.23.55

Vends disque dur D3I, 5 mega + cordon, utilisé 6 mois + manuel d'utilisation. Valeur à l'achat 23 600, vendu 10 000 cause double emploi. Verdera 5, rue Lafontaine, 69100 Villeurbanne.

Echange pour Yeno-Sega SC 3000 8 logiciels d'origine Faites vos Jeux + moniteur + reversi (K7) + congo + flipper + Monaco + Safari + Sinbad (Sega). Espinasse 196, rue C. de Beauregard, 73000 Chambéry.

Cse décès, vds kit MPF1-Plus (uP Z-80) avec alim. et notices d'ut., acheté 1 995 F déc. 84, pas servi, vendu 1 500 F TTC. J.C. Bauche, 143 av. B. Buyer, 69005 Lyon.

Vds ordinateur de jeux Philips Vidéopac C52 + 6 cassettes de jeux variés (n°s 1, 4, 11, 12, 34, 36). Complet, t.b.e. Prix : 1 000 F. Ou ordinateur seul : 650 F + les cassettes : 90 F pièce. Tél. : (16-61) 72.27.58 Serge.

Vds T.R.S. mod. 3 48 K + 2 disques avec nbx prog. (200), Newdos 80, Ldos, Cobol, Accel 3/4, Superscripts, etc. 9 000 F + imprimante Logabax 120 Cps : 1 000 F. Dürr M. 18, rue La Pérouse, 31120 Portet/Garonne. Tél. : (61) 72.23.18.

PME 92 Courbevoie, recherche pour Apple IIe opérateur compétent pour mise en route et travaux temps partiel. Tél. Bernard : (1) 788.47.02.

Vends Goupil 3 (fév. 84) état neuf + banc + tt de txt + pub. post. : 19 000 FF. Tél. après 19 h : 636.04.60.

Vds DAI (1984) 64 Ko + Mémocom + Paddle + Moniteur Taxan RGB Vision 1 + imprimante Seikosha RS 232 avec son câble + nombreux logiciels + revue DAInamic, 12 000 F le tout. Possib. décomposition. Tél. : (6) 086.11.23 le soir.

Vends ZX 81 (achet. 6/84) + 16 K + clavier ABS. Livres de jeux + logiciel tri, multifichiers, compte bancaire + 2 jeux + magnéto + 2 revues ORDI + tous les câbles. Le tout prêt à fonctionner, 1 500 F. Tél. : (94) 57.72.84 Chevalier, J. Hyères.

Vends Thomson T0 7 avec mémo et interpréteur Basic et extension mémoire ; garanti jusqu'au 1 mars 1985 : 3 000,00 F. (65) 48.21.62 h.r.

Vends Thomson T0 7 (déc. 83) + manettes jeux + 3 cartouches : Basic, Pictor, Trap + 2 manuels. Prix : 2 200 F. Tél. : 854.64.70.

Vds ordinateur Lynx 48 K état neuf. Extensible à 192 Ko. Prg. assembleur. Jeux. Prix : 2 100 F. F. Runavot 65, rue de Mulhouse, 02100 Saint-Quentin.

Vends TI 99/4A + cordon magnéto + magnéto + joysticks + 4 K7 (échecs, football, Tombstone City et conseil financier). Prix : 2 200 F. Boîtier d'extension périphérique : 800 F. Imprimante Axion EX 801 (compatibilité : TRS-80, PET, Apple II) : 3 000 F. Daniel Stéphane : (1) 205.05.56 (après 19 h).

Vds Oric Atmos 48 K complet + livres + cassettes logiciels. Pratiquement pas servi ; achat 1984. Px : 2 600 F, vendu : 1 800,00 F. Bouteville. Tél. : 904.97.40.

Vends Aicro Newbrain AD avec moniteur vidéo et cordons magnétos. Mr de Popow 104, rue Fontaine St Germain, 36000 Chateauroux. 16 (54) 27.04.96.

Vends Epson HX 20 + micro K7 + 48 K RAM + importante documentation + fournitures. Acheté neuf le 10 novembre 1983. Prix : 7 000 F à déb., crédit poss. Gilles au (38) 33.81.98.

**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

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.

112 pages
PRIX : 65 F

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



éditions fréquences
COLLECTION **Led** LOISIRS

DÉJA PARUS
DANS LA MÊME COLLECTION

«Les lecteurs de compact-discs»
au prix de **140 F**
(130 F + 10 F de port)

«Conseils et tours de main»
au prix de **75 F**
(68 F + 7 F de port)

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