

LOISIRS TECHNIQUES D'AUJOURD'HUI

hors série

Led

MICRO

PROGRAMMATION

COURS 2^{ème} CYCLE

COURS

N°32

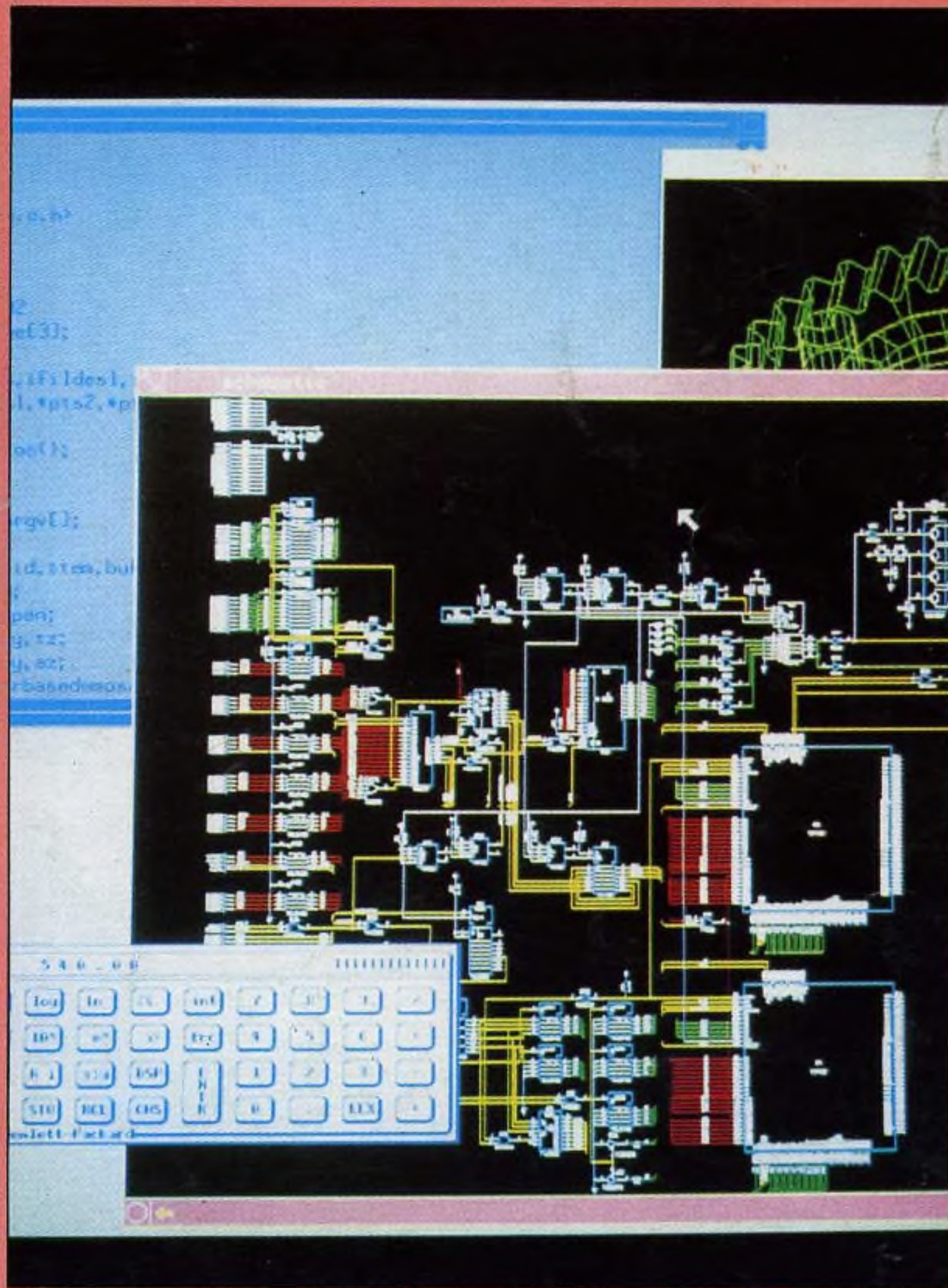
Suite
2^e cycle

N°12

**COURS DE
PASCAL**
la syntaxe
pascalienne

**COURS DE
PROGRAM-
MATION
APPROFONDIE :**
mise au point
d'un programme

**COURS
D'INITIATION AU
PROGICIEL
MULTIPLAN**



ISSN 0757-6889

TOUT SUR LES PÉRIPHÉRIQUES

NOUVEAU

dans la
**COLLECTION
«ETUDES»**
aux
éditions
fréquences



- 85 schémas
 - 20 tableaux
 - 136 pages
- Prix : 150 F

Les périphériques font partie intégrante d'un système informatique. En parallèle de l'unité centrale, qui gère et synchronise l'ensemble, ils sont responsables de différentes fonctions comme :

- la mémoire de masse : unités de disques souples et de disques durs, lecteur de cassettes ;
- le dialogue avec l'utilisateur : clavier, écran vidéo, imprimante ;
- les télécommunications : modem.

Tous ces périphériques sont décrits dans cet ouvrage avec, pour chacun d'eux, une partie technologie (principe de fonctionnement, caractéristiques techniques) et une partie interface (coupleurs d'entrées-sorties, connecteurs de liaison).

Dans chaque grande catégorie (mémoire, imprimante), une analyse comparative des différents produits existants est effectuée.

Philippe Faugeras, docteur ingénieur en électronique, est responsable matériel dans une entreprise d'informatique traitant des réseaux de P.C. Au préalable, il a acquis son expérience en travaillant sur des sujets comme les automatismes et les télécommunications dans deux grandes sociétés françaises (Bull, CGE). Philippe Faugeras est l'auteur d'un premier ouvrage «L'électronique des micro-ordinateurs» paru aux Editions Fréquences.

En vente chez votre libraire et aux Editions Fréquences.

BON DE COMMANDE

Je désire recevoir l'ouvrage «Périphériques interfaces et technologie» au prix de 160 F (150 F + 10 F de port).

Nom

Adresse

.....
A adresser aux EDITIONS FRÉQUENCES 1 boulevard Ney, 75018 Paris

Règlement ci-joint :

Par chèque bancaire

par chèque postal

par mandat

LOI SUR LES MARQUES D'AUJOURD'HUI

hors série

LED

MICRO

PROGRAMMATION COURS 2^e CYCLE

Société éditrice :
Éditions Fréquences
 Siège social :
 1, bd Ney, 75018 Paris
 Tél. : (1) 46.07.01.97 +
 SA au capital de 1 000 000 F
 Président-Directeur Général :
 Edouard Pastor

LED MICRO
 (cours 2^e cycle)
 Mensuel : 18 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) 46.07.01.97
 Lignes groupées

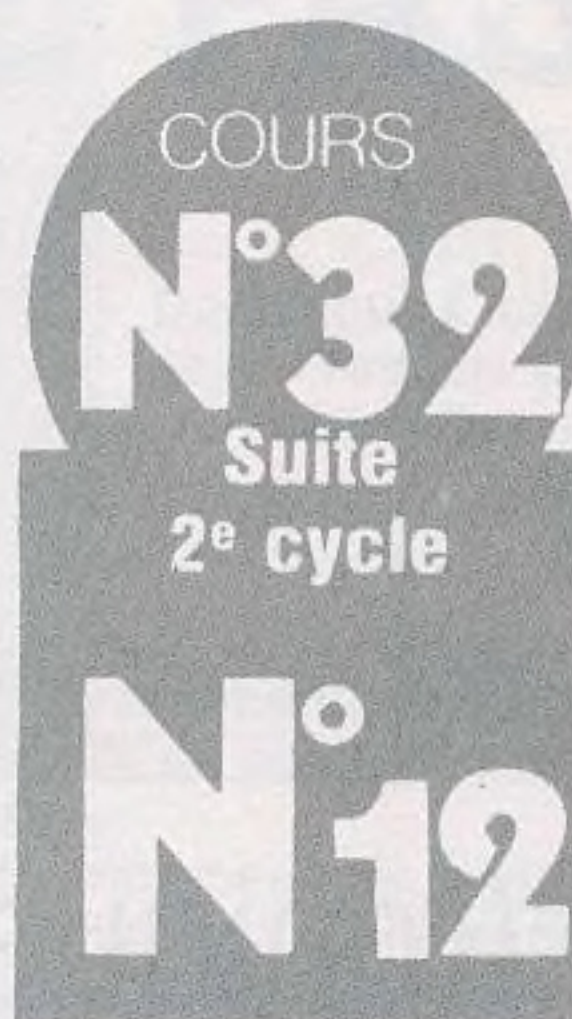
Comité de rédaction :
 Dominique Chastagnier
 Jean-François Coblenz
 Charles-Henry Delaleu
 Patrick Gueneau

Secrétaire de Rédaction
 Chantal Cauchois

Publicité, à la revue
 Tél. : 607.01.97
 Secrétaire responsable
 Annie Perbal

Abonnements
 10 numéros par an
 France : 160 F
 Etranger : 240 F

Réalisation
 Composition-Photogravure
 Edi Systèmes
 Impression
 Berger-Levrault - Nancy



SEPTEMBRE 86

COURS DE PASCAL de la page 5 à la page 16

- La syntaxe p. 6
 - Dis, comment ça marche ?
 - Les diagrammes
 - L'utilisation
 - La compréhension des exceptions
 - Exercice
 - Conclusion
- L'algèbre booléenne p. 8
 - Introduction
 - Les bouclages
 - Utilisation des booléens, ou opérateurs logiques
 - Les commandes spécifiques
 - Des exemples simples
 - Conclusion
- Complément au cours précédent p. 12

- Solution des exercices p. 12

**Dominique Chastagnier
 Jean-François Coblenz
 Patrick Gueneau**

COURS DE PROGRAMMATION APPROFONDIE de la page 18 à la page 25

- La mise au point des programmes p. 19
 - Par où commencer ?
 - Comment progresser dans la mise au point
 - Choix en cas d'erreur (1)
- Conclusion p. 25

**Dominique Chastagnier
 Jean-François Coblenz
 Patrick Gueneau**

DIALOGUE AVEC NOS LECTEURS de la page 26 à la page 32

C'EST ARRIVÉ DEMAIN de la page 36 à la page 38

COURS D'INITIATION AU PROGICIEL MULTIPLAN de la page 39 à la page 49

- Présentation sommaire p. 40
- Introduction aux tableurs p. 40
- Notions de base p. 40
- Mise en route de Multiplan p. 41
- Configuration sur disques durs p. 41
- L'écran de commandes Multiplan p. 42
- Déplacement du pointeur de cellule p. 43
- La ligne état p. 44
- Défilement de la feuille de calcul p. 45
- Commandes clavier Multiplan .. p. 46
- Touches relatives à d'autres terminaux et ordinateurs p. 47

Charles-Henry Delaleu

NOTRE COUVERTURE : Une illustration d'une application tournant sur Unix. Le DOS récent est l'un des meilleurs systèmes en ce qui concerne les applications à fenêtre et les entrées/sorties.

Microprocesseurs un cours essentiellement pratique !



Philippe Duquesne, ingénieur électronicien (I.S.E.N.) est chargé du cours de microprocesseurs au C.N.A.M. de Paris. Depuis plus de dix ans, il a pris goût à l'enseignement et il est l'auteur d'un ouvrage didactique sur l'électronique digitale et notamment d'un cours pratique de microprocesseurs. Fervent pratiquant du « dialogue » école/industrie, après avoir exercé les fonctions de chef de département électronique chez Burroughs, second constructeur mondial en informatique, il est actuellement chef du service Etudes Electroniques au sein de la direction technique chez Messier Hispano Bugatti (groupe SNECMA) avec, pour principal objectif l'introduction des microprocesseurs dans les trains d'atterrissage.

Pour ceux qui veulent aborder la micro-informatique en désirant en connaître les éléments essentiels ; ceux pour qui la « puce » ne doit pas rester un mythe.



Electronique digitale ?

Notre temps aura témoigné d'une nouvelle technique, une autre façon de communiquer avec l'électronique digitale. Philippe Duquesne, professeur chargé de cours au CNAM, a su dans cet ouvrage en expliquer clairement les fondements.



Diffusion auprès des libraires assurée exclusivement par les Editions Eyrolles.

Bon de commande à adresser aux EDITIONS FREQUENCES 1, bd Ney 75018 PARIS

Je désire recevoir le(s) ouvrage(s) suivant(s) :

- INITIATION A L'ELECTRONIQUE DIGITALE au prix de **105 F** (95 F + 10 F de port).
- INITIATION AUX MICROPROCESSEURS au prix de **105 F** (95 F + 10 F de port).

Ci-joint mon règlement par : CCP Chèque bancaire Mandat

Nom Prénom

Adresse

Code postal Ville

COURS DE PASCAL

Dominique Chastagnier
Jean-François Coblenz
Patrick Gueneau

Après avoir étudié une rudimentaire introduction du langage Pascal, nous allons, ce mois-ci, approfondir un point très précis, commun à beaucoup de langages structurés : la syntaxe.

Cette syntaxe est la grammaire du langage, imposant ses règles, ses exceptions (ou parfois ce qui semble être une exception et se justifie très bien à travers cette même syntaxe) et sa logique. Il est donc intéressant de bien maîtriser cette grammaire pour utiliser plus à fond les possibilités d'un tel langage. C'est le but de ce cours.

Dans une deuxième partie, nous affronterons (victorieusement, il faut bien le dire) le monde de la logique en présentant les variables de type booléennes. Enfin, nous corrigerons les exercices proposés la fois dernière, et en proposerons d'autres.

COURS N° 3

PLAN DU COURS

1. La syntaxe
 - 1.1. Dis, comment ça marche ?
 - 1.2. Les diagrammes
 - 1.3. L'utilisation
 - 1.4. La compréhension des «exceptions»
 - 1.5. Exercice
 - 1.6. Conclusion
2. L'algèbre booléenne
 - 2.1. Introduction
 - 2.2. Les bouclages
 - 2.3. Utilisation des booléens, ou opérateurs logiques
 - 2.4. Les commandes spécifiques
 - 2.5. Des exemples simples
 - 2.6. Conclusion
3. Complément au cours précédent
4. Solution des exercices

1. LA SYNTAXE

1.1. Dis, comment ça marche ?

Nous allons ici vous apporter la révélation pure et simple. Ce n'est peut-être pas la révélation au sens religieux du terme, mais ce n'est déjà pas si mal, et n'en demandez pas trop. Vous allez enfin comprendre le pourquoi des points-virgules, le pourquoi de leurs absences à certains endroits... Vous le voyez, vous le sentez, vous le vivez, un grand pas va être franchi.

Nous allons décrire la méthode de fabrication des instructions Pascal, méthode qui vous permettra, en la suivant scrupuleusement, de construire un programme de bonne qualité syntaxique.

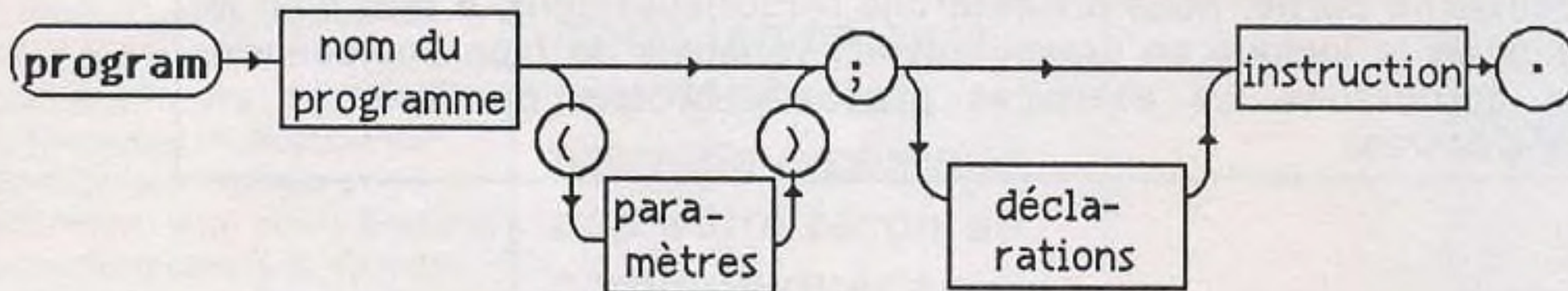
Ceci ne le rendra pas obligatoirement performant pour autant. En effet, syntaxe et algorithmique sont des sciences sans relations directes. Un exemple en langage courant est simple à proposer : «le soleil se lève à l'ouest» est correct, mais faux. Une déclaration Pascal relève du même schéma.

Partant du principe que les petites causes peuvent avoir de grands effets, il est bon de connaître la syntaxe du Pascal, pour éviter des petits bugs (que l'académie me pardonne, je ne le dirai plus) de rien du tout empêchant un programme de tourner et délicats à comprendre. Pour cela, un outil existe, le diagramme syntaxique.

1.2. Les diagrammes

Imaginez que vous ayez mélangé un puzzle et que, face à ses pièces, vous en soyez à reconstruire l'image originelle. Un programme Pascal peut être vu de cette façon, en prenant comme représentation du programme un diagramme syntaxique. Prenons un premier exemple avant de décrire plus en détail le fonctionnement.

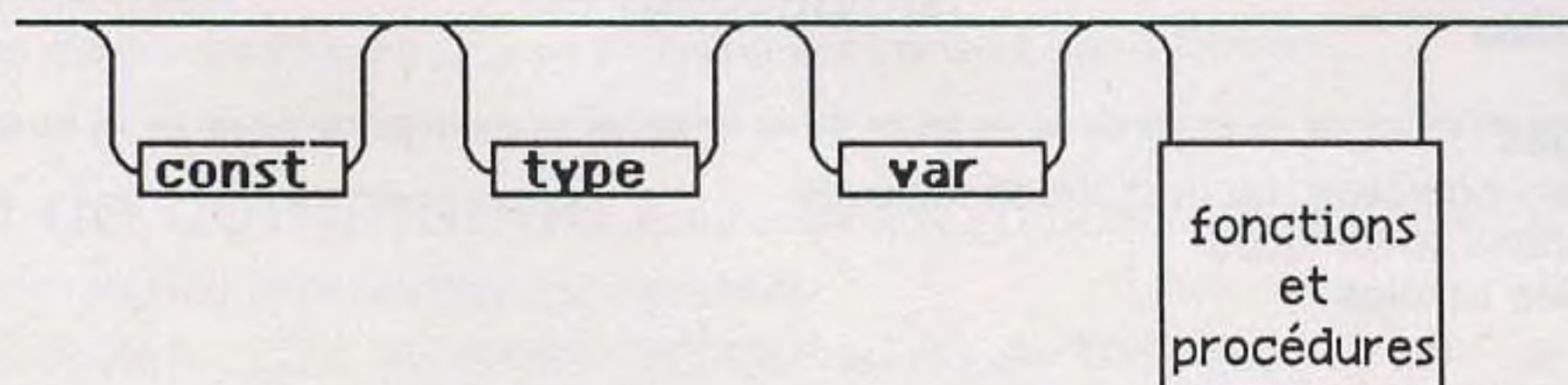
Voici le diagramme syntaxique d'un programme Pascal. Dans la théorie pascalienne, il aurait suffi de vous donner ce graphique en vous disant : «Voici la forme générale d'un programme» puis de vous donner le même type de diagramme pour chaque partie et commande de Pascal. Il semble tout de même assez vraisemblable que nous aurions reçu quelques lettres d'insultes. Donc, nous avons donné des précisions, et nous continuerons.



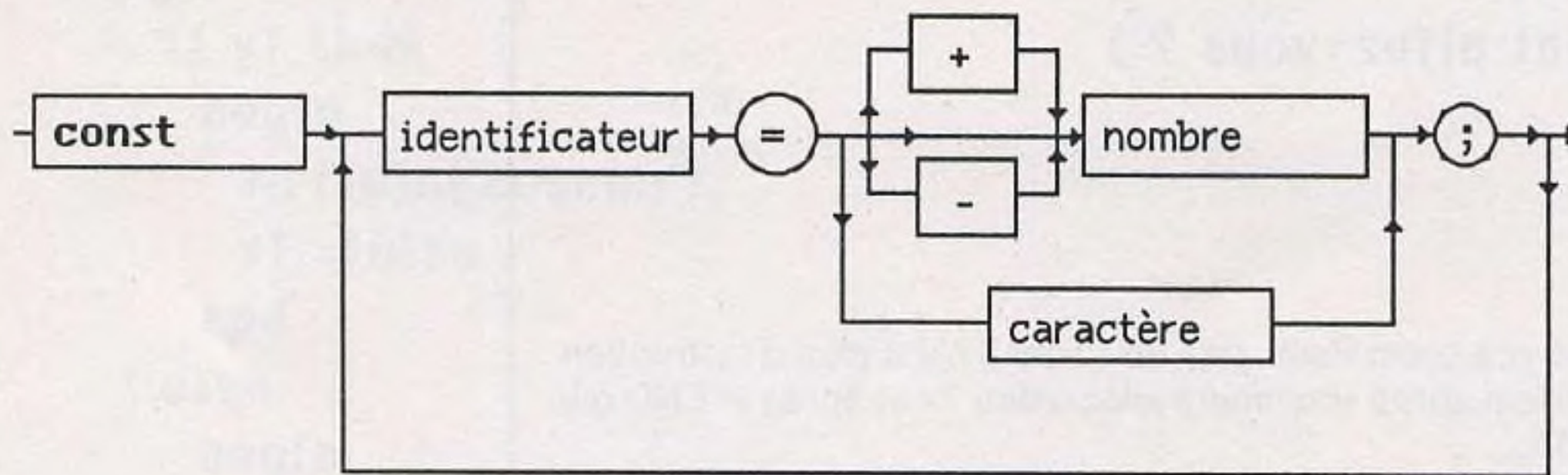
Vous voyez par ce diagramme qu'un programme commence par le mot **program** suivi du nom du programme suivi lui-même des paramètres comme **input** ou **output** si besoin est. Viennent ensuite les éventuelles déclarations (et il peut ne pas y en avoir) puis l'instruction commençant par **begin** et finissant par **end**. (Attention au point.)

1.3. L'utilisation

Il est maintenant possible de regarder chaque partie en détail. Par exemple, les déclarations :



Sur la ligne principale, vous ne trouvez rien, parce qu'un programme peut n'avoir aucune déclaration. Nous en avons vu des exemples lors des premiers cours. Nous pouvons aussi décrire la déclaration des constantes plus en détail :

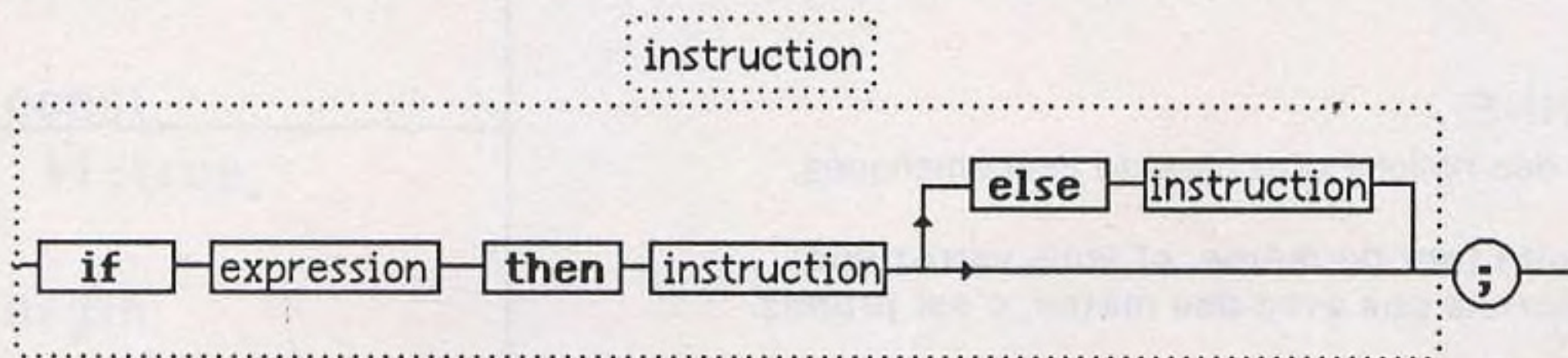


Comment lire ce diagramme. Rien n'est plus simple. Après le mot const, il doit y avoir au moins une constante déclarée, donc la ligne principale est non vide. Une constante est un nombre ou un caractère, ce qui explique les deux voies possibles. Enfin, le nombre peut être signé, et ne rien mettre est comme toujours équivalent à un plus (+).

1.4. La compréhension des «exceptions»

Les diagrammes semblent offrir un moyen clair et détaillé de descriptions pour peu qu'on sache les lire. Mais surtout, ces représentations expliquent la présence (ou surtout) l'absence, dans certains cas, de ; , ou d'autres repères habituels du langage. Un exemple immédiat : dans un test de type **if... then... else**, nous avons dit que vous ne deviez pas mettre de ; avant le **else**.

Le graphique ci-dessous l'explique fort bien :



Ceci permet d'ailleurs de commencer une intéressante dissertation sur la différence entre la notion de séparateur et celle d'instruction.

Un ; est un séparateur entre deux instructions, ou deux déclarations

Il est donc indispensable lorsque deux de ces structures se suivent, mais pas autrement. Si nous considérons le morceau de programme :

```

repeat
begin
for i:=1 to 10 do
begin
writeLn(i);
writeLn(' ça c est Paris')
end;
for j:=1 to 10 do
begin
  
```



```

writeln('hello');
writeln('coucou');
writeln('comment allez-vous ?')
end
end
until ok;

```

Regardons ceci en détail. Après «ça c'est Paris, pas de ;», car il n'y a plus d'instruction qui suit, mais la fin d'un bloc. Idem après «comment allez-vous ?» et après le END qui le suit. Idem enfin avant le UNTIL.

Ce programme n'est peut-être pas passionnant quant à son résultat mais son intérêt était purement descriptif. Nous y avons vu que

les commandes BEGIN et END représentent uniquement un identificateur de bloc-instruction, et rien d'autre.

1.5. Exercice

Nous vous proposons à titre d'exercice de réaliser les diagrammes des structures que nous avons vues jusqu'à ce point du cours. Nous les corrigerons le prochain cours.

1.6. Conclusion

Nous venons de voir un moyen puissant de décomposer un programme Pascal, autorisant la compréhension du fonctionnement interne. Il est très important de bien comprendre ce que sont ces diagrammes, car nous les utiliserons assez souvent, et en tout cas à chaque fois qu'ils nous permettront de lever une ambiguïté, ou de faire mieux passer l'information.

2. L'ALGÈBRE BOOLEENNE

Aïe aïe, nous voilà reparti vers des notions obscures de mathématiques.

Stop... lisez la suite tout de même, et vous verrez que nous ne vous ennuerons pas avec des maths, c'est promis.

2.1. Introduction

Maintenant que les choses sont bien mises au point, voilà ce qu'il faut savoir de la logique de Boole. Tout se résume à ceci : une chose est vraie ou fausse. Il n'y a pas à sortir de là et pas de tergiversations ni de rouspétances. Pas de demi-vérités. Ça permet de vivre avec un monde possédant deux états, vrai faux, 1 et 0, oui et non...

Voilà tout ce qu'il est indispensable de savoir.

Partant de là, le Pascal gère un type de données, déclarées par le mot **BOOLEAN**, comme il y a les **REAL** et les **INTEGER**. Ce type est formé par deux éléments, TRUE et FALSE (qui sont la traduction de VRAI et FAUX). Donc, si vous écrivez le programme suivant, tout est bon :

```

program vf;

var
  vf:boolean;

begin
  vf:=true;

```



```

repeat
  begin
    if vf then
      begin
        writeln('coucou');
        vf:=false
      end
    else
      begin
        vf:=true;
        writeln('c est moi')
      end
    end
  until vf
end.

```

Pourriez-vous dire ce qu'il fait ?

Vous avez aussi la possibilité de mettre en constante, comme ici :

```

program vf;

```

```

const

```

```

  vf=true;

```

```

begin

```

```

  writeln('ici, vf est toujours VRAIE ');

```

```

end.

```

2.2. Les bouclages

Un tel programme doit faire attention de ne pas aboutir à un paradoxe qui est mal géré par une telle structure. Un paradoxe est une phrase, ou un groupe de phrases, posant des problèmes de logique. Un exemple :

la phrase suivante est fausse

la phrase précédente est vraie.

Démêler le vrai du faux devient délicat, n'est-ce-pas ?

Pour un programme, cela peut l'amener à des bouclages magnifiques, par exemple si une boucle est régie par un test qui ne se réalise jamais. Pour que le programme se termine, il n'y a que la panne de courant, ou une petite main sur l'interrupteur, le RESET ou autre CONTROLE-C.

2.3. Utilisation des booléens, ou opérateurs logiques

Lorsque vous écrivez :

```

  if x=1 then writeln('x=1');

```

vous utilisez une relation logique qui est soit vraie soit fausse, qui est donc booléenne. Voilà que vous faisiez du booléen sans le savoir. M. Jourdain n'est pas loin. Mais de

plus, cette relation peut être remplacée par une variable logique déclarée, comme nous l'avons vu dans le programme précédent, où nous avons écrit :

```
if vf then ...
```

Ceci est très commode, et puissant, car cela permet d'utiliser des opérateurs entièrement spécialisés pour les boucles. Un exemple du même type

```
repeat  
begin  
  writeln('encore une fois ? '); readln(n)  
  if ((encore=false) and (n<>'0')) then encore=true;  
  writeln('c'est parti !!');  
end  
until encore;
```

Cela amène une écriture de tests logiques plus propre, plus concise et plus explicite qu'en Basic.

2.4. Les commandes spécifiques

Des commandes pour travailler avec les booléens sont disponibles. Ce sont des opérateurs qui permettent de rendre leurs utilisations plus performantes. Nous avons rencontré dans l'exemple précédent l'opérateur AND (ET en anglais) permettant de juxtaposer deux booléens ou plus, et dont le résultat sera vrai si les deux sont vrais, et seulement dans ce cas. Il est logique que, si vous écrivez A AND B, le résultat soit vrai si A est vrai et si B est vrai également. De la même façon, il existe aussi les opérateurs OR et NOT (OU et NON anglais). Pour simplifier les explications, nous vous donnons leur table de vérité, c'est-à-dire la façon de les utiliser sur le plan logique.

Voici d'abord pour AND.

AND	TRUE	FALSE
TRUE	TRUE	FALSE
FALSE	FALSE	FALSE

Vous pouvez donc voir que la seule condition pour que deux booléens réunis donnent un résultat vrai est qu'ils soient vrais tous les deux. Maintenant, l'opérateur OR.

OR	TRUE	FALSE
TRUE	TRUE	TRUE
FALSE	TRUE	FALSE

Ici, il suffit que l'un des deux soient vrais. C'est donc bien l'un ou l'autre. Les habitués de la logique booléenne reconnaîtront ici le OU inclusif et non exclusif qui stipule que l'un des deux doit être vrai, mais pas les deux. Si les deux sont vrais, le résultat est alors faux.

En exercice, nous vous proposons de créer un tel opérateur (ex. 2).

Enfin, le NOT (négation).

NOT	TRUE	FALSE
	FALSE	TRUE

Ce qui était vrai devient faux et réciproquement.

2.5. Des exemples simples

Soient des variables booléennes A, B et C. Elles prennent par exemple les valeurs suivantes :

A : = false ;
B : = true ;
C : = true.

On a alors :

A and B → false

A or B → true

not (A) → true

not (A) and (B or C) → true

A est faux

B est vrai, ça suffit

A faux, sa négation vraie

tous les termes vrais

2.6. Conclusion

Comme promis, tout ceci est compréhensible sans connaissances mathématiques de haut niveau mais vous vous rendrez compte que le résultat est d'une puissance stupéfiante et d'un intérêt permanent.

3. COMPLEMENT AU COURS PRECEDENT

Vous aurez sans doute noté que dans la liste des commandes fournie le mois dernier, nous avons omis les commandes spécifiques au Pascal UCSD, contrairement à ce qui était annoncé. Les voilà aujourd'hui.

ATAN	ATTACH	BLOCKREAD	BLOCKWRITE
CLOSE	CONCAT	COPY	DELETE
EXIT	FILLCHAR	GOTOXY	HALT
IDSEARCH	INSERT	INTERACTIVE	IORESULT
KEYBOARD	LENGTH	LOG	MARK
MEMAVAIL	MEMLOCK	MEMSWAP	MOVELEFT
MOVERIGHT	PMACHINE	POS	PROCESSID
PWROFTEN	RELEASE	SCAN	SEEK
SEMAPHORE	SEMINIT	SIGNAL	SIZEOF
START	STR	STRING	

Elles sont en nombre important, mais apportent en général une amélioration très sensible au langage. Nous consacrerons plusieurs cours, ou parties de cours, à la description de ces commandes spécifiques, pour que ceux qui en disposent puissent les exploiter au mieux.

4. SOLUTION DES EXERCICES

1. Construire un programme qui écrive la somme des 100 premiers entiers.

```
program unacent(input,output);
```

```
  const
```

```
    premier=1;
```

```
    dernier=100;
```

```
  var
```

```
    somme,i:integer;
```

```
begin
```

```
  for i:=premier to dernier do somme:= somme+i;
```

```
  writeln('la somme vaut : ',somme)
```

```
end.
```

Notons que nous avons trouvé que cette somme valait 5050. L'utilisation de la boucle est la plus simple possible. Deux variables sont nécessaires. Le programme tourne en moins de deux secondes sur notre système, à comparer aux six en Basic sur le même système. Pour aller encore plus vite, le lecteur peut utiliser, s'il la connaît, la formule mathématique donnant la somme des nombres de 1 à n, qui est :

$$\text{somme}(n) = n * (n + 1) / 2$$

Ceci donne un programme que voici :

```
program unacentbis(input,output);
```

```
  const
```

```
    nombre=100;
```

```
  var
```

```
    somme:integer;
```

```
begin
```

```
  somme := nombre*(nombre+1)/2
```

```
end.
```

Rapide, non !!


```

    write(tab_lettre[j]);
  writeln
end
end.

```

La pauvreté du langage est très nette ici, où nous devons en passer par des caractères à chaque étape. Ceci nous amènera à vous proposer des exercices dont le but sera de programmer des extensions de ces possibilités.

3. Ecrire un programme écrivant tous les mots de deux lettres imaginables. Pour cela, sachez que les lettres sont, en Pascal, un ensemble ordonné, de «A» à «Z».

```

Program mots(input,output);
var
  i,j : char;
begin
  for i:= 'a' to 'z' do
    begin
      for j:= 'a' to 'z' do
        begin
          write(i);
          writeln(j)    ( ou de façon équivalente : WriteLn(i,j); )
        end
      end
    end
  end.

```

Le fait que l'ensemble des lettres soit un ensemble ordonné est ici d'un grand intérêt. Il est très simple de faire des boucles sur cet ensemble, aussi simple que pour des nombres. Ceci vient de la structure d'ensembles que nous verrons prochainement.

4. Ecrire un programme qui calcule tous les produits de deux entiers entre 1 et 100. Le schéma est le même que dans le programme précédent, c'est-à-dire deux boucles imbriquées. Chaque boucle représente l'un des deux nombres dont on fait le produit. Le programme fort court lui aussi devient :

```

Program produits(input,output);
var
  i,j : integer;
begin
  for i:= 1 to 100 do
    begin
      for j:= 1 to 100 do
        begin
          writeln(i*j)
        end
      end
    end
  end.

```

5. Ecrire un programme qui, après avoir lu, disons cinq chiffres, fasse leur somme, leur produit et toute opération qui vous intéresse (moyenne, variance...).

Nous vous proposons une solution qui demande le nombre de chiffres à considérer. Le programme est très modulaire, une fonction par type de résultat. Le voici :

```

Program calcul(input,output);
type
  tab = array[1..20] of real;
var
  nb,i:integer;
  a : tab;
function somme(x : tab ; y : integer) : real;
var
  j : integer;
  s : real;
begin
  for j:= 1 to y do s:=s+x[j];
  somme:= s
end;
function prod(x : tab ; y : integer) : real;
var
  j : integer;
  p : real;
begin
  for j:= 1 to nb do p:=p*x[j];
  prod:= p
end;
function moyenne(x : tab ; y : integer) : real;
var
  j : integer;
  s : real;
begin
  for j:= 1 to y do s:=s+x[j];
  moyenne:= s/nb
end;
function variance(x:tab; y:integer):real;
var
  k : integer;
  v : real;
begin
  for k:= 1 to y do v:=v+sqr(x[j]);
  variance:= (v/y-sqr(moyenne(x,y)))
end;

```



```

begin
write('combien de chiffres à traiter :');
readln(nb);
for i:=1 to nb do readln(a[i]);
writeln('somme = ',somme(a,nb));
writeln('produit = ',prod(a,nb));
writeln('moyenne = ',moyenne(a,nb));
writeln('variance = ',variance(a,nb));
end.

```

Le programme ne demande qu'à être augmenté de toutes les fonctions dont vous pourrez avoir besoin. Notez que la fonction moyenne fait appel à la fonction somme. C'est un exemple d'appels d'un sous-programme à un autre. Pour cette raison, la fonction somme a été décrite avant. Dans le cas contraire, cela aurait occasionné une erreur de compilation car c'est une erreur de syntaxe. Il en va de même de la fonction Variance qui appelle la fonction Moyenne qui, nous venons de le voir, appelle la fonction Somme. Ceci est un peu un cas de démonstration, car ce n'est pas très efficace, en terme de temps d'exécution, de travailler ainsi. Mais c'est tellement beau. Des exercices pour la prochaine fois :

1. Réaliser les diagrammes syntaxiques des structures étudiées. Ce sont :

- Variables
- Types
- Une instruction
- Repeat... Until
- For... Next
- Sous-programme (procédure ou fonction)
- ...

2. Créer l'opérateur OU exclusif, que nous noterons OR selon les normes habituelles, et dont voici la table de vérité :

XOR	TRUE	FALSE
TRUE	FALSE	TRUE
FALSE	TRUE	FALSE

3. Si nous vous proposons les déclarations A, B, C suivantes, évaluer D :

A : «Les vaches sont vertes»

B : «Les petits rats laveurs naissent avec sept pattes»

C : «Le soleil se lève au Sud».

D : (NOT (A AND B)) OR ((NOT C) AND A)

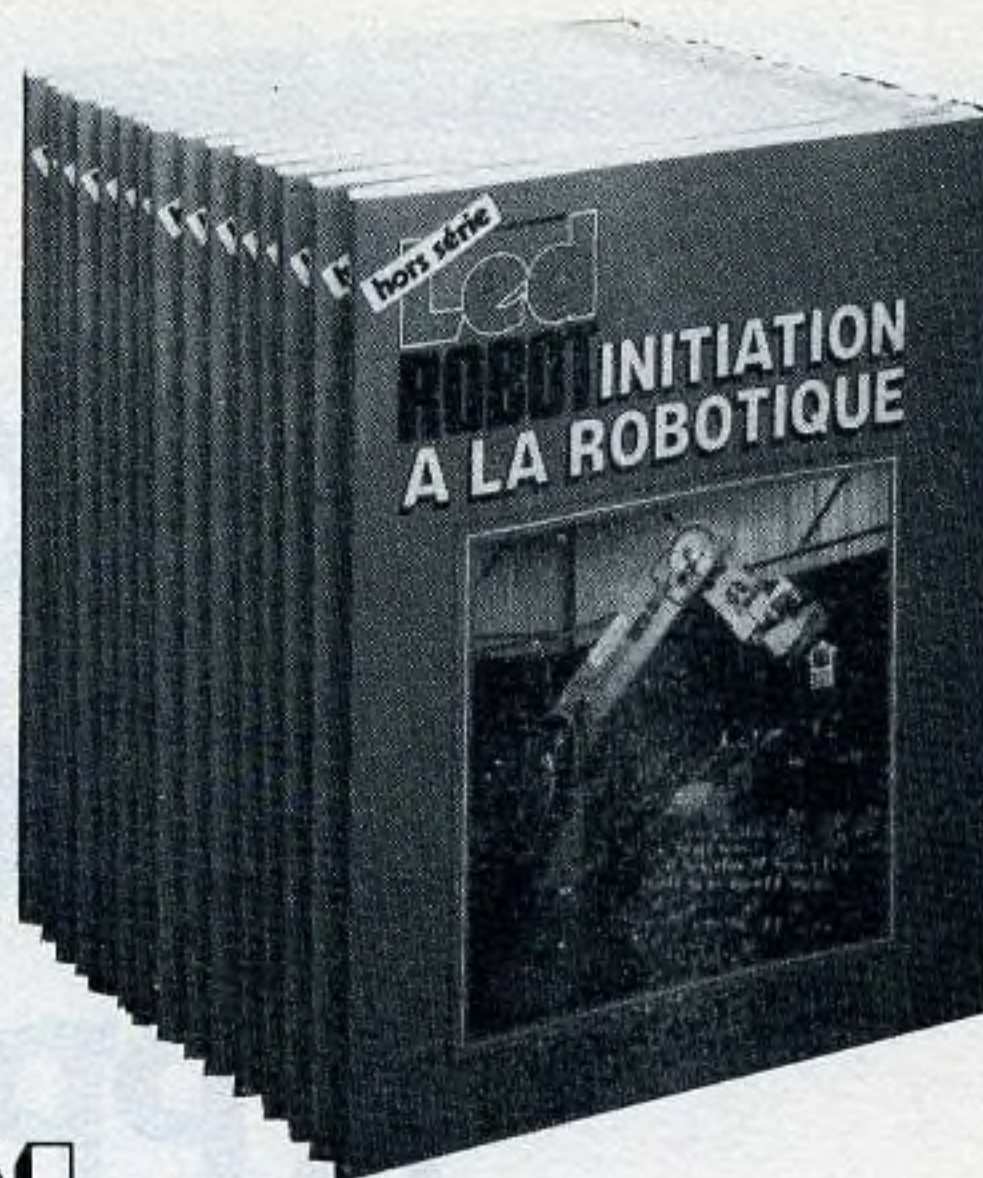
4. Etudiez la valeur de A OR NOT (A)

5. Idem avec A AND NOT (A)

6. Idem avec A OR A selon la valeur de 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
LEDES 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



Diffusion auprès des libraires assurée exclusivement par les Editions Eyrolles.

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.

COURS DE PROGRAMMATION APPROFONDIE

Dominique Chastagnier
Jean-François Coblentz
Patrick Gueneau

Après avoir vu, le mois précédent, les deux premières parties de cette série, nous abordons maintenant la mise au point des programmes. Quant aux phases de finition et d'optimisation, nous les réservons pour le mois prochain ; vous aurez ainsi le temps d'approfondir et de développer les idées que nous vous soumettons dans ce numéro.

COURS N° 12

PLAN DU COURS

La mise au point des programmes

- 1.1. Par où commencer ?
 - Morceaux par morceaux
 - Approche globale
- 1.2. Comment progresser dans la mise au point
 - Vérification en profondeur
 - Propagation module par module
- 1.3. Choix en cas d'erreur(s)
 - Utiliser les outils à votre disposition
 - Sonder l'environnement
 - Affiner la détection de l'erreur
 - Choisir le type d'intervention

1. LA MISE AU POINT DES PROGRAMMES

La mise au point conditionne, comme nous l'avons vu à maintes reprises, le succès ou l'échec de tout programme quelle que soit l'ambition du projet. Il est en effet difficile de déterminer à quel moment il est préférable de tout arrêter plutôt que de coller rustine après rustine à chaque point sensible ; et encore faut-il avoir découvert l'origine de ou des erreurs et de savoir y remédier. Notre ambition est de vous aider à acquérir un sens de la mise au point, puis à l'affiner au fur et à mesure de vos investigations. Plus on programme et plus il y a d'erreurs à corriger, donc plus on développe cette vision d'ensemble qui nous fait découvrir l'erreur en un rien de temps. Nous sommes passés par là et nous y passons toujours d'ailleurs, seulement l'expérience s'accumule.

```

1000 REM Sous programme de calcul d'une moyenne
1010 REM
1020 REM variables utilisées:
1022 REM
1025 REM 1) en appel
1040 REM   T      : tableau unidimensionnel
1050 REM           contenant les valeurs à moyenner
1060 REM   N      : est le nombre de valeurs
1065 REM           (on part de T(1) )
1067 REM
1070 REM 2) en retour
1080 REM   MO     : est la moyenne des N valeurs de T
1090 REM   SO     : est la somme des N valeurs de T
1095 REM   ERREUR: le code d'erreur en retour
1097 REM
1100 REM 3) variables locales
1110 REM   I      : indice de boucle
1120 REM
1130 REM Remarques: traite le cas où N=0 et renvoie dans ce
1140 REM           cas l'erreur -1 dans ERREUR (sinon 0)
.....
.....

```

FIGURE I

Abordons le concret, le premier conseil bien qu'il puisse paraître évident est de procéder méthodiquement, étape par étape. Libre à vous de choisir par quel bout commencer, mais surtout évitez de vous éparpiller entre les différents modules du programme. Enfin, il est toujours plus agréable à l'appel d'une procédure de connaître avec précision ce que fait cette procédure ; ainsi la figure I vous propose un exemple d'en-tête descripteur d'un sous-programme. A l'aide de ces commentaires vous savez précisément ce qu'il fait (ou plutôt est censé faire), les variables qu'il utilise, et les résultats en retour. On dépistera donc d'autant plus facilement les erreurs comme celles présentées le mois-précédent (figure de la page 20) qu'il ne subsiste aucune ambiguïté dans le rôle alloué au-dit sous-programme (la définition de l'entête vous poussera peut-être à vérifier avec plus de soin, son travail effectif et, notamment, l'utilisation des variables locales).

1.1. Par où commencer ?

Tout ceci est bien gentil, mais on ne sait toujours pas par où débiter cette mise au point (il est fréquent que rien ne tourne comme on l'aurait souhaité). Deux approches complémentaires nous sont offertes, soit commencer par l'âme du programme la partie

sans laquelle rien ne tourne, soit travailler module par module si ceux-ci peuvent être testés indépendamment.

1.1.1. Morceaux par morceaux

C'est en fait appliquer la devise «Diviser pour régner» tant une erreur peut quelquefois nous apparaître comme notre plus mortel ennemi. Il faut donc distinguer les morceaux sains du programme en opposition avec ceux soit suspects soit complètement fautifs. Réitérez cette division et vous obtiendrez vite la ou les quelques lignes responsables de l'erreur ; elle sera plus aisément repérable donc corrigable.

1.1.1.1. Au niveau d'un module

Il est plus facile de procéder de cette façon, mais, allez-vous dire, ce n'est pas toujours possible ! Et pourtant si, le plus souvent. L'idée fondamentale consiste à découper le module, que l'on croit former un ensemble indissociable, en autant de séquences dépendantes l'une de l'autre, et de tester chaque séquence à l'aide soit des résultats de la séquence précédente, soit en «trichant par l'ajout d'une routine parasite qui adaptera l'environnement faisant ainsi croire que tout est OK ! La figure II illustre cette méthode et montre bien comment dériver le flot du programme et en réguler le débit à la manière d'un cours d'eau.

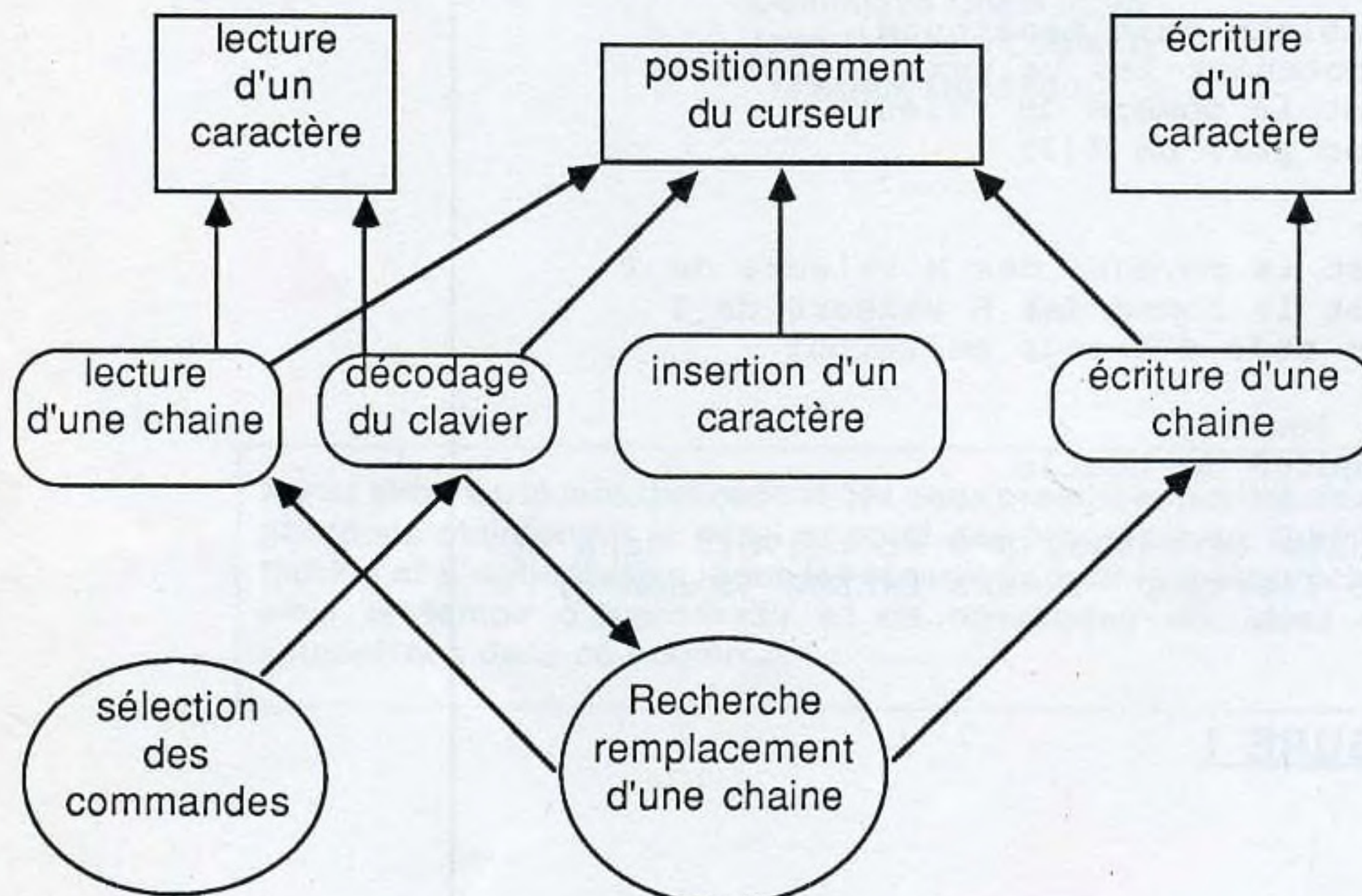


Figure II

1.1.1.2. Module par module

La situation usuelle est celle du menu d'option permettant d'accéder à différentes fonctions pour par exemple, la gestion d'un fichier (lecture, modification, ajout, etc...), des statistiques sur un lot de données. Dans ce cas, chacun des modules est en fait un programme à part entière si ce n'est des contraintes supplémentaires dans l'utilisation des variables et constantes communes (là encore la connaissance précise de ce que fait et ne fait pas chaque module est primordiale). On a donc toute liberté pour tester séparément ce sous-programme à l'aide d'un programme appelant plus simple (il ne fera que déclarer les variables nécessaires aux jeux d'essais et qu'appeler le sous-programme).

1.1.1.2.1. Remarque importante

En procédant ainsi, vous ne pourrez pas valider l'ensemble du programme, connaître la mémoire utilisée, l'interaction entre les modules, ni sa fiabilité.

1.1.2. Approche globale

Si votre application tourne autour d'un noyau central chargé de gérer l'ensemble des

fonctions utilisées, par exemple la gestion d'un fichier, ou encore un jeu interactif, prenez votre courage à deux mains et lancez-vous, il n'y a pas d'autres solutions que d'exécuter globalement le programme et de vérifier ce qui fonctionne ou pas en améliorant petit à petit la progression de l'exécution. Voici une liste d'erreurs communément rencontrées quelques solutions dans les trois situations les plus fréquentes.

1.1.2.1. La programmation est complexe

Cette approche est la plus adaptée lorsque la phase de programmation est délicate. C'est le cas des gros calculs (manipulations de tableaux...), des traitements richement paramétrés, ou des gestions multi-fichiers par exemple. On remarque qu'en général l'analyse n'a pas posé de gros problèmes mais plutôt sa mise en œuvre.

Erreurs fréquentes :

1.1.2.1.1. Erreur signalée

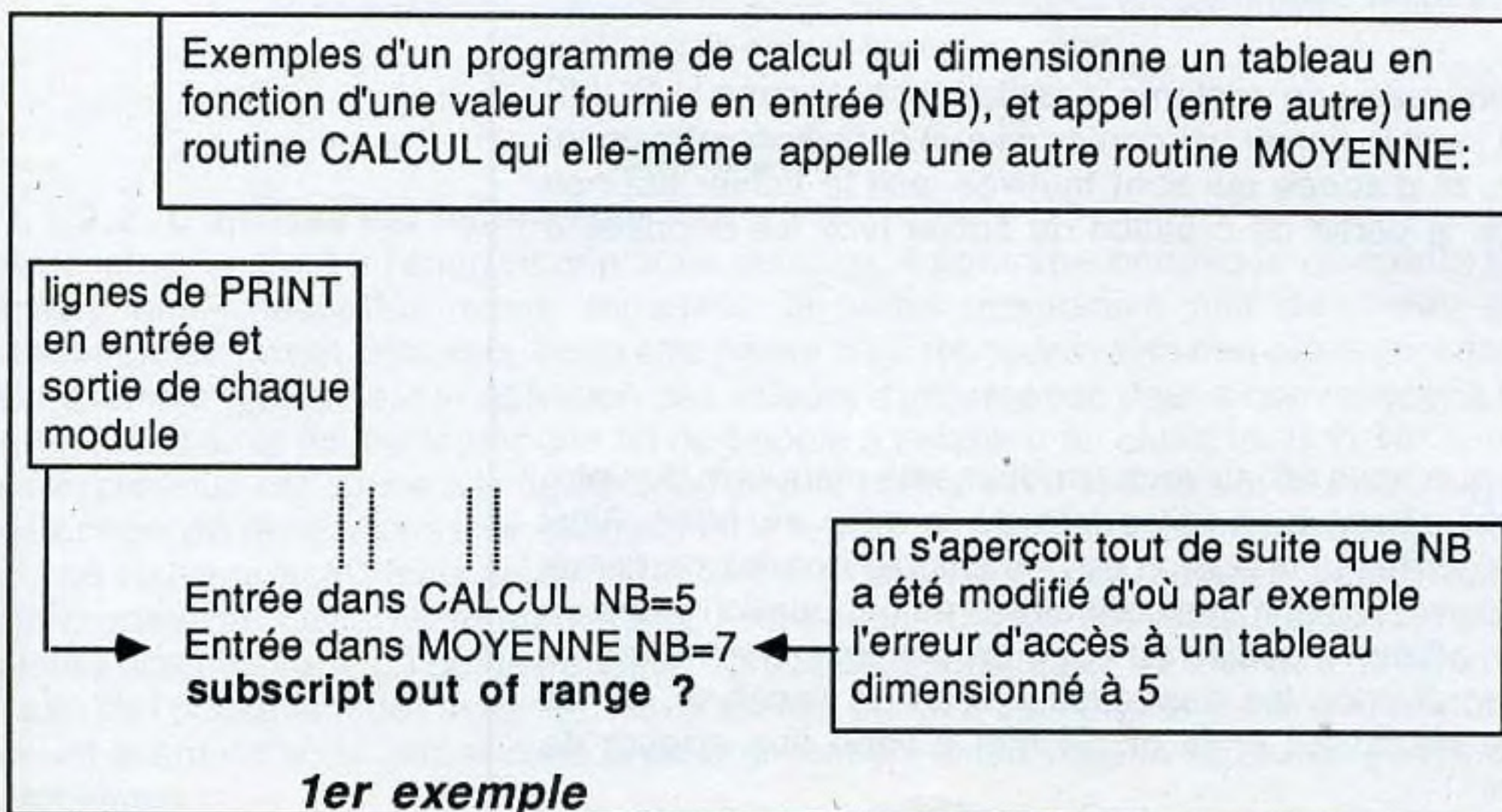
L'erreur est signalée par l'interpréteur BASIC :

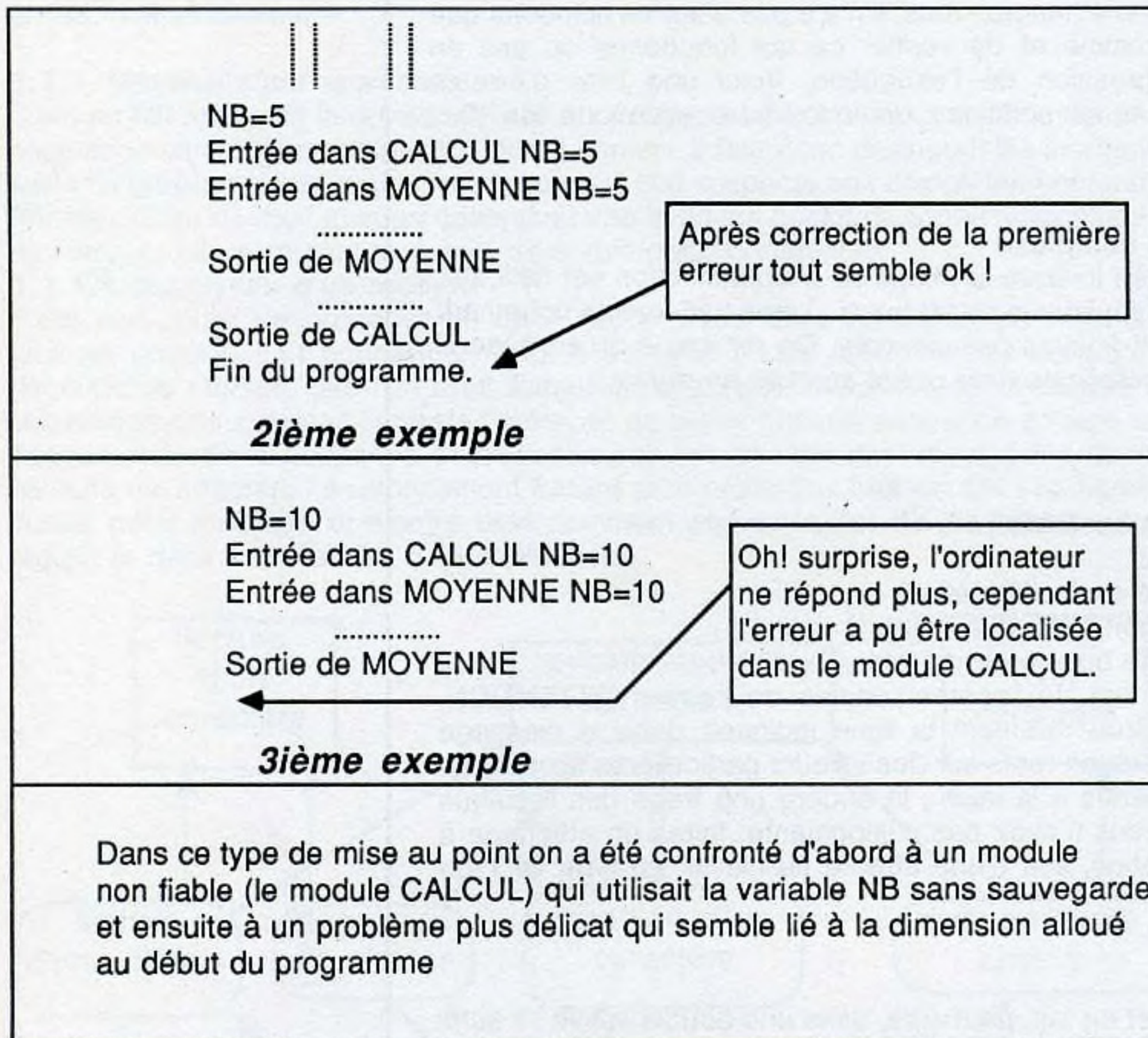
Par exemple : - division par zéro,
- dépassement de capacité (calcul),
- erreur de précision (arrondi à zéro),
- erreur d'indice de boucle ou de tableaux.

Une solution souvent applicable est de repérer l'origine de l'erreur (ATTENTION, malheureusement, ce n'est que trop rarement la ligne indiquée dans le message d'erreur à l'écran), d'effectuer quelques tests sur des valeurs particulières simples en comparant avec les résultats obtenus à la main ; là encore une trace des résultats intermédiaires peut être utile (si vous n'avez pas d'imprimante, faites un affichage à l'écran suivi, soit d'une temporisation, soit d'une attente au clavier du type GET ou INKEY\$).

1.1.2.1.2. «Plantage» complet

Si l'ordinateur semble bloqué, il est en fait, peut-être, dans une boucle infinie ; il suffit alors pour repérer les instructions responsables d'interrompre l'exécution pour connaître tout au moins le numéro de la ligne où le «BREAK» s'est produit, il suffit alors de procéder de façon similaire au cas a ; sinon (la tentative d'interruption a échouée) l'erreur est sans aucun doute plus grave (dépassement de capacité mémoire, accès à une zone réservée par un POKE malencontreux etc..). La signalisation, par exemple à l'aide de PRINT (cf figure 4), des entrées et sorties des sous-programmes doit déjà vous indiquer à quel moment dans la séquence de programme s'est produit l'erreur fatale.



**FIGURE 4****1.1.2.1.3. Résultats erronés**

Distinguons trois cas :

- erreur de calcul on procède alors comme en 1.2.1.1.1 ;
- erreur d'affichage (décalage, troncature de chaînes etc...) :
La méthode la plus simple est alors de progresser par tâtonnement, par essai-erreur, jusqu'à l'obtention du résultat souhaité.
- gestion de fichiers :
Utiliser les outils disponibles au niveau du système d'exploitation, comme le DUMP (cf. le numéro du mois dernier) ; soit le fichier est conforme aux définitions et ce sont les instructions de lecture ou, et d'accès qui sont fautives, soit le fichier est non conforme, et il faut alors revoir la partie de création du fichier (voir les données à introduire).

1.1.2.1.4. Remarques :

Comme dans d'autres situations que nous allons vous exposer, une maquette de votre programme apportera des simplifications bien utiles lors de la mise au point. Ainsi réduisez la taille de vos tableaux, limitez la précision de vos calculs simulez certaines opérations complexes, et vous pourrez, à l'abri des contraintes en occupation mémoire et des temps infinis de bouclage, affiner la qualité de votre programme. C'est avec ce genre de méthodes que l'on appréhende les capacités réelles de l'analyse, de sa programmation, mais aussi de la machine ; enfin on se met à l'abri des erreurs de « plantage » si difficiles à analyser.

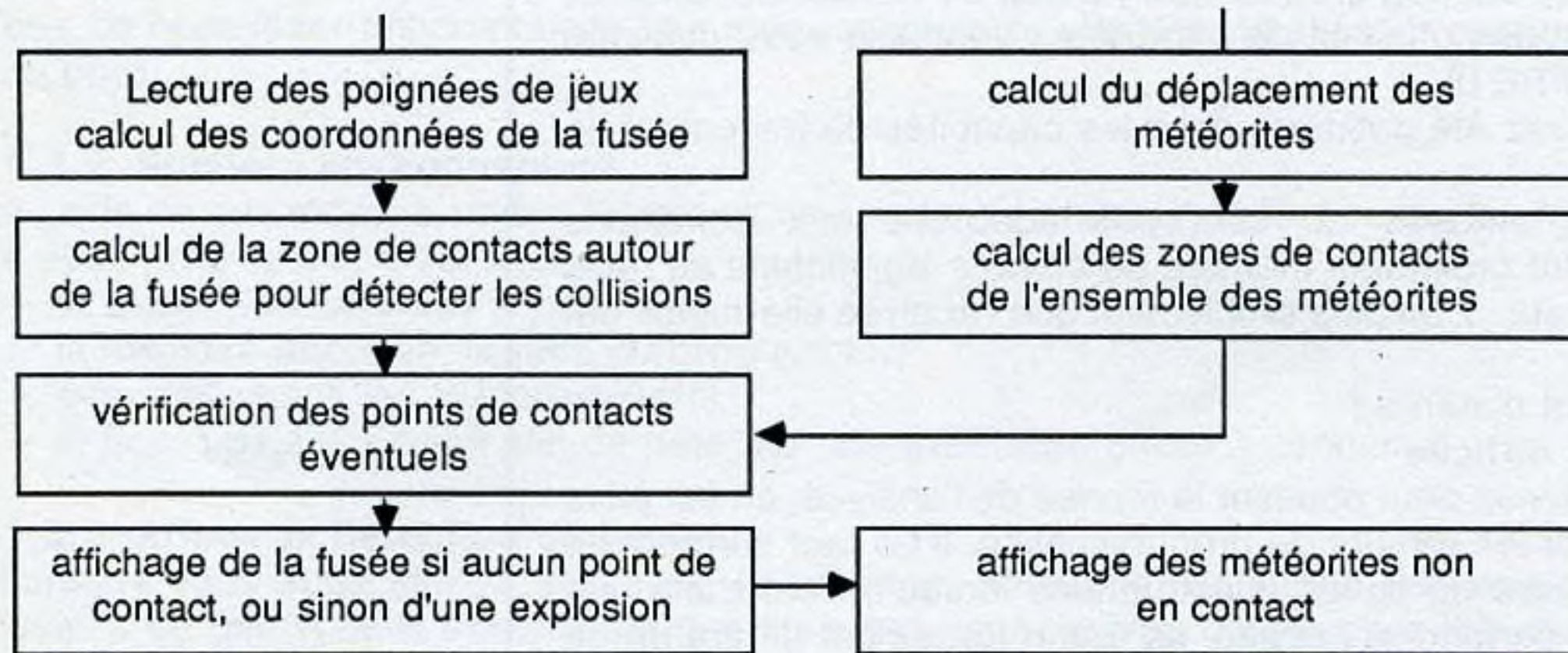
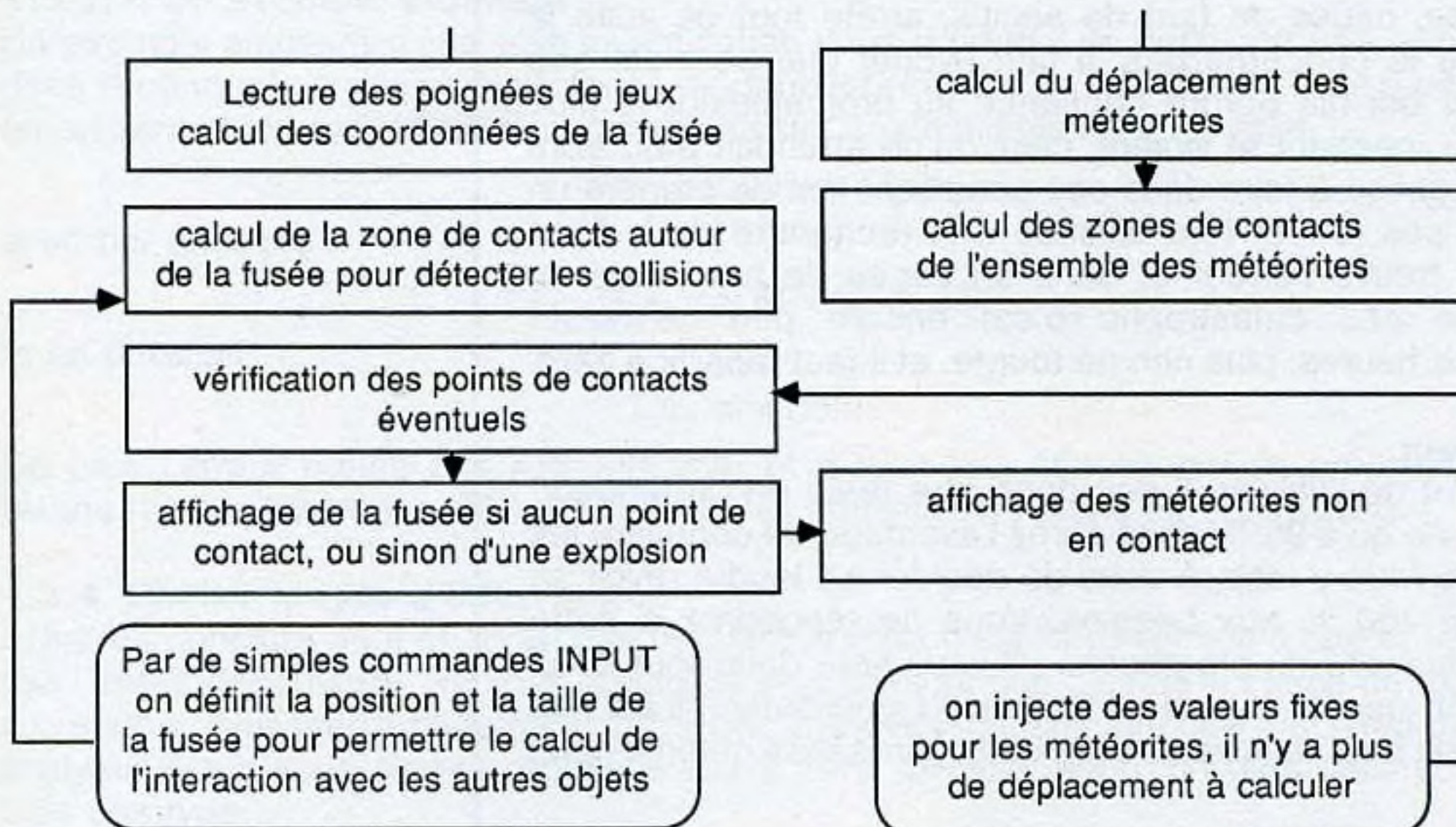


Diagramme d'une séquence d'opérations à effectuer dans un jeu de réflexes
(déplacement d'une fusée au milieu d'objets à éviter)



Pour tester les modules d'affichages et de détection des collisions
on a simplifié une partie du programme, cette partie pouvant être
non terminée ou pas encore au point

FIGURE 3

1.1.2.2. L'analyse est incomplète

A l'inverse, si c'est à l'analyse que vous souffrez, il peut être bénéfique de mettre en œuvre une «maquette» même imparfaite de votre programme afin de mettre en évidence les zones délicates, peut-être même d'en résoudre certaines par la pratique. Un exemple typique est la définition des valeurs d'initialisation (faut-il commencer à 0, à 1 ?) ou d'arrêt (faut-il tester une fin de boucle à l'étape n ou plutôt $(n-1)$?). Un autre exemple vous est donné à la figure 3, où un test partiel est effectué à la fois pour juger des choix de réalisations (par exemple ici le temps de réponse très important pour jeu qui se veut interactif), mais en outre faciliter la mise au point des modules d'affichages. Un programme devient vite un monstre avant même d'avoir été testé ; parce qu'on y ajoute une multitude de petites choses agréables à l'utilisateur... En un mot on vend la peau de l'ours avant de l'avoir tué. Aussi prenez garde à bien cadrer les limites de votre projet avant de vous lancer dans la programmation, sinon vous avez toutes sortes de problèmes :

- taille mémoire, si le programme est trop gros ou qu'il y a trop de variables (tableaux), ou encore si l'utilisation abusive des chaînes de caractères encrasse votre mémoire (souvenez-vous de la fonction FRE ()).

- temps de réponse, si vous avez été optimiste dans les capacités de traitement de votre merveille !

- précision si les méthodes utilisées ne sont pas adaptées aux conditions d'exploitation particulières à votre ordinateur (nombre de chiffres significatifs en réels taille maximale en entier ou réel etc...), ou plus simplement que l'analyse elle-même est en cause.

... Vous en connaissez sûrement d'autres !

1.1.2.3. La mise au point est difficile

L'incohérence des résultats obtenus peut pousser la reprise de l'analyse, on fait alors machine arrière afin de retrouver les erreurs de programmation. Il ne faut surtout pas hésiter à jeter plusieurs centaines de lignes d'instructions lorsqu'on est dans une impasse, ou appeler un copain étranger à l'affaire, sa vision forcément différente de votre prose lui permettra peut-être de trouver plus facilement que vous les points noirs.

1.2. Comment progresser dans la mise au point

Que celui qui n'est jamais resté des heures durant devant un listing apparemment clair à rechercher une erreur vicieuse, cause de tant de soucis, arrête tout de suite la lecture de ce chapitre ; cela ne le concerne pas. Il faut avouer que ce n'est que rarement une erreur de syntaxe qui fait perdre confiance au programmeur le plus acharné, mais bien le «plantage» méchant et violent, celui qu'on attendait pas, alors que tout allait bien. La première chose à faire dans ces conditions est de prendre un peu de recul, et surtout de ne pas foncer tête baissée à la recherche de la ligne fautive ; on croit souvent avoir trouvé l'erreur et dans un excès de confiance on corrige, relance le programme et... catastrophe c'est encore pire qu'avant ! Assurément, au bout de quelques heures, plus rien ne tourne, et il faut repartir à zéro.

1.2.1. Vérification en profondeur

Il faut être sûr d'un module avant de l'utiliser. Il doit donc être testé en profondeur. Cependant même s'il ne fonctionne qu'à 95 % vous aurez l'avantage de connaître les conditions limites et vous saurez vous y tenir. A vous de décider s'il faudra revoir sa conception afin qu'il réponde à 100 % aux besoins. Vous ne répondrez à cette question qu'à la mise au point du reste du programme ; il vous sera donc tout à fait possible de différer toute décision jusqu'à l'étape de finition, si l'amélioration n'est pas trop coûteuse en modifications, ou encore jusqu'à celle de l'optimisation comme nous le verrons plus tard.

1.2.2. Propagation module par module

La figure 2 illustre l'interaction entre modules, c'est-à-dire l'imbrication ou le parallélisme qui différencie les lignes de conduite à tenir pour faire évoluer la mise au point. Ainsi, dans ce cas précis, on conçoit aisément qu'il faut d'abord s'assurer à 100 % du bon fonctionnement des trois routines essentielles à traitement de texte, à savoir lire ou écrire un caractère, et se positionner dans l'écran. Sans ces trois routines, aucun des modules de niveau supérieur décrits sur ce schéma ne peut fonctionner. De même la fonction de recherche et remplacement d'une chaîne ne fonctionne que si 3 des 4 routines du niveau 2 (à savoir : lecture et écriture d'une chaîne, décodage du clavier) sont au point. Bien sûr, ce diagramme est très simpliste, nous vous laissons imaginer celui d'une application complète de traitement de texte !

1.3. Choix en cas d'erreur(s)

Nous voilà entrés dans le vif du sujet (enfin !). L'erreur est là qui nous nargue ; que faire ? Tout dépend bien évidemment de sa gravité. Mais avant d'aller plus loin, voici quelques conseils sur la façon de «combattre».

1.3.1. Utiliser les outils à votre disposition

Il est difficile de traiter les possibilités de chaque ordinateur mais sachez les utiliser au maximum, étudier particulièrement les documentations relatives aux erreurs de votre BASIC, mais aussi de votre système d'exploitation.

Les outils dont vous disposez iront des seules fonctions de TRACE (du BASIC), aux

riches «DEBUGGERS» symboliques qui à tout moment permettent d'effectuer du pas à pas, de consulter n'importe quelle valeur de variable etc... Alors faites avec les moyens du bord.

1.3.2. Sonder l'environnement

A l'aide de ces moyens, vous devez avoir une vision d'ensemble de tout ce qui touche à votre programme, c'est-à-dire :

- la valeur des variables importantes,
- la mémoire occupée, la taille du programme,
- éventuellement, les fichiers ouverts,
- le travail déjà accompli afin de détecter une éventuelle erreur antérieure.

1.3.3. Affiner la détection de l'erreur

Lorsque vous vous sentez maître de la situation, affinez la recherche de l'erreur, jusqu'à ce que vous la trouviez, ou que vous arriviez à un bloc d'instruction (boucle, test, instruction simple appel de fonction prédéfinie, calcul...). A partir de ce moment, et si l'erreur ne vous est pas évidente, il faut examiner en détail tout ce qui intervient dans cette zone ainsi délimitée.

1.3.3.1. Un exemple étonnant

Un exemple amusant d'une telle investigation nous a permis de découvrir une erreur dans la conception des multiplications en double précision en FORTRAN. Petit à petit en resserrant les mailles du filet, on aboutissait à une simple ligne du style

$$T(i) = T(i-1) * T(j),$$

avec par exemple :

$$T(i-1) = 1,0 \text{ et } T(j) = 2,0,$$

et on obtenait

$$T(i) = 4,0 !!!!$$

De plus l'erreur n'était pas reproductible et n'avait pas lieu en simple précision ; bizarre, bizarre (le bug a été corrigé avec les nouvelles versions du compilateur).

1.3.4. Choisir le type d'intervention

Avant de conclure pour ce mois, une dernière précision sur l'intervention à opérer une fois l'erreur découverte. Ne vous précipitez pas pensant que c'est là la cause de tous vos soucis, mais bien au contraire, continuez vos investigations tant que tout n'est pas analysé. Sinon vous devrez recommencer à zéro. Grosso modo, on peut distinguer trois cas type :

- l'erreur de syntaxe, à moins qu'elle ne vous paraisse incompréhensible, vous relancerez la machine après correction, sans trop de souci à se faire ;
- l'erreur délicate, au niveau d'une structure (boucle, sous-programme), le mieux est alors de reprendre parallèlement l'analyse et le test de la zone incriminée avec des méthodes comme celles décrites dans le numéro de juin, en examinant en détail les variables, les branchements, etc.
- l'erreur tenace, le résultat obtenu après plusieurs corrections n'est toujours pas satisfaisant, c'est à ce moment qu'il faut repartir dans la phase d'analyse, en espérant qu'elle ne remettra pas en cause trop de parties (voire l'ensemble) du programme. Attention, ne soyez cependant pas trop exigeants si vous êtes inexpérimentés dans le domaine d'application de votre projet : il sera toujours possible de réaliser un autre logiciel plus performant, après l'expérience acquise sur la ou les premières versions.

CONCLUSION

Nous en avons terminé pour ce mois, il nous reste à aborder les ultimes phases de finition et d'optimisation, et nous en profiterons pour développer plus en détails la façon de procéder à l'aide de deux exemples classiques : un traitement de texte ainsi qu'un jeu interactif, vous découvrirez ainsi ce que plus pratiquement un programmeur «amateur» est capable de réaliser sans développer des trésors d'ingéniosité. Si vous avez d'autres choix à nous soumettre, n'hésitez pas à nous écrire.

DIALOGUE AVEC NOS LECTEURS

1. CORRECTION DU PROGRAMME DE MORSE

Nous avons reçu plusieurs réponses ; nous en détacherons deux qui présentent aussi bien des analogies que des différences. La plus remarquable des ressemblances reste que les deux auteurs habitent la même charmante bourgade de Saint-Sauveur-le-Vicomte. La première est de M. Truffert et la seconde de M. Lepetit.

Dans la première solution, son auteur a choisi de séparer la phase de constitution du dictionnaire et celle de son emploi. La méthode a du bon puisqu'elle évite à chaque usage de refaire un travail déjà fait et autorise au chargement les fautes de frappe puisqu'une touche de correction est prévue. L'attitude choisie par son compatriote est de tout saisir dans des DATAs insérés à la fin du programme, cela évite un fichier de données à conserver séparément en mémoire mais cela augmente la taille du programme et la saisie des données doit être bonne du premier coup. Hormis cela, les deux programmes emploient le même type de raisonnement, à savoir : un tableau de caractères conserve la phrase (en français ou en morse) à traduire et la change en utilisant les valeurs ASCII des lettres allant de 65 (a) à 90 (z). Passons donc aux programmes eux-mêmes.

```
10 PRINT"CONSTITUTION DU DICTIONNAIRE MORSE-ALPHABET FRANCAIS";: GOSUB 350
20 DIM D$(26)*5
30 PRINT "ENTREZ LES ELEMENTS DU DICTIONNAIRE COMME SUIT : .- A; SOIT MORSE LETTRE" ;
40 GOSUB 350
50 PRINT"POUR CORRIGER TAPEZ * , AVANCEZ , CORRIGEZ ." : GOSUB 350
60 FOR I = 0 TO 25
70 PRINT I;: INPUT ")" , D$(I)
80 IF D$(I) = "*" THEN I = I - 2
90 REM LIGNE 80 PERMET DE CORRIGER EN REVENANT EN ARRIERE.
100 NEXT I
110 PRINT "LECTURE DU DICTIONNAIRE. POUR CORRIGER TAPEZ LE N";CHR$(223);" DE ' '";
120 FOR I = 0 TO 300 : NEXT I
130 PRINT "L";CHR$(96);"ERREUR AVANT DE CHANGER DE LETTRE , AVANCEZ , CORRIGEZ " ;
140 GOSUB 350 : PRINT
150 FOR I = 0 TO 25 : PRINT I;"); D$(I);: GOSUB 350
160 IF H$ <> " " THEN PRINT "D$("; VAL(H$); : INPUT ")" ,D$ ( VAL (H$) )
170 NEXT I : END

350 INPUT " ",H$: RETURN
```


Le premier programme est une simple boucle où l'utilisateur entre successivement le signe morse puis le caractère, boucle comportant au milieu un test de correction du caractère précédent si la nouvelle chaîne de caractères commence par une "*", cela signifie une faute de frappe sur la lettre immédiatement antérieure, il faut donc enlever 2 à I car le nouveau passage à next i va rajouter 1. Seul écueil, il ne faut pas faire de faute sur le z puisqu'on ne saisit rien après.

```

10 REM CHOIX DU PROGRAMME DE TRADUCTION D' APRES LA PREMIERE LETTRE DU MESSAGE
20 CLS : DIM F$(1)*79 , M$(160)*5
30 REM FRANCAIS = F$ , MORSE = M$ , DICTIONNAIRE = D$
40 PRINT " ENTREZ LA PREMIERE LETTRE OU LE PREMIER SIGNE DU MESSAGE "
50 INPUT P$ : P$ = LEFT$(P$ , 1)
60 REM CECI AU CAS OU PLUSIEURS SIGNES AURAIENT ETE ENTRES
70 IF P$ = "-" THEN 400 ELSE IF P$ = "." THEN 400 ELSE 100
100 REM TRADUCTION FRANCAIS EN MORSE
110 PRINT " ENTREZ VOTRE MESSAGE FRANCAIS , JUSQU' A DEUX
130 GOSUB 300 : CLS
140 INPUT F$(0) , F$(1)
150 PRINT "TRADUCTION"
160 FOR N = 0 TO 1
170 A = LEN ( F$(N) ) : IF A = 0 THEN 240
180 FOR I = 1 TO A
190 IF MID$ ( F$(N) , I , 1 ) = " " THEN PRINT " ";CHR$(150) ;: GOTO 230
200 M$(I) = D$(ASC(MID$(F$(N),I,1)) - 65)
210 M$(I) = LEFT$( M$(I), LEN(M$(I)) - 1
220 PRINT M$(I); CHR$(150);
230 NEXT I
240 NEXT N: INPUT " ",H$: END

300 FOR I = 0 TO 300 : NEXT I
310 RETURN
320 FOR I = 0 TO 120 : NEXT I : RETURN

400 REM TRADUCTION DU MORSE EN FRANCAIS
410 CLS : PRINT " ENTREZ VOTRE MESSAGE MORSE , LETTRE PAR LETTRE ":GOSUB 300
420 PRINT " ENTRE CHAQUE MOT TAPPEZ ' '": GOSUB 300
430 PRINT " POUR CORRIGER , ENTREZ * , AVANCEZ , CORRIGEZ " : GOSUB 300
440 PRINT " EN FIN DE MESSAGE , AVANCEZ DEUX FOIS " : GOSUB 300 : CLS
450 FOR I = 0 TO 160
460 INPUT M$(I)
470 IF M$(I) = "*" THEN I = I - 2
480 IF M$(I) = " " THEN A = I : I = 160
490 NEXT I : CLS
500 PRINT " TRADUCTION " : GOSUB 320
510 FOR J = 0 TO A
520 V$ = M$(J)
530 FOR I = 0 TO 25
540 IF V$ = " " THEN PRINT " " ;: GOTO 580
550 IF V$ = LEFT$(D$(I),LEN(D$(I))-1) THEN PRINT RIGHT$(D$(I),1);: GOTO 580
560 IF V$ = " " THEN 580
570 NEXT I
580 NEXT J : INPUT " ",H$: END
224 FOR J = 1 TO LEN (M$(I))
225 IF MID$(M$(I),J,1) = "." THEN BEEP 1 ELSE BEEP 0
226 NEXT J

```

Le second programme effectue la traduction, il interprète la signification du premier caractère puis, selon le résultat, identifie le texte comme étant du morse ou du français (ligne 70). Il boucle sur la i-ème lettre du texte français

(mid\$(f\$(n), i, 1),

en prend le code ASCII, enlève 65 pour obtenir le rang dans le tableau des traductions et prend l'élément désigné du tableau (ligne 200).

La traduction morse-français se fait en deux étapes, il morcelle le message en une suite de séries de signes puis essaie chaque élément de la suite avec chacune des séries correspondant à une lettre (ligne 550), ce programme est suffisamment commenté mais quelquefois à contre-temps. Les lignes 224 à 226 peuvent être

ajoutées par qui sait reconnaître le morse à l'oreille.

```

10 PRINT "TRADUCTION LETTRE-MORSE"
20 DIM MO$(26,2),ST$(150)
30 FOR I = 1 TO 26
40 FOR J = 1 TO 2
50 READ MO$(I,J)
60 NEXT J,I
70 GOTO 600
100 REM TEXTE LITTERAL
110 CLS: INPUT "TEXTE :";TL$
120 N = LEN (TL$) + 1
130 IF N < 2 THEN GOTO 110
140 FOR I = 2 TO N
150 ST$(I) = MID$(TL$,I - 1,1)
160 NEXT I
170 FOR I = 2 TO N
180 C = ASC (ST$(I))
190 IF (C > 96) AND (C < 123) THEN C = C - 32
195 IF C < 65 OR C > 90 THEN C = 64
200 C = C - 64
210 ST$(I - 1) = MO$(C,2):ST$(I) = ""
220 NEXT I
250 FOR I = 1 TO N - 1
260 M = LEN (ST$(I))
270 IF M = 0 THEN PLAY"R3": GOTO 330
280 FOR K = 1 TO M
290 E$ = MID$(ST$(I),K,1)
300 X = ASC (E$) - 45:Y = 20 * X + 5
310 PLAY"T250L=Y:06AR10"
320 NEXT K:PLAY"R5"
330 NEXT I
340 RETURN
400 REM TEXTE EN MORSE
410 CLS:I = 1:ST$(1) = ""
420 I = I + 1
430 INPUT ST$(I)
440 IF ST$(I) < > "FIN" THEN 420
450 N = I - 1
460 FOR I = 1 TO N
470 IF ST$(I) = "" THEN ST$(I - 1) = CHR$(32)
480 FOR J = 1 TO 26
490 IF MO$(J,2) < > ST$(I) THEN 510
500 ST$(I - 1) = MO$(J,1)
510 NEXT J
520 NEXT I
530 FOR I = 0 TO N - 1
540 PRINT ST$(I);
550 NEXT I: PRINT : PRINT
560 RETURN
600 PRINT : PRINT TAB(6);"MENU": PRINT
610 PRINT " TEXTE EN LETTRES....1"
620 PRINT " TEXTE EN MORSE.....2"
630 PRINT : INPUT "VOTRE CHOIX:";E
640 ON E GOSUB 100,400
650 PRINT : PRINT "FIN DE TRADUCTION ?"
660 INPUT "OUI OU NON";CH$
670 IF CH$ = "OUI" THEN END
680 GOTO 600
700 DATA A,-,B,-.,C,-.-.,D,-...E,..F,..-,G,---.,H,.....I
,.,J,----,K,-.-.,L,-...M,---,N,-.,O,----,P,---.,Q,---.,R,-.,
S,....,T,-,U,..-,V,....,W,---,X,-.-.,Y,-.--,Z,---.

```


M. Lepetit a nettement moins soigné les commentaires mais a beaucoup plus aéré son texte, la méthode employée est analogue si l'on excepte l'absence de correction d'erreurs à laquelle ce programme substitue une interprétation systématique de tous les caractères, si celle-ci est défailante il passe outre n'imprimant rien, mais interprète majuscule et minuscule. Les lignes 250 à 340 fabriquent le fond sonore. La traduction est légèrement différente car la machine est différente. Enfin, le programme principal est situé à la fin comme en Pascal s'il ne fait pas l'interprétation lui-même.

2. REPONSES AU COURRIER DE JUIN

M. Ara de Mourenx nous a envoyé trois programmes fort détaillés qui montrent que leur auteur n'a pas immédiatement succombé au besoin de saisir son programme et a précédé la rédaction des instructions d'une analyse qui a rendu son programme très agréable à lire et facile à comprendre, vous commencez à connaître notre amour de la simplicité.

Ces beaux programmes sont de philosophies diverses, deux d'entre eux visent à faire un logiciel de traduction entre l'anglais et le français, entreprise qui n'est pas sans utilité dans le domaine de l'informatique, mais dont l'emploi sur un micro-ordinateur Amstrad PCW 8256, nous semble frustrant dans la mesure où la taille mémoire est limitée. M. Ara a réussi l'exploit de stocker sur une disquette de 175 Ko 3 700 mots et nous sommes admiratifs devant le temps qu'il a dû passer face à son écran pour entrer tout cela sans erreur. Chapeau bas. Mais nous préférons vous proposer, pour rester dans la note britannique de concevoir plutôt un jeu vous demandant les différentes formes des verbes irréguliers de notre jeunesse, le plus mauvais de nous trois ne se souvient que de cut, cut, cut. C'est vous dire si le programme lui sera utile !

Quant à son troisième programme, c'est une résolution des équations du second degré lorsque les racines sont réelles, les racines étant alors exprimées sous leur forme exacte.

EQUATIONS DU SECOND DEGRE

$$ax^2 + bx + c = 0$$

$$a=2 \quad b=-9 \quad c=3$$

$$\text{delta} = (b^2 - 4ac) \geq 0$$

$$\dots\dots\dots \text{delta} = 81 - 4 \times 2 \times 3 = 57$$

deux racines

$$x' = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x'' = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

$$x' = -(-9) + (\text{Rc } 57) / 2 \cdot 2 = (x') = 4.14$$

$$x'' = -(-9) - (\text{Rc } 57) / 2 \cdot 2 = (x'') = 0.36$$

$$ax^2 + bx + c = 0$$

$$a=1 \quad b=-5 \quad c=6.25$$

$$\text{delta} = (b^2 - 4ac) = 0$$

$$\dots\dots\dots \text{delta} = 25 - 4 \times 1 \times 6.25 = 57$$

$$\text{l'équation s'écrit } \dots \left(x + \frac{b}{2a}\right)^2 = 0$$

$$\text{ou } x + \frac{b}{2a} = 0$$

$$\text{on dit que l'on a une racine double : } x = -\frac{b}{2a}$$

$$x = -(-5) / 2 \times 1 = (x) = 2.5$$

EQUATIONS DU SECOND DEGRE

$$ax^2 + bx + c = 0$$

$$a = 7 \quad b = 2 \quad c = 5$$

$$\Delta = b^2 - 4ac < 0 \quad \dots \quad \Delta = (4 - 4 \cdot 7 \cdot 5) = -136$$

On ne peut extraire la racine carrée de $(b^2 - 4ac)$

L'équation n'a pas de racines

IMPOSSIBILITE

Mais nous avons trouvé que vous n'êtes pas allé au bout de vos intentions en ne présentant pas la solution alors que vous le faites juste au-dessus. De plus, vous ne cherchez pas à réduire le discriminant si par exemple Δ est égal à 8, la solution exacte s'exprime généralement en

$$X = -b + 2\sqrt{2} \quad \text{plutôt que} \quad X = -b + \sqrt{8}$$

Ne croyez pas que nous faisons la fine bouche, cela nous est plutôt dicté par la beauté des résultats qui nous semblent si peu éloignés du meilleur que nous pensons qu'il est à votre portée. Et puis, si nous ne vous faisons des réserves, que vous resterait-il à faire ? Le challenge est ouvert si d'autres se sentent intéressés par ces détails, nous pensons qu'il est toujours possible de mieux faire. Sinon, il demeure le troisième degré qui, là, n'est accessible qu'avec un honnête bagage mathématique, nous rappellerons donc que l'équation

$$ax^3 + bx^2 + cx + d = 0$$

peut se mettre sous la forme $X^3 + pX + q = 0$ avec $X = x - p/3$, l'équation originelle étant préalablement divisée par a .

Ce qui donne :

$$x^3 + Bx^2 + Cx + D = 0$$

$$p = 1/3 * (3 * C - B * B)$$

$$\text{et } q = 1/27 * (2 * B * B * B - 9 * B * C + 27 * D)$$

nous avons alors le discriminant qui vaut :

$$\frac{p^3}{27} + \frac{q^2}{4}$$

Si le discriminant est négatif, les trois solutions sont réelles. S'il est nul, il y a une racine double réelle. Dans le cas où il est positif, deux solutions sont des complexes conjuguées. Nous employons là des notions de mathématiques abordées en fin de secondaire mais si vous avez quelques difficultés pour les comprendre, il existe de très bons cours de classes terminales où vous trouverez tous les éclaircissements désirés.

Pour obtenir les solutions, il convient de calculer les deux valeurs suivantes :

$$X_1 = \sqrt[3]{-b/2 + \sqrt{\Delta}} \quad X_2 = \sqrt[3]{-b/2 - \sqrt{\Delta}}$$

A partir de là, on obtient :

$$x = \frac{X_1 + X_2}{2} \pm \frac{X_1 - X_2}{2} * \sqrt{-3}$$

ce qui nous fait bien trois solutions, cette méthode a été découverte par un Italien de la Renaissance du nom de Cardano (1501-1576) qui ne savait pas exprimer la racine carrée de -3 et donc obtenait la solution sans savoir prouver qu'elle était bonne. Il fallut attendre les nombres en $A + iB$ pour le montrer. Evariste Galois (1811-1832, enseigna à Polytechnique moins de deux ans après avoir été «collé» au concours ! Est mort en duel après avoir passé la nuit à mettre au point la théorie des groupes qui porte son nom) a démontré que la recherche de la solution passe nécessairement par des calculs complexes même si les trois solutions sont réelles («casus irreductibilis»). Pour plus de précisions, nous vous conseillons de vous adresser à votre mathématicien habituel, de préférence spécialisé en algèbre.

3. COMMENT PRATIQUER LE PASCAL ?

Cela fait maintenant trois numéros que vous pouvez suivre le cours d'initiation au langage PASCAL. Il est donc temps de passer à la pratique autrement que sur une feuille de papier ; aussi, et sur la demande de nombreux lecteurs (pardon à M. Pierre Lamy de ne pas avoir répondu plus tôt), nous sommes-nous décidés à vous parler des implantations du langage PASCAL sur les ordinateurs les plus répandus. Le PASCAL est un langage standardisé, toute implémentation sur un ordinateur suit donc ce standard. Malheureusement, ce dernier est incomplet pour permettre au PASCAL d'être, sans extensions, un langage de développement. C'est pourquoi les premières versions commercialisées eurent pour origine l'adaptation faite par l'université de San Diego, nommée PASCAL UCSD (U.C.S.D. pour University of California San Diego). En plus des nombreuses extensions (vous en avez eu un aperçu ce mois-ci), elle est fondée sur l'utilisation d'un pseudo-code (Pcode), associé à une machine virtuelle (Pmachine), le tout géré sous un système d'exploitation particulier (le Psystem) : les programmes sources sont compilés pour obtenir du Pcode. La réalisation de l'adaptation d'une machine à une autre se réduit donc à la façon de traduire ce code intermédiaire beaucoup plus simple que les instructions du langage. Heureusement, depuis quelques années, il existe plusieurs versions sur la plupart des micro-ordinateurs à vocations professionnelles ou semi-professionnelles disposant elles aussi d'extensions. En revanche, l'ordinateur familial ne dispose le plus souvent que de versions limitées et non standard : néanmoins elles suffisent à l'apprentissage du langage, au moins au début.

Le tableau ne prétend pas, loin de là, recenser toutes les, désormais, nombreuses versions disponibles sur le marché. Ainsi, il nous est difficile, dès à présent, de vous préciser les caractéristiques complètes des versions existant sur les Thomson T09, T07, M05 ou les ordinateurs au standard MSX ou encore les Oric, Commodore, etc. Par contre, n'hésitez pas à nous écrire, si vous disposez d'une de ces versions, vos commentaires intéresseront sans aucun doute de nombreux lecteurs. De même, nous n'avons pas cité l'ensemble des versions disponibles sur IBM et compatibles, ni sur Macintosh ou Apple II. Le plus important est de choisir le meilleur rapport qualité/prix, surtout que les tarifs ne sont généralement pas à la portée de toutes les bourses. C'est pourquoi le Turbo-Pascal a si rapidement conquis le marché car, en plus de ses performances et de sa facilité d'utilisation, il est parmi les moins chers. Il a en plus le mérite d'exister sur de nombreuses machines et de reprendre une partie des extensions du Pascal UCSD ; avantage donc au Turbo-Pascal.

Dans les prochains numéros, nous essaierons de compléter cette liste afin surtout de vous éviter de vous lancer dans un investissement important pour un produit pas toujours performant. En attendant, gardez votre stylo et votre feuille de papier, et... économisez.

LANGAGE PASCAL	ORDINATEURS	AVANTAGES	INCONVENIENTS
TURBO PASCAL (Borland) actuellement la star du marché.	1) IBM et compatibles 2) CP/M (APPLE 2) 3) AMSTRAD	- bon marché (600 à 1000 F) - performant (en compilation et en exécution) - graphiques sur IBM et AMSTRAD - facile à utiliser - dispose de nombreux outils en sus.	- non standard pour certaines fonctions. - pas de librairies compilées ni d'édition de liens (link).
PASCAL UCSD plus qu'un langage, un envi- ronnement complet (gestion de fichiers, éditeur, debugger...).	1) APPLE (Version 2) 2) MACINTOSH (Version 4) 3) IBM et compatibles	- la référence de tous les ou- vrages (et de notre cours). - un système d'exploitation commun à toutes les versions - possibilités graphiques - outil de développement complet (librairies, modules).	- onéreux (2000 à 4000 F) - compilation lente - code semi- compilé (PCODE) d'où un temps d'exécution moyen.
MICROSOFT PASCAL le premier disponible sur IBM, et le standard IBM.	IBM et compatibles, environ- nement MS-DOS.	- complet et doté de nomb- reuses extensions. - compatible avec tous les langages de MICROSOFT. - plus rapide que la version UCSD	- aussi peu abordable que la version UCSD. - moins performant que le TURBO PASCAL.

SEPTEMBRE 1986
Vient de paraître



● INITIATION A LA VIDÉO LÉGÈRE

(THÉORIE ET PRATIQUE)

Claude Gendre.

– Choix d'un standard ? – Caméscopes VHS, VHS-C ou 8 mm ? – Connexion ? Compatibilité ? – Accessoires ? Montage ? Enfin... Comment filmer.

Le nouveau livre de Claude Gendre répond à toutes ces questions, cet ouvrage essentiellement pratique qui n'a pas d'équivalent en librairie aujourd'hui s'adresse (sans formules mathématiques) à tous ceux passionnés (déjà ou à venir) de vidéo et aux amateurs de belles images.

Des illustrations en couleur donnent une excellente idée des possibilités de «filmage» et de montage.

L'avenir du cinéma d'amateurs et celle de la création par l'image passera par la vidéo légère...

Ce livre devenait urgent.

● LE 3^e TOME de INITIATION A LA MICRO-INFORMATIQUE

(COURS 1^{er} CYCLE)

Claude Polgar. (Enfin paru !)

Non, on ne s'initie pas à la micro-informatique en 5 leçons !

Si vous croyez au Père Noël vous pouvez espérer apprendre l'Informatique en lisant les innombrables «Cours de BASIC pour débutants» qui ont poussé comme des champignons dans les années 1980. Votre ordinateur risque de finir ses jours au-dessus de votre armoire.

Mais si vous voulez vraiment apprendre à programmer il faut avoir le courage de commencer par A pour arriver à Z. Programmer est un loisir intelligent et peut devenir un métier passionnant, mais l'étude de la programmation nécessite un minimum de travail et de méthode.

Etre sérieux – c'est le pari que fit la revue LED-MICRO en publiant à partir de 1985 les 20 premiers cours de C. Polgar. Plus de 40 000 lecteurs les ont suivis. Ce succès nous a conduit à demander à C. Polgar de remettre son cours à jour et de le compléter. Le résultat : un ouvrage épais (3 tomes, plus de 700 pages format 21 x 27), permettant d'acquérir agréablement des connaissances solides.

● INITIATION A L'ÉLECTRICITÉ ET A L'ÉLECTROTECHNIQUE

Roger Friedérich.

Vous trouverez aisément en librairie des ouvrages d'initiation à l'électronique ou aux techniques les plus avancées des circuits intégrés, etc. Mais si vous désirez une initiation aux bases de l'électricité et de l'électrotechnique sans vous en remettre à des ouvrages scolaires, alors vous ne trouverez pas ! Nous avons demandé à un spécialiste de ces disciplines de tenter d'expliquer de la manière la plus claire tout ce qui se rapporte à l'électricité et ses applications ainsi qu'à l'électrotechnique. Il a réussi et nous sommes certains que dans ce domaine il fallait oser recommencer par la loi d'Ohm et répondre à la question : Comment ça marche ?

Chaque mois, les nouveautés seront signalées.

VOIR AU DOS NOTRE COLLECTION COMPLÈTE AINSI QUE LES PRIX DE CHAQUE
OUVRAGE ET SES CARACTÉRISTIQUES

Diffusion auprès des libraires assurée exclusivement par les Editions Eyrolles.

Bon de commande à retourner aux Editions Fréquences 1, boulevard Ney 75018 Paris.

Je désire recevoir le(s) ouvrage(s) ci-dessous référencé(s) que je coche d'une croix :

E 01 <input type="checkbox"/>	E 02 <input type="checkbox"/>	E 03 <input type="checkbox"/>	E 04 <input type="checkbox"/>	E 05 <input type="checkbox"/>	E 06 <input type="checkbox"/>	L 07 <input type="checkbox"/>	P 08 <input type="checkbox"/>	L 09 <input type="checkbox"/>	L 10 <input type="checkbox"/>
L 11 <input type="checkbox"/>	E 12 <input type="checkbox"/>	E 13 <input type="checkbox"/>	L 14 <input type="checkbox"/>	E 15 <input type="checkbox"/>	P 16 <input type="checkbox"/>	P 17 <input type="checkbox"/>	P 18 <input type="checkbox"/>	P 19 <input type="checkbox"/>	L 20 <input type="checkbox"/>
P 21 <input type="checkbox"/>	E 22 <input type="checkbox"/>	P 23 <input type="checkbox"/>	P 24 <input type="checkbox"/>	E 25 <input type="checkbox"/>	P 26 <input type="checkbox"/>	P 27 <input type="checkbox"/>	P 28 <input type="checkbox"/>	P 29 <input type="checkbox"/>	

Frais de port : + 10 F par livre commandé, soit la somme totale ci-jointe, de Frs par CCP Chèque bancaire Mandat-lettre

Nom..... Prénom.....

Adresse.....

Ville..... Code postal.....



BIBLIOTHEQUE TECHNIQUE

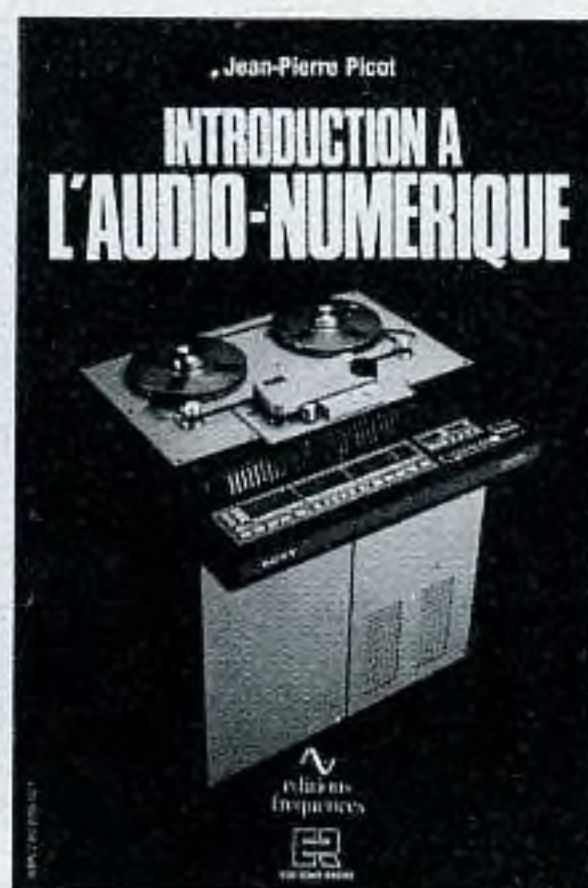
Collection études (format 165 x 240)



E 15. 184 p. Prix : 140 F TTC
Face au développement spectaculaire des synthétiseurs, grâce à l'électronique numérique, le besoin d'un ouvrage complet, accessible, et surtout bien informé des dernières ou futures techniques, se faisait sentir. Le vœu est comblé, en 180 pages... à dévorer.



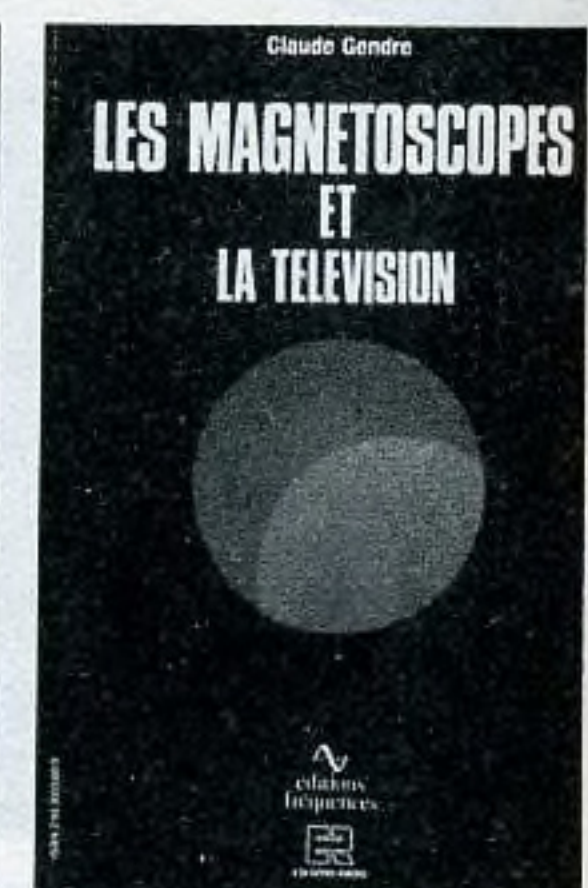
E 01. 320 p. Prix : 165 F TTC
Un gros volume qui connaît un succès constant : bien plus qu'un traité, il s'agit d'une véritable encyclopédie, alliant théorie et pratique, histoire, en une mine inépuisable d'informations, reconnue dans le monde entier !



E 05. 160 p. Prix : 155 F TTC
C'est le premier ouvrage paru en langue française traitant de l'audio numérique : écrit par un professionnel, avec rigueur, simplicité, il explique brillamment les bases de cette technique : quantification, conversion, formats, codes d'erreurs.



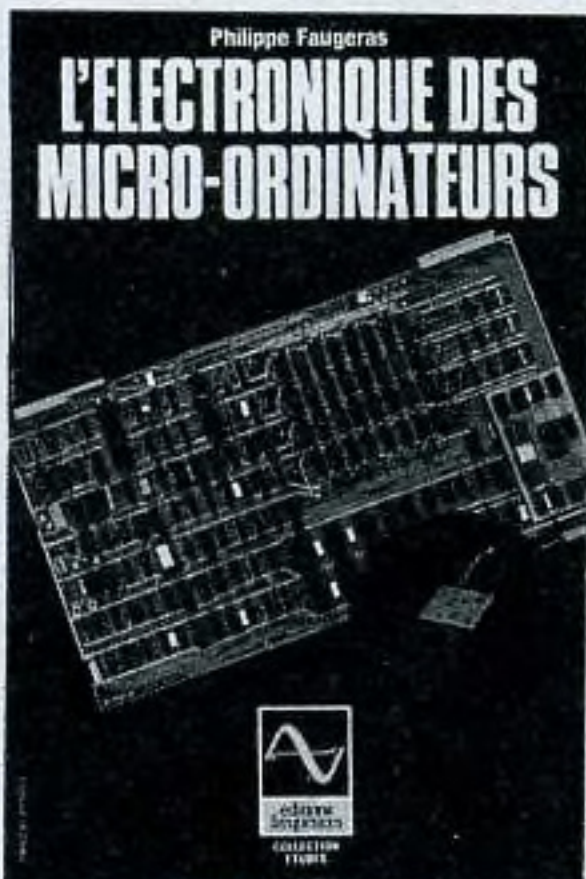
E 04. 240 p. Prix : 154 F TTC
Seconde édition passionnée d'un ouvrage fort attendu des amateurs d'électroacoustique. Ce livre permet aux amateurs et aux professionnels de se familiariser avec les rigoureuses techniques de modélisation des haut-parleurs et enceintes acoustiques et d'en mener à bien la réalisation.



E 03. 256 p. Prix : 155 F TTC
Complément direct des «Magnétophones», les «Magnétoscopes et la Télévision» débute par un bel historique de la télévision et la description des premiers magnétoscopes. La théorie et la pratique de la capture et de l'enregistrement moderne des images vidéo en sont la teneur essentielle.



E 22. 136 p. Prix : 150 F TTC
Faisant suite à la parution de «L'électronique des micro-ordinateurs», cet ouvrage s'adresse aux électroniciens qui désirent s'initier aux montages périphériques des micro-ordinateurs, interfaces en particulier, qui permettent la communication avec le monde extérieur.



E 06. 128 p. Prix : 150 F TTC
Cet ouvrage est destiné aux électroniciens désireux d'aborder l'étude du «hard» des micro-ordinateurs. Cette étude s'articule autour du microprocesseur Z-80, très répandu, et en décrit les éléments périphériques : mémoires, clavier, écran, interfaces de toutes sortes.



E 02. 160 p. Prix : 92 F TTC
Pour tout savoir sur le magnétophone, depuis l'avènement de cette mémoire des temps modernes, jusqu'aux enregistreurs numériques en passant par la cassette. «Les magnétophones» est un ouvrage pratique, complet, indispensable à l'amateur d'enregistrement magnétique.



E 12. 256 pages. Prix : 155 F TTC
E 13. 256 pages. Prix : 165 F TTC
Introuvable aujourd'hui une sélection des meilleurs articles de la célèbre revue «L'Audiophile». Le tome 1 traite de l'électronique audio à tubes, et le tome 2. Dans un esprit identique, le tome 2 traite du domaine passionnant que constituent les transducteurs en audio.



E 25. 160 pages. Prix : 140 F TTC
Le monde de l'audio évolue... Un secteur d'activité entièrement neuf vient d'apparaître : les Mini-Studios. L'ouvrage de Denis Fortier, ingénieur du son, aborde le sujet de la manière la plus globale. Après les données indispensables, le choix des maillons, la manière d'installer et d'exploiter.

Collection loisirs (format 135 x 210)



L 07. 160 p. Prix : 68 F TTC
Le «dernier coup de patte» apporté à un lecteur, celui qui fait la différence entre la réalisation approximative et le kit bien fini, ce savoir-faire s'acquiert au fil des ans... ou en parcourant «Conseils et tours de main en électronique».



L 10. 200 p. Prix : 130 F TTC
Tout beau, tout nouveau, le lecteur laser. Qu'en est-il réellement ? Pour en savoir plus, un livre traitant du sujet s'imposait. «Les lecteurs de compact-discs» permet de faire son choix parmi 37 modèles testés, analysés, examinés et écoutés.



L 09. 72 p. Prix : 65 F TTC
Pour la première fois en électronique, un lexique anglais-français est présenté sous une forme pratique avec en plus des explications techniques, succinctes mais précises. Ce sont plus de 1 500 mots ou termes anglais qui n'auront plus de secret pour vous.



L 11. 160 p. Prix : 85 F TTC
Finis les calculs fastidieux et erronés ! Grâce à cet ouvrage, les concepteurs d'enceintes acoustiques gagneront un temps appréciable durant la phase d'étude et de mise au point : 120 abaques et tableaux pour tous types de filtres et d'impédances de HP !

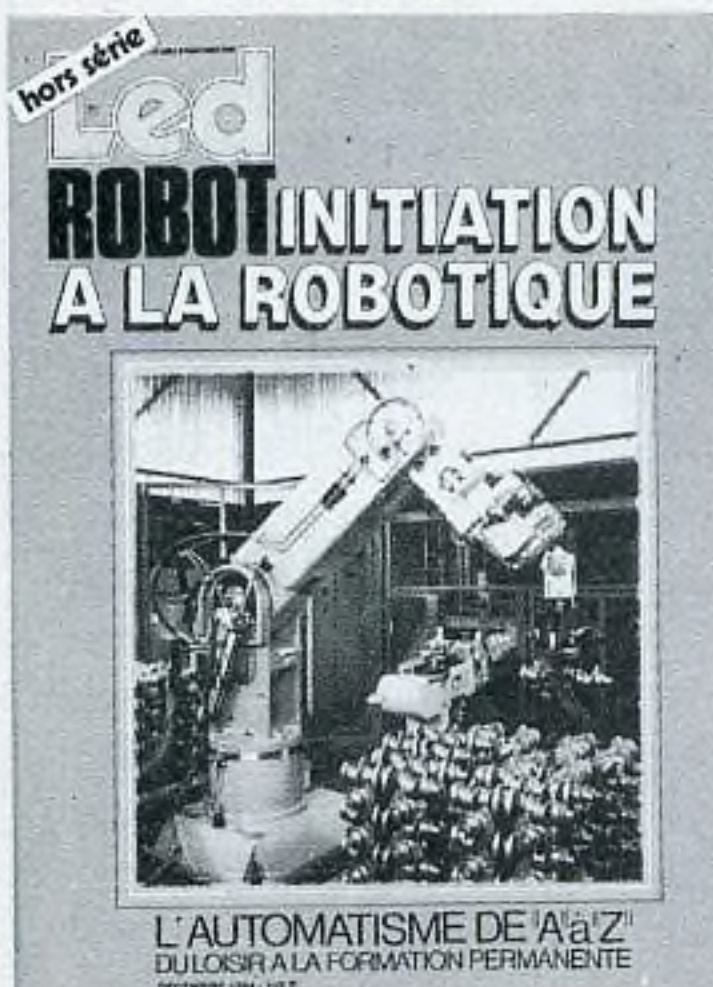


L 14. 128 p. Prix : 95 F TTC
Voici enfin réunies dans un même ouvrage, dix-sept descriptions complètes et précises de montages électroniques simples. Il s'agit de réalisations à la portée de tous, dont bon nombre d'exemplaires fonctionnent régulièrement. Les schémas d'implantation et de circuits imprimés sont systématiquement publiés.



L 20. 208 p. Prix : 130 F TTC
Accessible à tous, «Week-end photo» permet de découvrir de façon simple les différents aspects de la photographie actuelle. Vous y trouverez les bases indispensables pour vous perfectionner, un guide de choix des appareils 24 x 36 et des illustrations abondamment commentées.

Collection initiation (format 210x270)



P 08. 96 pages. Prix : 115 F TTC

Cet ouvrage eut un succès retentissant dès sa sortie. Bien plus qu'un cours d'initiation, il s'agit aussi du premier recueil d'informations données par les concepteurs, les utilisateurs de robots et les fans de cybernétique, enfin réunis !



P 16. 272 pages. Prix : 130 F TTC

Passé les premiers remous de la révolution que fut l'avènement de la micro-informatique, il fallut bien tenter d'en réunir les enseignements. Une lacune apparut : celle d'un ouvrage d'initiation à la programmation, universel et complet. En voici le premier tome.



P 17. 208 pages. Prix : 130 F TTC

Le tome 2 est la suite du tome 1 : l'esprit puissamment didactique de l'auteur s'y retrouve, le contenu du livre permettra d'acquérir un niveau suffisant pour exercer l'analyse, la programmation, la gestion, l'automatisme, la simulation et d'autres choses encore !



P 27. 250 pages. Prix : 190 F TTC

Le troisième volume du cours de Programmation, dû à Cl. Polgar, pédagogue apprécié de tous. Il continue dans la lignée d'un réel souci didactique, de haut niveau, maintenant, mais en conservant l'aspect progressif qui fit son succès initial.



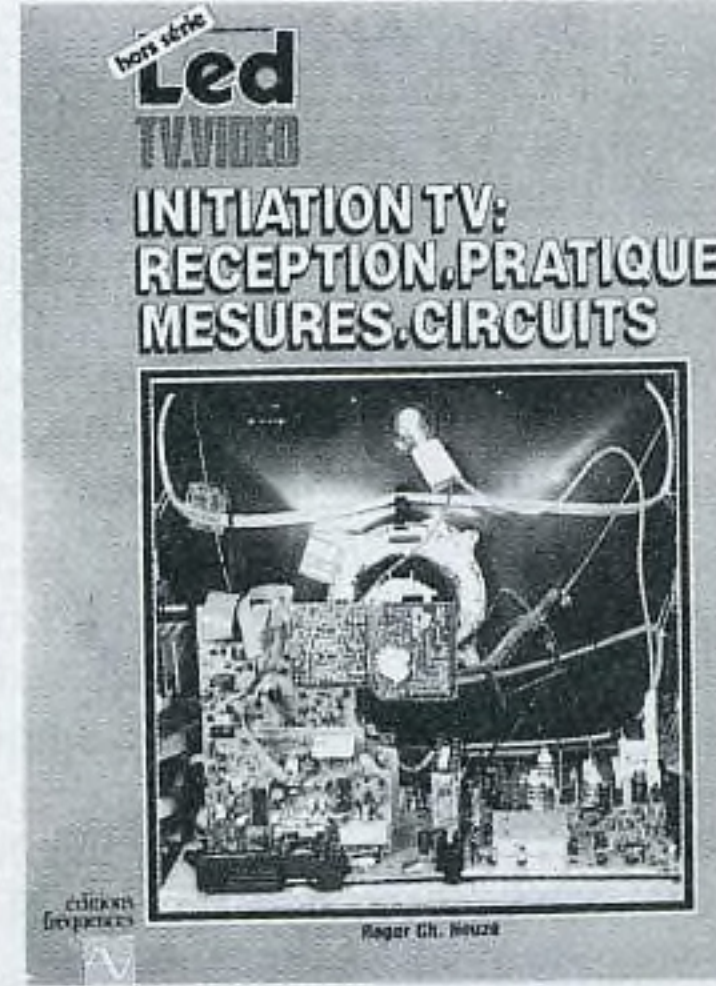
P 18. 136 pages. Prix : 95 F TTC

Du même auteur, Ph. Duquesne, on nous propose cette fois-ci, de pénétrer au cœur même de l'ordinateur, de comprendre le fonctionnement de l'élément vital qu'est le microprocesseur et enfin de maîtriser l'assembleur, langage du microprocesseur.



P 19. 104 pages. Prix : 95 F TTC

Ce cours d'Initiation à l'Electronique Digitale est dû à Ph. Duquesne, chargé de cours de microprocesseurs au CNAM. L'objet de cet ouvrage est de présenter les opérateurs logiques et leurs associations. La technologie est évoquée, brièvement, elle aussi.



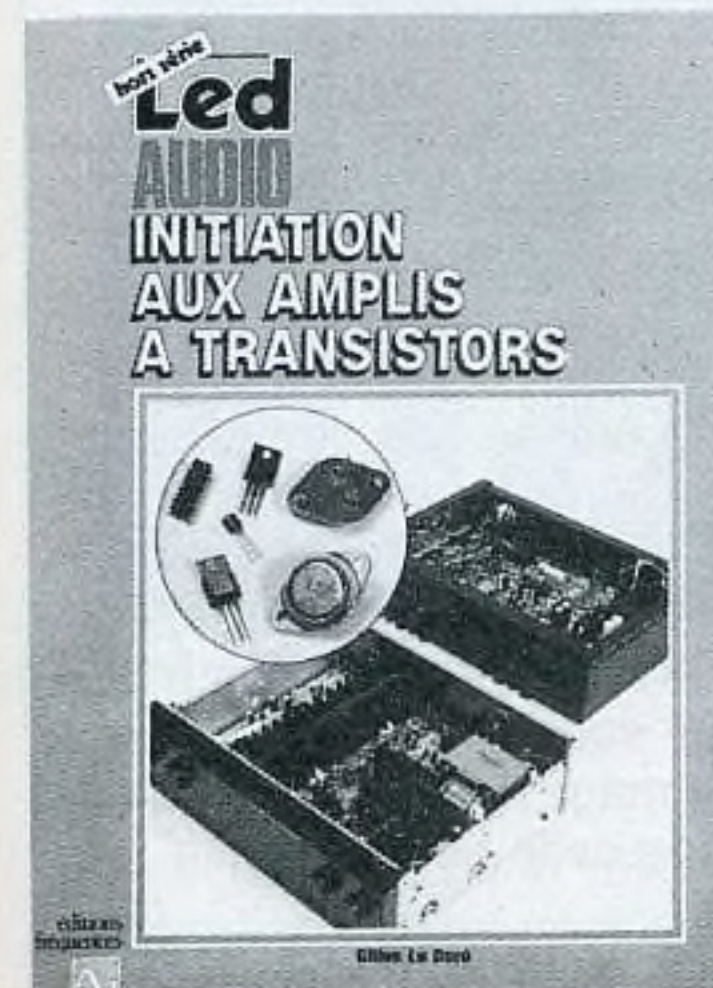
P 21. 136 pages. Prix : 135 F TTC

Issu d'un cours régulièrement remis à jour, ce livre permet à l'amateur comme au professionnel de se tenir au courant de l'état actuel de la technologie en télévision. De nombreux schémas explicatifs illustrent le contenu du livre.



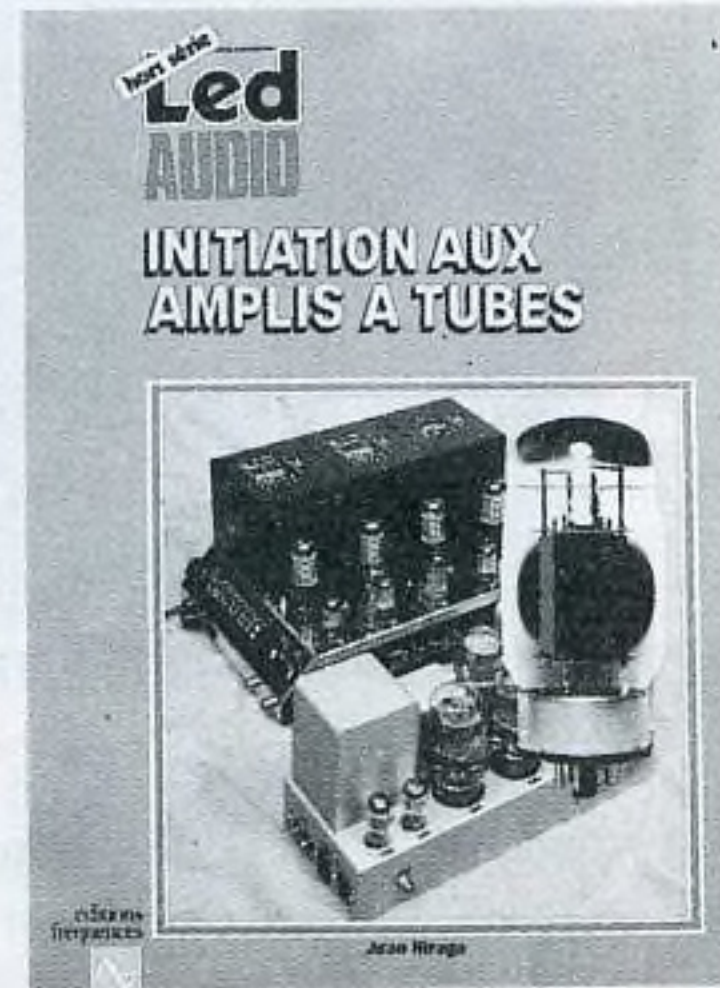
P 23. 120 pages. Prix : 140 F TTC

Il n'existait pas, jusqu'à présent, un ouvrage couvrant de manière générale mais précise, l'ensemble des problèmes relatifs à l'instrumentation et à la méthodologie du laboratoire électronique. C'est chose faite aujourd'hui avec ce volume récemment paru.



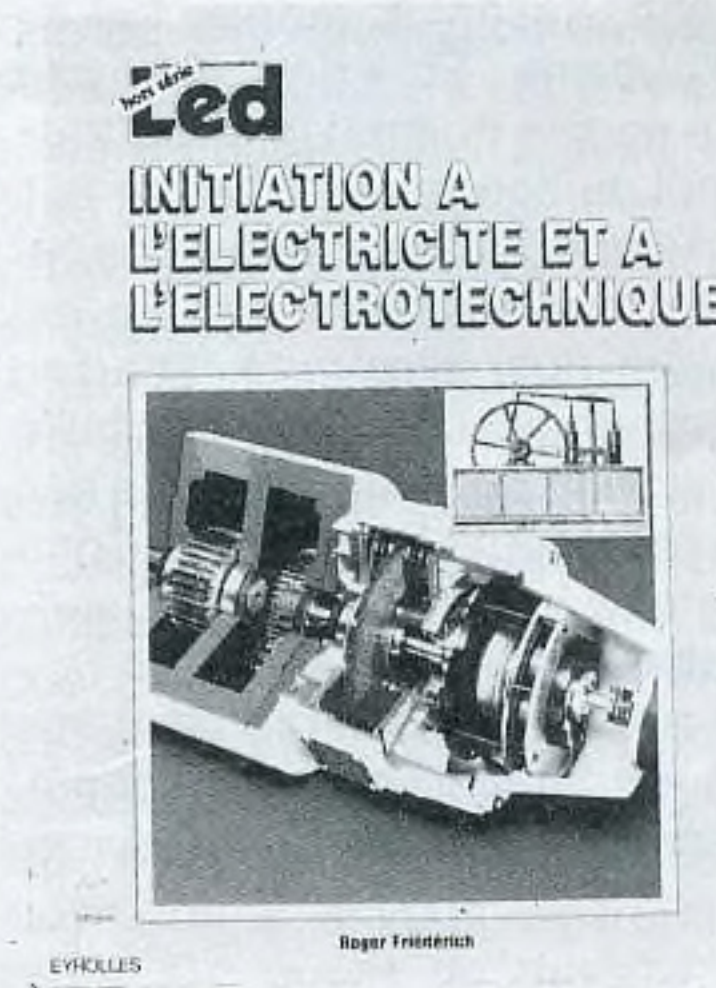
P 24. 96 pages. Prix : 130 F TTC

Après un bref historique du transistor, cet ouvrage traite essentiellement de la conception des amplificateurs modernes à transistors. La théorie est décrite de manière simple et abordable, illustrée d'exemples de réalisations commerciales. Le but du livre est de donner à chacun la possibilité de réaliser soi-même son amplificateur...



P 26. 152 pages. Prix : 155 F TTC

Complémentaires des «Amplis à transistors», les «Amplis à tubes» sera certainement une petite encyclopédie sur ce sujet : historique, mais aussi polémique, puisque les tubes sont encore d'actualité et parce que les arguments en faveur de cette technique et ses défenseurs sont encore nombreux.



P 28. 110 pages. Prix : 150 F TTC

Vous trouverez aisément en librairie des ouvrages d'initiation à l'électronique ou aux techniques les plus avancées des circuits intégrés, etc. Mais si vous désirez une initiation aux bases de l'électricité et de l'électrotechnique sans vous en remettre à des ouvrages scolaires, alors vous ne trouverez pas !



P 29. 64 pages. Prix : 100 F TTC

— Choix d'un standard ? — Caméscopes VHS, VHS-C ou 8 mm ? — Connexion ? Compatibilité ? — Accessoires ? Montage ? Enfin... Comment filmer. Le nouveau livre de Claude Gendre répond à toutes ces questions, cet ouvrage essentiellement pratique qui n'a pas d'équivalent en librairie aujourd'hui.

C'EST ARRIVE DEMAIN



(en direct de notre envoyé permanent dans la Silicon Valley)

Le saviez-vous ? IBM ne représente plus la moitié des PC et compatibles sur le marché international, mais seulement 40 %, en ne tenant compte que du standard que cette société a imposé. Juste retour des choses pour IBM, qui a voulu s'imposer avec un système obsolète dès sa sortie. Aujourd'hui, le marché est tenu par Compaq, Laser, et autres, qui se frottent les mains de l'aubaine. En effet, si la part d'IBM diminue, le marché de ses compatibles est toujours fort fructueux, et nul ne songe à cacher ce fait. IBM perd des parts fort importantes néanmoins, et tous les observateurs le reconnaissent, ce qui est suffisamment peu fréquent pour être noté. Leading Edge et Epson représentent ainsi 4 % de la totalité des ordinateurs vendus au détail, ce qui n'est pas rien pour des débutants, et représente autant que Compaq, et encore plus fort, ces trois firmes et ATT vendent plus de PC-compatibles que IBM.

Le point le plus frappant est que c'est la première fois qu'IBM perd des places sur un marché électronique. Et cette perte ne se passe plus seulement sur le mar-

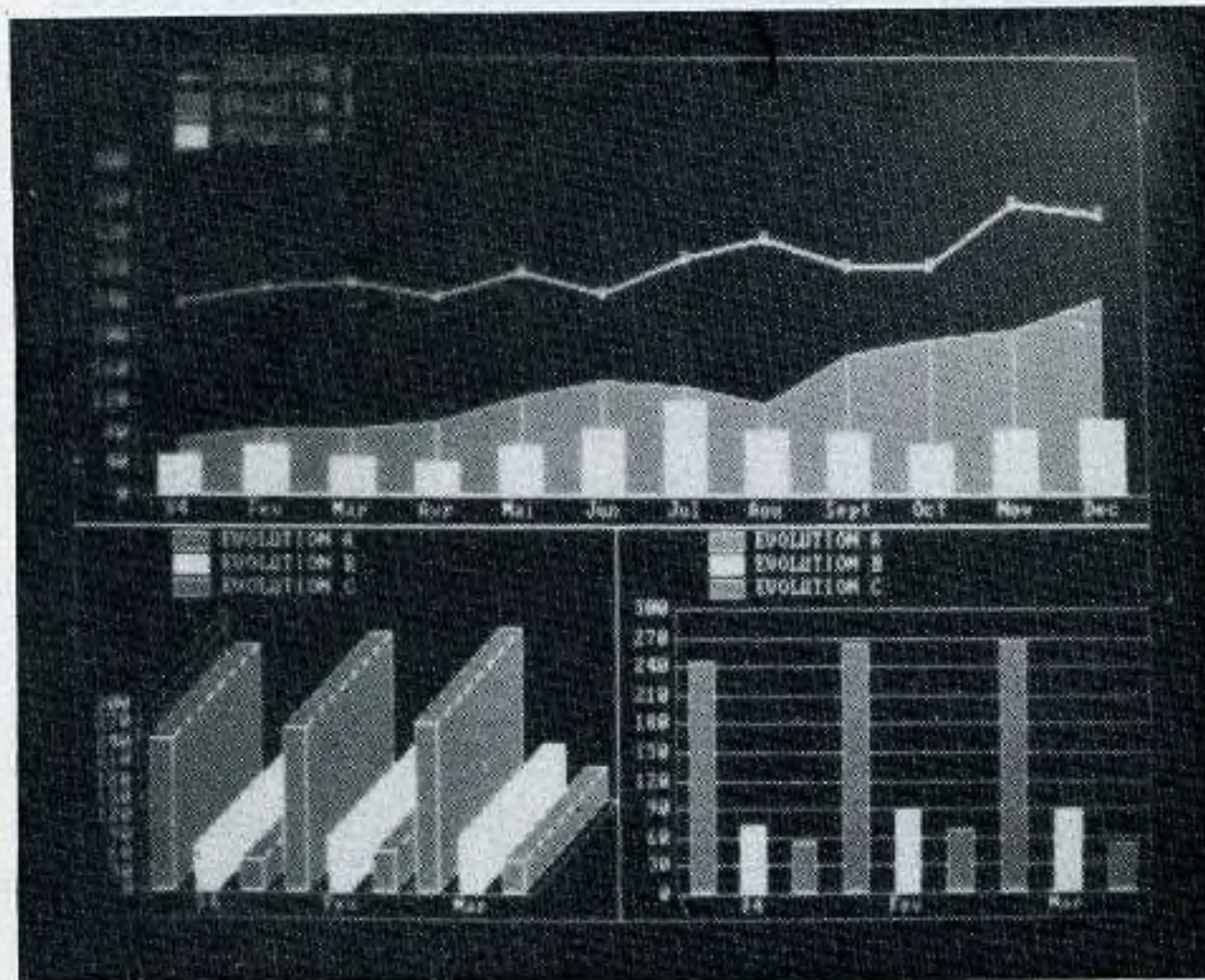
ché des petites machines, mais aussi sur de plus grosses.

Les analystes américains s'intéressent aux micros selon trois grandes catégories. Tout d'abord, le plus rapide, puis le moins cher, puis les autres. Un seul ordinateur dans chacune des deux premières catégories. Les autres, dans le marais de la troisième catégorie. Et l'on voit ainsi que IBM se place toujours dans ce marais, pas mal d'ailleurs, mais le plus souvent, le choix est fait sur les premières catégories, où la différence est plus aisément compréhensible, et promotionnée.

Enfin, IBM a fait quelques erreurs de promotion, par exemple pour le PC Junior. Lors de son lancement aux USA, il fut annoncé en Europe comme devant être vendu six mois plus tard. Aux USA, tout le monde critiqua les faibles performances, la vente fut nulle, et le produit retiré du marché. Puis IBM commença à le diffuser en Europe, ce qui souleva une véritable vague d'indignation dans les magazines. La publicité pour IBM n'était peut-être pas celle souhaitée !!

Toujours est-il que maintenant IBM est regardé ici comme un simple membre de la communauté micro, et non plus comme le leader, qu'il est pourtant toujours. Les gens ont décidément la mémoire courte, mais c'est en général la contrepartie d'une (trop ?) grande activité. On ne pardonne rien aux anciens dieux déçus, ou considérés comme tels.

La grande mode actuellement parmi les cadres américains ce sont les simulateurs. Ils peuvent être ludiques ou professionnels, graphiques ou analytiques, ils sont les rois du moment. Ils sont d'ailleurs encouragés par les chefs de service, comme soupapes de sécurité contre le surmenage. Ce peuvent être des simulations bancaires, militaires, ou de pilotage, ou encore un moyen de comprendre et d'analyser un marché en évolution. Il est amusant de voir que ce qui représentait deux marchés totalement séparés a pu se fondre sur le simple motif que l'on manipule le futur, l'inconnu, quitte à recommencer pour obtenir les résultats souhaités. Voilà qui est motivant. Les leaders de cette branche sont les simulateurs de vols d'avions, et les simulations bancaires financières, qui vous proposent en général, et sans risque pour votre portefeuille, de devenir riche et de prendre une retraite aisée (comprenez multimillionnaire) vers 18 ans. Le rêve américain en quelque sorte. Le plus drôle est que certains appliquent ensuite leurs connaissances au marché boursier réel, et que cela marche parfois. Pour les autres, le seul vrai risque est le prix du Kilo-Watt-Heure, car le jeu dure souvent plusieurs dizaines d'heures.



Sortie graphique typique d'une simulation boursière.

Je vous avais parlé voilà quelques mois de la grande utilisation de traitements de textes, à titre professionnel, par les ingénieurs américains, au grand dam des professions de secrétaire, dactylo... Le résultat de

cette utilisation massive est la sortie de traitements de textes toujours plus perfectionnés, et simples à utiliser, ce qui semble ne plus être toujours antinomique, Dieu est grand. Des programmes comme Word 3.0, NewWord 3, sur PC, ou Write Plus sur Macintosh sont des bijoux chacun dans leur domaine. Sur PC, les environnements de type multi-fenêtres, souris se développent, avec les possibilités de menus, et autres. Sur le Mac, les colonnages, insertions de graphiques sous formatage variable deviennent enfin possibles. Le Mac continue à avoir deux ans d'avance dans ce domaine, principalement grâce à son menu «pomme», et le meilleur produit de l'instant pour cet appareil est français. Cocorico, Writer Plus est le fruit du travail d'un ingénieur de chez nous et distribué par ACI. Vous savez, 4^e dimension, c'est eux.

Un progrès dans l'utilisation des communications entre ordinateurs. Avec les mémoires incorporées aux Modems, il est possible de travailler en différé, donc de choisir une heure à tarif réduit pour utiliser la ligne, et ainsi diminuer sa facture de manière significative, souvent plus que ne le permettrait un Modem plus rapide. Ainsi, vous pouvez dormir tranquillement à des heures franchement nocturnes, pendant que votre ligne et le modem travaillent sans se douter de votre absence. Ne leur dites pas, ils en seraient peut-être vexés.

Nous débutons ce mois-ci une nouvelle évolution de cet article, avec la description des possibilités des programmes et des systèmes les plus avancés sortant aux USA, ce qui vous permettra de vous faire une idée de ce que vous trouverez dans les boutiques de l'Hexagone sous peu (six mois, un an, ...!). Il s'agira ici, non de décrire l'utilisation d'un programme, mais de voir comment évolue le marché, quel programme apporte des solutions originales, etc...

Suite à ce que nous avons dit dans la première partie, il nous a semblé logique de débiter par une description des nouveautés en matière de traitements de textes.

La nouvelle génération de traitements de textes sur PC et compatibles s'est donnée pour but de simplifier sans dérouter. Cela signifie que les techniques de type souris, menus,... font leur apparition, mais que les présentations générales ont été, le plus possible, conservées.

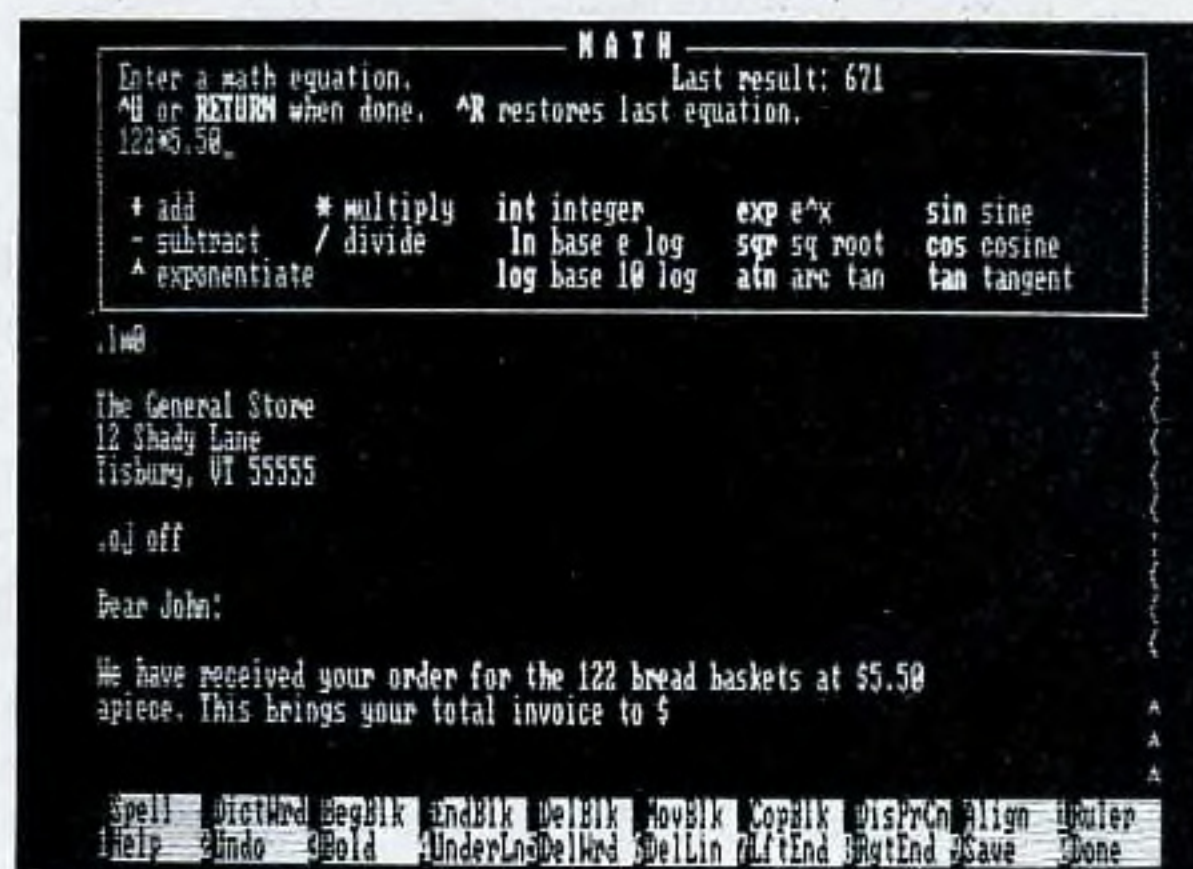
Ainsi, un programme, du nom de NewWord 3, propose une sorte de super WordStar, avec des commandes très proches de celles de son glorieux ancêtre, et la possibilité de lire les fichiers de celui-ci. La compatibilité vraie, en quelque sorte. Mais en plus de la réplique de l'environnement WordStar, c'est-à-dire que les commandes du premier sont reprises sur le second de la même façon, il y a de nombreux plus sur NewWord 3. Ainsi, il est possible de récupérer du texte effacé, si vous avez changé d'avis en cours de route. Ce genre de possibilités, trop rares, sont à signaler, car elles sont le noyau d'un système souple

et confortable. Dans la même veine, il est possible de se créer des touches de macros, que vous aurez définies entièrement vous-même, permettant ainsi de ne taper qu'une touche pour sauvegarder le fichier courant, le fermer, en ouvrir un autre, etc... New Word 3 propose également un module de calcul assez complet, le tout tenant en 300 Ko. Le plus drôle est que ce programme est proposé 1 \$ moins cher que WordStar, 350 \$ tout de même.

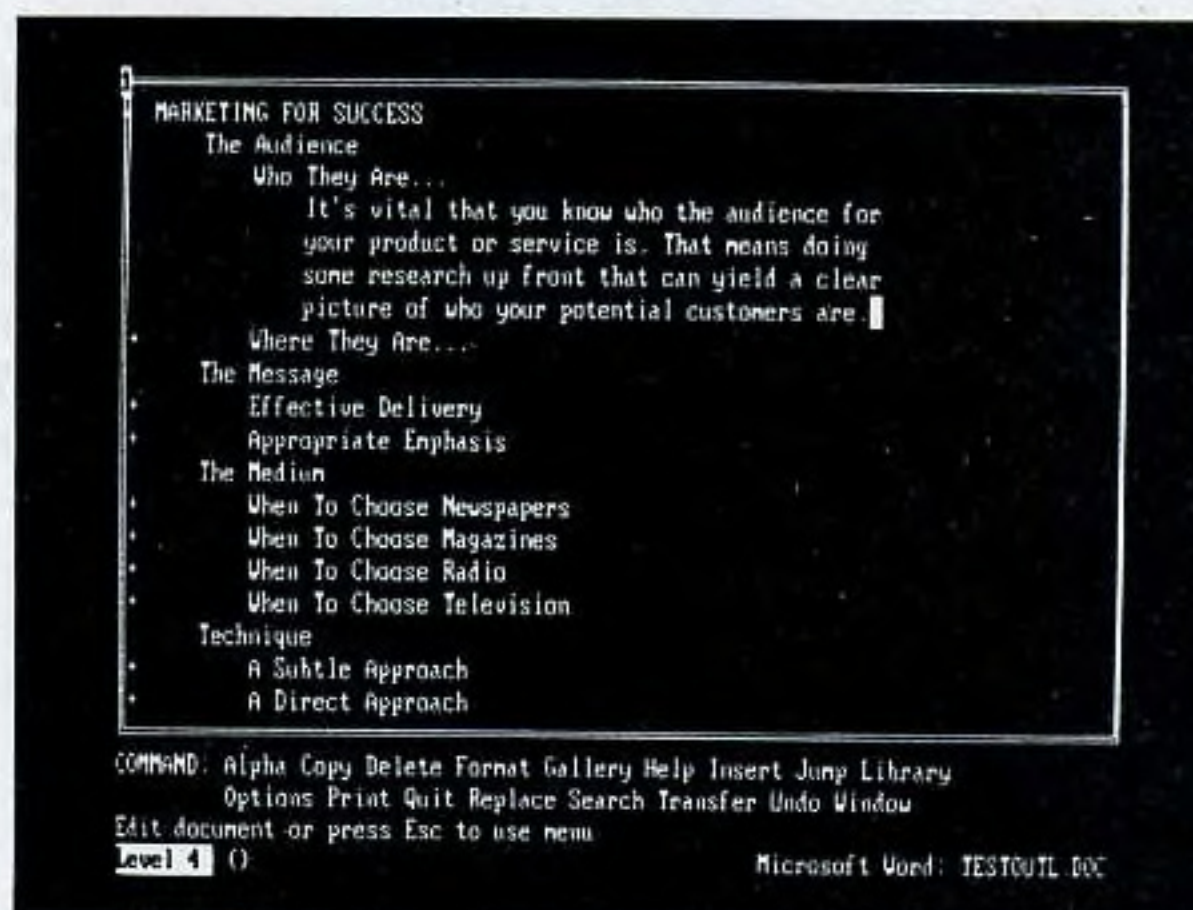
Il ne peut être question pour NewWord de nouvelle génération, car il est trop proche de son aîné pour cela. En clair, un utilisateur ne connaissant pas WordStar trouvera NewWord bien peu séduisant, avec ses commandes peu naturelles, et tellement plus complexe qu'un programme utilisant les menus déroulants.

Un autre programme fait son apparition depuis peu. Il s'agit de Word 3.0.

Les nouveautés que l'on trouve dans cette version du populaire traitement de textes sont principalement des facilités pour créer des zones de commentaires, et pour jongler avec la typographie, sur la sortie papier. Enfin, il n'y a plus de protection sur la disquette. Mais ce dernier point est sans doute réservé aux USA, l'Europe, et principalement la France ayant la réputation de piratage outrancier. De toute façon, cette confiance (toute relative) vis-à-vis de l'utilisateur local se paie par un coût très supérieur au précédent modèle. Quoi d'autre sur cette nouvelle version ? Une zone de texte pour travailler, et une zone de commande en pied d'écran, ce qui permet de visualiser les principales actions possibles, et la façon de les obtenir. Ainsi, plus de trou de mémoire, plus de référence trop fréquente au manuel, tout est là. Simplement, il est nécessaire de savoir sous quel alinéa, quelle commande se trouve la vôtre. Par exemple, vous ne trouverez pas Sauvegarde, ou Chargement, car il faut d'abord activer Transfert. Ce genre de chose ne facilite en rien l'utilisation du programme, et lui est grandement dommageable. Des progrès par contre au niveau de certaines possibilités. Ainsi, il est possible d'annuler la dernière commande réalisée, et donc de récupérer un morceau de texte effacé. Enfin, l'adjonction de macros permet de limiter les touches à frapper dans des cas fréquents, comme le passage de texte d'un fichier à un autre; ou autres actions souvent répétées. Le multi fenêtrage est maintenant possible, et vous pourrez travailler sur huit fichiers au maximum, ce qui, en texte, est généralement suffisant. Un vérificateur de texte est disponible, mais sa lenteur est telle que son emploi est à peine conseillé. De même, un module de calcul rudimentaire semble ne devoir servir qu'à être utilisé pour la publicité de Microsoft, tellement il est limité et de peu d'intérêt.



NewWord 3.



Word 3.0.

Parmi les commandes les plus utiles, vous aurez en pied de page la possibilité de réarranger le texte en triant diverses parties, paragraphes, le plus simplement du monde. Ceci est très commode, et demande à être généralisé, mais semble gourmand en taille mémoire.

Il faut signaler que ce programme est l'un des premiers qui suivent sur un PC la méthode WYSIWYG. Qu'est-ce que c'est que cette bête étrange ? Cela signifie «What You Is What You Get», ou encore, «Ce que vous voyez (sur l'écran) est exactement ce que vous aurez (sur votre imprimante)». Pour cela, il faut tout de même une carte graphique sur votre PC, qui soit de plus compatible avec le format Hercules. Alors, Italiques, Gras, Soulignés sont possibles, comme sur un Mac.

Au mois prochain.

COURS D'INITIATION AU PROGICIEL MULTIPLAN

Charles-Henry Delaleu

1^{re} PARTIE

L'acquisition d'un micro-ordinateur s'accompagne dans la quasi-totalité des cas de l'achat d'un certain nombre de progiciels. Rappelons qu'un progiciel est un logiciel dupliqué à grande échelle et destiné à la vente en masse. Parmi les grands progiciels vendus, notons Visicalc (tableur), dBase II (gestion de fichiers), Wordstar (traitement de texte). De nos jours, il existe dans tous les domaines d'applications courantes, un nombre non négligeable de progiciels en tout genre. Dans ce numéro de Led-Micro, nous nous pencherons sur Multiplan. Multiplan est un tableur. Un tableur est aussi appelé «feuille de calcul électronique». Ce progiciel est l'un des programmes les plus vendus à travers le monde. Il existe de nombreux tableurs, les plus classiques sont : Visicalc, Supercalc, Calc-Result, etc. Les applications des tableurs sont nombreuses. Multiplan a l'avantage de s'appliquer à la majorité des besoins, de plus sa souplesse est grande. Il est disponible sur la quasi-totalité des machines (petites ou plus grosses). Son prix reste abordable. Enfin, il s'agit d'un produit bien rôdé qui n'a plus à faire ses preuves. Sa seule lacune était son espace mémoire un peu limité dans certains cas, qui vient d'être comblé par la version étendue disponible depuis peu sur IBM PC et compatibles. Notons qu'un certain nombre d'ouvrages ont été écrits sur ce progiciel et que certains d'entre eux sont même accompagnés de programmes d'essais.

Ce cours d'initiation sur Multiplan devrait se prolonger sur quatre numéros. Il serait intéressant que les lecteurs connaissant déjà ce produit et ayant réalisé des applications originales se manifestent afin de faire profiter de leurs expériences l'ensemble du lectorat.

MULTIPLAN : PRESENTATION SOMMAIRE (CARACTERISTIQUES TECHNIQUES)

Fonction : tableur.

EDITEUR : Microsoft.

Mémoire vive requise : 64 koctets au minimum.

Capacité en version de base : 255 lignes, 63 colonnes.

INTRODUCTION AUX TABLEURS

Un tableur sert essentiellement à résoudre des problèmes de gestion. Qu'il s'agisse de comptabilité ou de la simulation d'un réseau de vente, d'une équipe commerciale, la feuille électronique de calcul s'adapte assez facilement à tous les cas de figure. Que ce soit dans une petite ou une grosse entreprise, le tableur est un outil indispensable à un certain nombre de services (comptabilité, service commercial, direction, achat, contrôle de gestion, etc.). Il convient de noter qu'il existe même des versions spécialisées de tableur disposant de fonctions mathématiques évoluées pour techniciens et ingénieurs. Multiplan est un instrument de travail puissant. Il aide le comptable dans la préparation de ses états, le commercial peut, grâce à lui, réaliser des statistiques, des simulations de campagnes. C'est un outil de précision pour le responsable des achats, il assiste le responsable du personnel dans ses plans d'embauche, etc.

Enfin, un tableur est un outil d'aide à la décision pour le chef d'entreprise qui peut, grâce à lui, simuler et étudier son chiffre d'affaires, sa marge possible, l'amortissement de ses investissements, etc.

NOTIONS DE BASES

Un tableur est architecturé à partir d'une matrice de cases appelée cellules. Il existe donc un certain nombre de cellules rangées par ligne. Une ligne de cellules est composée d'un certain nombre de cellules rangées sous forme de colonnes. De ce fait, chaque cellule porte une adresse qui correspond à son numéro de ligne et à son numéro de colonne. De ce fait, la cellule L5C7 se trouve à la 7^e colonne de la 5^e ligne, la cellule L3C12 est en 12^e colonne, 3^e ligne. Chaque cellule porte donc un nom qui correspond à son numéro de ligne et son numéro de colonne. Chaque cellule peut être formatée en largeur. Une cellule peut recevoir :

- une chaîne de caractères (alphanumérique) ;
- un nombre ;
- une équation.

Suivant l'application à résoudre, les cellules seront programmées en alphanumérique pour faciliter la présentation et la lecture de la feuille de calcul, en nombres et équations pour ce qui concerne les calculs.

Soit l'exemple suivant : une ménagère désire vérifier le montant du ticket de caisse qu'elle a reçu du super-marché. Elle a acheté : 3 kg de sucre, 2 l d'huile, 1 boîte de petits pois, 1 litre de jus de fruit et 5 kg de pommes de terre. Sous une feuille de calcul électronique, les choses se présenteront de la façon suivante :

		1	2	3	4	N° de colonne
N° de ligne	1		Qté	P. unitaire	S-Total	
	2	Sucre	3	7	21	
	3	Huile	2	10	20	
	4	Petit pois	1	5	5	
	5	Jus de fruit	1	12	12	
	6	P. de terre	5	6	30	
	7					
	8	TOTAL			88	

En L3C1, nous avons marqué le type de produit (en alphanumérique).
 En L3C2 sa quantité (en numérique).
 En L3C3 son prix unitaire (en numérique).
 En L3C4, nous avons effectué un calcul équivalent à la quantité multipliée par le prix unitaire. Nous avons donc programmé la cellule L3C4 comme étant le produit de L3C2 par L3C3 (mode calcul).
Nota : La programmation des cellules sera l'objet du second cours.

CONCLUSION

Chaque ligne est ainsi préparée. Une fois le travail accompli, le tableau est programmé. Le fait de changer le prix unitaire de nos petits pois en L4C3 entraînera le recalcul du sous-total en L4C4 et du total en L8C4.

$$(L8C4 = L2C4 + L3C4 + L4C4 + L5C4 + L6C4)$$

Conclusion, le fait de changer une valeur provoque le recalcul automatique de l'ensemble de la feuille. C'est ici que le tableur prend toute sa signification. Bien évidemment notre exemple est sûrement un peu simplifié. Mais il suffit de porter ce concept à un ensemble de plusieurs dizaines ou centaines de cellules pour comprendre la puissance d'un tel programme.

MISE EN ROUTE DE MULTIPLAN

Généralement Multiplan se présente sous la forme de deux disquettes. La première disquette contient les programmes et la seconde un ensemble des utilitaires permettant en outre de configurer Multiplan en fonction de l'imprimante utilisée. Comme toujours, il convient avant toute chose de préparer des copies de deux disquettes : INSTALL et MULTIPLAN. (Une erreur est vite arrivée !) Dans le cas de Multiplan 1^{re} version, les disquettes sont copiables. Pour la nouvelle version étendue, il n'y a pas de copie possible. Si vous endommagez votre disquette, il faudra revoir votre revendeur afin d'obtenir une nouvelle disquette sans perdre votre licence.

Après avoir chargé votre système d'exploitation lors de la première exécution, lancer le programme INSTALL qui vous permet de configurer votre système. Ensuite, lorsque vous serez sous DOS, il suffira de taper MP (MP pour Multiplan). Dès lors, le programme Multiplan se chargera. Pour les exécutions suivantes, taper MP directement pour accéder à Multiplan.

Une fois Multiplan commencé, vous verrez apparaître à l'écran votre image de travail qui se présente en trois parties.

La zone tableau. C'est la feuille de calcul proprement dite. Elle occupe la majeure partie de l'écran.

La zone commande. Placée en bas de l'écran, la zone commande contient l'ensemble des commandes et options utilisables sous Multiplan.

La zone état. Il s'agit de la dernière ligne en bas de l'écran. Elle affiche des informations sur votre travail.

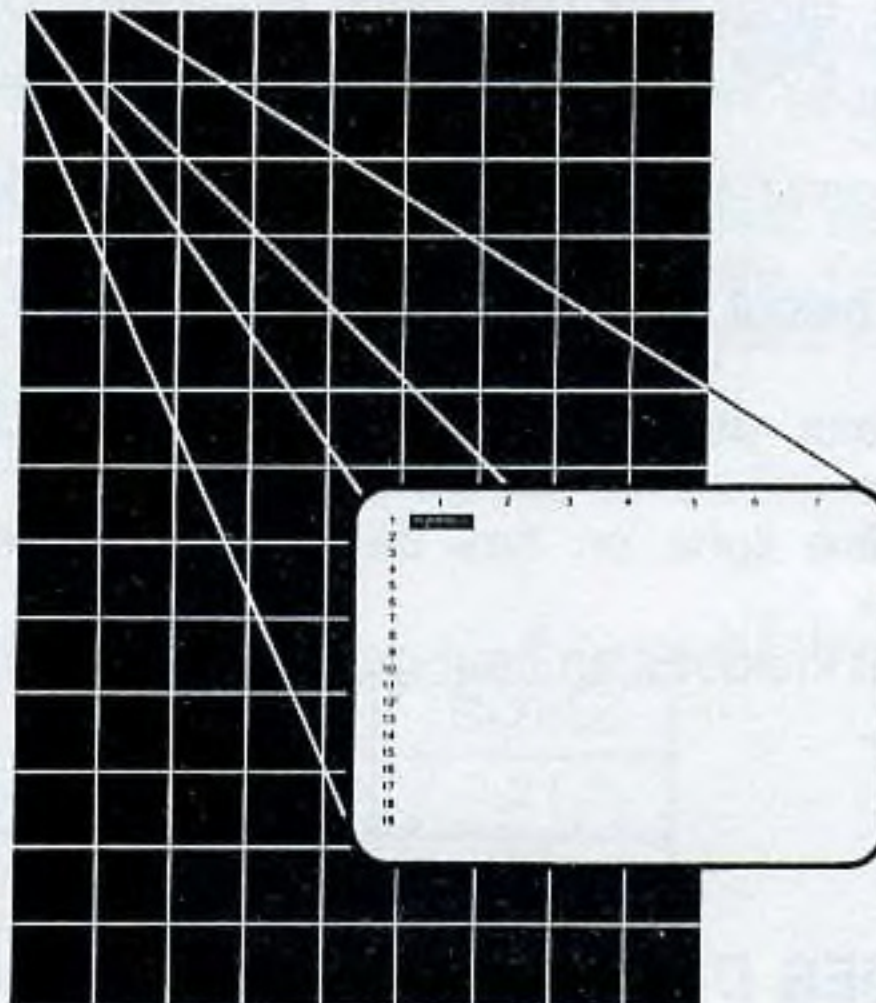
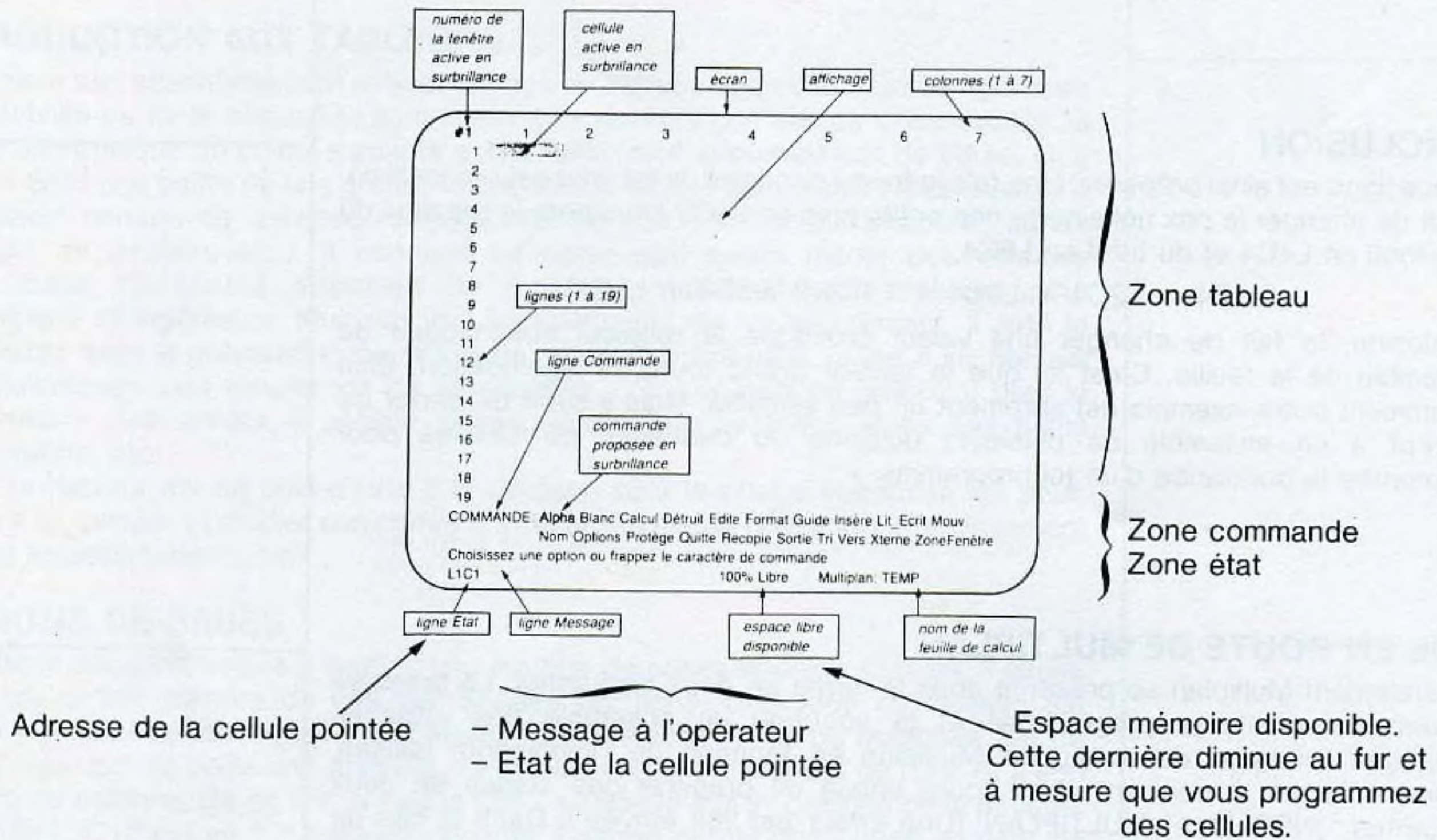
Nota : Les numéros de colonnes sont indiqués en haut de l'écran et les numéros de lignes sur la partie gauche de l'écran.

CONFIGURATION SUR DISQUES DUR

Si vous désirez utiliser Multiplan sur un disque dur, il faudra préparer un sous-dossier afin d'y placer votre progiciel : ne jamais placer plusieurs progiciels dans un seul et unique volume. Si vous employez plusieurs progiciels, il est souhaitable d'implanter tous les progiciels sur disque dur et d'effectuer la sauvegarde sur disquette. Attention à ne pas oublier de réaliser des copies de vos sauvegardes.

L'ECRAN DE COMMANDES MULTIPLAN

L'écran de Multiplan tel qu'il apparaît lors de la mise en route du progiciel.



Multiplan est composé en version de base de 255 lignes de 63 colonnes chacune. Il n'est donc pas possible de voir à l'écran toute la feuille de calcul. De ce fait, seule une petite partie est visible de l'écran. Il est possible de se déplacer dans l'ensemble de la feuille. De même, il est possible de visualiser en même temps des informations se trouvant à différents endroits de la feuille. (Ceci fera l'objet d'un prochain cours).

DEPLACEMENT DU POINTEUR DE CELLULE

#1	1	2	3	4	5	6	7
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

COMMANDE: Alpha Blanc Calcul Détruit Edite Format Guide Insère Lit_Ecrit Mouv.
 Nom Options Protège Quitte Recopie Sortie Tri Vers Xterne ZoneFenêtre
 Choisissez une option ou frappez le caractère de commande
 L1C2 100% Libre Multiplan: TEMP

Il est possible de se déplacer dans la feuille de calcul. Le pointeur de cellule peut être amené vers la droite, puis vers la gauche (ou de le changer de ligne). Il existe pour cela deux techniques. La première consiste à se servir des flèches de déplacement du clavier pour amener le pointeur à la position désirée. Un clavier de micro-ordinateur est généralement équipé de quatre flèches de direction. Dans le cas de Multiplan, ces flèches sont programmées pour déplacer le pointeur de cellule. La seconde technique consiste à utiliser la commande VERS (étudiée dans le prochain cours).

#1	1	2	3	4	5	6	7
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

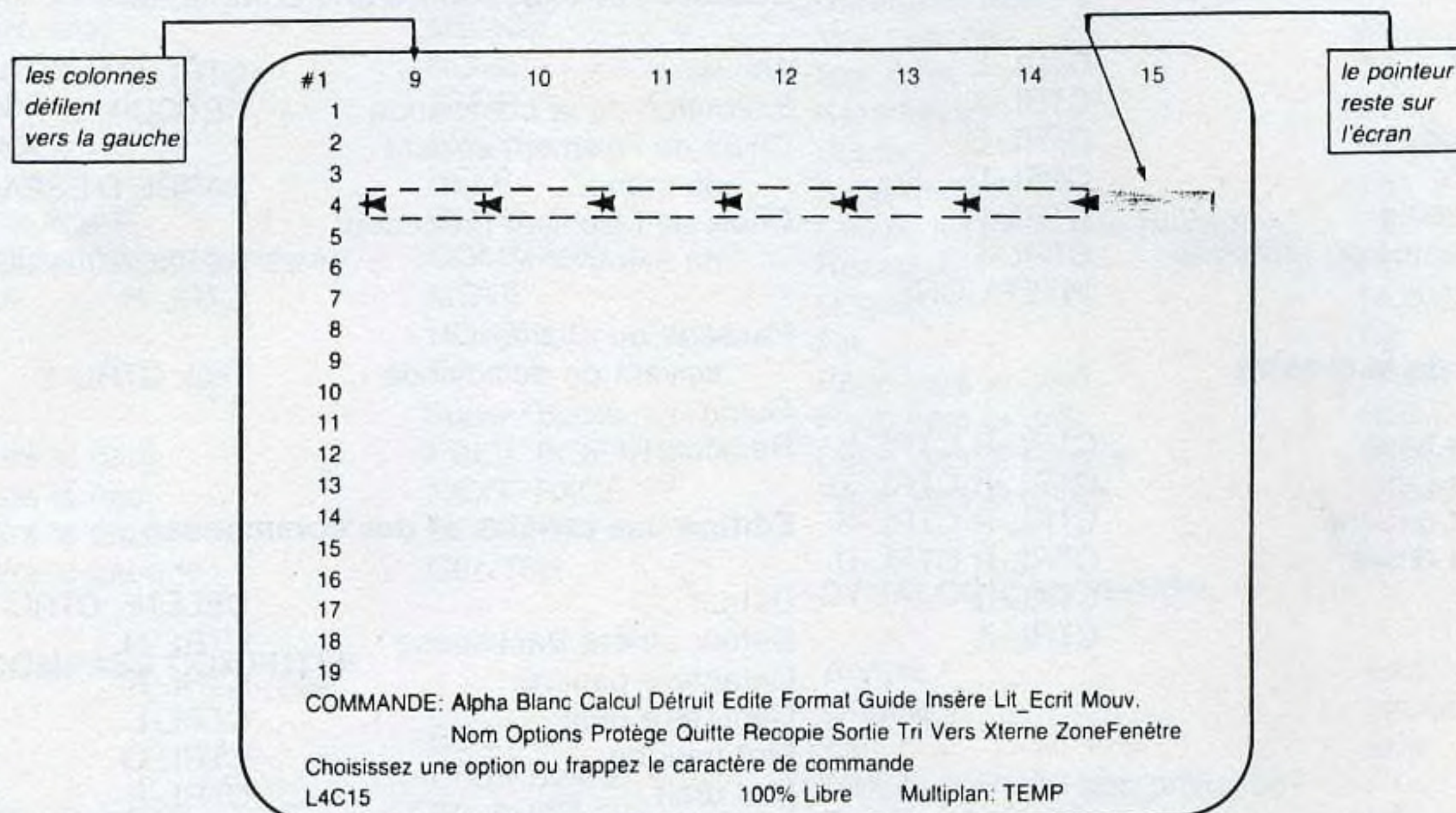
COMMANDE: Alpha Blanc Calcul Détruit Edite Format Guide Insère Lit_Ecrit Mouv.
 Nom Options Protège Quitte Recopie Sortie Tri Vers Xterne ZoneFenêtre
 Choisissez une option ou frappez le caractère de commande
 L4C4 100% Libre Multiplan: TEMP

ligne Etat :
indique la cellule active ligne 4 colonne 4

Le fait de déplacer le pointeur de cellule provoque automatiquement une modification de la ligne état qui indique la cellule active (ex. : L4C4).

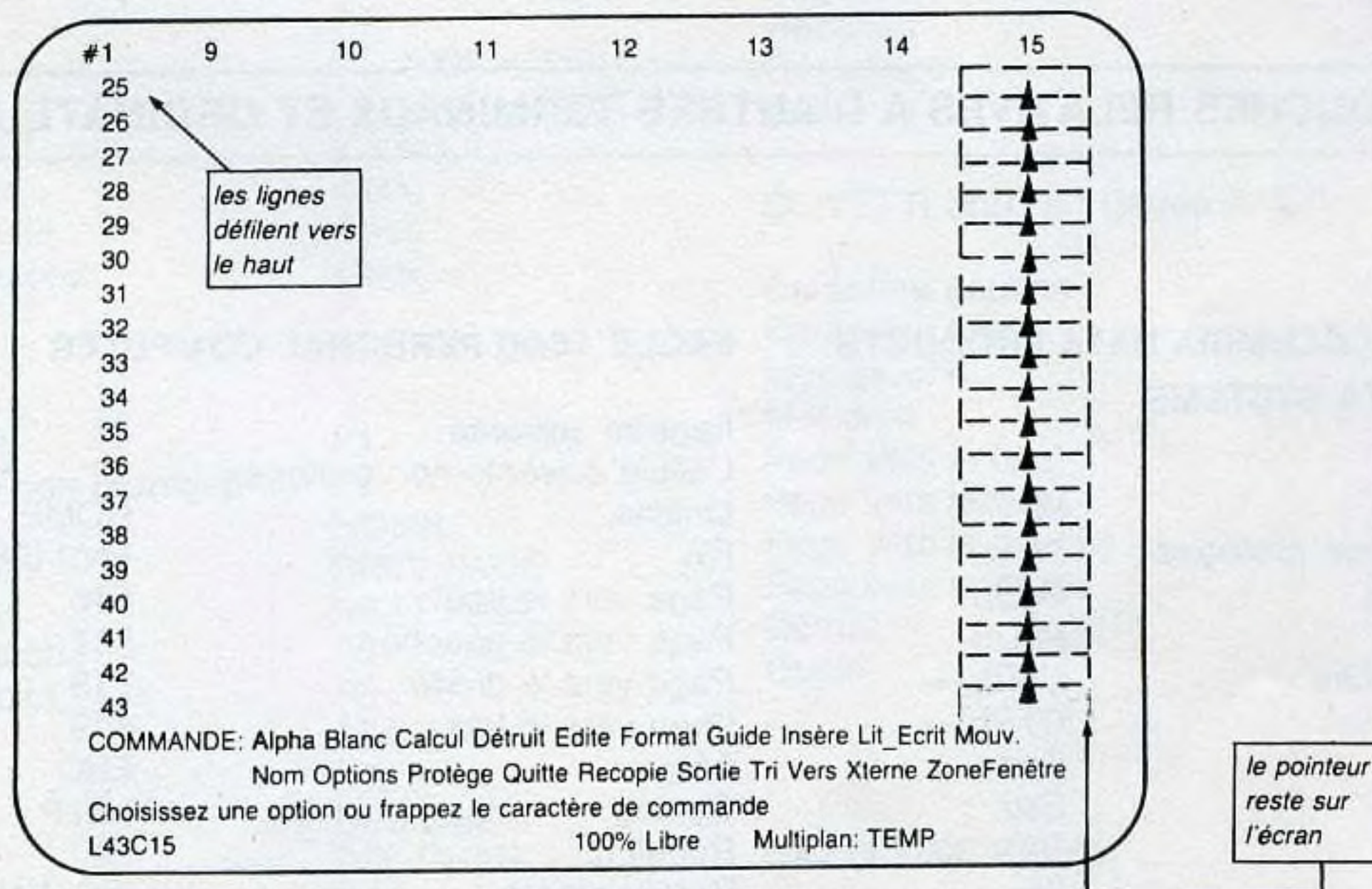
Nota : Pour programmer une cellule désirée, il suffit de la pointer grâce aux flèches de déplacement. Il convient alors d'entrer au clavier l'information souhaitée.

DEFILEMENT DE LA FEUILLE DE CALCUL



VERS LA DROITE

Nous avons vu qu'il était impossible de visualiser toute la feuille de calcul en une seule fois. Il est possible de déplacer le pointeur de cellule vers la droite de l'écran (et inversement). En continuant d'appuyer sur la touche →. Les numéros de colonnes se déplacent vers la gauche. C'est ce qu'on appelle le défilement. En appuyant sur la touche de direction vers la droite jusqu'à la colonne 15, les colonnes 1 à 8 ont disparu de l'écran par la gauche. Vous êtes maintenant sur la deuxième partie droite de la feuille de calcul.



VERS LE BAS

Pour passer vers le bas de la feuille de calcul, le processus est le même que précédemment. Il suffit de faire défiler la feuille à l'aide de la touche ↓.

Nota : Pour revenir rapidement en ligne 1 colonne 1 (L1C1), il suffit d'appuyer sur la touche HOME ou ORIGINE du clavier. Cette touche est parfois représentée par une flèche inclinée vers la gauche.

COMMANDES CLAVIER MULTIPLAN

Déplacement du pointeur de cellule

Vers le haut	CTRL-E
Vers le bas	CTRL-X
Vers la gauche	CTRL-S
Vers la droite	CTRL-D
Fenêtre suivante	CTRL-W
Cellule suivante non protégée	CTRL-F INTERLIGNE

Défilement de la fenêtre

Page vers le haut	CTRL-R CTRL-E
Page vers le bas	CTRL-R CTRL-X
Page vers la gauche	CTRL-R CTRL-S
Page vers la droite	CTRL-R CTRL-D
Origine	CTRL-Q
Fin	CTRL-Z

Sélection et exécution d'une commande

Annule	CTRL-C
Exécution de la commande	RETOUR
Choix de l'élément suivant du menu	BARRE D'ESPACEMENT
Choix de l'élément précédent du menu	RETOUR ARRIERE, CTRL-H
Passage au champ suivant de commande	TAB, CTRL-1
Guide	?
Recalcul	!

Edition des cellules et des commandes

Détruit	DELETE, CTRL-Y
Retour arrière Backspace	CTRL-H
Caractère gauche	CTRL-K
Caractère droit	CTRL-L
Mot gauche	CTRL-O
Mot droit	CTRL-P
Référence	@, à

Nota : Les déplacements du pointeur de cellule peuvent être réalisés de trois façons : par CTRL, par les flèches, soit par une souris si votre machine en est équipée.

TOUCHES RELATIVES A D'AUTRES TERMINAUX ET ORDINATEURS

IBM, COMPAQ, COLUMBIA DATA PRODUCTS et CORONA DATA SYSTEMS

Fenêtre suivante	F6
Cellule suivante non protégée	F5
Page vers le haut	PgUp
Page vers le bas	PgDn
Page vers la gauche	CTRL--←
Page vers la droite	CTRL--→
Fin	End
Annule	Esc
Guide	F10
Recalcul	F8
Caractère gauche	F1
Caractère droit	F2
Mot gauche	F3
Mot droit	F4
Référence	F7

EAGLE 1600 PERSONAL COMPUTER

Fenêtre suivante	F5
Cellule suivante non protégée	F6
Origine	HOME, TOP (F7)
Fin	END (F8)
Page vers le haut	F16
Page vers la gauche	F17
Page vers la droite	F18
Page vers le bas	F19
Annule	ESC
Guide	HELP
Recalcul	F9
Retour arrière	INS, F15
Caractère gauche	F1
Caractère droit	F2
Mot gauche	F3
Mot droit	F4
Référence	F10

WANG PROFESSIONAL COMPUTER

Annule	CANCEL
Exécution de la commande	EXEC
Caractère gauche	FORMAT
Caractère droit	MERGE
Mot gauche	NOTE
Mot droit	STOP
Référence	SRCH
Guide	HELP
Fenêtre suivante	GOTO
Cellule suivante non protégée	COMMAND
Recalcul	MOVE
Origine	HOME
Fin	Touche
	Super/Subscript
Page vers le haut	PREV. INDENT
Page vers le bas	NEXT. PAGE
Page vers la droite	DECTAB
Page vers la gauche	CENTER

GRID COMPASS COMPUTER

Annule	ESC
Détruit	Code-Delete
Caractère gauche	F9, SHIFT - ←
Caractère droit	F10, SHIFT - →
Mot gauche	F7, code - SHIFT - ←
Mot droit	F8, code - SHIFT - →
Guide	Code - ?
Fenêtre suivante	F1
Cellule suivante non protégée	F2
Référence	F3
Recalcul	F4
Origine	F5, code - SHIFT - ↑
Fin	F6, code - SHIFT - ↓
Page vers le haut	Code - ↑
Page vers le bas	Code - ↓
Page vers la droite	Code - →
Page vers la gauche	Code - ←

ALTOS II

Fenêtre suivante	F1
Cellule suivante non protégée	F2
Origine	HOME
Page vers le haut	PREV PAGE
Page vers le bas	NEXT PAGE
Page vers la gauche	F5
Page vers la droite	F6
Annule	ESC
Guide	HELP
Détruit	DEL LINE, DEL CHAR
Caractère gauche	F13
Caractère droit	F14
Mot gauche	F15
Mot droit	F16
Recalcul	F3
Référence	F4

BURROUGHS ET-2000 COMPUTER

Annule	ESC
Caractère gauche	F1
Caractère droit	F2
Mot gauche	F3
Mot droit	F4
Référence	F7
Guide	HELP, F10
Fenêtre suivante	F6
Cellule suivante non protégée	F5
Recalcul	F8
Origine	HOME, CLR
Fin	F9
Page vers le haut	BACK
Page vers le bas	NEXT
Page vers la gauche	6 (pad numérique 6)
Page vers la droite	4 (pad numérique 7)

DYNALOGIC HYPERION

Annule	Esc
Origine	Home
Fin	End
Cellule suivante non protégée	F1
Fenêtre suivante	F2
Caractère gauche	F6
Caractère droit	F7
Mot gauche	F8
Mot droit	F9
Détruit	Del
Page vers le haut	Pg Up
Page vers le bas	Pg Dn
Recalcul	F3
Référence	F4
Guide	F10

OLIVETTI M20 L1 (Mode ANSI)

Caractère gauche	Command-1
Caractère droit	Command-2
Mot gauche	Command-3
Mot droit	Command-4
Page vers le haut	Command-5
Page vers le bas	Command-6
Page vers la gauche	Command-7
Page vers la droite	Command-8
Détruit	Command-9
Guide	Command-0

HAZELTINE ESPRIT

Retour arrière	CTRL-U
Choisit l'élément précédent du menu	CTRL-U
Mot gauche	CTRL-V
Mot droit	CTRL-B

HONEYWELL MICROSYSTEM 6/10

Annule	Esc
Détruit	Delete
Caractère gauche	F1
Caractère droit	F2
Mot gauche	F3
Mot droit	F4
Référence	F7
Guide	F10
Fenêtre suivante	F6
Cellule suivante non protégée	F5
Recalcul	F8
Origine	Home
Fin	End
Page vers le haut	Page Up
Page vers le bas	Page Down
Page vers la gauche	F11
Page vers la droite	F12

**ADM-3A, ADM-5
TELEVIDEO 910/912/920**

Détruit	CTRL-U
Choisit l'élément précédent du menu	CTRL-U
Caractère gauche	CTRL-V
Caractère droit	CTRL-B

TELEVIDEO 910 +

Détruit	CTRL-B
Choisit l'élément précédent du menu	CTRL-B
Caractère gauche	CTRL-U
Caractère droit	CTRL-

TELEVIDEO 925/950

Cellule suivante non protégée	F6
Page vers le haut	F8
Page vers le bas	F7
Page vers la gauche	F9
Page vers la droite	F10
Annule	F1
Détruit	CTRL-B
Choisit l'élément précédent du menu	CTRL-B
Guide	F11
Caractère gauche	F2, CTRL-U
Caractère droit	F3, CTRL-J
Mot gauche	F4
Mot droit	F5

DEC RAINBOW 100 PERSONAL COMPUTER

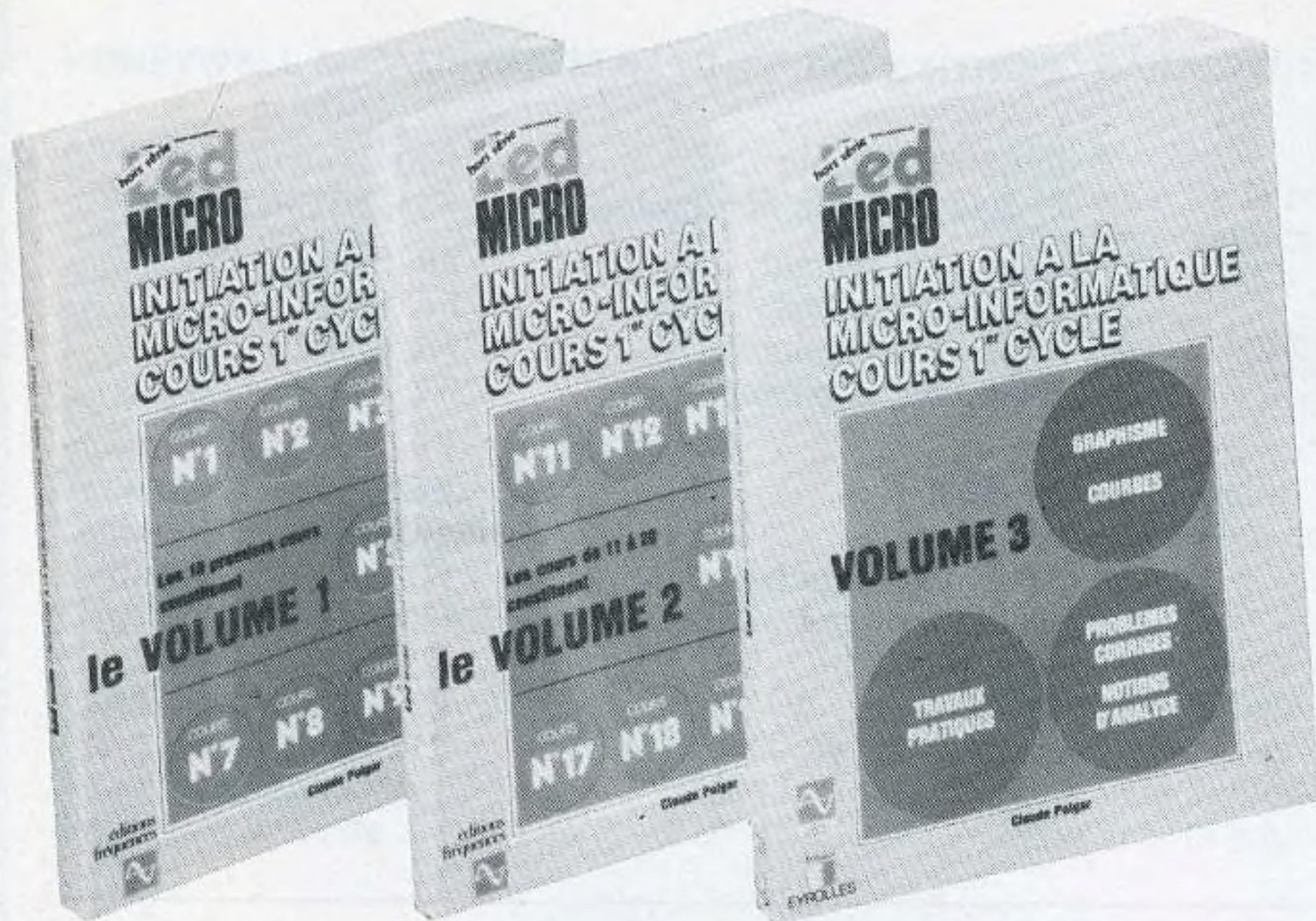
Annule	CANCEL
Exécution de la commande	DO
Caractère gauche	PF1
Caractère droit	PF2
Mot gauche	PF3
Mot droit	PF4
Référence	F17
Guide	HELP
Fenêtre suivante	F19
Cellule suivante non protégée	FIND
Recalcul	F18
Origine	MAIN SCREEN
Fin	EXIT
Page vers le haut	PREV SCREEN
Page vers le bas	NEXT SCREEN
Page vers la droite	INSERT HERE
Page vers la gauche	REMOVE

HEATH/ZENITH H19

Utilisez la touche majuscules (SHIFT) pour entrer des nombres à partir du pavé (ou pad) numérique du clavier.	
Caractère gauche	IC (pad numérique 7)
Caractère droit	DC (pad numérique 9)
Mot gauche	IL (pad numérique 1)
Mot droit	DL (pad numérique 3)
Page vers le haut	f1
Page vers le bas	f2
Page vers la gauche	f3
Page vers la droite	f4
Fin	f5
Annule	ERASE
Guide	ROUGE
Recalcul	BLEU
Référence	BLANC

TEXAS INSTRUMENTS PROFESSIONAL COMPUTER

Fenêtre suivante	F6
Cellule suivante non protégée	F5
Page vers le haut	ALT + ↑
Page vers le bas	ALT + ↓
Page vers la gauche	ALT + ←
Page vers la droite	ALT + →
Fin	F9
Annule	ESC
Guide	F10
Recalcul	F8
Caractère gauche	F1
Caractère droit	F2
Mot gauche	F3
Mot droit	F4
Référence	F7



**Le cours
d'initiation
le plus
complet
+ de 700 pages**

Non, on ne s'initie pas à la micro-informatique en 5 leçons !

Si vous croyez au Père Noël vous pouvez espérer apprendre l'Informatique en lisant les innombrables «Cours de BASIC pour débutants» qui ont poussé comme des champignons dans les années 1980. Votre ordinateur risque de finir ses jours au-dessus de votre armoire.

Mais si vous voulez vraiment apprendre à programmer il faut avoir le courage de commencer par A pour arriver à Z. Programmer est un loisir intelligent et peut devenir un métier passionnant, mais l'étude de la programmation nécessite un minimum de travail et de méthode.

Etre sérieux – c'est le pari que fit la revue LED-MICRO en publiant à partir de 1985 les 20 premiers cours de C. Polgar. Plus de 40 000 lecteurs les ont suivis. Ce succès nous a conduit à demander à C. Polgar de remettre son cours à jour et de le compléter. Le résultat : un ouvrage épais (3 tomes, plus de 700 pages format 21 x 27), permettant d'acquérir agréablement des connaissances solides.



Diffusion auprès des libraires assurée exclusivement par les Editions Eyrolles.

Initiation à la micro-informatique C. Polgar

Bon de commande à retourner aux Editions Fréquences
1, boulevard Ney 75018 Paris.

Je désire recevoir le tome 1 140 F (130 F + 10 F de frais de port)
le tome 2 140 F (130 F + 10 F de frais de port)
le tome 3 200 F (190 F + 10 F de frais de port)

Ci-joint mon règlement par :

CCP Chèque bancaire Mandat

Nom

Prénom

Adresse

Code postal Ville

Une seule
parmi près de 600 lettres
de lecteurs :

J'enseigne les mathématiques dans une Université de Sciences Humaines et j'ai été amenée, alors que n'avais moi-même reçu aucune formation à la micro-informatique, à initier des étudiants de 1^{re} année de Mathématiques et Sciences Sociales (MASS) à la programmation en S-BASIC (sur Goupil-3), dans le but de faire avec eux de l'analyse numérique élémentaire. Ce que j'ai fait, tant bien que mal, cette année, en collaboration avec deux autres collègues. Nous sommes conscientes d'avoir commis un certain nombre d'erreurs pédagogiques et nous souhaitons tenter d'y remédier l'an prochain.
J'ai découvert votre revue tout récemment, alors que j'arrivais quasiment au bout de mon enseignement. J'ai été très sensible à votre démarche pédagogique et je me sens personnellement tout à fait en accord avec votre manière de procéder. Je me suis procurée l'ensemble des n^{os} de la revue et me permettrai de puiser dans votre cours certains exemples ou certaines façons de présenter les choses l'an prochain. Donc merci à vous...
C.L. St Cloud, le 22/5/85

Initiation à la Micro-Informatique 1^{er} Cycle Tome 3 (enfin paru !)

3.16 (Suite et fin) L'affichage

- ★ Etude des instructions permettant d'effectuer des présentations « évoluées » : PRINT TAB - PRINT USING - LOCATE - COLOR en mode texte.
- ★ Présentation en tableaux de toutes sortes grâce à la pratique des opérateurs MODULO et DIVISION ENTIERE.
- ★ Beaucoup de programmes utilisent des assemblages de ces instructions et opérateurs... dont la combinaison n'est pas toujours facile.

3.17 Compléments

- ★ Etude des dernières instructions, fonctions et variables du cycle 1 : FILES, KILL, AUTO, ON ERROR GOTO, RESUME, ERR, ERL, DELETE, EDIT, RENUM TRON, TROFF, STOP, CONT, KEY ON, KEY OFF, FIX, BEEP.
- ★ Compléments de cycle 1 qui sont maintenant accessibles aux élèves : sur la précision et les erreurs dues à l'arrondi, sur la sélection, les boucles.

3.18 Graphisme

- ★ Une étude complète et détaillée sur les instructions graphiques en haute résolution : SCREEN, PSET, PRESET, STEP, LINE, CIRCLE, COLOR, POINT, PAINT, sans éluder aucune des difficultés et « pièges » classiques : l'incrustation de texte dans le dessin, les « bavures » dues au PAINT mal utilisé.
- ★ Une étude détaillée du langage graphique DRAW, avec ses subtilités et ses pièges (sous-chaînes X, paramètres variables dans le DRAW, etc.).
- ★ De nombreux exercices avec leurs solutions (80) et leurs illustrations sur des photos d'écran en couleur (48 photos).

3.19. Dessin des courbes

- ★ Un chapitre séparé du graphisme général (chapitre 3.18) de façon à ce que les « non matheux » puissent le sauter sans remords : ils ne seront pas punis !
- ★ Pour les matheux : une excellente révision et illustration des courbes de toutes sortes : $Y = f(x)$, courbes paramétrées, courbes en coordonnées polaires, avec des exemples utiles : courbes d'amortissement, astroïde, cardioïde, décomposition d'une fonction périodique par une série de Fourier.

3.20. Révision générale

- ★ L'enchaînement des notions selon l'ordre « pédagogique » qui a été utilisé jusqu'ici est bien différent de l'ordre « logique ». Autant qu'un cours d'anglais suit un ordre différent de celui (plus logique !) d'une grammaire anglaise.
- ★ Tout ce qui a été enseigné jusqu'ici résumé en 30 pages. Une référence pour retrouver la notion dont on a besoin à travers le cours et ses exercices. Mais aussi une réflexion sur la structure d'un langage informatique, d'où une préparation à la lecture des cours de PASCAL (par exemple !).

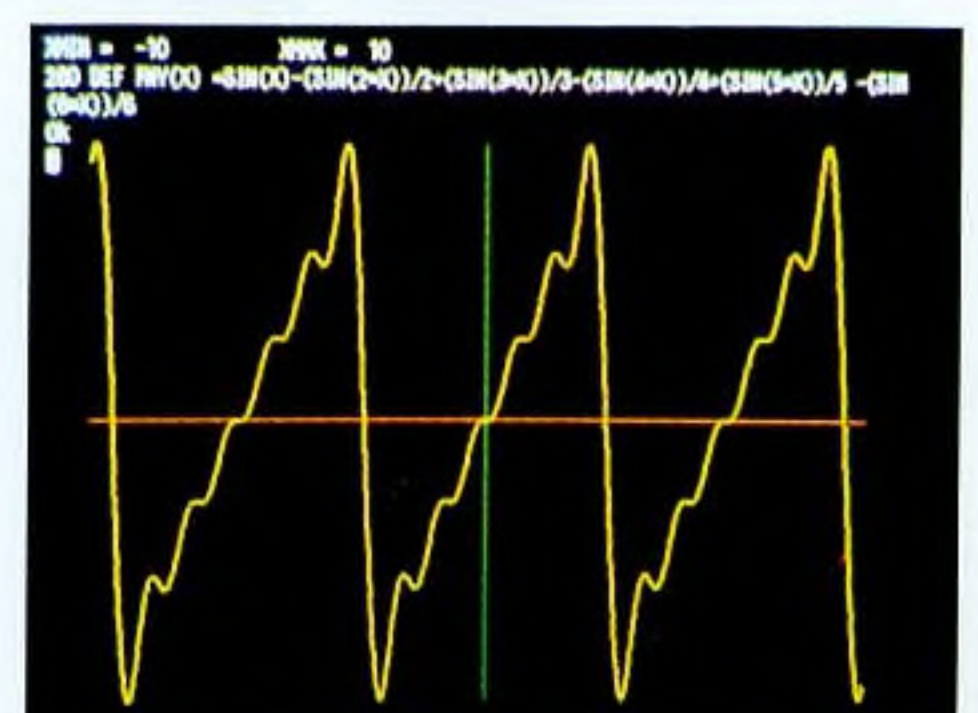
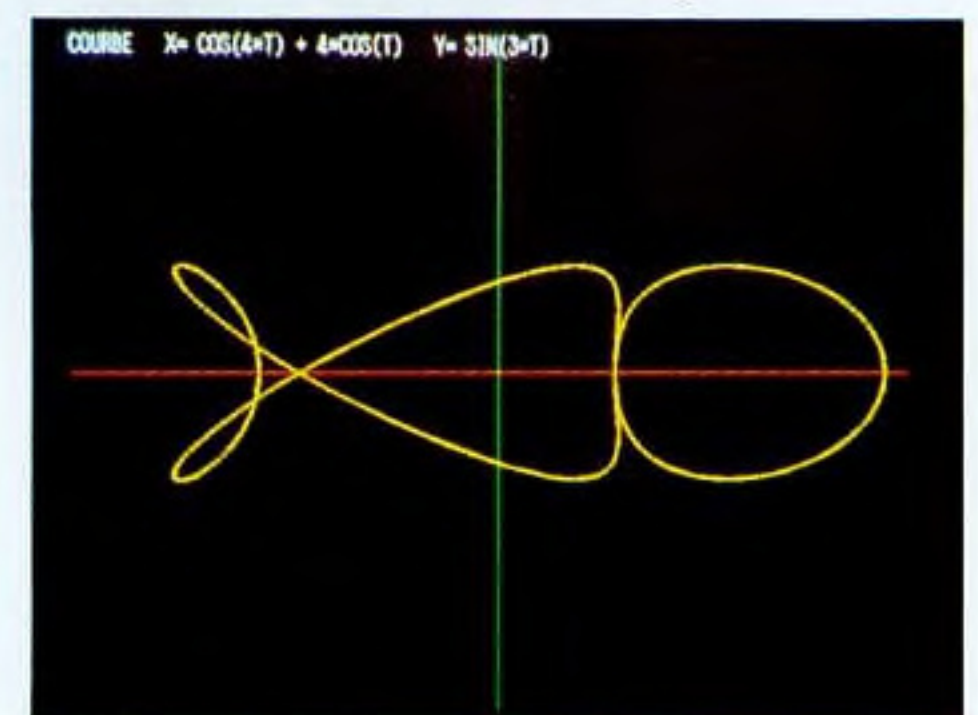
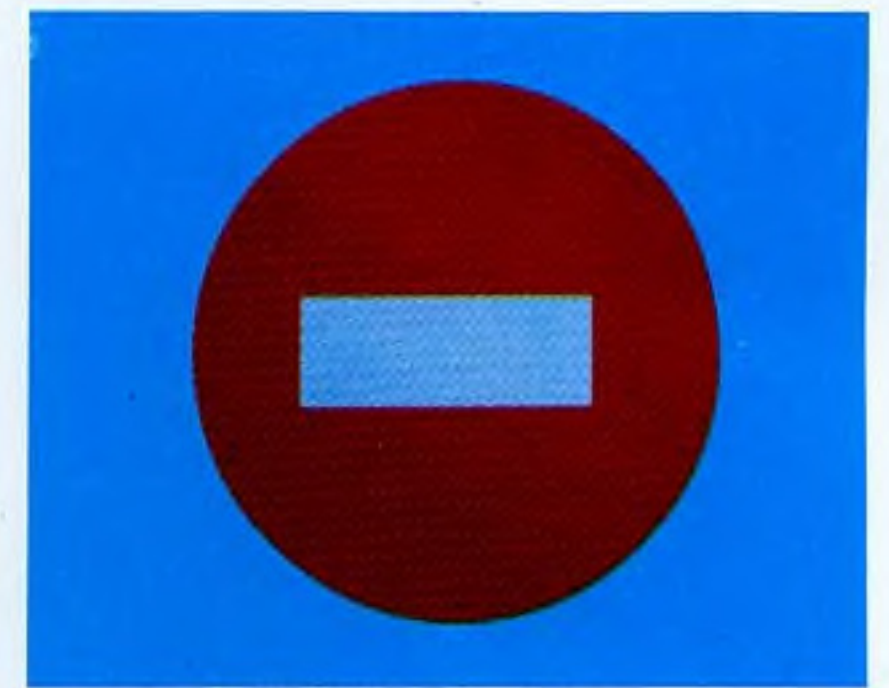
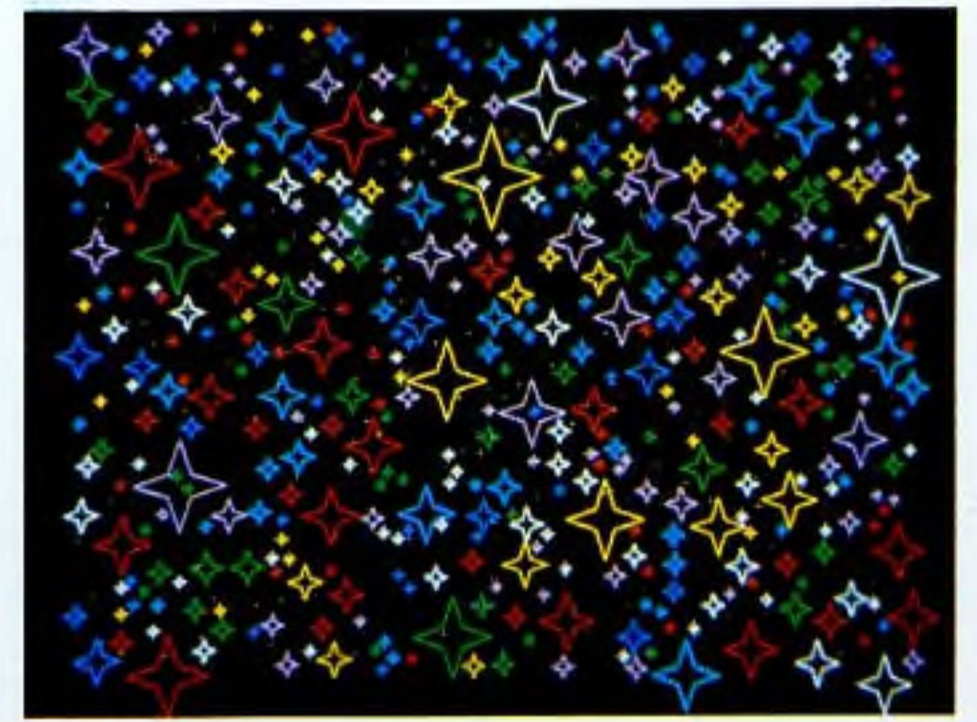
3.21. Techniques de mise au point

- ★ Les outils de base : étude des éditeurs de texte, connaissance et interprétation des messages d'erreur.
- ★ Comment rechercher et corriger ses erreurs.
- ★ La représentation du dialogue homme-machine, pour noter l'expérience que vous acquérez par la pratique.

3.22. Problèmes de synthèse - Notions d'analyse

C'est à la fois la conclusion, la partie la plus originale et la plus utile de ce cours. L'auteur ne se contente pas de fournir une liste de problèmes avec leur solution : il se met à la place du programmeur débutant en essayant de décortiquer le « processus de réflexion » qui fait passer de l'énoncé d'un problème à sa solution : une initiation pratique à l'analyse.

1 livre broché de 248 pages pages 21 x 27, dont 8 pages en couleur



NOUVEAU
Offre de lancement

METTEZ LE MONDE DES ORDINATEURS à portée de votre main

Initiation à l'informatique

**EXAMINEZ
GRATUITEMENT
CE LIVRE UNIQUE
PENDANT
10 JOURS!**

Des textes
simples,
un langage
quotidien

Des illustrations
parlantes

informatique complète qui répondra à toutes les questions que vous vous posez.

Parmi les volumes de cette collection :
Le Logiciel • Les Images électroniques
• Les Entrées/Sorties • La Protection des données • Les Langages de programmation, etc.

COMPRENEZ SANS PEINE LE LANGAGE ET LE FONCTIONNEMENT DES ORDINATEURS

LE MONDE DES ORDINATEURS, nouvelle collection publiée par les Éditions TIME-LIFE vous permet de comprendre comment fonctionnent ces ordinateurs qui vous entourent, chez vous, dans votre cuisine, dans votre voiture ou à votre travail...

Grâce à *Initiation à l'informatique*, premier volume de cette collection unique, grâce à son langage clair, à ses photographies parlantes, à ses schémas détaillés :

- Vous suivrez la fascinante évolution des ordinateurs depuis leur naissance ;
- Vous apprendrez sans peine le langage des ordinateurs ;

- Vous examinerez le fonctionnement intérieur d'une "puce" ;
- Vous observerez la circulation des informations dans toutes les parties de l'ordinateur.

DÉCOUVREZ TOUS LES ÉLÉMENTS D'UN MONDE FASCINANT

A partir du premier volume *Initiation à l'informatique*, avec **LE MONDE DES ORDINATEURS**, vous vous constituerez une collection

DE SUPERBES OUVRAGES

- Grand format : 23,5 x 28 cm
- Couverture rigide argentée
 - Environ 128 pages
 - Des schémas clairs
- Des photographies couleurs spectaculaires

EN CADEAU POUR VOUS

ce boulier inventé il y a 1500 ans

Avec votre volume *Initiation à l'informatique* en examen gratuit, vous recevrez, en cadeau, cet authentique boulier (24,7 x 12,3 x 2,1 cm d'épaisseur) reproduction fidèle de la première calculatrice du monde.

Le mode d'emploi qui y sera joint vous permettra de l'utiliser comme le firent nos lointains ancêtres, avec une déconcertante facilité.



EDITIONS
**TIME
LIFE**

BON D'EXAMEN GRATUIT

à retourner sous enveloppe affranchie
à Time-Life International B.P. 83-08 - 75362 Paris cedex 08

A RETOURNER

OUI, veuillez accepter ma demande de consultation du volume *Initiation à l'informatique* et envoyez-le moi pour un examen gratuit de 10 jours, en même temps que mon cadeau : le boulier inventé il y a 1500 ans. Si je décide de garder *Initiation à l'informatique*, je réglerai la facture qui accompagne ce volume, soit 135 FF + 14 FF de frais d'envoi. Vous m'enverrez alors les volumes suivants de la collection **LE MONDE DES ORDINATEURS**, à raison d'un livre toutes les six semaines environ, toujours pour un examen gratuit de 10 jours. Je ne suis nullement tenu d'acheter un nombre minimum de livres et je suis en droit d'arrêter ma collection à tout moment en vous le faisant savoir par écrit. Si le volume *Initiation à l'informatique* ne répond pas exactement à mon attente, je vous le retournerai dans les 10 jours suivant sa réception. Vous cesserez toute autre expédition de cette collection et je ne vous devrai rien.

INSCRIVEZ EN MAJUSCULES VOS NOM ET ADRESSE

F2 ZAA V7

NOM _____ Prénom _____

N° _____ Rue _____ Code Postal _____

Ville _____

Signature _____