

LOISIRS TECHNIQUES D'AUJOURD'HUI

hors série

Leed

MICRO

PROGRAMMATION

COURS 2^{ème} CYCLE

COURS
N°36
Suite
2^e cycle

N°16

COURS DE PASCAL
les commandes

**COURS DE PROGRAM-
MATION**
APPROFONDIE :
données
graphiques

LA LIGNE MAJOR
DE CHEZ SAARI



ISSN 0757-6889

M 1988 - 36 - 18,00 F



3791988018008 00360

VOYAGE AU COEUR DES MICRO-ORDINATEURS

dans la
COLLECTION
«ETUDES»
aux
éditions
fréquences



une véritable schémathèque

- 128 pages
 - 101 schémas
 - 34 tableaux
- Prix : 150 F

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

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

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

En vente chez votre libraire et aux Editions Fréquences

BON DE COMMANDE

Je désire recevoir l'ouvrage **L'électronique des micro-ordinateurs** au prix de **160 F** (150 F + 10 F de port).

Nom

Adresse

.....

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

Règlement ci-joint :

Par chèque bancaire par chèque postal par mandat

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

LO...QUES D'AUJOURD'HUI

hors série

Led

MICRO

PROGRAMMATION COURS 2^e CYCLE

Société éditrice :
Editions 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 Coblentz
 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



JANVIER 87

COURS DE PASCAL
 de la page 5 à la page 21

- Rappel des mots réservés disponibles en Pascal standard. . p. 6
 - Les commandes
 - Les fonctions intégrées
 - Les opérateurs
- Description détaillée. p. 6
 - Les commandes
- Conclusion sur les commandes standard du Pascal. p. 14
- Introduction aux sous-programmes. p. 14
 - Notions de bases
 - Deux sortes de sous-programmes
 - Les procédures
 - Les fonctions, ou fonctions procédurales
 - Description de l'utilisation d'une procédure

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

**COURS DE PROGRAMMATION
 APPROFONDIE**

de la page 22 à la page 27

- Structuration des données graphiques. p. 23
 - Représentation d'un solide
 - Les arêtes
 - Les faces
 - Les volumes
- Visualisation 3D. p. 27
 - Représentation 2D

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

DIALOGUE AVEC NOS LECTEURS
 de la page 28 à la page 31

C'EST ARRIVE DEMAIN
 de la page 32 à la page 34

QUOI DE NEUF EN 87 ?
 pages 38-39

DES PROGICIELS PRO
 La ligne Major de chez Saari
 de la page 41 à la page 48

- La gestion commerciale Saari. . . p. 42
- La comptabilité. p. 42
- La paie. p. 42
- La gestion commerciale ligne Major. p. 43
 - Gestion des stocks
 - Gestion des clients
 - Facturation
 - Utilitaires programme
 - Utilitaires système

Charles-Henry Delaleu

NOTRE COUVERTURE : Un compatible AT, le PCA de Tandon. La guerre des prix continue, exemple Tandon fabricant US propose des matériels US 30 % moins cher.

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.

Diffusion auprès des libraires assurée exclusivement par les Editions Eyrolles.

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.



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 Coblentz
Patrick Gueneau

Deux gros morceaux ce mois-ci dont la difficulté n'est pas à la hauteur de la taille. La description exhaustive des commandes Pascal vous permettra de découvrir une bonne partie des avantages de la programmation dans ce langage. Le deuxième pavé concernera les sous-programmes et les passages de variables qu'ils requièrent.

COURS N° 7

PLAN DU COURS

1. Introduction
2. Rappel des mots réservés disponibles en pascal standard
 - 2.1. Les commandes
 - 2.2. Les fonctions intégrées
 - 2.3. Les opérateurs
3. Description détaillée
 - 3.1. Les commandes
 - 3.1.1. Les opérateurs portant sur les booléens
 - 3.1.2. Les tests
 - 3.1.3. Les boucles
 - 3.1.4. Les commandes à utilisation mathématique
4. Conclusion sur les commandes standard du pascal
5. Introduction aux sous-programmes
 - 5.1. Notions de bases
 - 5.2. Deux sortes de sous-programmes
 - 5.3. Les procédures
 - 5.4. Les fonctions, ou fonctions procédurales
 - 5.5. Description de l'utilisation d'une procédure
 - 5.5.1. Règles générales
6. Conclusion très provisoire

1. INTRODUCTION

Le mois dernier, nous avons effectué un rappel de toutes les notions déjà vues. Il est temps de décrire les commandes principales, maintenant que les structures et la syntaxe sont connues. Nous allons donc ce mois-ci donner une longue, mais, nous l'espérons, pas trop fastidieuse, liste des commandes les plus courantes du langage, avec leur description détaillée. La dernière partie de ce cours est le début de la partie vraiment intéressante des possibilités du Pascal, avec une introduction à l'utilisation des sous-programmes.

2. RAPPEL DES MOTS RESERVES DISPONIBLES EN PASCAL STANDARD

2.1. Les commandes

AND	ARRAY	BEGIN	CASE	CONST	DIV
DO	DOWNTO	ELSE	END	EXTERN	FILE
FOR	FUNCTION	GOTO	IF	IN	LABEL
MOD	NIL	NOT	OF	OR	PACKED
PROCEDURE	PROGRAM	RECORD	REPEAT	SET	THEN
TO	TYPE	UNTIL	VAR	WHILE	WITH

2.2. Les fonctions intégrées

ABS	ARCTAN	CHR	COS	EOLN	EOF
EXP	LN	ODD	ORD	PRED	ROUND
SIN	SQR	SQRT	SUCC	TRUNC	

2.3. Les opérateurs

+	-	*	/	DIV	MOD
AND	OR	NOT	<	>	=
<=	>=	< >	IN	TRUE	FALSE
MAXINT	NIL				

3. DESCRIPTION DETAILLEE

Nous ne reviendrons pas sur les mots connus, comma par exemple PROGRAM, BEGIN,...

Par contre, nous essaierons d'être aussi précis et complet que possible sur les autres.

3.1. Les commandes

3.1.1. Les opérateurs portant sur les booléens

AND : Liaison de plusieurs opérateurs logiques. Le résultat de cette liaison est TRUE si, et seulement si, les deux termes entourant le AND sont TRUE.

Ex. : $N = 2$

$M = 4$

$(M > N) \text{ AND } (N > 0)$ est vrai

Pour résumer les résultats, voici la table de vérité du mot :

AND	TRUE	FALSE
TRUE	TRUE	FALSE
FALSE	FALSE	FALSE

Un exemple programmé :

```

program et;
  var
    m, n, o, p : integer;
    x, y, z, t : real;
  begin
    x := 2.5;
    y := 1.23;
    z := 4.9876;
    t := 0.1;
    m := 2;
    n := 1;
    o := 4;
    p := 0;
    if (x > m) and (t < z) and (o = p) then
      writeln('surprenant, car o est different de p');
    if (x > m) and (t < z) and (o <> p) then
      writeln('voila qui est mieux');
  end.

```

OR : Liaison de plusieurs opérateurs logiques. Cette fois-ci, le fait que l'un des deux opérateurs soit TRUE est suffisant pour rendre le résultat TRUE. OR signifie «OU», ce qui explique la définition.

Ex. : N = 2
M = 4

(M > N) OR (N = 0) est vrai car le premier terme est vrai

La table de vérité du mot :

OR	TRUE	FALSE
TRUE	TRUE	TRUE
FALSE	TRUE	FALSE

Un programme :

```

program ou;
  var
    m, n, o, p : integer;
    x, y, z, t : real;
  begin
    x := 2.5; y := 1.23;
    z := 4.9876; t := 0.1;
    m := 2; n := 1;
    o := 4; p := 0;
    if (x > m) or (t = z) or (o = p) then
      writeln('oui car x > m suffit');
    if (x > m) and (t < z) and (o <> p) then
      writeln('ici aussi, mais de plus dans ce cas, les autres sont vrais aussi');
  end.

```

NOT : NOT sert à nier un test logique. Si le test avait été vrai, NOT le rend faux, et réciproquement. Par exemple :

```
n = 2
n > 1 est vrai, donc
NOT (n > 1) est faux (FALSE)
```

La table de vérité est très simple :

NOT	TRUE	FALSE
	FALSE	TRUE

Un exemple programmé :

```
program non;
var
  m, n, o, p : integer;
  x, y, z, t : real;
begin
  x := 2.5;
  m := 2;
  if (x > m) then
    writeln('oui');
  if not (x = m) then
    writeln('ici aussi');
end.
```

Rappelons que les mots TRUE et FALSE représentent des constantes booléennes. Elles sont prédéfinies en standard, et leurs noms sont des mots réservés. Quelques précautions à prendre avec les opérateurs booléens : en général, il n'est pas possible d'écrire les tests dans n'importe quel ordre. Voici un programme qui illustre ceci :

```
program Tout_sauf_cela;
var
  m, n, q, p : integer;
  x, y, z, t : real;

begin
  x := 2.5; y := 1.23;
  z := 4.9876; t := 0.1;
  m := 2; n := 1; q := 4; p := 0;

  if (x > m) and (z <> t) or (y = q) and (n < m) then
    writeln('que penser de cette expression');

  if ((x > m) and (z <> t)) or ((y = q) and (n < m)) then
    writeln('et de celle-ci');

  if (((x > m) and (z <> t)) or (y = q)) and (n < m) then
    writeln('tout se complique');

  if ((x > m) and ((z <> t) or (y = q))) and (n < m) then
    writeln('peut-etre vouliez-vous cela');
```



```
if ((x > m) and (z <> t)) or ((y = q) and (n < m)) then
  writeln('ou cela');
```

```
if (x > m) and ((z <> t) or ((y = q) and (n < m))) then
  writeln('ou encore ceci');
```

```
end.
```

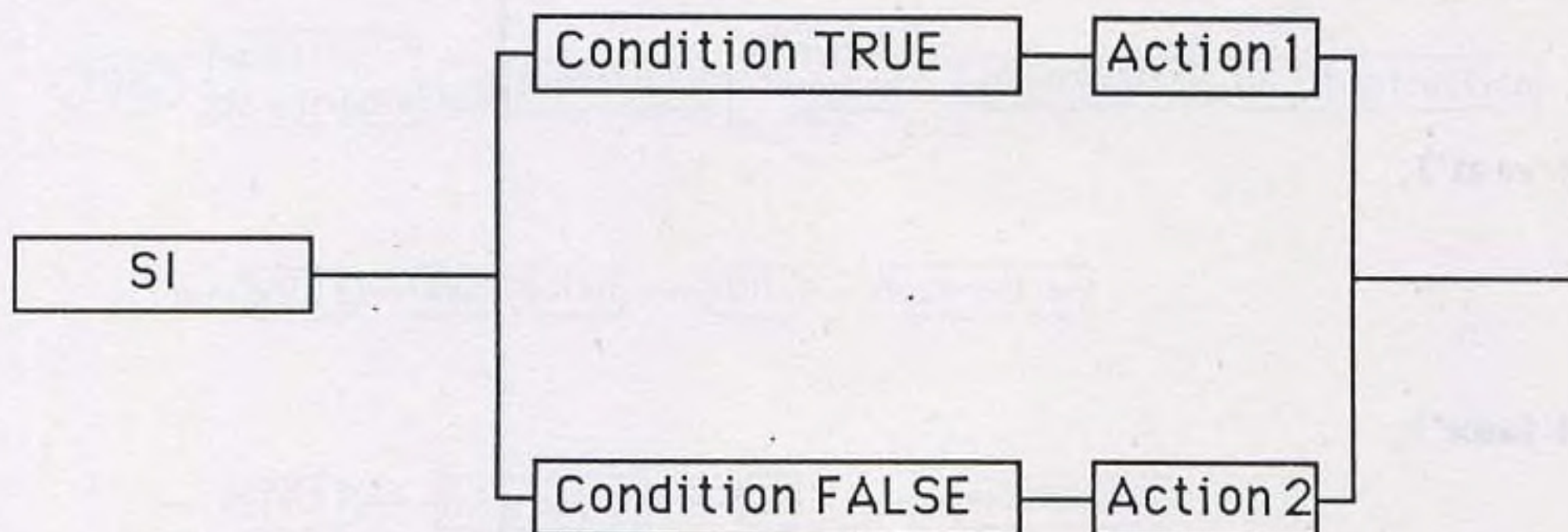
Il faut donc être très vigilant, car les défauts de parenthèses sont parmi les erreurs les plus délicates à détecter.

3.1.2. Les tests

Il n'est sans doute pas nécessaire de revenir sur le test classique, du type :

```
IF condition THEN action1 ELSE action2.
```

Lorsque la condition est TRUE l'action1 est effectuée, sinon, l'action2 est exécutée.



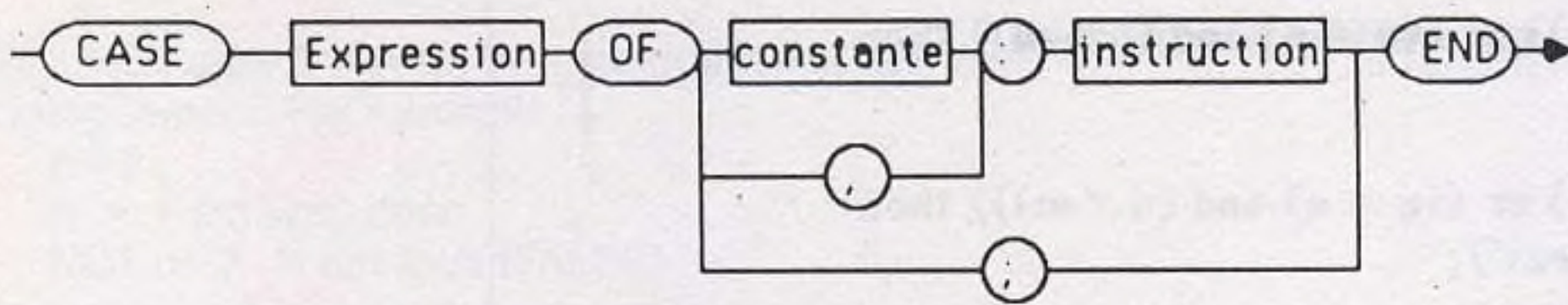
Cette façon de tester des conditions est bien connue, mais une autre est sans doute nouvelle pour vous. Voici le problème : supposons que vous ayez à tester une condition sur une série de valeurs et que, pour chaque valeur, une action différente soit souhaitée. Par exemple, vous testez des nombres, et s'ils valent 0, vous faites une chose, s'ils valent 1, une autre... Comment faire proprement un tel programme ? Vous connaissez bien sûr une méthode, à base de IF THEN ELSE imbriqués :

```
program Comment_faire;
var
  m : integer;

begin
  if m = 0 then
    writeln('m=0')
  else if m = 1 then
    writeln('m=1')
  else if m = 2 then
    writeln('m=2');
end.
```

Il est clair que si vous avez 100 tests imbriqués, le programme devient illisible, et un peu long. Pour éviter cela, une commande rend tout très simple, il s'agit de CASE.

La commande CASE, dite commande de sélection multiple, permet de stocker différentes possibilités, et de les traiter quand l'occasion se présente. Voici sa forme générale :



Cela signifie que vous pourrez avoir par exemple :

```

program cas;
var
  n : boolean;
begin
  n := true;
  repeat
    case n of
      true :
        begin
          writeln(n, ' est vrai');
          n := false;
        end;
      false :
        begin
          writeln(n, ' est faux');
          n := true;
        end;
    end;
  until (n = true);
end.
  
```

Cette commande est très commode, car une fois de plus en Pascal, tout est rendu simple et clair. Ainsi, on peut avoir :

```

program autre_cas;
var
  n : integer;
begin
  repeat
    case n of
      1 :
        begin
          writeln(n, ' = 1');
          n:=n+1;
        end;
      2 :
        begin
          writeln(n, ' = 2');
          n:=n+1;
        end;
      3 :
        begin
          writeln(n, ' = 3');
          n:=n+1;
        end;
    end;
  until n > 10;
end;
  
```

```

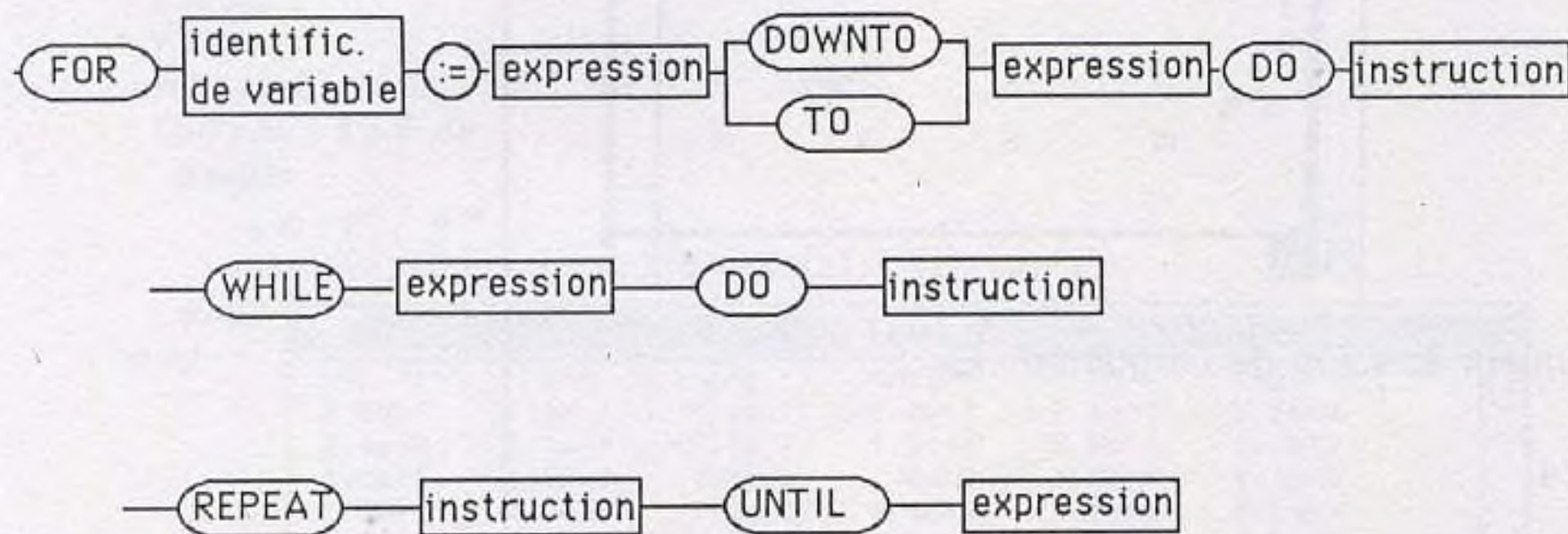
4 :
begin
writeln(n, ' = 4');
n:=n+1;
end;
end;
until (n = 5);
end.

```

Aussi artificiel que soit ce programme, il illustre bien notre propos.

3.1.3. Les boucles

Les boucles en Pascal sont traitées selon trois schémas différents. Il y a la traditionnelle boucle FOR... DO, qui est proche du FOR... NEXT du Basic. Il y a aussi les boucles REPEAT... UNTIL, et WHILE... DO. Les formes de ces boucles sont :



La principale différence entre ces deux dernières boucles réside dans la position du test réalisé pour recommencer la boucle ou sortir. Dans la boucle REPEAT... UNTIL, le test est réalisé en sortie de boucle, ce qui signifie que si la condition est fautive avant, la boucle est effectuée une fois quand même, alors que ce n'est pas le cas pour la boucle WHILE DO, puisque le test est réalisé avant l'entrée. Ceci rend leur utilisation plus adaptée à certains cas ou à d'autres.

Enfin, un détail : dans le cas de la boucle utilisant FOR, TO sert à indiquer que l'incrément de boucle est positif, DOWNTO indique qu'il est négatif :

```
for i : = 1 to 1000 do
```

```
for j : = 1000 downto 1 do
```

Notons que le problème n'existe pas avec les autres types de boucles puisque dans celles-ci, c'est le programmeur qui définit explicitement la modification du coefficient permettant le test de sortie. Contrairement à la boucle FOR, où le coefficient est toujours entier (INTEGER), dans les deux autres, il est possible de tester n'importe quel type de variable (attention, elle doit bien sûr être déclarée). Il est d'ailleurs assez facile de se tromper dans la modification de la variable de contrôle, et de ne jamais sortir de ce type de boucles. A vous de faire très attention.

3.1.4. Les commandes à utilisation mathématique

MOD : donne le reste de la division entière de deux nombres. Ex. :

$$7 \text{ MOD } 3 = 1 \quad |$$

DIV : donne le quotient de la division entière de deux nombres. Ex. :

$$7 \text{ DIV } 3 = 2 \quad |$$

Remarquons que ces deux commandes sont liées par une relation immédiate :

$$a = (a \text{ DIV } b) * b + (a \text{ MOD } b)$$

On retrouve bien là la notion de quotient et de reste. Un exemple :

```

File Edit Search Run Windows
Untitled Text
program entiere;
var
  i, n, m : integer;
begin
  for i := 1 to 10 do
  begin
    readln(n, m);
    writeln(n, m, n div m, n mod m);
  end;
end.
10 4
100 12
9 3
14 50
17 3
8 7
23456 123
2
1
3
2
15 5
4 2 2
12 8 4
9 3 3 0
14 50 0 14
17 3 5 2
8 7 1 1
23456 123 190 86
2 1 2 0
3 2 1 1
15 5 3 0
  
```

ABS : ABS permet de trouver la valeur absolue de l'argument. Ex. :

$ABS(-7) = 7$
 $ABS(124) = 124$
 $ABS(0) = 0$

Programme :

```

Untitled Text
program ex;
var
  z : real;
  i : integer;
begin
  for i := 1 to 4 do
  begin
    readln(z);
    writeln(abs(z));
  end;
end.
-17
1.7e+1
-1.414
1.4e+0
0
0.0e+0
6
6.0e+0
  
```

TRUNC et **ROUND** : Ces deux fonctions permettent de trouver pour la première la partie entière, pour la seconde l'entier le plus proche du nombre fourni en argument. Un programme pour bien voir :

```

Untitled Text
program ex;
var
  z : real;
  i : integer;
begin
  for i := 1 to 6 do
  begin
    readln(z);
    writeln(trunc(z), round(z));
  end;
end.
-12.8
-12 -13
-12.1
-12 -12
-0.08
0 0
0.08
0 0
0.9
0 1
-1.50001
-1 -2
  
```

ARCTAN, COS, SIN, EXP et LOG : Ces fonctions calculent les valeurs des fonctions mathématiques correspondantes. Pour certains environnements, (UCSD), il est nécessaire de préciser que le programme les utilisera, par la déclaration :

USES TRANSCEND ;

qui charge la bibliothèque des fonctions transcendantales (les fonctions transcendantales sont les fonctions du type fonctions trigonométriques, logarithmiques...). Cette bibliothèque n'est pas en ligne en permanence pour ne pas charger inutilement l'exécution. Pour la plupart des autres Pascal, l'utilisation de ces fonctions se fait automatiquement. Un petit programme pour voir ce que sont ces fonctions :

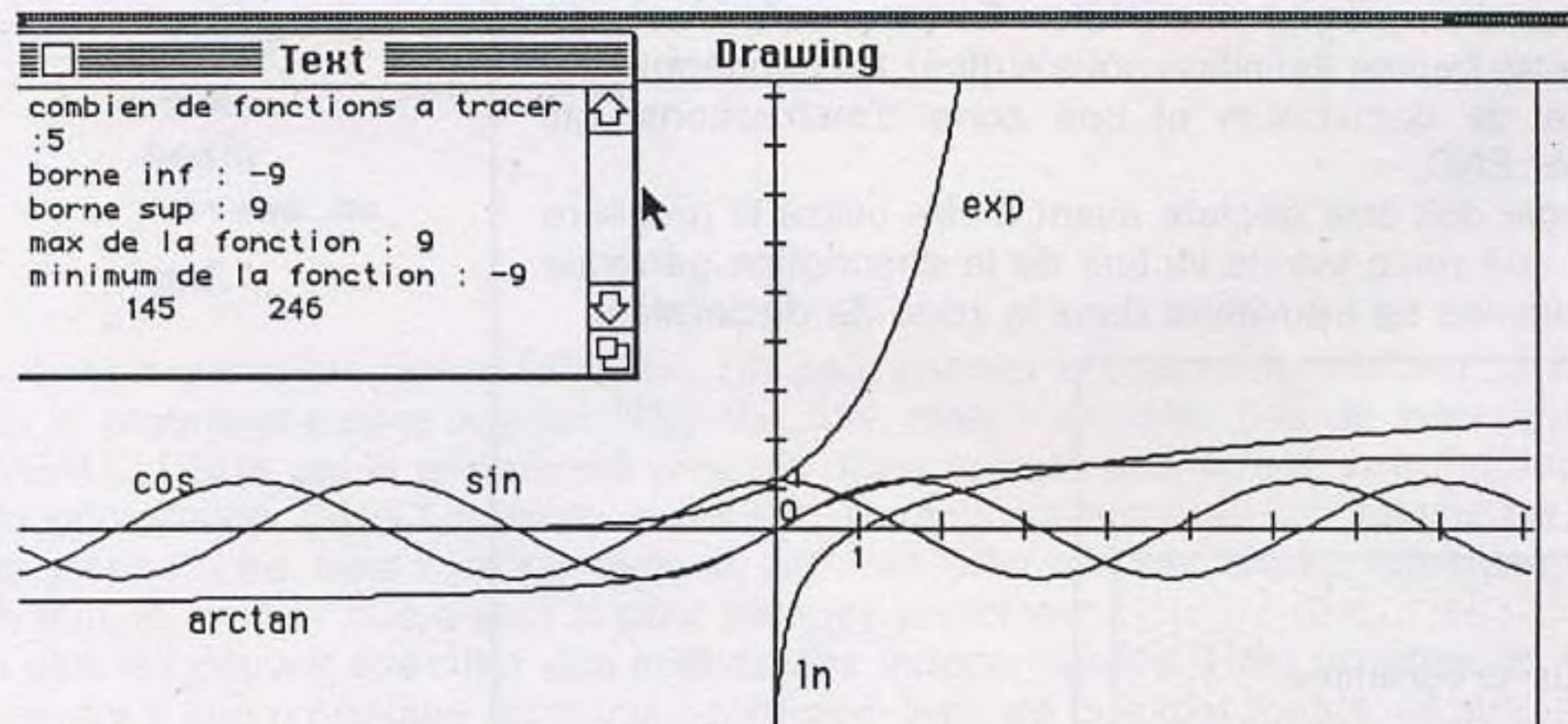
```

program ex;
var
  z : real;
  i : integer;
begin
  z := 0;
  write('  ', z, '  ', 'sin', '  ', 'cos', '  ');
  writeln('arctan', '  ', 'ln', '  ', 'exp');
  for i := 1 to 6 do
  begin
    z := z + pi / 4 * i;
    writeln(z, sin(z), cos(z), arctan(z), ln(z), exp(z));
  end;
end.

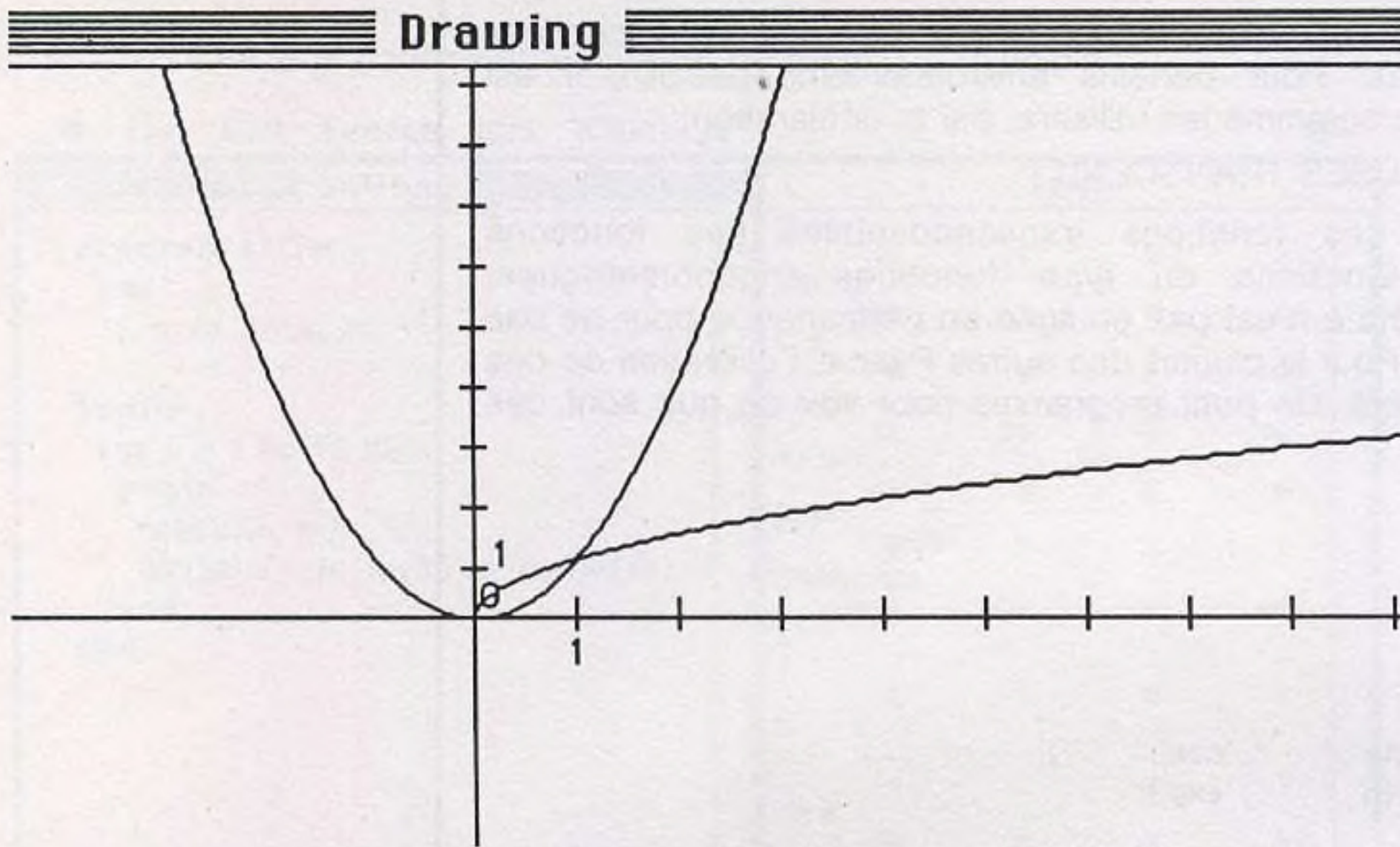
```

z	sin	cos	arctan	ln	exp
7.9e-1	7.1e-1	7.1e-1	6.7e-1	-2.4e-1	2.2e+0
2.4e+0	7.1e-1	-7.1e-1	1.2e+0	8.6e-1	1.1e+1
4.7e+0	-1.0e+0	1.2e-8	1.4e+0	1.6e+0	1.1e+2
7.9e+0	1.0e+0	1.4e-7	1.4e+0	2.1e+0	2.6e+3
1.2e+1	-7.1e-1	7.1e-1	1.5e+0	2.5e+0	1.3e+5
1.6e+1	-7.1e-1	-7.1e-1	1.5e+0	2.8e+0	1.5e+7

et leurs courbes :



SQR et SQRT : Ces deux fonctions donnent le carré et la racine carrée d'un nombre. Il faut noter que ce sont les deux seules qui sont offertes en standard, et que pour calculer une racine cubique par exemple, il faut programmer une routine. Les courbes de ces fonctions :



4. CONCLUSION SUR LES COMMANDES STANDARD DU PASCAL

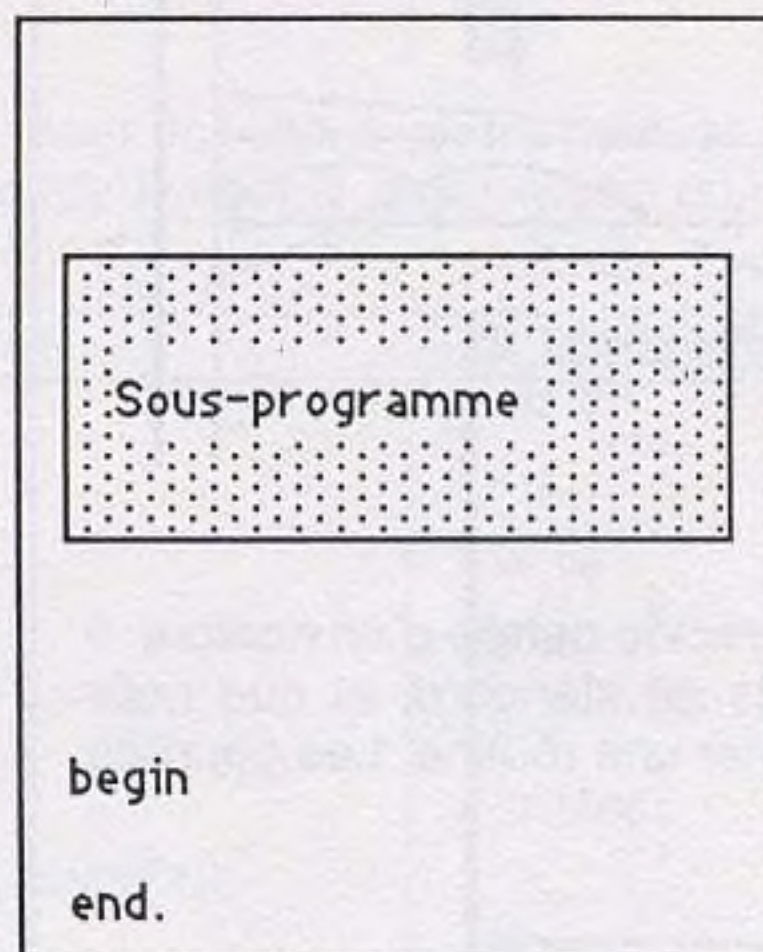
Ceci marque la fin de la partie classique de ce cours. A partir de maintenant, les notions que nous aborderons seront nouvelles comparées à ce qui avait été fait en Basic niveau 1. Ceux qui avaient suivi le cours niveau 2 retrouveront des notions déjà vues, et implémentées, souvent avec la plus grande difficulté. Ainsi, les sous-programmes, la récursivité, les pointeurs sont utilisés naturellement en Pascal. Ce mois-ci, nous allons voir ce qu'est un sous-programme, comment l'utiliser, et les notions à connaître pour ne pas être surpris par des résultats aberrants.

5. INTRODUCTION AUX SOUS-PROGRAMMES

5.1. Notions de bases

Un sous-programme est un programme (!!!). Bien sûr, cela semble commencer assez mal, mais la suite va permettre de bien différencier les deux structures. Un sous-programme est en effet à l'intérieur d'un programme. (Note : En plus, cela n'est pas toujours vrai, mais pour l'instant, cette fausse définition nous suffira). En conséquence, un sous-programme a une zone de déclaration et une zone d'instructions, qui commencent par BEGIN et finit par END.

Il est clair que tout sous-programme doit être déclaré avant d'être utilisé la première fois. Rappelez-vous le graphique que nous avons vu lors de la description générale d'un programme. Les sous-programmes se trouvaient dans la zone de déclaration.



5.2. Deux sortes de sous-programmes

En Pascal, deux types de sous-programmes sont accessibles. Ils ont des propriétés communes et d'autres très différentes. Parmi les propriétés communes, la structure visuelle, nous venons de le dire. Parmi les différences notables, la structure interne de leur représentation en machine, et la manière de les appeler, de les utiliser.

5.3. Les procédures

Une procédure (du nom de ce sous-programme en Pascal) est un sous-programme, un bloc indépendant. On peut donc le tester indépendamment, puis l'intégrer dans un plus gros programme. On voit ici la structuration réapparaître. Toute instruction, ou bloc d'instructions, peut être séparée pour former une procédure. Il suffit de créer le bloc de déclarations approprié. On peut donc donner une définition, encore assez générale il est vrai : une procédure est un bloc d'instructions utilisant des données qui lui sont transmises ou que le bloc va chercher dans la déclaration du bloc appelant.

5.4. Les fonctions, ou fonctions procédurales

Les fonctions est un bloc d'instructions dont le but est de fournir une valeur d'un type déclaré. Cela signifie que le rôle d'une telle structure est exclusivement de calculer. Ce qui est renvoyé au programme ou sous-programme appelant est une valeur, rien d'autre.

5.5. Description de l'utilisation d'une procédure

Pour utiliser une procédure, deux impératifs. Tout d'abord, quitte à se répéter, elle doit être entièrement écrite dans la zone de déclaration de la structure qui se trouve au dessus. Un exemple :

```

program un;
  var
    x, y, z : integer;
  procedure niveau_un;
  procedure niveau_deux;
  begin
    niveau_deux;
  end;
  begin
  end;
  begin
    niveau_un;
  end.

```

Ici, on voit que la procédure NIVEAU__UN peut appeler la procédure NIVEAU__DEUX, que le programme peut appeler NIVEAU__UN, mais n'essayez pas de faire appeler NIVEAU__DEUX par le programme principal. Cela occasionnerait une erreur détectée à la compilation. Dans l'exemple ci-dessus, les procédures et le programme ne font pas grand chose, mais il est possible de leur faire effectuer des tâches intéressantes. (Ah bon, je croyais que c'était là pour faire joli seulement.)

En plus de pouvoir exécuter des instructions indépendantes, il est possible de faire exécuter à une procédure un travail coordonné avec ce que font toutes les autres, en lui envoyant des informations venues d'ailleurs (un peu comme GINI) (est-ce bien raisonnable d'imprimer cela ?). Pour cela, plusieurs solutions que nous allons décrire maintenant, et dont il ne faut pas oublier qu'elles sont très différentes.

5.5.1. Règles générales

Pour passer des informations dans la procédure, il faut qu'au moment de l'appel, vous puissiez indiquer quelles informations vous voulez transmettre.

Pour cela, **à l'endroit de l'appel**, les informations sont groupées dans une liste parenthésée, qui suit le nom de la procédure appelée.

Par exemple, si la procédure ESSAI est déclarée avant de façon adéquate, il est possible de réaliser l'appel à ESSAI par :

```
ESSAI (x,z,t) ;
```

qui signifie un appel à ESSAI, qui utilisera les données qui se trouvent dans x, y et z. Un exemple :

```
program premier_appel;
var
  x, y, z : integer;

procedure essai (a, b, c : integer);
begin
  writeln(a + b + c);
end;

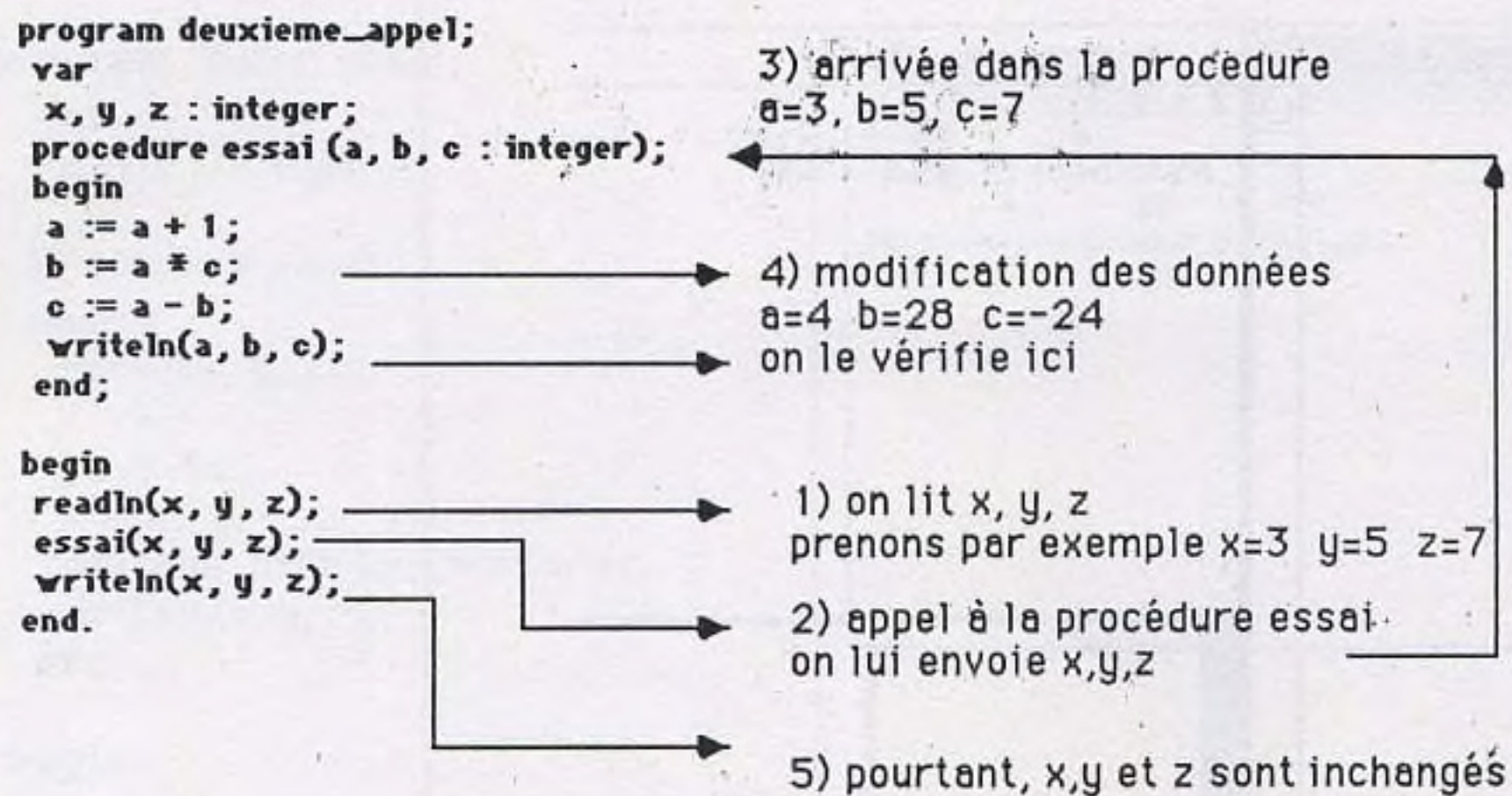
begin
  readln(x, y, z);
  essai(x, y, z);
end.
```

Ici, x, y, z sont envoyées à ESSAI, qui travaille dessus sans les modifier. Notons que pour éviter toute confusion dans la procédure appelée, le nom de chaque variable est différente. Ainsi, on sait où l'on se trouve, et il n'y a pas de mélange avec les variables du programme appelant. Nous vous recommandons de toujours faire cela.

Depuis le début de ce chapitre, nous insistons sur le fait qu'ainsi, les variables passées sont inchangées. Pourquoi ? Voilà un exemple qui illustre ce qui se passe :

```
program deuxieme_appel;
var
  x, y, z : integer;
procedure essai (a, b, c : integer);
begin
  a := a + 1;
  b := a * c;
  c := a - b;
  writeln(a, b, c);
end;
begin
  readln(x, y, z);
  essai(x, y, z);
  writeln(x, y, z);
end.
```

Regardons en détail les étapes de ce programme :



Il est normal que certains ne soient pas convaincus par cette démonstration. Par exemple, supposons que nous ayons nommé les variables dans ESSAI du nom des autres variables. Essayons :

```

program troisieme_appel;
var
  x, y, z : integer;
procedure essai (x, y, z : integer);
begin
  x := x + 1;
  y := x * z;
  z := x - y;
  writeln(x, y, z);
end;

begin
  readln(x, y, z);
  essai(x, y, z);
  writeln(x, y, z);
end.

```

Si vous faites tourner ce programme, il fonctionne **exactement** comme le précédent. Alors, comment interpréter ceci ? Tout simplement, les variables d'un programme sont locales, sauf mentions particulières, et leur utilisation ne modifie pas celles du programme appelant (nous le répétons), sauf mention particulière. Autrement dit, cela revient à utiliser les variables de manière purement locales. Mais alors, jusqu'où une variable est-elle utilisable sans problème ? C'est-à-dire, jusqu'où une variable est-elle comprise par le programme comme cette variable et non une autre déclarée ailleurs ? Voici une série d'exemples commentés qui vont faciliter la réponse :

```


program rayon_d_action;
var
  x, y, z : integer;
procedure essai;
begin
  x := x + 1;
  y := x * z;
  z := x - y;
  writeln(x, y, z);
end;

begin
  readln(x, y, z);
  essai;
  writeln(x, y, z);
end.

```

3 5 7
4 28 -24
4 28 -24

Ici, la procédure se sert des variables déclarées au-dessus, dans le programme qui l'appelle. Pas de problèmes particuliers, c'est donc une action valide. On peut utiliser des variables déclarées à un niveau supérieur (niveau appelant, ou encore plus haut, niveau qui appelle le niveau appelant, etc.).

 The name "x" has not been declared yet.

```

program rayon_d_action;

procedure essai;
var
  x, y, z : integer;
begin
  readln(x, y, z);
  x := x + 1;
  y := x * z;
  z := x - y;
  writeln(x, y, z);
end;

begin
  essai;
  writeln(x, y, z);
end.

```

Ici, le message est clair, la variable x est inconnue au niveau du programme principal, bien que déclarée au niveau du sous-programme. Merci Mr Macintosh pour la clarté de vos messages d'erreurs (et en plus ils sont jolis).

```

program autre_essai;
var
  x, y, z : integer;

procedure essai;
begin
  readln(x, y, z);
  x := x + 1;
  y := x * z;
  z := x - y;
  writeln(' dans la procedure');
  writeln(x, y, z);
end;

begin
  essai;
  writeln('dans le programme principal');
  writeln(x, y, z);
end.

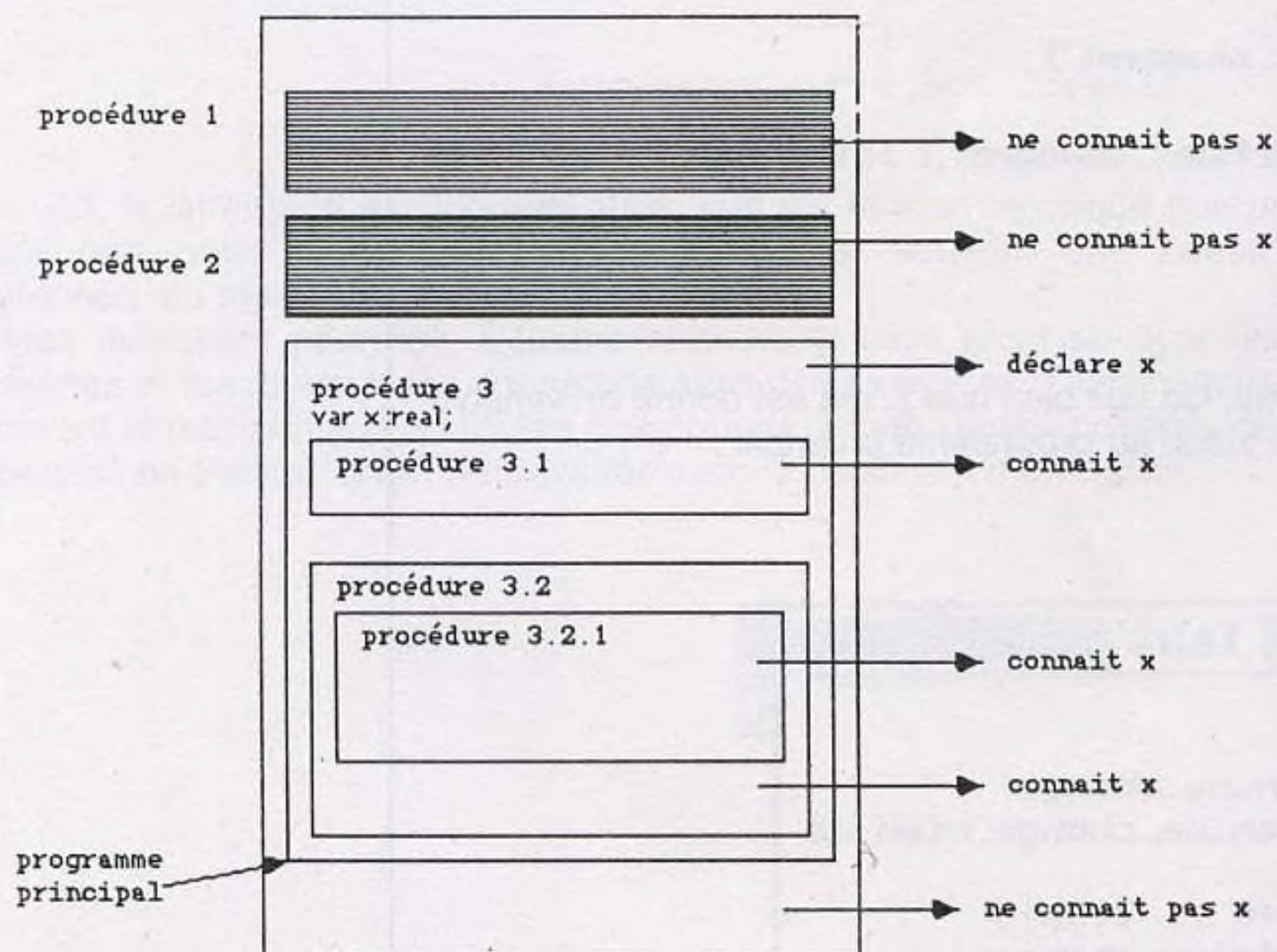
```

```

===== Text =====
3 5 7
dans la procedure
 4 28 -24
dans le programme principal
 0 0 0

```

Ici, les variables sont déclarées dans le programme appelant, utilisées dans le sous-programme et reviennent intactes (non modifiées) dans le programme appelant. Quelle est l'interprétation donnée à des programmes ? Une variable déclarée dans une procédure A est locale à cette procédure et à toutes les procédures qui sont déclarées dans la première. Par contre, cette variable n'est pas connue par les procédures qui sont au-dessus de A, par exemple celle qui appelle A. En résumé, un graphique qui explique la zone «d'influence» d'une variable :



Jusqu'à présent, nous avons dit qu'une variable n'était pas modifiée par un appel, c'est-à-dire que si l'on appelle une procédure A qui utilise une variable x déclarée dans la procédure appelante, à la fin de A, x est restituée avec la valeur qu'elle avait avant l'appel. Mais comment faire si l'on veut changer cette valeur de x dans A ? Rien de plus simple, il suffit de le dire. Comment ? Par le mot VAR dans la déclaration de la procédure, qui indique que ce paramètre pourra varier, et la modification être conservée dans le programme appelant :

```

program variations;
var
  x, y, z : real;
procedure rien_ne_change (x, y, z : real);
begin
  writeln(' en entree de rien_ne_change, les valeurs sont : ',x,y,z);
  x := x + 1;
  z := x - y;
  y := 5 * y;
  writeln(' en sortie de rien_ne_change, les valeurs sont : ',x,y,z);
end;
procedure certains_changent (x, y : real;
  var z : real);
begin
  writeln(' en entree de certains_changent, les valeurs sont : ',x,y,z);
  x := x + 1;
  z := x - y;
  y := 5 * y;
  writeln(' en sortie de certains_changent, les valeurs sont : ',x,y,z);
end;
begin
  readln(x, y, z);
  writeln('entree dans rien_ne_change');
  rien_ne_change(x, y, z);
  writeln('apres la sortie de rien_ne_change,rien de change');
  writeln(x, y, z);
  writeln('entree dans certains_changent');
  certains_changent(x, y, z);
  writeln('apres la sortie de certains_changent,z est change');
  writeln(x, y, z);
end.

```

Voici la sortie partielle de ce programme. On voit bien que z, qui est donné en variable de la seconde procédure, est modifié jusqu'au programme principal :

```

3 5 7
entree dans rien_ne_change
en sortie de rien_ne_change,rien de
change
 3.0e+0  5.0e+0  7.0e+0
entree dans certains_changent
en sortie de certains_changent,z
est change
 3.0e+0  5.0e+0  -1.0e+0

```

6. CONCLUSION TRES PROVISoire

On voit donc que pour qu'une variable soit affectée durablement (après la sortie de la procédure qui l'a modifiée), il faut la déclarer explicitement en variable à l'entrée de cette procédure. C'est là la distinction fondamentale entre deux types très différents de passages de paramètres à des sous-structures. Si aucun changement n'est souhaité, le passage est fait par la valeur de la variable, et pas par son nom. Ainsi, la sous-structure n'a accès qu'à la valeur, pas à l'emplacement où la variable est stockée, si elle ne peut la modifier. Dans le deuxième cas, le passage de l'adresse de la variable est réalisé, permettant sa modification en mémoire. Le graphique ci-dessous résume ces diverses options :

Passage par valeur

```

x=1.2345 ←
sous-prog(x,y,...)
      ↑
sous-prog(a,b ...:real);
  
```

Passage par variable

```

      ↓
x=1.2345
sous-prog(x,y,...)
      ↑
sous-prog(var a,b ...:real);
  
```

On le voit, la différence est fondamentale, tant sur le plan théorique que pratique. Il s'avère que cette multiplicité très puissante est souvent une cause d'erreur d'inattention, du moins dans les débuts.

La place manquant pour finir, il faudra attendre le mois prochain pour finir sur les procédures et fonctions. Nous donnerons alors des exercices d'applications qui vous montreront la manipulation de toutes ces notions. Il sera «enfin» possible de travailler proprement en Pascal, sinon de travailler avec le maximum d'efficacité.

COURS DE PROGRAMMATION APPROFONDIE

Dominique Chastagnier
Jean-François Coblentz
Patrick Gueneau

Comment représenter une pièce de mécanique ou un voilier de la coupe de l'America sur un écran d'ordinateur ? Dans un premier temps nous allons voir les différentes méthodes employées pour conserver à moindre frais les caractéristiques de la pièce en mémoire. Puis, nous verrons comment transcrire sur un écran plat une pièce en relief.

COURS N° 16

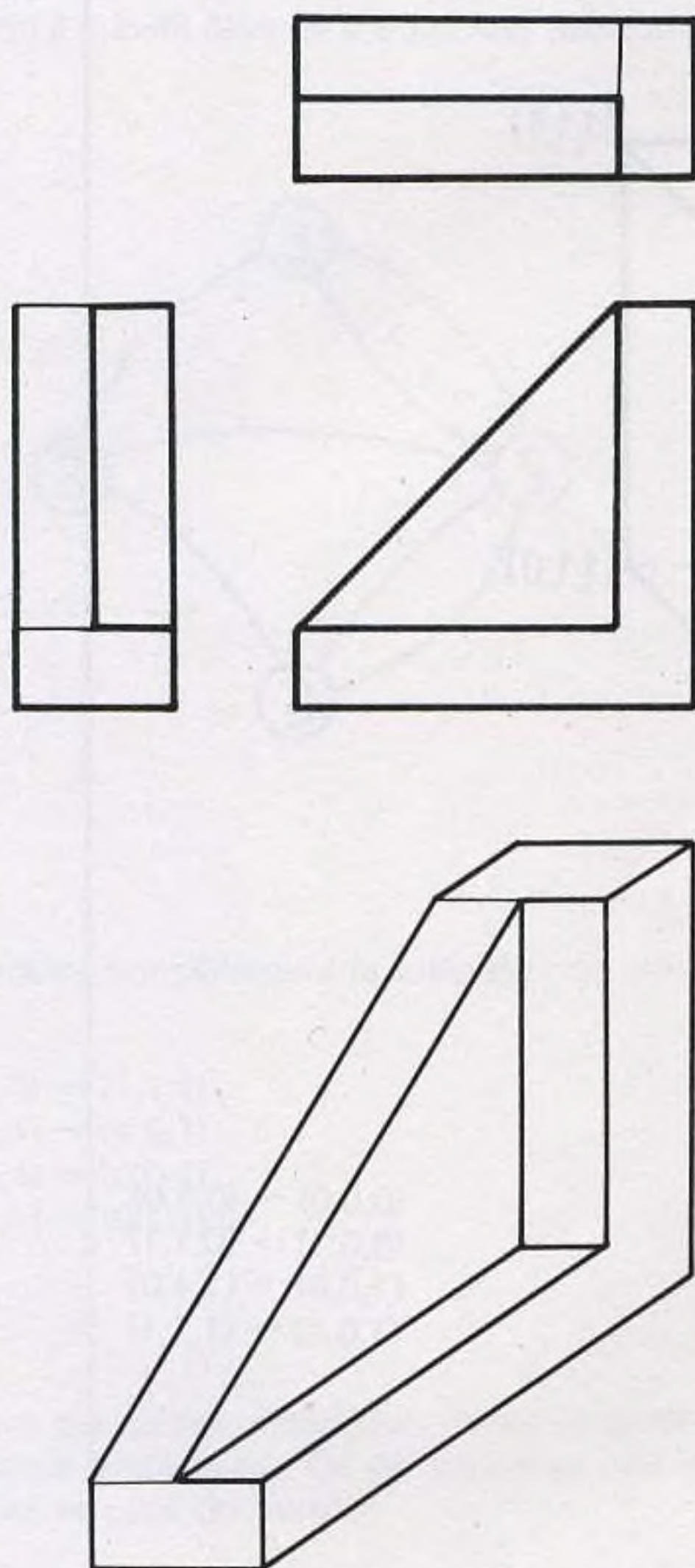
PLAN DU COURS

1. Structuration des données graphiques
 - 1.1. Représentation d'un solide
 - 1.1.1. Les arêtes
 - 1.1.2. Les faces
 - 1.1.3. Les volumes
2. Visualisation 3D
 - 2.1. Représentation 2D

I. STRUCTURATION DES DONNEES GRAPHIQUES

I.1. Représentation d'un solide

On appelle solide toute pièce que l'on souhaite représenter à l'écran. Cela va du dé à coudre jusqu'au moteur de voiture et il faut bien conserver dans une structure de données les caractéristiques de la pièce afin de la restituer sous diverses formes, selon les besoins du programme. Cela peut aller depuis différentes représentations statiques jusqu'à une évolution en trois dimensions de la pièce que l'on fait tourner telle un dessin animé.

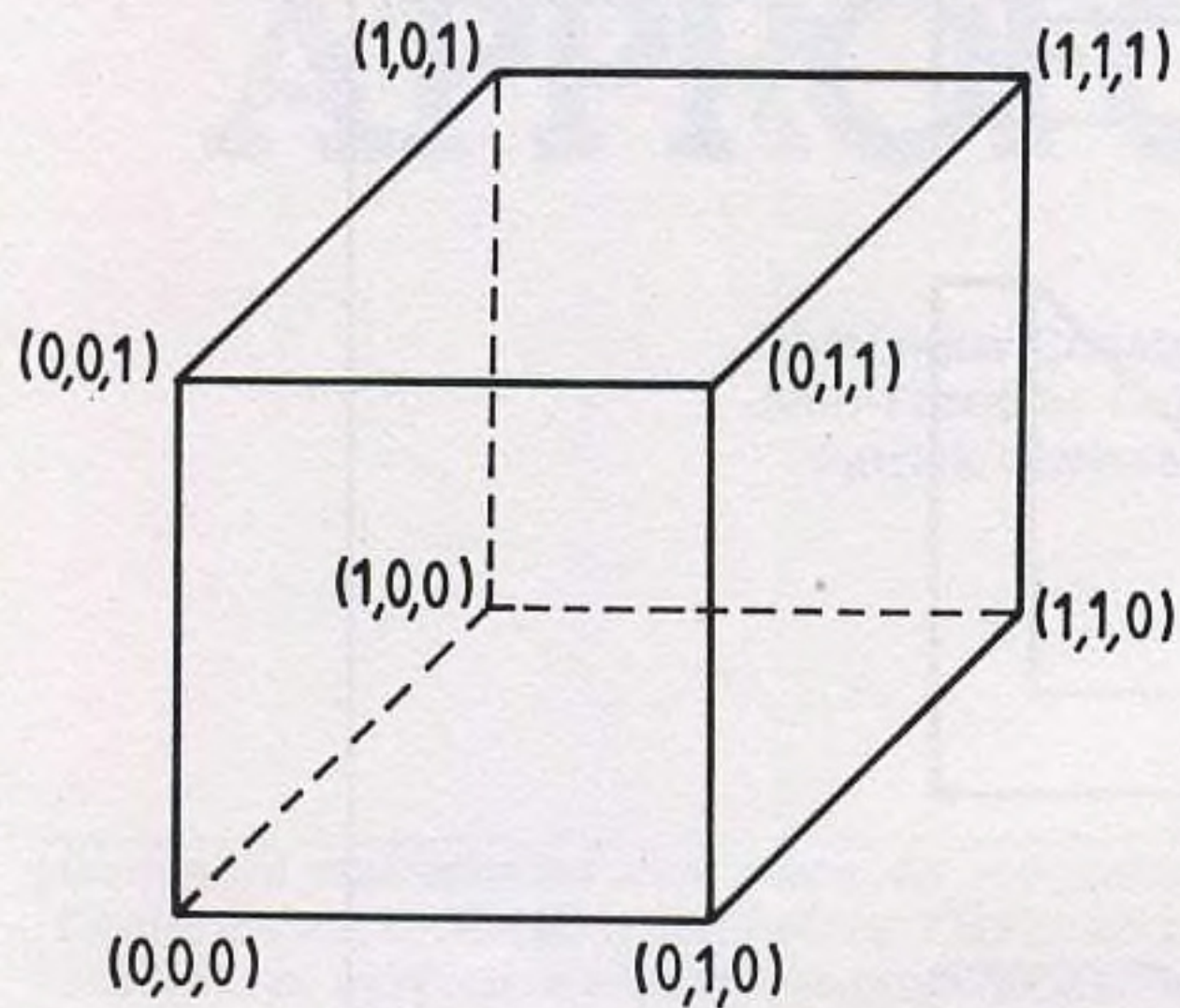


Pour conserver cette pièce dans une base de données, il existe bien des façons différentes de la symboliser, chacune a ses adeptes, et correspond à une démarche particulière de définition du problème. En effet, dans l'étude d'un solide, chacun veut voir midi à sa porte et souhaite définir le solide de la façon la plus facile à exploiter ultérieurement. Il existe donc trois grandes écoles pour la définition des solides :

- les arêtes,
- les faces,
- les volumes.

I.1.1. Les arêtes

Cette définition est souvent surnommée la définition fil de fer. Il est en l'occurrence très facile de suivre la construction du solide en construisant avec un fil de fer la suite des arêtes formant l'ossature du solide. Si l'on considère comme exemple un cube de côté de taille 1, il peut être symbolisé de la façon suivante :

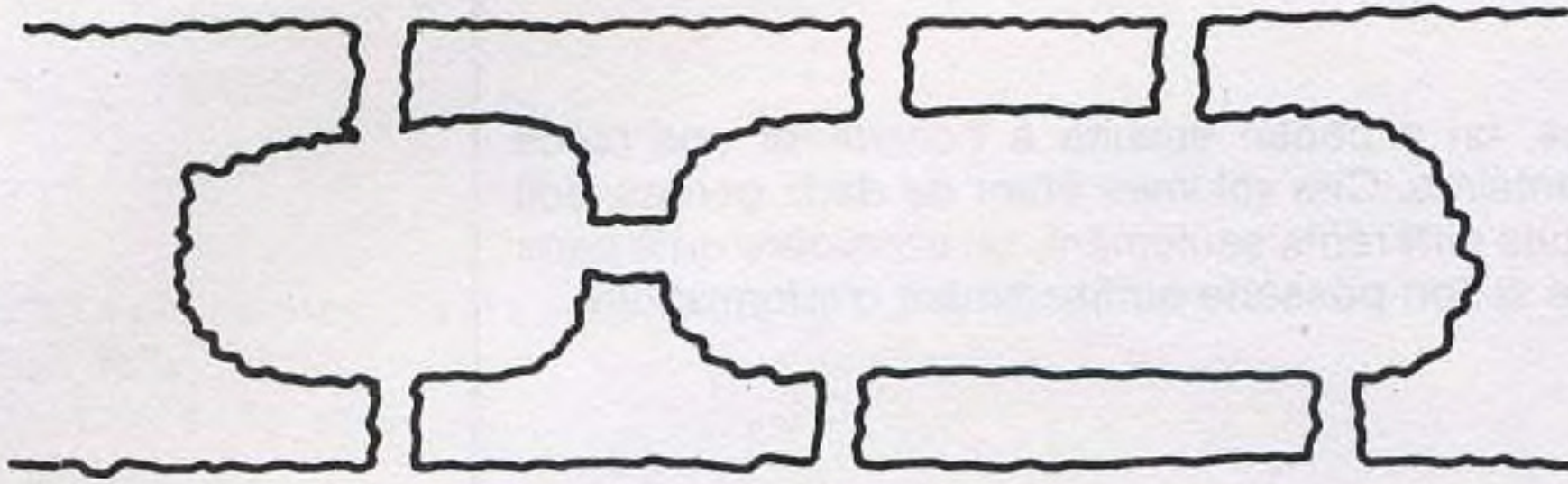


$(0,0,0) \rightarrow (0,0,1)$
 $(0,1,0) \rightarrow (0,1,1)$
 $(1,0,0) \rightarrow (1,0,1)$
 $(1,1,0) \rightarrow (1,1,1)$

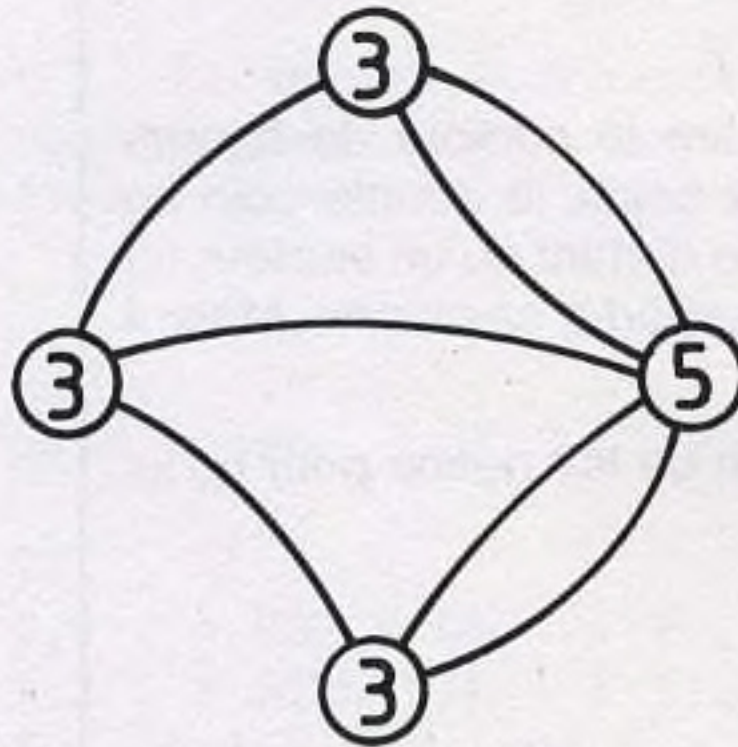
$(0,0,0) \rightarrow (1,0,0)$
 $(0,0,1) \rightarrow (1,0,1)$
 $(0,1,0) \rightarrow (1,1,0)$
 $(0,1,1) \rightarrow (1,1,1)$

$(0,0,0) \rightarrow (0,1,0)$
 $(0,0,1) \rightarrow (0,1,1)$
 $(1,0,0) \rightarrow (1,1,0)$
 $(1,0,1) \rightarrow (1,1,1)$

où nous avons la suite de ses arêtes avec, à chaque fois, une origine et une extrémité. Il existe d'autres façons plus condensées qui consistent à considérer que l'origine du second segment est l'extrémité du premier. Mais alors, on se trouve parfois face à des problèmes pour définir le nombre minimal de sauts nécessaires. Mais qu'est-ce qu'un saut ? Lorsque l'on arrive à un sommet dont toutes les arêtes ont été définies, on est obligé soit de repasser par une arête déjà définie soit de « sauter » à un autre point du solide sans tracer d'arête. Or, ceci n'est qu'une version moderne du problème des ponts de Königsberg. Pour les non-mathématiciens, il s'agissait d'un mathématicien du XVIII^e (non pas arrondissement, mais siècle) qui, habitant Königsberg, avait pour habitude de se promener le soir et il souhaitait, lors de sa promenade, passer sur tous les ponts de la ville sans repasser par le même. Il était arrivé à la conclusion que cela n'était pas possible.



Cela provenait du fait qu'il y avait plus de 2 sommets avec un nombre impair d'arêtes.



Nous pouvons donc réduire sensiblement la taille de nos données en mémoire :

$(0,0,0) \rightarrow (0,1,0) \rightarrow (0,1,1) \rightarrow (1,1,1)$
 $(0,1,0) \rightarrow (1,1,0) \rightarrow (1,1,1) \rightarrow (1,0,1)$
 $(1,1,0) \rightarrow (1,0,0) \rightarrow (1,0,1) \rightarrow (0,0,1)$
 $(1,0,0) \rightarrow (0,0,0) \rightarrow (0,0,1) \rightarrow (0,1,1)$

1.1.2. Les faces

Plutôt que de conserver les arêtes, il est plus facile de garder uniquement les faces pour un cube, c'est même simplissime. On ne conserve que les centres des faces, le fait que ce soit un carré et côté du carré.

On se trouve confronté à la question suivante : quelles sont les figures élémentaires nécessaires pour conserver toutes les faces car un polyèdre quelconque peut être fait de polygones quelconques. Le bon sens nous indique tout de suite que tout peut être ramené à un nombre plus ou moins grand de triangles, exception faite généralement des carrés et autres rectangles qui sont conservés, étant généralement d'un emploi facile. Mais une question brûle les lèvres de beaucoup d'entre vous depuis le début du paragraphe sur les arêtes : que se passe-t-il pour une sphère ? Il n'y a ni arête ni face sur une sphère, donc, il n'est pas possible de la symboliser telle quelle. On a donc recours à un expédient et on la « discrétise », c'est-à-dire qu'on la remplace à l'image d'un ballon de football, par une succession de facettes. Il est bien évident que ce n'est plus une sphère mais la solution de remplacement suffit dans la majeure partie des cas.

I.1.3. Les volumes

Comme les faces ne suffiraient pas, on a pensé ensuite à considérer une pièce comme la réunion de volumes élémentaires. Ces volumes étant de deux genres, soit pleins, soit vides, et surtout de 10 types différents seulement, on considère qu'à partir de ceux-ci, on peut tout reconstruire si l'on possède suffisamment d'informations :

- Le parallélépipède
- Le coin
- Le double-coin
- Le cylindre
- La portion de cylindre
- La section de cylindre
- Le tronc de cône
- La sphère
- La calotte sphérique
- Le tore

Vous remarquerez, bien sûr, que l'on pourrait encore réduire le nombre de figures élémentaires, le parallélépipède pouvant être fait avec deux coins, le double-coin en enlevant deux coins au parallélépipède, la section de cylindre n'étant qu'un secteur ou une portion de cylindre particulier, de même, la sphère par rapport à sa calotte. Mais, il est plus facile de posséder ces volumes supplémentaires.

Nous verrons le mois prochain en détail comment on se doit de les définir pour qu'ils occupent le moins de mémoire possible.

II. VISUALISATION 3D

Quelle que soit la méthode choisie pour représenter les pièces, nous sommes obligés, à un moment donné, de transformer la vision spatiale de la pièce en une simple vision planaire. En d'autres termes, les trois coordonnées en X, Y et Z de la pièce ne doivent en devenir que deux en I et J de l'écran.

II.1. Représentation 2D

Pour situer un point sur l'écran à partir de ses coordonnées 3D, il convient de recourir à un système d'équations comprenant, outre les trois coordonnées, les deux angles paramétrés θ et φ .

$$I = X \cdot \cos \varphi - Y \cdot \sin \theta$$

$$J = -X \cdot \sin \theta \sin \varphi - Y \cos \theta \sin \varphi + Z \cos \varphi$$

Ensuite, il ne reste plus qu'à donner différentes valeurs à θ et φ pour obtenir une vue de la pièce considérée.

- | | |
|-------------------|---|
| - vue de face | $\theta = 0^\circ, \varphi = 0^\circ$ |
| - vue de dessus | $\theta = 0^\circ, \varphi = 90^\circ$ |
| - vue de côté | $\theta = 90^\circ, \varphi = 0^\circ$ |
| - vue isométrique | $\theta = 45^\circ, \varphi = 35^\circ 26'$ |

II.2. Maintenant que vous avez votre système de transformation en 2D, vous avez certainement envie de faire évoluer vos pièces dans l'espace. Nous avons vu, le mois dernier, tous les déplacements possibles, à vous de les simuler.

Pour cela, il suffit de modifier les valeurs X, Y, Z de chacun de vos points dans l'espace puis, une fois la modification faite de conserver la projection 2D que vous avez choisie initialement et de tracer sur votre écran l'ensemble de droites que vous avez entre vos points.

Toutes les modifications à apporter se mettent sous forme de matrices que l'on multiplie aux coordonnées précédentes.

Print file "BASIC"

Page 1

```

10 REM Lecture des Suites de Points
20 READ NF%
30 FOR I = 1 TO NF%
40 READ NP%(I)
50 FOR J = 1 TO NP%(I)
60 READ X ( I,J ) , Y ( I,J ) , Z ( I,J )
70 NEXT J
80 NEXT I

.....

500 REM Projection sur l' Ecran des Coordonnees 3D
510 REM si calcul en degres theta = TD , en radians theta = TR
511 REM phi = PD ou phi = PR
520 TD = 45
521 PD = 35,26
530 TR = TD * 3.141592653 / 180
531 PR = PD * 3.141592653 / 180
540 FOR I = 1 TO NF%
550 FOR J = 1 TO NP%(I)
560 XP ( I,J ) = X ( I,J ) * COS ( PD ) - Y ( I,J ) * SIN ( TD )
570 YP ( I,J ) = X ( I,J ) * SIN ( TD ) * SIN ( PD )
      - Y ( I,J ) * COS ( TD ) * SIN ( PD ) + Z ( I,J ) * COS ( PD )
580 NEXT J
590 NEXT I

```

Vous avez remarqué que nous transformons les points et donc, par là même, les droites sans nous préoccuper de savoir si celles-ci sont visibles ou non. En effet, les algorithmes de «parties cachées» qui éliminent les arêtes non visibles sont beaucoup trop coûteux en temps de calcul pour être ajoutés. Voici donc les coordonnées complètes de plusieurs pièces.

pyramide

$$(0,0,0) \rightarrow (0,2,0) \rightarrow (1,1,2) \rightarrow (2,0,0) \rightarrow (2,2,0)$$

$$(0,2,0) \rightarrow (2,2,0) \rightarrow (1,1,2) \rightarrow (0,0,0) \rightarrow (2,0,0)$$

à vous de trouver

$$(0,0,0) \rightarrow (3,1,1) \rightarrow (6,0,0) \rightarrow (4,2,2) \rightarrow (2,2,2) \rightarrow (3,3,8)$$

$$(6,0,0) \rightarrow (5,3,1) \rightarrow (6,6,0) \rightarrow (4,4,2) \rightarrow (4,2,2) \rightarrow (3,3,8)$$

$$(6,6,0) \rightarrow (3,5,1) \rightarrow (0,6,0) \rightarrow (2,4,2) \rightarrow (4,4,2) \rightarrow (3,3,8)$$

$$(0,6,0) \rightarrow (1,3,1) \rightarrow (0,0,0) \rightarrow (2,2,2) \rightarrow (2,4,2) \rightarrow (3,3,8)$$

DIALOGUE AVEC NOS LECTEURS

LES ENVIRONNEMENTS DE DEVELOPPEMENT

INTRODUCTION

Vous connaissez maintenant le vocabulaire, et les outils mis à la disposition du programmeur pour développer dans son langage préféré. Nous allons décrire brièvement quelques uns des environnements les plus connus afin de vous permettre de bien situer les performances respectives des langages proposés par les revendeurs.

Environnements types

Que ce soit pour le PASCAL ou le BASIC, on rencontre généralement trois grandes catégories de systèmes de développement. Il y a les interpréteurs, les compilateurs et les compilateurs-interpréteurs. Il existe néanmoins de nombreux utilitaires permettant d'améliorer votre langage habituel. Ils se présentent soit sous la forme de bibliothèques de programme source ou machine (le plus souvent pour la langage PASCAL), soit sous forme de modules qui se greffent au langage standard et ajoutent des commandes animations graphiques, sons, 3D, etc (sous BASIC en général).

INTERPRETEURS

Pour ce qui est du BASIC, vous en disposez déjà un, tout au moins si vous possédez un ordinateur (évidemment s'il s'agit d'un MACINTOSH vous faites exception à la règle, celui-ci est en effet un des seuls à ne pas avoir en ROM, ni même sur disquette un BASIC en standard). Inutile donc d'entrer dans les détails. Cependant, si vous envisagez de passer à un BASIC compilé, souvenez-vous des facilités d'utilisation dont vous disposez jusqu'alors.

De même, vous pourrez être amené à changer votre environnement BASIC parce qu'il ne vous satisfait pas (ainsi beaucoup critiquent le BASIC standard de l'ATARI 1040ST) pour reprendre une version interprétée plus complète et qui disposera un jour (mais si, il faut croire les annonces des nouveaux produits, on n'est pas toujours déçu !) d'un compilateur.

A l'inverse, il vous sera difficile de comparer les interpréteurs PASCAL du marché. Nous n'en connaissons qu'un seul et qui, de plus, est réservé aux utilisateurs fortunés du MACINTOSH (c'est une bien belle machine mais bien chère pour un usage exclusivement familial). Ceci est d'ailleurs bien dommage car en plus de prouver que la réalisation d'un tel produit n'est pas impossible, il représente l'outil quasi-idéal pour apprendre le PASCAL sans se perdre dans les complications des compilations, éditions de liens et autres. La plupart des exemples PASCAL des derniers numéros de LED (copie d'écrans MACINTOSH) proviennent de ce langage.

COMPILATEURS

L'éventail des compilateurs pour quelque langage que ce soit dépend bien évidemment du succès de votre machine. Si elle tourne sous CP/M ou CP/M+, MS-DOS ou PC-DOS (i.e. IBM PC ou compatible), pas de problème, vous n'aurez que l'embarras du choix. De même, il est maintenant facile de trouver des compilateurs PASCAL ou BASIC pour les AMSTRAD CPC ou PCW et ATARI 520ST ou 1040ST. Les

plus connus sont le TURBO-PASCAL de Borland (pour son prix intéressant, mais aussi ses performances) et le MS-PASCAL de Microsoft (parce qu'il est compatible avec les environnements standards) (éditeur de liens, librairies...) des systèmes d'exploitation CP/M ou MSDOS ou encore le PASCAL MT+ disponibles sur les CPC et PCW AMSTRAD.

Le TURBO-PASCAL : Dans sa version de base, il se présente comme un ensemble intégrant éditeur, compilateur et éditeur de liens. Le menu principal propose l'édition d'un fichier, sa compilation en mémoire (compilation rapide, sans sauvegarde sur disquette), ou avec création d'un fichier, l'exécution du programme en mémoire et le retour au système. Bien évidemment, en cas d'erreur, il est possible de revenir facilement à la ligne de programme fautive, avant que cette erreur n'intervienne à la compilation ou à l'exécution. Il existe sur PC et compatible, AMSTRAD CPC, CP/M et bientôt sur MACINTOSH.

Le MS-PASCAL : Il représente le compilateur type tel qu'il existe sur les plus gros ordinateurs. Ce n'est en fait, qu'une application qui accepte du texte en entrée (i.e. un fichier de caractère créé par n'importe quel éditeur et produit s'il n'y a pas d'erreur un fichier objet. Ce fichier possède une structure compatible avec les standards du système d'exploitation donc compatible avec les outils existants (créateur de bibliothèques et éditeurs de liens «LINKER» en anglais). Il est donc possible, en théorie tout au moins, de lier ensemble des modules PASCAL FORTRAN BASIC pour créer une seule application. Il est disponible sur toute machine MS-DOS.

Le PASCAL MT+ : C'est le seul PASCAL disponible à notre connaissance sur AMSTRAD PCW8256. Il a l'avantage d'être bon marché, mais, bien qu'il ait bonne réputation (sur d'autres machines notamment), n'ayant jamais eu l'occasion de le voir fonctionner sur AMSTRAD, nous ne pouvons donc en dire plus (si quelqu'un l'utilise sur sa machine, qu'il nous envoie ses appréciations et nous les publierons).

Pour les autres :

Il est hélas impossible d'énumérer toutes les versions disponibles ou annoncées pour toutes les machines. Ainsi, il existe deux PASCAL sur ATARI 1040ST disponibles en France, dont un provient de la même société METACOMCO créatrice du système d'exploitation du COMMODORE AMIGA, et l'autre le PASCAL OSS ; là encore, nous ne connaissons pas leurs caractéristiques. Pour les THOMSON T07 et T09, une version complète de PASCAL UCSD se fait toujours attendre chez FIL ; elle comblera alors le vide qui existe dans la gamme THOMSON, mais attention, il semble que ce PASCAL ne tournera que sur T07/70 (avec 128K de ram) T08, T09 et T09+.

Les compilateurs BASIC :

Le choix d'un nouvel environnement BASIC ne se pose pas dans les mêmes termes que celui d'un premier langage PASCAL. Votre choix dépendra des objectifs que vous vous fixez :

- soit améliorer la performance de programmes existants, il faut donc un compilateur 100 % compatible avec le BASIC résident.
- soit changer complètement d'environnement, le plus important étant alors les performances ou le rapport qualité/prix de ce BASIC.

Voici quelques Compilateurs recensés :

sur THOMSON :

SPEEDY WONDER compilateur basic pour T07-T09

sur AMSTRAD CPC :

Vous n'avez que l'embarras du choix, avec de nombreuses extensions, compilateurs et autres utilitaires disponibles soit en cassettes (CPC464 de base) soit sur disquette pour le haut de la gamme CPC (C BASIC COMPILER, LASER BASIC, SPEEDY WONDER comme pour T09, 3D MEGACODE, etc.) ; pour la série PCW le «C BASIC COMPILER» est aussi disponible.

sur PC et compatible :

Il y en a trop pour les citer. Ils sont malheureusement très souvent beaucoup plus

chers que sur les micro-ordinateurs dont on vient de parler (C BASIC, MS-QUICK BASIC, ZBASIC).

sur ATARI 520ST :

Essentiellement deux solutions, soit l'ajout de la cartouche FAST BASIC qui est un basic compilé, soit l'utilisation du GFA BASIC qui disposera prochainement d'une version compilée.

sur MACINTOSH :

La version compilée du fameux MS-BASIC de Microsoft se faisant attendre (la verra-t-on jamais un jour), il reste le compilateur ZBASIC qui a outre l'avantage d'exister sous plusieurs machines et de proposer des commandes graphiques indépendantes de l'ordinateur, pratique pour «porter» un logiciel d'une machine vers une autre. Le code généré est de plus très rapide.

sur APPLE II :

Les plus connus sont «TASC» et «Einstein compiler».

Le premier lit un programme BASIC à partir d'une disquette et sauvegarde le «code généré» sur cette disquette. Il faut charger au moment de l'exécution un module machine contenant toutes les routines systèmes nécessaires à l'exécution du programme (BLOAD RUNTIME). On exécute ensuite le programme comme n'importe quel module machine (BRUN <nom du programme>). Ce basic est compatible avec l'APPLESOFT, et dispose de nombreuses extensions ; il est donc relativement délicat de mise en œuvre.

Le second est on ne peut plus simple. La compilation se déroule en trois étapes.

a) chargement du programme basic en mémoire (LOAD)

b) exécution du compilateur (par un BRUN)

c) en fin de compilation, on sauve le programme en mémoire par un SAVE classique (mais ce n'est plus du BASIC).

INTERPRETEURS / COMPILATEURS

Peu de BASIC ont cette caractéristique. On peut citer le C BASIC de Digital Research (le créateur de CP/M), encore qu'il existe une version «compilateur seul» de ce langage, ce qui pose d'ailleurs quelques problèmes d'adaptation lorsque l'on passe de la version semi-compilée à la version complètement compilée (par exemple l'allocation et la désallocation dynamique des variables chaînes de caractères et les dimensionnements de tableaux).

Pour les PASCAL, il est difficile dans de nombreux cas de connaître avec précision s'il s'agit de «true compiler» comme le disent les américains, c'est-à-dire de vrais compilateurs, ou bien d'environnements mixtes. Cette caractéristique apparaît rarement dans les publicités des produits. Seul le PASCAL UCSD est connu pour être un véritable interpréteur/compilateur. Il est disponible sur APPLE II, sur MACINTOSH, les IBM PC et compatibles, et le sera donc bientôt sur les THOMSON T07/70 à T09+. Nous en avons déjà parlé il y a quelques temps, et nous reviendrons sur ces particularités dans le cours de PASCAL. C'est en effet un peu grâce au PASCAL UCSD que ce langage est passé de la pure théorie au développement réel d'applications.

CONCLUSION

Pour être complet, il faut ajouter que les prix des langages de développement baissent. Est-ce l'effet BORLAND ? Toujours est-il qu'il est maintenant possible pour le prix de 2 à 3 jeux de programmer en PASCAL sur AMSTRAD, ATARI et cela le sera bientôt sur d'autres ordinateurs familiaux. Cette remarque n'est plus tout-à-fait vraie pour un compatible PC (on est le plus souvent au dessus de 1000 F ttc), ne parlons pas du MACINTOSH qui est une merveilleuse machine ... inabordable en France (aux Etats-Unis les prix sont nettement plus raisonnables, par exemple le ZBASIC à moins de 50 \$ vaut ici à peu près 1000 F).

Nous continuerons à vous tenir régulièrement au courant des évolutions des produits disponibles sur le marché, et nous publierons toutes les expériences des lecteurs qui disposeraient de certaines versions dont nous vous avons parlé mais aussi qui seraient susceptibles d'en connaître d'autres ; nous n'avons nullement la prétention de tout connaître dans ce domaine : aussi faites-nous parvenir vos commentaires.

LANGAGE PASCAL	ORDINATEURS	AVANTAGES	INCONVENIENTS
<p>TURBO PASCAL (Borland) actuellement la star du marché.</p>	<p>1) IBM et compatibles 2) CP/M (APPLE 2) 3) AMSTRAD CPC 4) MACINTOSH</p>	<ul style="list-style-type: none"> - 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. 	<ul style="list-style-type: none"> - non standard pour certaines fonctions. - pas de librairies compilées ni d'édition de liens (link).
<p>PASCAL UCSD plus qu'un langage, un environnement complet (gestion de fichiers, éditeur, debugger...).</p>	<p>1) APPLE (Version 2) 2) MACINTOSH (Version 4) 3) IBM et compatibles 4) THOMSON TO770 à TO9+</p>	<ul style="list-style-type: none"> - la référence de tous les ouvrages (et de notre cours). - un système d'exploitation commun à toutes les versions - possibilités graphiques - outil de développement complet (librairies, modules). 	<ul style="list-style-type: none"> - onéreux (2000 à 4000 F) sauf pour THOMSON (?). - compilation lente - code semi- compilé (PCODE) d'où un temps d'exécution moyen.
<p>MICROSOFT PASCAL le premier disponible sur IBM, et le standard IBM.</p>	<p>IBM et compatibles, environnement MS-DOS.</p>	<ul style="list-style-type: none"> - complet et doté de nombreuses extensions. - compatible avec tous les langages de MICROSOFT. - plus rapide que la version UCSD 	<ul style="list-style-type: none"> - aussi peu abordable que la version UCSD. - moins performant que le TURBO PASCAL.
<p>PASCAL MT + sous CP/M</p>	<p>AMSTRAD CPC et PCW</p>	<ul style="list-style-type: none"> - sous environnement CP/M - bon marché. 	<ul style="list-style-type: none"> - moins connu que le TURBO-PASCAL.

C'EST ARRIVE DEMAIN



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

Et voilà !!! L'abominable s'est produit. Un programme vient de sortir accompagné d'une plaque destinée à fournir des impressions sensibles. Le problème : il s'agit d'odeurs à gratter, et sur le plan des odeurs, les USA sont vraiment très en retard sur nous. C'est tout simplement immonde... franc succès en perspective ! D'autant que ce n'est pas le jeu lui-même qui captivera suffisamment pour que les odeurs passent au second plan. Alors, Leather Goodesses of Phobos sera une réussite pour Noël. L'imagination, bien que sans frontières, semble correspondre à la civilisation qui l'inspire. Ceci va dans la droite ligne du film, lui aussi odorant, dont je ne peux me rappeler le nom, et dont le seul succès fut cette plaque à gratter. Dans quelques cinémas, devant le succès totalement inattendu, le manque de ces cartes fit se détourner les clients. Il y a donc un public pour l'horrible (et non l'horreur). Complément d'information : cela se passe en inter-Galactique. (Ah bon, j'ai eu peur que ce ne

soit ringard en plus). En fait, je ne sais pas comment on fait lorsque la plaque est entièrement grattée. Vivement le programme qui sent sur la disquette, ou fait produire des odeurs par la machine (après le langage machine, l'odeur machine. Pourvu que cela soit dans un langage structuré). Tout de même, il faut revenir sur le jeu lui-même. Il s'agit de récupérer le père d'une séduisante demoiselle en maillot de bain en cuir (???), abominablement retenu par d'effrayantes sorcières du genre amazone.

Lors de la première exposition consacrée à l'édition électronique (traduction approximative de l'expression locale «Desktop Publishing»), il m'a été donné de voir le passé lointain de cette technologie (les machines de l'an dernier), le présent (celles de l'an prochain), mais rien de neuf en fait, et nul ne semble savoir ce que demain sera. Personne ne m'aide pour cette rubrique. J'ai tout de même bien ri. Un monsieur,

que nous qualifierons de bon chic bon genre en France, imprimeur de son état, a mis les exposants en transes en les menaçant de ses foudres si ces derniers continuaient à «tenter» de lui faire de l'ombre avec leurs jouets, «qui ne valent même pas un million de dollars», a-t-il ajouté. En France, tout le monde en rirait encore. Là, une conférence fut aussitôt organisée pour discuter de cet aspect des choses. Non, vous ne m'avez pas compris, pas la mise au chômage de l'imprimeur, mais savoir s'il est moral de gêner un marché de plusieurs milliards de dollars, avec des outils qui en valent quelques milliers. Au moins, même si rien n'est sorti de cette causerie, la morale est sauve.

La grande lutte des ordinateurs (autour des ordinateurs plutôt) s'est encore déplacée... inévitable sans doute. Maintenant, il ne s'agit plus que de 68020, 68030, 6840, 80386, etc. Mais alors, mon pauvre 68000 est complètement perdu, dépassé.

Bon, il est vrai qu'il l'était à sa sortie, mais enfin, que reste-t-il face à la vague des encore-plus-modernes, pour les pas-encore-classiques. Toujours est-il que de plus en plus de compagnies dévoilent des ordinateurs à base de 68020, 80386, le plus souvent sous forme de stations de travail autonomes et graphiques. La plupart (sinon tous) travaillent sous Unix (voir article du mois dernier), et possèdent un mode pour émuler... PC DOS. Décidément, ces systèmes lourds et limités ont encore de beaux jours devant eux. Les nouvelles machines ont donc de vrais 32 bits, permettent le multi-tâche et sont, ma foi, fort rapides. Mais, les vrais 64 bits sont déjà en étude poussée, sur des systèmes de recherche. Alors, encore une fois, le passage à de plus gros processeurs est une simple étape. Peut-être faudra-t-il un jour ralentir le processus, car on s'y perd un peu.

La société Maynard vient d'annoncer un contrôleur de disque dur permettant de doubler la capacité de stockage d'un disque. Cela permettrait, sans changer de disque, de passer par exemple de 100 Mb à 197 Mb. Un beau progrès si cela se confirme être fiable. En effet, la course aux stockages immenses commençait à s'essouffler un peu, avec des pointes vers le Gigabit. En théorie, cette interface permet directement de passer à 2 Gb.

Le COMDEX vient d'avoir lieu et à cette occasion, deux points importants peuvent être soulignés. Tout d'abord, ces expositions, comme en France le SICOB, ne sont plus suivies avec autant de ferveur que par le passé (proche). Les gens n'y vont que s'ils sont motivés, et des compagnies importantes (Apple et Ashton-Tate cette année par exemple) ne se déplacent plus, leur cible commerciale se déplaçant peu. Il y a donc une prise de conscience de ces sociétés, qui ne croient plus indispensable d'investir

temps et argent dans ces salons. La baisse de fréquentation s'en est d'ailleurs légèrement trouvée augmentée. Le second point est, à l'inverse, la présence de nouveaux produits sur de nombreux stands. Beaucoup de sociétés, à tendance «hard» confirmée trouvent encore au COMDEX l'audience qu'elles recherchent, et en profitent. Il faut donc voir là un retour aux sources de ce COMDEX, très électronique à l'origine. Citons, au hasard des allées, quelques-uns de ces produits :

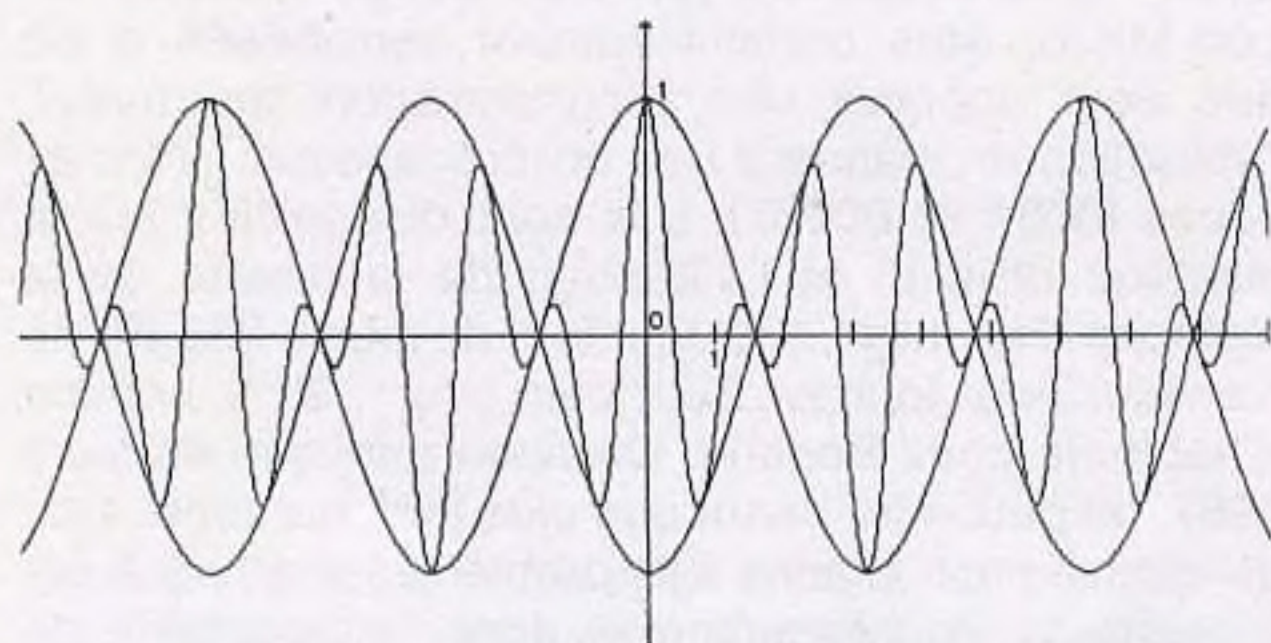
- Microsoft annonce une nouvelle gamme d'intégrés, et deux nouveaux langages, ou nouvelles versions de langages, le tout pour PC et compatibles.

- Intel propose sur son stand plusieurs réalisations d'autres sociétés à base de ses processeurs 80386 et 82786. Ils représentent le futur pour cette société qui, rappelons-le tout de même, est en général plagée par Nec quelques semaines plus tard et ce, à moindre prix. Parmi les produits, des Compaq, Convergents, Digital Research, Microsoft et Quadram. Certains sont déjà commercialisés, pour moins de 4 000 \$, le prix d'un AT de base.

- Borland, la société de Philippe Kahn, annonce et montre *Turbo Basic*, pour PC et compatibles. De plus, deux autres programmes de moindre intérêt y sont annoncés. Turbo Basic est interactif, mais très proche de l'esprit Borland, qui avait pris naissance avec le Turbo Pascal. La récursivité est implantée dans ce Basic, la structuration (et vous autres lecteurs de Led-Micro, êtes particulièrement sensibilisés à ce que peut apporter une programmation structurée), l'utilisation en standard des co-processeurs arithmétiques (8087 et 80287), s'ils sont disponibles sur la machine utilisée, et l'utilisation de la totalité de la mémoire disponible. Il lui faut tout de même 256 Ko de mémoire pour tourner. Tout cela pour... 99 \$, comme d'habitude chez Borland. Livraison effective en mars 1987, ou peut-être beaucoup plus tard, car repousser de quatre mois signifie des problèmes majeurs à éliminer dans le programme et donc l'impossibilité de proposer une vraie date. Simplement, cela permet de faire attendre les gens, sans qu'ils soient trop tentés d'acheter un autre produit. Borland montrait aussi un programme de résolution d'équations, *Eureka The Solver*, assez proche de ce qu'était TK Solver, c'est-à-dire un programme puissant, simple et agréable. prix : 99 \$. Enfin, *Turbo Pascal Numerical Methods* est une bibliothèque d'utilitaires mathématiques, avec démonstrations sur la disquette. Prix :... 99 \$. Le vrai Prix-Unique, c'est Kahn !!

Le développement de l'utilisation des micros chez les ingénieurs et scientifiques rendait obligatoire une révision de certaines politiques de logiciels chez les gros distributeurs. C'est presque chose faite, puisque Lotus et Borland viennent de faire l'annonce de pro-

grammes scientifiques de différents types. Lotus commercialise un logiciel de traitement de texte scientifique, *Manuscript*, et un programme de gestion de données scientifiques provenant d'appareils de mesures, *Measure*. Borland proposera bientôt un résolveur d'équations, nous venons de le voir. Cela montre que le marché potentiel de ce type de programmes, qui restaient jusqu'alors l'apanage de minuscules sociétés, avec le risque de manque de longévité qui s'y rattache, devient enfin suffisant pour intéresser de grosses sociétés, et donc permettre d'acheter des produits sans trop se poser la question du suivi. Enfin, cela pourrait aussi signifier à court terme une standardisation des formats et des algorithmes, ce qui ne pourrait qu'être bénéfique dans un premier temps, pour assainir les traitements scientifiques toujours un peu «aléatoires» sur micros. Enfin, cela pourrait aussi signifier la fin du règne des tableurs, qui sont de bien pauvres outils de travail, masquant les algorithmes, ne permettant pas de voir les hypothèses de départ et les résultats obtenus simultanément. Alors, on peut au moins espérer que de nouveaux efforts soient fournis dans ce domaine. Mais les petits fournisseurs de logiciels, on s'en doute, sont très inquiets, et les certitudes qu'ils affichent ne peuvent masquer leur crainte quand au proche avenir.



Un exemple de sortie avec TK Solver.

Ce mois-ci, notre intérêt sera concentré sur les services électroniques, messageries et autres connexions proposées par les réseaux à accès libre (entendez : libres si vous payez). Le développement spectaculaire de ces services aux USA ne peut masquer la déconvenue des gestionnaires, qui voient plafonner le nombre de leurs adhérents depuis maintenant au moins deux ans, malgré des tentatives dans tous les domaines pour allécher les clients potentiels. Voilà les données de base du problème, nous allons maintenant voir un peu plus les détails. Tout d'abord, que vous faut-il pour vous connecter ? Un ordinateur, de n'importe quel type, un modem, un programme de

communication, une ligne téléphonique et une carte de crédit, sur un compte (très) bien approvisionné. Vous êtes maintenant membre potentiel d'une des centaines de services électroniques, bases de données, conférences privées ou publiques, informations horaires pour les compagnies aériennes, programmes du domaine public, courrier électronique... Cela paraît alléchant, mais pourtant, chez les non-utilisateurs (possédant un ordinateur et un modem bien sûr), les réticences augmentent chaque jour. 37 % des propriétaires d'ordinateurs ont un modem, et moins de la moitié de ceux-ci ont utilisé un des ces services électroniques. Pourquoi ? Il semble que le point capital soit la difficulté apparente qui rebute les non-initiés. En effet, ces systèmes semblent parfois d'un autre âge, à l'époque des multi-fenêtres, du graphisme et de la couleur. Que répondre à un message du type :

- QXY please :

Alors, la plupart calent dès ce point. On les comprend. D'autant que ceux qui s'entêtent voient arriver par la suite le prix de cet acharnement, la facture du téléphone et du service lui-même. En effet, une heure est vite passée, et à ce prix, Crésus y réfléchirait. Le prix moyen est de 15 à 20 \$ l'heure, pour la première heure, et pour ceux qui continuent (??), les suivantes sont environ à 10 \$. La réponse des sociétés qui gèrent ces services est que cela coûte moins cher d'obtenir des informations par ce biais que de les chercher durant des heures ailleurs. C'est sans doute vrai, mais seuls quelques professionnels sont concernés, et les particuliers peuvent se passer de la plupart des services offerts. Ils sont souvent intéressés par les discussions techniques concernant leur matériel, ou des nouvelles en général, et c'est tout. Alors, payer 100 \$ par mois de liaison n'est vraiment pas justifié. En résumé, les clients actuels sont souvent des cadres qui sont connectés depuis l'entreprise qui les emploie, et dont ce système représente une amélioration de leurs possibilités de travail.

Les services les plus connus (mais ils ne sont que l'arbre qui cache la forêt) sont CompuServe, The Source, Dow Jones, GENie. Ils comportent des informations générales et spécialisées (business, scientifiques), des boîtes aux lettres électroniques, des cours de la bourse, des programmes du domaine public, des conférences spécialisées (les fameux SIG, Special Interest Groups). En général, l'adhésion coûte 20 \$ par an (The Source : 10 \$ par mois), et la connexion est ensuite à l'heure ou au forfait (Dow Jones : 50 \$ à chaque connexion).

En conclusion, il semble qu'à ce prix, et devant le peu de simplicité des services, le client potentiel continue à être réticent. C'est bien dommage, car comme d'habitude, il est des occasions qui ne se rattrapent pas et la perte est immense pour les services électroniques. Alors, un petit effort, messieurs.



LA BIBLIOTHEQUE TECHNIQUE DES EDITIONS FREQUENCES

offre des ouvrages techniques très actuels rédigés par des auteurs passionnés et impliqués complètement dans le sujet qu'ils traitent.

Vous trouverez soit des études approfondies sur les techniques ou les technologies de votre métier, soit des initiations théoriques et pratiques de techniques ou technologies que vous désirez approcher ou mieux cerner. Vous découvrirez au verso la description des ouvrages récemment parus ainsi que les commentaires sur les additifs d'éventuelles rééditions.

Les titres dont la parution est prochaine sont également mentionnés.

La page suivante comporte la liste complète des titres, leurs codes et leurs prix.



Vient de paraître :

LES MONTAGES ELECTRONIQUES

(SAVOIR, PRATIQUER, INVENTER) de Jean-Pierre Lemoine

Une nouvelle manière d'aborder l'étude et la pratique de l'électronique

La finalité de cet ouvrage est de permettre à chacun, passionné d'électronique, de concevoir ses montages lui-même.

Or, à notre époque où la technologie évolue sans cesse, il importe, principalement dans ce domaine, de bien connaître la majorité des composants mis en œuvre et les différentes façons de les utiliser. Devant l'ampleur du sujet, peu d'alternatives possibles, le savoir passant par la prise de conscience nette et délibérée de stockage, voire de mémorisation d'un maximum d'informations relatives aux différents matériels et schémas types d'exploitation.

Par quelques 1 000 dessins et représentations divers, répartis sur plus de 380 figures, ainsi que par la description de 25 montages originaux, cet ouvrage représente un véritable outil de travail, permettant de familiariser l'électronicien avec les études électroniques, afin de l'amener, grâce à la somme de renseignements fournis, à concevoir et élaborer lui-même des réalisations personnalisées.

Le sommaire :

Première partie : Connaître		Troisième partie Inventer	
Chapitre I : Les principaux composants, technologie et emploi	9	Chapitre I : Naissance d'une idée	207
Chapitre II : La documentation nécessaire	47	Chapitre II : De la théorie à la pratique	209
Chapitre III : Ces composants qui reviennent souvent - Caractéristiques principales	51	Chapitre III : Matérialisation de l'idée	215
Chapitre IV : Principaux brochages	59	Chapitre IV : Optimisation des montages d'étude	219
Chapitre V : Ces schémas de base qu'il faut connaître ou posséder	65	Chapitre V : Les relevés de schéma	229
Chapitre VI : La détermination simple des inductances et transformateurs	117	Chapitre VI : Réalisation de l'appareil	233
Chapitre VII : Ces matériels exotiques qui nous entourent	127	Quatrième partie Conclusion	
Deuxième partie Pratiquer		Chapitre I : Naissance de l'idée	239
Chapitre I : L'outillage	149	Chapitre II : De la théorie à la pratique	241
Chapitre II : Le matériel d'expérimentation	151	Chapitre III : Matérialisation de l'idée	247
Chapitre III : Le matériel de réalisation des circuits imprimés	157	Chapitre IV : Expérimentation sur table d'essais	251
Chapitre IV : Les appareils de laboratoire vraiment indispensables	161	Chapitre V : Réalisation pratique	255
Chapitre V : Les appareils de mesure et de laboratoire bien utiles	167	Chapitre VI : Finition - Essais - Réglages	259
Chapitre VI : La table et le support travail	169	Annexes	263
Chapitre VII : Le stock - Classement et rangement du matériel	173		
Chapitre VIII : Le classement et le rangement de la documentation technique	177		
Chapitre IX : Les sigles et autres repères des broches des circuits intégrés	179		
Chapitre X : Les lots de composants du commerce spécialisé	181		
Chapitre XI : Les transformations et équivalences toujours possibles	183		
Chapitre XII : Par où commencer ? Quelques montages d'initiation	185		
Chapitre XIII : L'expérimentation - Ces cartes standards qui font toute la différence	195		
Chapitre XIV : Les différents contrôles et mises au point	203		

LE TELEPHONE ET LES RADIOTELEPHONES

Roger-Ch. Houzé

Le sommaire :

Préface	3	Chapitre VI : Les fibres optiques	27
Chapitre I : La liaison téléphonique duplex - Le réseau commuté	5	Chapitre VII : Le combiné téléphonique	33
Chapitre II : Le multiplexage des voies téléphoniques par courant «porteur» et par MIC	9	Chapitre VIII : Le radiotéléphone	49
Chapitre III : La ligne téléphonique	15	Chapitre IX : Le récepteur d'appel (Eurosignal)	67
Chapitre IV : La ligne aérienne	17	Chapitre X : Historique du téléphone	73
Chapitre V : Le réseau câblé	19	Chapitre XI : L'actualité du téléphone - La nouvelle numérotation	79

PROCHAINEMENT

**Collection
jaune**

Les Bases de l'Electronique de Raymond Breton
Etude autour du 6809 (constructions et logiciels) de Claude Vicidomini

**Collection
noire**

Les Techniques du Son (1^{er} tome) : collectif d'auteurs dirigé par Denis Mercier -
La Synthèse musicale par ordinateur de Frédéric Levé -

Collection noire (format 165 x 240)

LES SYNTHETISEURS, UNE NOUVELLE LUTHERIE de Claude Gendre - 184 p. - 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 ressentir. Le vœu est comblé, en 180 pages

LES HAUT-PARLEURS de Jean Hiraga - 320 p. - 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

INTRODUCTION A L'AUDIO-NUMERIQUE de Jean-Pierre Picot - 160 p. - C'est le premier ouvrage paru en langue française traitant de l'audio numérique ; écrit par un professionnel, avec rigueur et simplicité, il explique brillamment les bases de cette technique : quantification, conversion, formats, codes d'erreurs

L'OPTIMISATION DES HAUT-PARLEURS ET ENCEINTES ACOUSTIQUES de Charles-Henry Delaleu - 240 p. - Seconde édition améliorée d'un ouvrage fort attendu des passionnés 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

LES MAGNETOPHONES de Claude Gendre - 160 p. - 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

LES MAGNETOSCOPES ET LA TELEVISION de Claude Gendre - 256 p. - 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

L'ELECTRONIQUE DES MICRO-ORDINATEURS de Philippe Faugeras - 128 p. - 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 : mémoire, clavier, écran, interfaces de toutes sortes

PERIPHERIQUES : INTERFACES ET TECHNOLOGIE de Philippe Faugeras - 136 p. - Faisant suite à la parution de «L'électronique des micro-ordinateurs», cet ouvrage s'adresse aux électroniciens désireux de s'initier aux montages périphériques des micro-ordinateurs, interfaces en particulier, qui permettent la communication avec monde extérieur

SELECTION DE L'AUDIOPHILE - TOME 1 : L'ELECTRONIQUE 256 p.

SELECTION DE L'AUDIOPHILE - TOME 2 : LES TRANSDUCTEURS 256 p.

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 transistors. Dans un esprit identique, le tome 2 traite du domaine passionnant que constituent les transducteurs en audio.

LE MINI STUDIO de Denis Fortier - 160 p. - 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 physiques indispensables, le choix des maillons, la manière d'installer et d'exploiter

Réf.	Prix TTC
E 15	140 F
E 01	165 F
E 05	155 F
E 04	154 F
E 02	92 F
E 03	155 F
E 06	150 F
E 22	150 F
E 12	155 F
E 13	165 F
E 25	140 F

Collection rouge (format 135 x 210)

CONSEILS ET TOURS DE MAIN EN ELECTRONIQUE de Jean Hiraga 160 p. - Le «dernier coup de patte» apporté à un montage, 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»

LES LECTEURS DE COMPACT-DISCS 200 p. - 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

LEXIQUE DE L'ELECTRONIQUE ANGLAIS-FRANÇAIS de Jean Hiraga - 72 p. - 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

FILTRES ACTIFS ET PASSIFS POUR ENCEINTES ACOUSTIQUES de Charles-Henry Delaleu - 160 p. - 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 !

17 MONTAGES ELECTRONIQUES de Bernard Duval - 128 p. - 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

WEEK-END PHOTO de Philippe Folie-Dupart - 208 p. - 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

L 07	68 F
L 10	130 F
L 09	65 F
L 11	85 F
L 14	95 F
L 20	130 F

Collection jaune (format 210 x 270)

INITIATION A LA ROBOTIQUE 96 p. - 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 et les fans de cybernétique enfin réunis ! (épuisé)

INITIATION A LA MICRO-INFORMATIQUE COURS 1^{er} CYCLE - LE VOLUME 1 de Claude Polgar - 272 p.

INITIATION A LA MICRO-INFORMATIQUE COURS 1^{er} CYCLE - LE VOLUME 2 de Claude Polgar - 208 p.

INITIATION A LA MICRO-INFORMATIQUE COURS 1^{er} CYCLE - LE VOLUME 3 de Claude Polgar - 250 p.

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.

INITIATION A L'ELECTRONIQUE DIGITALE de Philippe Duquesne - 104 p. - Ce cours d'initiation à l'électronique digital 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

INITIATION AUX MICROPROCESSEURS de Philippe Duquesne - 136 p. - 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

INITIATION TV : RECEPTION, PRATIQUE, MESURES, CIRCUITS de Roger-Charles Houzé - 136 p. - 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

INITIATION A LA MESURE ELECTRONIQUE de Michel Casabo - 120 p. - 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

INITIATION AUX AMPLIS A TRANSISTORS de Gilles Le Doré - 96 p. - 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.

INITIATION AUX AMPLIS A TUBES de Jean Hiraga - 152 p. - 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

INITIATION A L'ELECTRICITE ET A L'ELECTROTECHNIQUE de Roger Friederich - 110 p. - 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 !

INITIATION A LA VIDEO LEGERE - THEORIE ET PRATIQUE de Claude Gendre - 72 p. - 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 n'a pas d'équivalent en librairie aujourd'hui

LES MONTAGES ELECTRONIQUES de Jean-Pierre Lemoine - 276 p. - Véritable encyclopédie. Plus de 1 000 dessins. 25 montages originaux

LE TELEPHONE ET LES RADIOTELEPHONES de Roger-Charles Houzé - 96 p., 73 schémas

P 08	115 F
P 16	130 F
P 17	130 F
P 27	190 F
P 19	95 F
P 18	95 F
P 21	135 F
P 23	140 F
P 24	130 F
P 26	155 F
P 28	150 F
P 29	100 F
P 30	250 F
P 31	130 F



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"/>	(épuisé) 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"/>	P 30 <input type="checkbox"/>
P 31 <input type="checkbox"/>									

Frais de port : + 12 F par livre commandé, soit la somme totale ci-jointe, de Frs par CCP Chèque bancaire Mandat-lettre

Nom Prénom

Adresse

Ville Code Postal

Micro 1

QUOI DE NEUF EN 87 ?

S'il ne fait plus aucun doute que le standard IBM est devenu la Référence, il faut bien admettre que les premiers PC et autres XT étaient des machines somme toute ordinaires.

En fait, la fin de l'année 86 annonce de bien bonnes choses pour l'acheteur de 87. Deux nouveautés marquantes se présentent à l'horizon. Les premières machines équipées du tout nouveau microprocesseur Intel arrivent. En même temps, le BASIC devient un langage noble.

Ils sont plusieurs fabricants à effectuer toutes sortes de premières, avant la première. Des bancs d'essai de machines pilotées par 80386 sont même présentées au public. Toutefois, les premières machines ne seront pas livrées avant un ou deux mois. Le microprocesseur Intel 80386 est un processeur 32 bits qui hisse les performances d'un micro à la hauteur de celles d'un très bon mini. Les différences de vitesse entre un PC version originale et une machine équipée d'un 80386 sont très importantes. De plus, la plupart des constructeurs en lice proposent toutes sortes d'architectures qui vont en direction d'une machine multitâche. Les blocs mémoire se retrouvent câblés en page mémoire, apparition d'antimémoires, suppression des files d'attente au niveau du microprocesseur, etc. Ainsi, grâce à ces améliorations, un ordinateur de bureau (on n'ose plus dire, un micro-ordinateur) peut offrir des performances allant de 3 à 4 millions d'instructions par seconde, ceci est comparable à un VAX 11/780 de DEC qui ne passe pas pour un petit mini-ordinateur.

Malheureusement, les premières machines seront livrées avec le MS-DOS 3.1. Ce système d'exploitation ne correspond en aucun cas à ce type de machine. Microsoft a annoncé la version 4.0 de MS-DOS au dernier SICOB, et l'on parle déjà de MS-DOS 5.0. Ces systèmes d'exploitation sont des multitâches. Espérons donc leur diffusion rapide. Il convient de noter que MS-DOS 5.0 ne pourra tourner que sur les microprocesseurs 80286 et 80386. Il faudra donc encore quelques mois pour passer à la vitesse supérieure.

Comme vous l'avez sûrement lu dans beaucoup de livres et autres articles, le BASIC est un langage nul qu'il ne faudrait jamais avoir appris. Pourquoi cette provocation ? Tout simplement à cause des GOTO 50 et autre GOSUB 300. Eh bien, c'est fini ! Microsoft vient de sortir le QUICK BASIC. On retrouve ici les mêmes problèmes que pour les machines. Si les 8086 et 8088 étaient devenu indétrônables en face des 68000 beaucoup plus performants, il faut bien admettre que le BASICA et le GWBASIC sont bien lourdauds. Si le BASIC est apparu dès 1965, son architecture n'a pas beaucoup évolué depuis. Les

seules améliorations dont on puisse parler concernent l'augmentation quotidienne du nombre d'instructions. Il aura fallu attendre vingt ans pour qu'arrive la structuration.

Pourtant il faut être honnête, Hewlett Packard disposait déjà depuis plus de dix ans d'un BASIC hyper-structuré et très puissant. Il faut croire qu'il fut réservé uniquement au client de la marque.

Vu d'un autre côté, on pourrait croire que Microsoft a voulu amortir au maximum son BASIC. Un BASIC Hewlett-Packard série 9845, 200, 300, etc., est aussi puissant qu'une addition du BASIC + FORTRAN + ALGOL + PL/I. A ceci il convient d'ajouter les très nombreux ordres graphiques, d'entrées-sorties, et autres contrôles diverses. Microsoft présente son QUICK BASIC comme le BASIC du troisième type. C'est sans doute un peu fort. Nous dirons plus simplement que le BASIC version Microsoft devient enfin structuré et modulaire.

Ce qui change avec le QUICK BASIC :

Dans un premier temps, l'utilisation du QUICK BASIC rappelle WINDOWS. Mais ce qui saute le plus aux yeux, c'est la disparition des numéros de lignes. Ensuite, on fera connaissance avec les nouvelles commandes et les instructions étendues. Notons, entre autres, l'apparition de tableaux dynamiques, l'instruction CALL ne sert plus à appeler uniquement des routines en langage machine, mais aussi des sous-programmes internes, externes, ou dans une librairie utilisateur. Enfin QUICK BASIC possède des métacommandes qui donnent des directives au compilateur pour l'allocation mémoire des tableaux, ou pour inclure un autre programme source dans le programme courant. Il existe toute une série d'options de compilation facilitant la mise au point.

Il ne fait aucun doute que 1987 annonce une nouvelle phase dans le monde des micro-ordinateurs. Avec la commercialisation des microprocesseurs 68020 et 80386, la naissance d'un BASIC structuré pour le grand public, et l'utilisation de langages évolués de type «C» sur petite machine, le fossé qui séparait la «grande informatique» de la micro est en train de disparaître. Que dire de ces informaticiens qui ne voyaient aucun avenir à la micro. Après deux, trois années où nous n'aurons connu aucune nouveauté technique spectaculaire, mais des baisses de prix impressionnantes, il faut bien admettre que nous repartons dans une folle course avec Monsieur Plus.

Ch.-H. Delaleu

TOUT SUR LES PÉRIPHÉRIQUES



- 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 reponsables 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 technique (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 Faugas, 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 Faugas est l'auteur d'un premier ouvrage «L'électronique des micro-ordinateurs» paru aux Editions Fréquences.

BON DE COMMANDE

Diffusion auprès des libraires assurée exclusivement par les Editions Eyrolles.

Je désire recevoir l'ouvrage «**Périphériques interfaces et technologie**» au prix de 162 F (150 F + 12 F de port).

Nom

Adresse

Règlement ci-joint :

Par chèque bancaire par chèque postal par mandat

Des progiciels PRO...

La ligne MAJOR de chez SAARI !

Charles-Henry Delaleu

La société SAARI propose, pour les sociétés, deux lignes de trois produits. Il s'agit de la ligne standard (5 000 F environ le progiciel) et de la ligne major (10 000 F le progiciel). Chacune de ces gammes comprend :

- un logiciel de COMPTABILITE
- un logiciel de GESTION COMMERCIALE
- un logiciel de PAIE.

Ces programmes existent pour le standard IBM PC, XT et AT, ainsi que pour la famille APPLE. Il s'agit de progiciels qui ont fait leurs preuves et qui se situent parmi les tout premiers en nombre d'exemplaires diffusés. Il nous a paru intéressant de faire une brève description de ces produits dans nos colonnes. En effet, il existe dans notre lectorat de nombreuses personnes qui travaillent dans des sociétés de petite taille ou de taille moyenne pour lesquelles les progiciels SAARI permettront une gestion saine et en temps réel de l'ensemble de la maison. Nous restons à la disposition des personnes qui souhaiteraient de plus amples renseignements.

Avertissement : Il convient de noter que pour toutes applications professionnelles, l'utilisation d'un disque dur en mémoire de masse est une nécessité. Dans le cas d'une utilisation intensive, l'implantation d'un streamer n'est pas un luxe.

LA GESTION COMMERCIALE SAARI

Ce progiciel s'adresse aux PME/PMI de 10 à 50 personnes. Il est divisé en trois parties distinctes :

- la gestion des stocks
- la gestion des clients
- la gestion des commandes et de la facturation.

Ce programme, qui est un intégré, traite les informations en temps réel. L'établissement d'une facture entraîne automatiquement la mise à jour du chiffre d'affaires, des quantités vendues, etc.

Il permet de gérer 2 500 articles, 1 000 clients et 500 factures par mois sur un XT. Cela passe à 5 000 articles, 2 000 clients et 1 000 factures/mois pour un AT. Si le système dispose d'un disque dur de grande capacité, les possibilités passent à 10 000 articles et 10 000 clients.

La gestion commerciale SAARI assure la gestion des fiches clients et des différents articles, la saisie et l'édition de devis, commandes, confirmations, bons de livraison et factures et la gestion complète des règlements clients. Elle autorise les statistiques de ventes, les coefficients de rotation des articles, les marges par client et par article.

Chaque donnée est paramétrable, ainsi ce progiciel peut s'adapter à de très nombreux cas. Un pont existe avec le progiciel de comptabilité, permettant ainsi des mises à jour automatiques.

Ce programme est très facile à utiliser. Il comporte des écrans multi-fenêtres, des aides à tous les niveaux, des fonctions de recherche rapide et une calculatrice intégrée au programme.

LA COMPTABILITE

Comme l'ensemble des programmes de la ligne major, la comptabilité peut être multi-société. Le nombre maximum de sociétés étant de 99. Cette comptabilité est très complète et assez puissante. Pour tenir votre comptabilité, vous disposez de journaux divisionnaires (ventes, achats, banque, caisse, etc.), des comptes généraux, des comptes auxiliaires, et si vous le désirez des comptes analytiques. Les comptes auxiliaires servent à gérer séparément les clients, les fournisseurs, etc.

La comptabilité n'est pas tenue en temps réel. Cela signifie que chaque transaction doit être validée avant d'être transférée dans la comptabilité. La fonction de clôture périodique permet d'arrêter définitivement un mois comptable. Il ne sera alors plus possible de saisir des écritures sur ce mois.

Comme pour les autres programmes de la ligne major, ce progiciel dispose d'un jeu d'essais permettant de se familiariser avec l'ensemble des commandes et leur utilisation.

Il existe d'origine un plan comptable livré avec le progiciel. Bien entendu, il est possible de réaliser son propre plan, ou d'y ajouter des comptes à ceux déjà existants.

LA PAIE

La paie GIPSI MAJOR de SAARI peut traiter jusqu'à 99 sociétés, 99 rubriques et 999 salaires par société. Bien entendu, il convient d'avoir un disque dur de taille nécessaire. Le progiciel est livré avec un plan de paie standard. Pour les paies spécifiques, il est possible de créer par société 99 rubriques entièrement paramétrables adaptées au type de paie à traiter : forfaitaire, horaire, à point, à la pièce, etc. Il est possible de définir plus de 100 constantes de paie. Les éléments de calcul des rubriques sont propres à chaque salaire.

L'ensemble des informations nécessaires à la DAS est réalisé automatiquement (présence et interruptions de travail, avantages en nature, frais professionnels, etc.).

La gestion complète des cotisations sociales (prise en compte des parts salariales et patronales et régularisation automatique des cotisations) est faite mensuellement ou en fin d'année au choix.

Un pont existe avec le progiciel de comptabilité afin de passer les informations nécessaires, sans refaire de saisie.

L'édition est très complète : bulletin de paie, livre de paie, états récapitulatifs, ordre de paiement, fiches individuelles de salaire, bilan social simplifié, DAS 1.

CONCLUSION

Il n'est en aucun cas possible de décrire en quelques pages l'ensemble des trois progiciels de la ligne MAJOR de SAARI. Aussi nous avons préféré les présenter sous forme d'organigrammes simplifiés sur les pages suivantes. Le but étant d'attirer le lecteur sur l'énorme avantage à automatiser un travail répétitif largement effectué en entreprise.

LA GESTION COMMERCIALE LIGNE MAJOR

GESTION DES STOCKS

1. Gestion du fichier articles
2. Mouvements de stocks
3. Gestion des tarifs
4. Edition des articles
 - 4.1. Etats généraux
 - 4.1.1. Fiches articles
 - 4.1.2. Catalogue des pièces
 - 4.1.3. Tarifs clientèle
 - 4.1.4. Etiquettes articles
 - 4.2. Etats de synthèse
 - 4.2.1. Catalogue général
 - 4.2.2. Catalogue par fournisseur
 - 4.2.3. Articles à réapprovisionner
 - 4.2.4. Articles à rotation lente
 - 4.2.5. Etat du stock réel
 - 4.3. Etats de gestion
 - 4.3.1. Marges brutes périodiques
 - 4.3.2. Marges brutes annuelles
 - 4.3.3. Ratios
 - 4.3.4. Balance périodique et annuelle
 - 4.3.5. Journaux des entrées et sorties
 - 4.3.6. Etat des mouvements des stocks
5. Clôtures périodique et annuelle

GESTION DES CLIENTS

1. Gestion du fichier clients
2. Mouvement des clients
3. Lettrage
4. Edition des clients
 - 4.1. Etats généraux
 - 4.1.1. Fiches clients
 - 4.1.2. Statistiques représentants
 - 4.1.3. Etiquettes clients
 - 4.1.4. Lettres de relance
 - 4.2. Etats de synthèse
 - 4.2.1. Répertoire général
 - 4.2.2. Répertoire par représentant
 - 4.2.3. Client en dépassement de crédit

- 4.2.4. Client à faible activité
- 4.2.5. Balance âge clients
- 4.3. Etats de gestion
 - 4.3.1. Marges brutes
 - 4.3.2. Balances périodique et annuelle
 - 4.3.3. Relevé des mouvements clients
 - 4.3.4. Journaux divers
 - 4.3.5. Etat des mouvements des clients
 - 4.3.6. Etat des mouvement lettres
- 5. Clôtures périodique et annuelle.

FACTURATION

- 1. Gestion manuelle des pièces
- 2. Gestion automatique des pièces
- 3. Edition de facturation :
 - 3.1. Liste résumée des pièces
 - 3.2. Relevés de factures
 - 3.3. Edition des pièces
 - 3.4. Edition des traites.

UTILITAIRES PROGRAMME

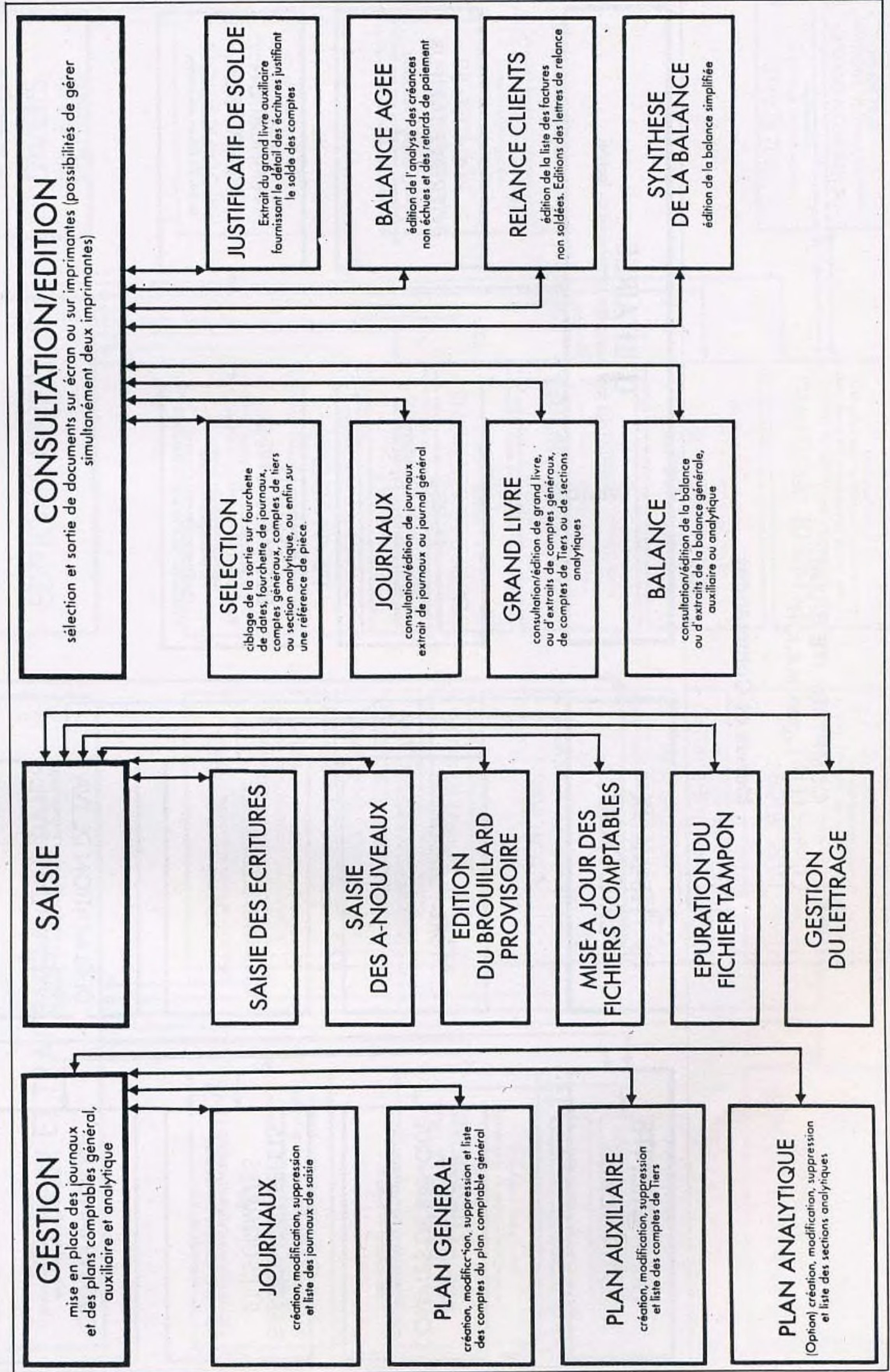
- 1. Sauvegarde
- 2. Pont comptabilité SAARI MAJOR
- 3. Occupation des fichiers :
- 4. Paramétrage des fiches articles :
 - 4.1. Catégories de TVA
 - 4.2. Taxes parafiscales
 - 4.3. Unités de vente
 - 4.4. Catégories d'articles
 - 4.5. Types d'articles
 - 4.6. Paramétrage des étiquettes
- 5. Paramétrage des fiches clients :
 - 5.1. Modes de règlement
 - 5.2. Modes d'expédition
 - 5.3. Représentants
 - 5.4. Catégories de clients
 - 5.5. Paramétrage des étiquettes
 - 5.6. Saisie des lettres de relance
- 6. Paramètres généraux :
 - 6.1. Edition des paramètres
 - 6.2. Séries de pièces
 - 6.3. Comptes comptabilité générale
 - 6.4. Numéro des journaux
 - 6.5. Libellés libres
 - 6.6. Purge des fichiers

UTILITAIRES SYSTEME

- 1. Paramètres système
- 2. Paramètres société
- 3. Mots de passe utilisateur
- 4. Sauvegarde et restauration
- 5. Organisation des fichiers
 - 5.1. Initialisation des fichiers
 - 5.2. Remise à zéro des fichiers
 - 5.3. Remise à zéro des valeurs
 - 5.4. Remise à zéro des représentants
 - 5.5. Modification des tailles des fichiers

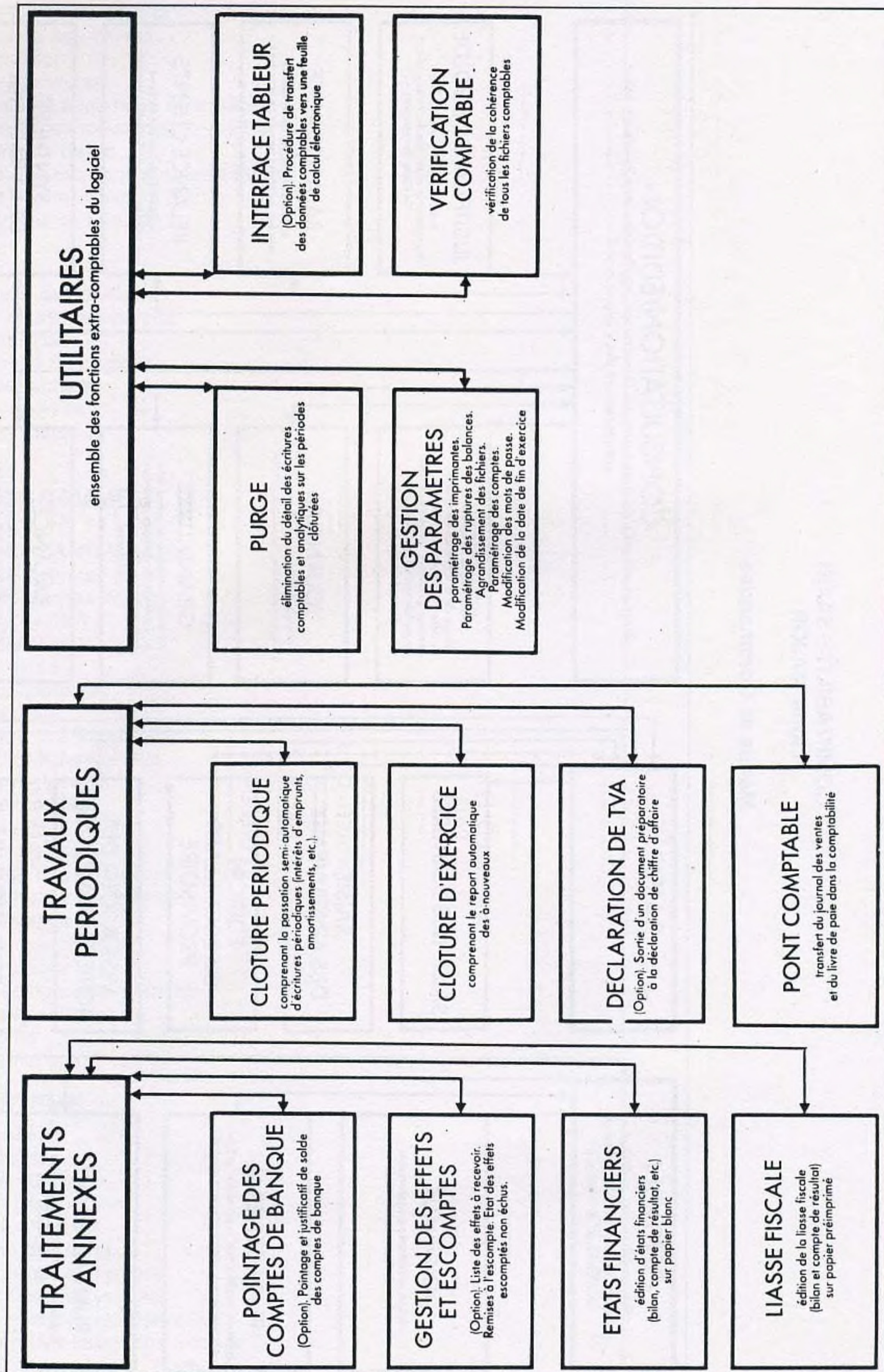
COMPTABILITE SAARI
Ligne MAJOR

Menus et Commandes



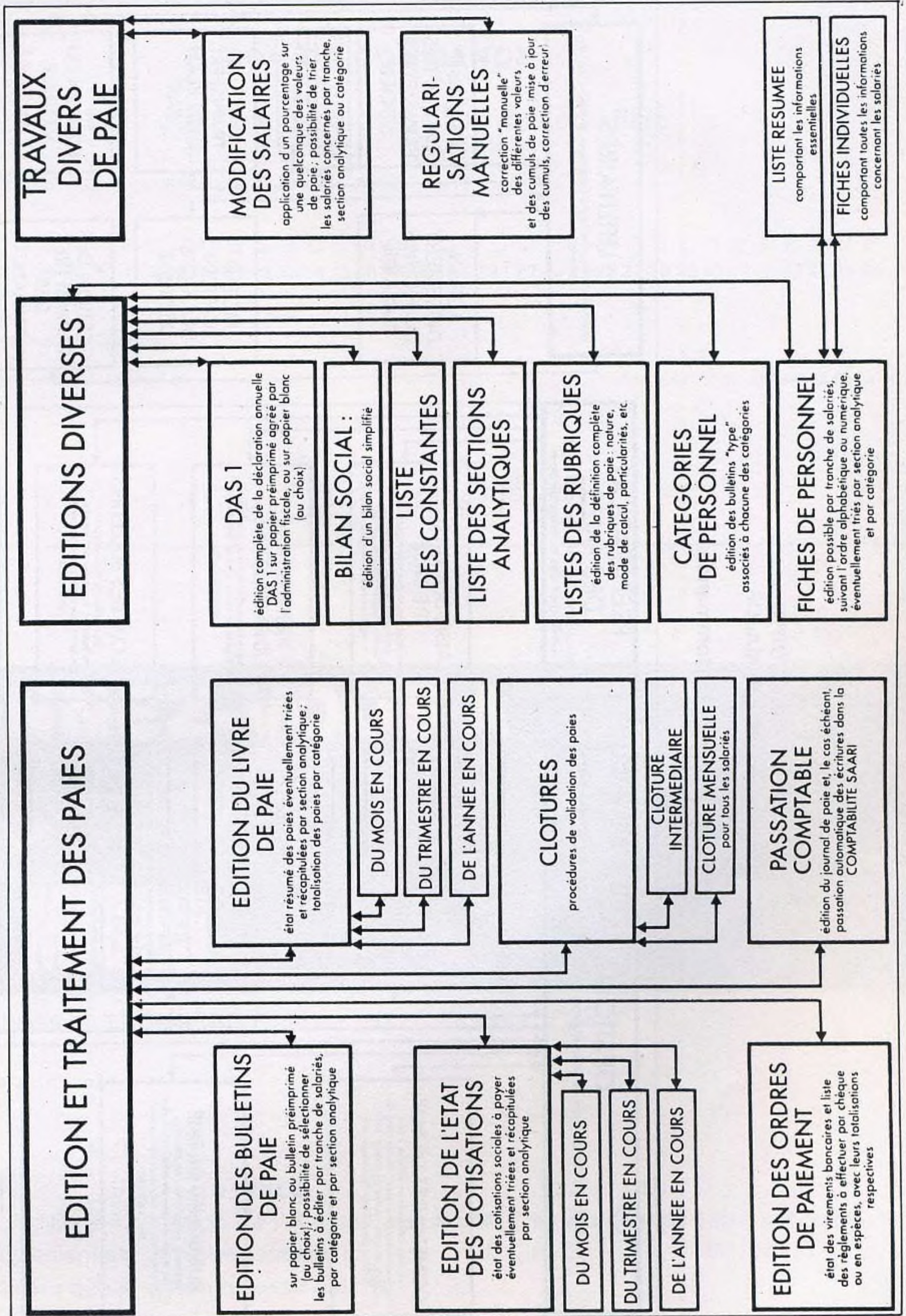
COMPTABILITE SAARI
Ligne MAJOR

Menus et Commandes



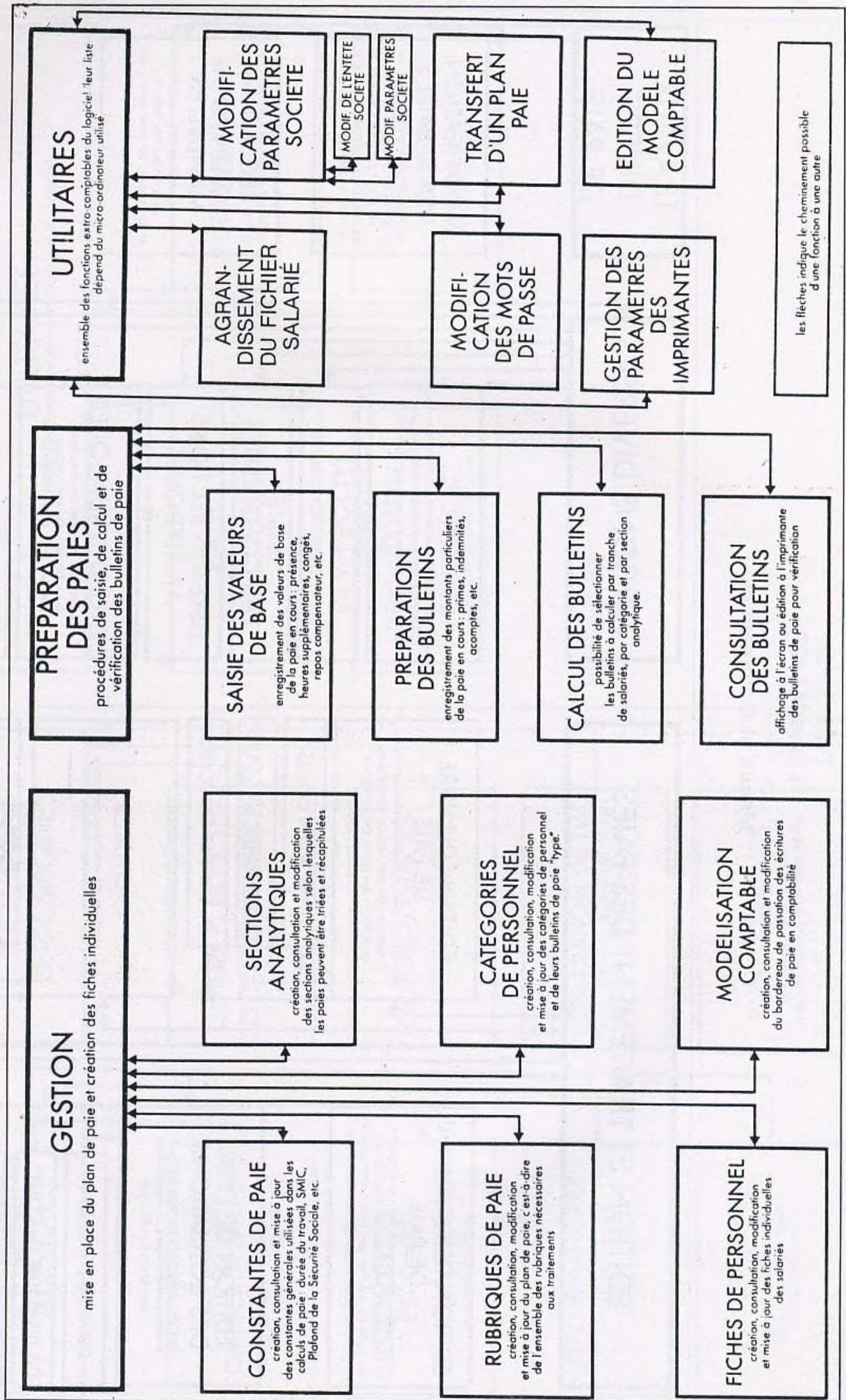
PAIE GIPSI
Ligne MAJOR

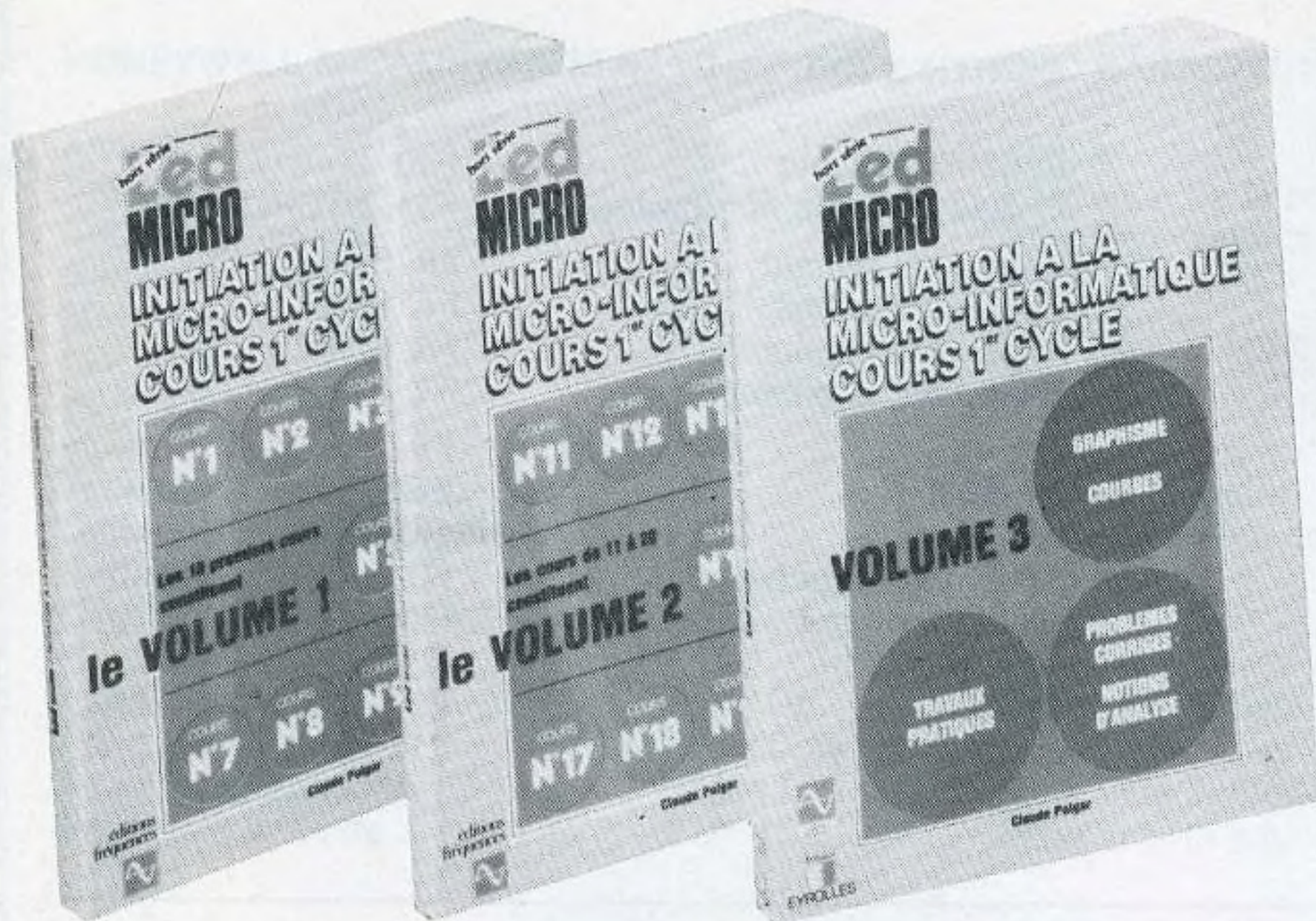
Menus et Commandes



PAIE GIPSI Ligne MAJOR

Menus et Commandes





**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 ENTIÈRE.
- ★ 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 !).

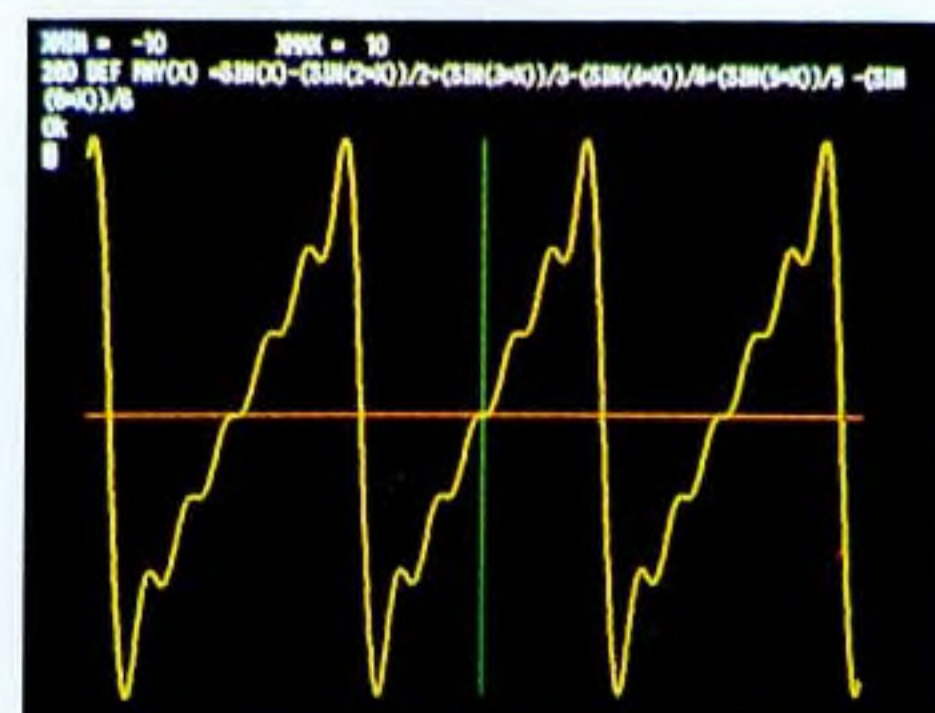
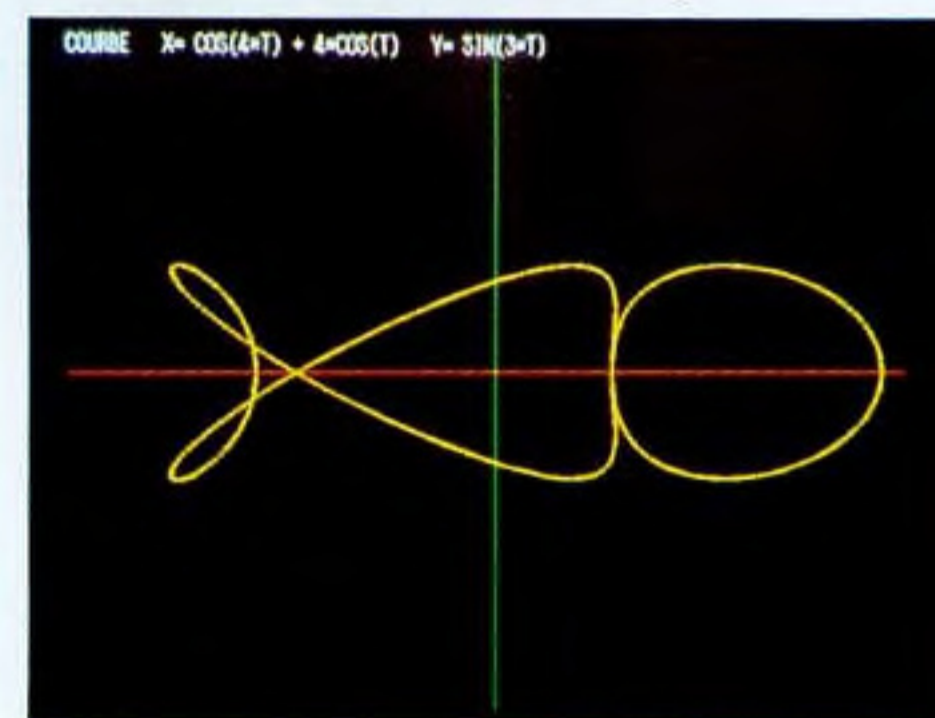
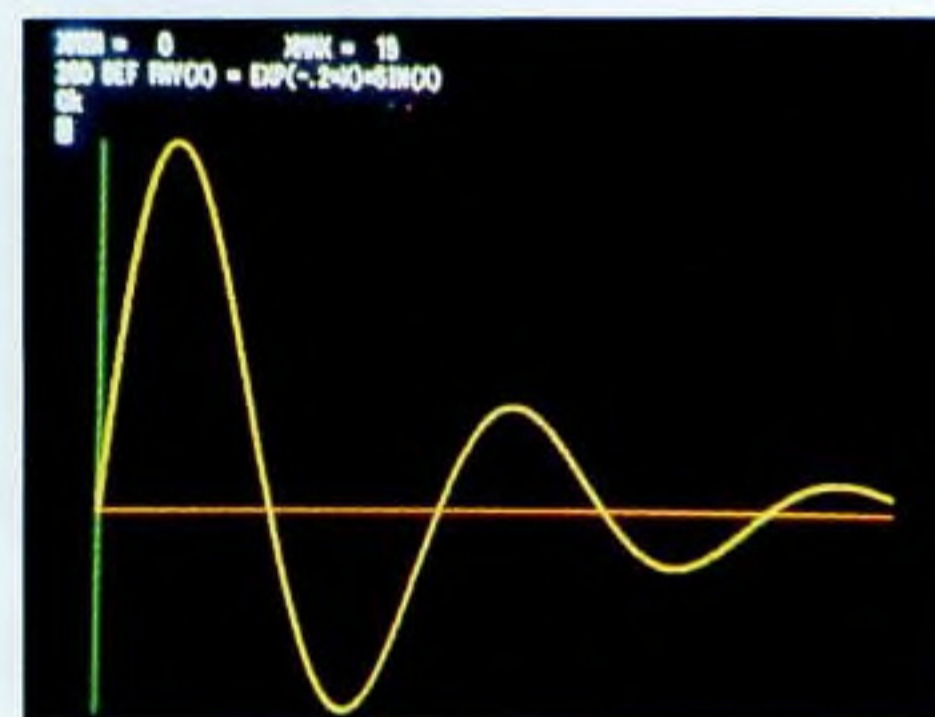
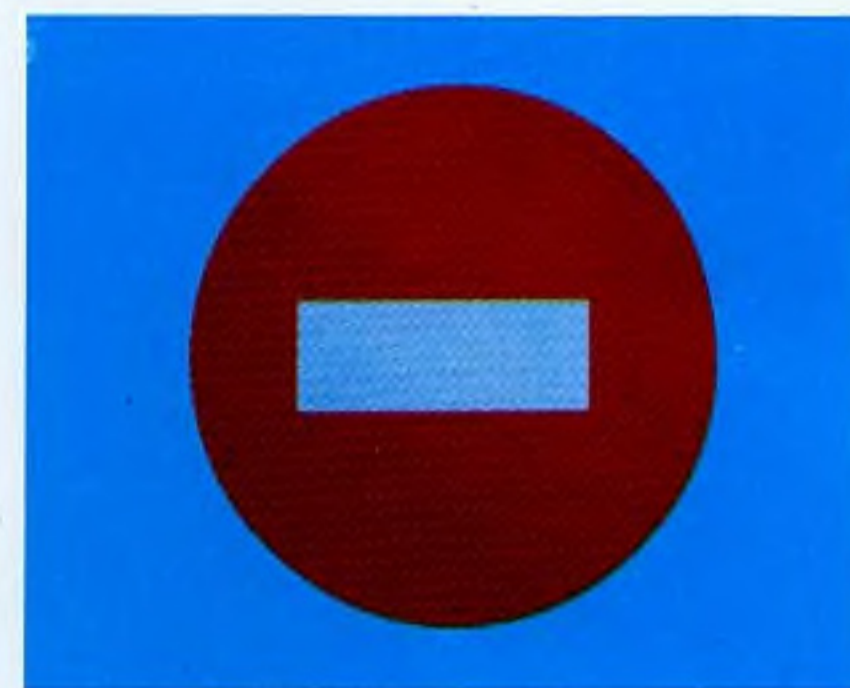
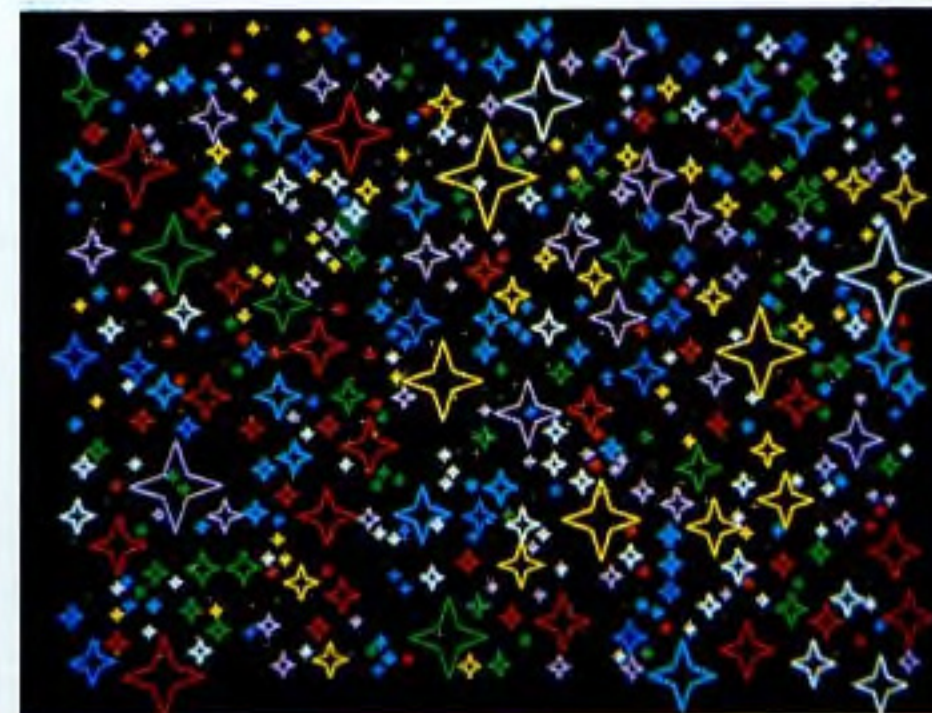
3.21. Techniques de mise au point

- ★ 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!



- *exploiter toutes les possibilités des systèmes MIDI*
- *réaliser vous-mêmes un clip vidéo*
- *tirer le maximum de vos synthétiseurs*
- *installer chez vous votre studio d'enregistrement*
- *tout savoir sur les nouveautés musique et vidéo créatives*

**Tout cela chaque mois
dans Music Vidéo Systèmes**

une publication des Editions Fréquences chez votre marchand de journaux

Editions Fréquences 1, boulevard Ney 75018 Paris - Tél. 46.07.01.97