

LOISIRS TECHNIQUES D'AUJOURD'HUI

hors série

Led

MICRO

PROGRAMMATION

COURS 2^{ème} CYCLE

ISSN 0757-6889

COURS

N°34

Suite
2^e cycle

N°14

**COURS DE
PASCAL**
la notion
de type

**COURS DE
PROGRAM-
MATION
APPROFONDIE :**
compilation et
interprétation

**COURS
D'INITIATION AU
PROGICIEL
MULTIPLAN**



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.

LOM...IQUES D'AUJOURD'HUI

hors série

LED

MICRO

PROGRAMMATION COURS 2^e CYCLE

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

LED MICRO
 (cours 2^e cycle)
 Mensuel : 18 F
 Commission paritaire : 64949
 Directeur de la publication :
 Edouard Pastor

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

**Services Rédaction-Publicité-
 Abonnements :**
 1, bd Ney, 75018 Paris
 Tél. : (1) 46.07.01.97
 Lignes groupées

Comité de rédaction :
 Dominique Chastagnier
 Jean-François 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



NOVEMBRE 86

COURS DE PASCAL de la page 4 à la page 16

- Rappel p. 5
 - Approximations
 - Les types non standard
 - Les types énumérés et intervalles
- Les types structurés p. 6
 - Introduction
 - Les tableaux en Pascal
 - Les tableaux compactés
- Les ensembles p. 14
 - Introduction
 - Le type d'ensemble
 - Déclaration d'un ensemble
 - Les opérations disponibles
- Des exercices p. 15

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

NOTRE COUVERTURE : La puissance de test et de mesure Hewlett-Packard associée à l'efficacité d'un ordinateur personnel. Cette dernière nouveauté transforme n'importe quel compatible en un laboratoire ultra-perfectionné tout en restant dans un budget raisonnable.

COURS DE PROGRAMMATION APPROFONDIE

de la page 20 à la page 29

- Avant-propos p. 21
- Les principes de bases p. 21
 - Editeur
 - Analyseur syntaxique
 - Interpréteur
 - Le compilateur
 - Semi-compilateur/interpréteur
 - Assembleur
 - Editeur de liens
 - Bibliothèque de procédures
- Conclusion p. 29

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

DIALOGUE AVEC NOS LECTEURS de la page 30 à la page 35

C'EST ARRIVÉ DEMAIN
 de la page 37 à la page 40

COURS D'INITIATION AU PROGICIEL MULTIPLAN de la page 41 à la page 49

- Fonction Somme p. 42
 - Fonction Nom p. 42
 - Les références relatives et les références absolues p. 43
 - Les exemples p. 44
 - Les données
 - Programmation en mode absolu
 - Programmation en mode relatif
- Charles-Henry Delaleu**

COURS DE PASCAL

Dominique Chastagnier
Jean-François Coblenz
Patrick Gueneau

L'approche du Pascal continue ce mois-ci, avec une récapitulation des notions vues le mois précédent, la description plus détaillée des types particuliers à ce langage, mais aussi l'étude des tableaux, les possibilités de compactages des données, et aussi de nouveaux exercices pour mettre à profit toutes ces nouveautés.

COURS N° 5

PLAN DU COURS

1. Rappel
 - 1.2. Approximations
 - 1.3. Les types non standard
 - 1.4. Les types énumérés et intervalles
2. Les types structurés
 - 2.1. Introduction
 - 2.2. Les tableaux en Pascal
 - 2.2.1. La déclaration
 - 2.2.2. Les tableaux multi-dimensionnels
 - 2.3. Les tableaux compactés
3. Les ensembles
 - 3.1. Introduction
 - 3.2. Le type d'ensemble
 - 3.3. Déclaration d'un ensemble
4. Des exercices

INTRODUCTION

Le mois dernier, nous avons décrit la création et l'utilisation de base des structures de types en Pascal. Comme nous l'avons alors signalé, nous continuerons ce mois-ci, en approfondissant ces connaissances.

1. RAPPEL

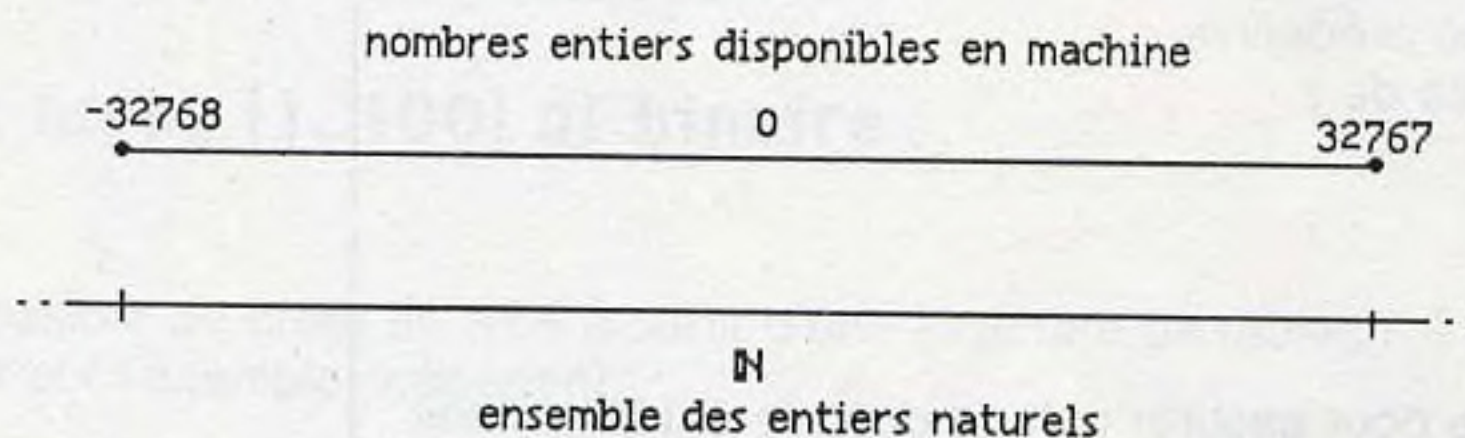
1.1. Il existe en Pascal quatre types standard qui sont :

- Integer
- Real
- Boolean
- Char

1.2. Approximations

Notons qu'en ce qui concerne les deux premiers, les interprétations des mots entiers et réels ne sont que de mauvaises approximations de ce que sont ces ensembles au sens mathématique du terme. En effet, ils sont en Pascal, comme pour tous les langages évolués, par essence, finis (pour les entiers), ou tout au moins bornés (les réels). Or, ceci est en opposition flagrante avec les structures mathématiques du même nom, mais ne doit plus vous surprendre, puisque nous l'avons déjà noté avec l'étude du BASIC.

Enfin, il faut ajouter que les réels correspondent plus en fait à la notion de nombres rationnels. Ceci est dû au fait que les nombres irrationnels, du type $\sqrt{2}$, sont représentés par des approximations numériques, par exemple 1.414... pour $\sqrt{2}$. Il est donc fréquent de trouver des aberrations dans les résultats numériques, si une très grande attention n'est pas donnée au programme. Pour plus de détails, l'article sur les implémentations des nombres en machine, des numéros 21 et suivants vous seront sans doute utiles, car ils expliquent en détail ces problèmes.



1.3. Les types non standard

En Pascal, il est possible de définir toute entité comme un type à part entière. Pour cela, il suffit d'appeler le mot TYPE, qui permet la définition de ce «type» de structure. Nous avons déjà vu comment le faire, et nous ne rappellerons que l'essentiel sur un exemple simple. Supposons que vous ayez besoin de manipuler les jours de la semaine, il vous sera alors possible de créer un type par :

```
type Jours_ouvrables = ( lundi, mardi, mercredi, jeudi,
vendredi )
```

Un autre exemple, pour qualifier une population :

```
type taille = ( petit, moyen, grand, tres_grand );
```

On le voit, la variété des types disponibles est alors sans fin.

1.4. Les types énumérés et intervalles

Ce sont ceux que nous avons étudiés le mois dernier. Un type énuméré est défini par une suite ordonnée de constantes. Nous en avons des exemples ci-dessus. Un type intervalle est défini par les deux bornes de l'intervalle, comme par exemple :

type binaire = 0..1;

ou

type décimal = 0..9;

Un autre exemple est fourni par le sous-ensemble des jours ouvrables, issu des jours de la semaine :

**Type Jours_semaine = (lundi, mardi, mercredi, jeudi,
vendredi, samedi, dimanche);
jours_ouvrables = lundi..vendredi;**

Nous avons vu que c'était d'ailleurs la formule à employer dans ce cas, puisque le Pascal interdit, à juste titre, la redéfinition de constantes dans des types.

Quelques règles de bases pour les types intervalles :

- Le type sur lequel est basé le nouveau type intervalle doit être décrit au même niveau, c'est-à-dire que tout type intervalle déclaré dans une procédure (sous-programme) doit voir le type principal déclaré dans cette procédure. Cette restriction ne s'applique naturellement pas aux types prédéfinis.
- L'ordre des bornes doit être correct. Pas de :

type binaire = 1..0;

Les types énumérés sont très commodes pour assurer qu'une donnée est bien dans un ensemble désiré. Nous avons vu le mois dernier comment faire pour contrôler ce type de validité.

2. LES TYPES STRUCTURÉS

2.1. Introduction

Le Pascal permet de plus la création de types particuliers, dits structurés, en raison de la forme qu'ils prennent, et de la structure qui en découle au niveau du programme entier. Ces types sont :

- les ensembles
- les tableaux
- les fichiers
- les enregistrements

Règlons immédiatement quelques questions quant à ces types, avant de les étudier plus en détail.

Les tableaux sont très proches de ceux que nous avons décrits en BASIC, adaptés à un langage plus structuré. Les fichiers sont un des points noirs du Pascal, et ce pour deux raisons. Tout d'abord, seuls les accès séquentiels sont disponibles. Ensuite, ils représentent une des limites de la standardisation. En clair, la gestion des fichiers est

hautement non standard. Méfiance donc, et ce que nous pourrons en dire sera le plus souvent adapté à une seule machine, que nous préciserons. Les fichiers seront étudiés plus tard. Nous décrirons à ce point du cours la notion d'ensembles, de tableaux et d'enregistrements. Si la notion de fichier vous est indispensable, écrivez-nous, nous traiterons les bases en quelques mots, mais le détail sera plus sujet à confusion à ce niveau du cours.

2.2. Les tableaux en Pascal

2.2.1 La déclaration

Un tableau est une suite, munie d'un indice, de valeurs de même type. Déclarer un tableau en Pascal comporte une étape de plus que pour la même opération en Basic. Rappelez-vous, pour décrire un tableau, nous faisons une déclaration de la forme :

```
dim a(100)
```

En Pascal, il faudra faire ceci :

```
var a : array [1..100] of real;
```

Donc, en plus de la taille exprimée en toutes lettres, le type est écrit de la manière la plus précise qui soit. En effet, il est possible de créer des tableaux d'éléments d'un type non standard, comme par exemple, en reprenant ce que nous venons de définir :

```
type binaire = 0..1;
```

Puis dans le programme utiliser un tableau :

```
Tab_bin : array [1..100] of binaire ;
```

Il est aussi possible de créer un type à partir d'une structure de tableau. Il vient alors, en continuant sur l'exemple précédent :

```
type binaire = 0..1;  
    typ_Tab_bin = array [1..100] of binaire
```

puis déclarer en variable :

```
var t_b : Tab_bin
```

Pour résumer tout cela, voici un exemple un peu plus complet, mais qui ne propose que la partie déclaration.

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

```
const  
    max = 26;  
    nb = 20;
```

```

type
  voyelle = (a, e, i, o, u, y);
  tab_lettre = array[1..max] of char;
  tab_voy = array[1..nb] of voyelle;

```

```

var
  vo : voyelle;
  t_v : tab_voy;
  t_l : tab_lettre;

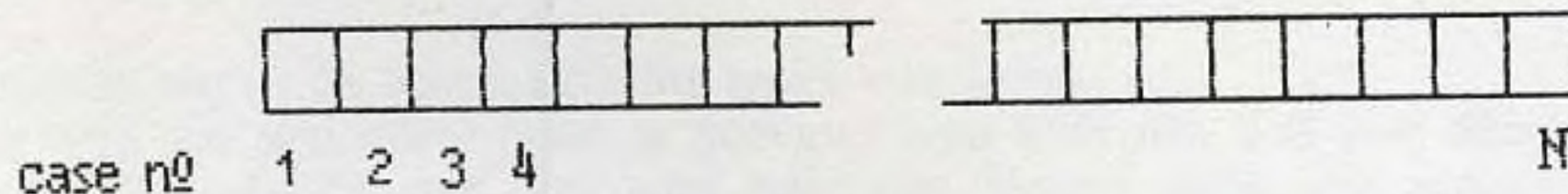
```

```

begin
end.

```

Nous n'avons manipulé pour l'instant que des tableaux à une seule dimension, ayant la forme générale suivante :



Ceci permet de faire toutefois quelques commentaires généraux :

- L'**index**, qui est le nombre permettant de repérer une case, doit être d'un type **énuméré**. Il est clair qu'un index du style 3.1415 n'a pas grand sens.
- L'index doit être déclaré auparavant, pour que le programme puisse comprendre de quoi il s'agit lorsqu'il lit la déclaration de tableau.

Voici un exemple qui reprend tout ceci :

```

program exemple (input, output);

```

```

type
  bin = 0..1;

```

```

var
  essai_tab : array[bin] of bin;

```

```

begin
end.

```

Bin est un type énuméré, déclaré avant que le tableau ne le soit. Puis le tableau est déclaré. A quoi ressemble ce tableau ? Il est composé de deux cases, qui pourront accueillir des nombres qui seront des 0 ou des 1. Rien d'autre ne sera accepté.

Une autre façon de voir le même exemple :

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

```
type
```

```
  bin = 0..1;
```

```
  bin_tab = array(bin) of bin;
```

```
var
```

```
  essai_tab : bin_tab;
```

```
begin
```

```
end.
```

Dans le même esprit, en reprenant l'exemple que nous avons traité juste avant, il est possible d'écrire :

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

```
const
```

```
  max = 26;
```

```
  nb = 20;
```

```
type
```

```
  voyelle = (a, e, i, o, u, y);
```

```
  tab_v = array [voyelle] of char;
```

```
  tab_lettre = array [1..26] of char;
```

```
var
```

```
  vo : voyelle;
```

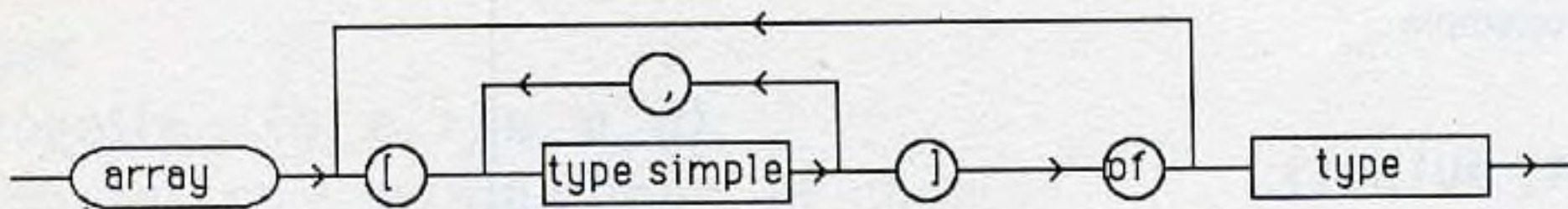
```
  t_v : tab_voy;
```

```
  t_l : tab_lettre;
```

```
begin
```

```
end.
```

Pour conclure, voici le diagramme syntaxique de la déclaration de tableaux :

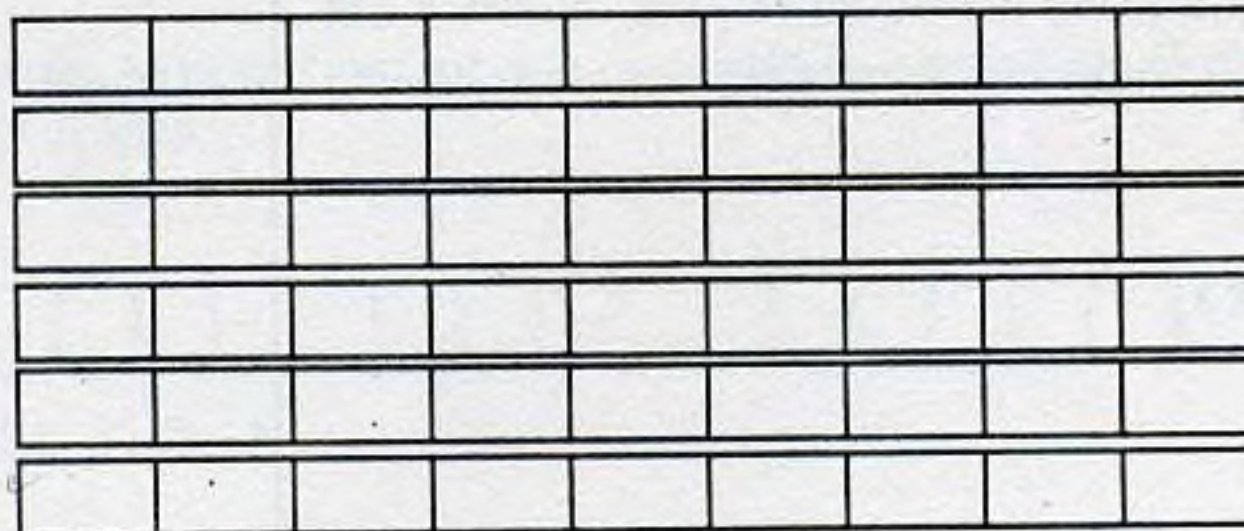


2.2.2. Les tableaux multi-dimensionnels

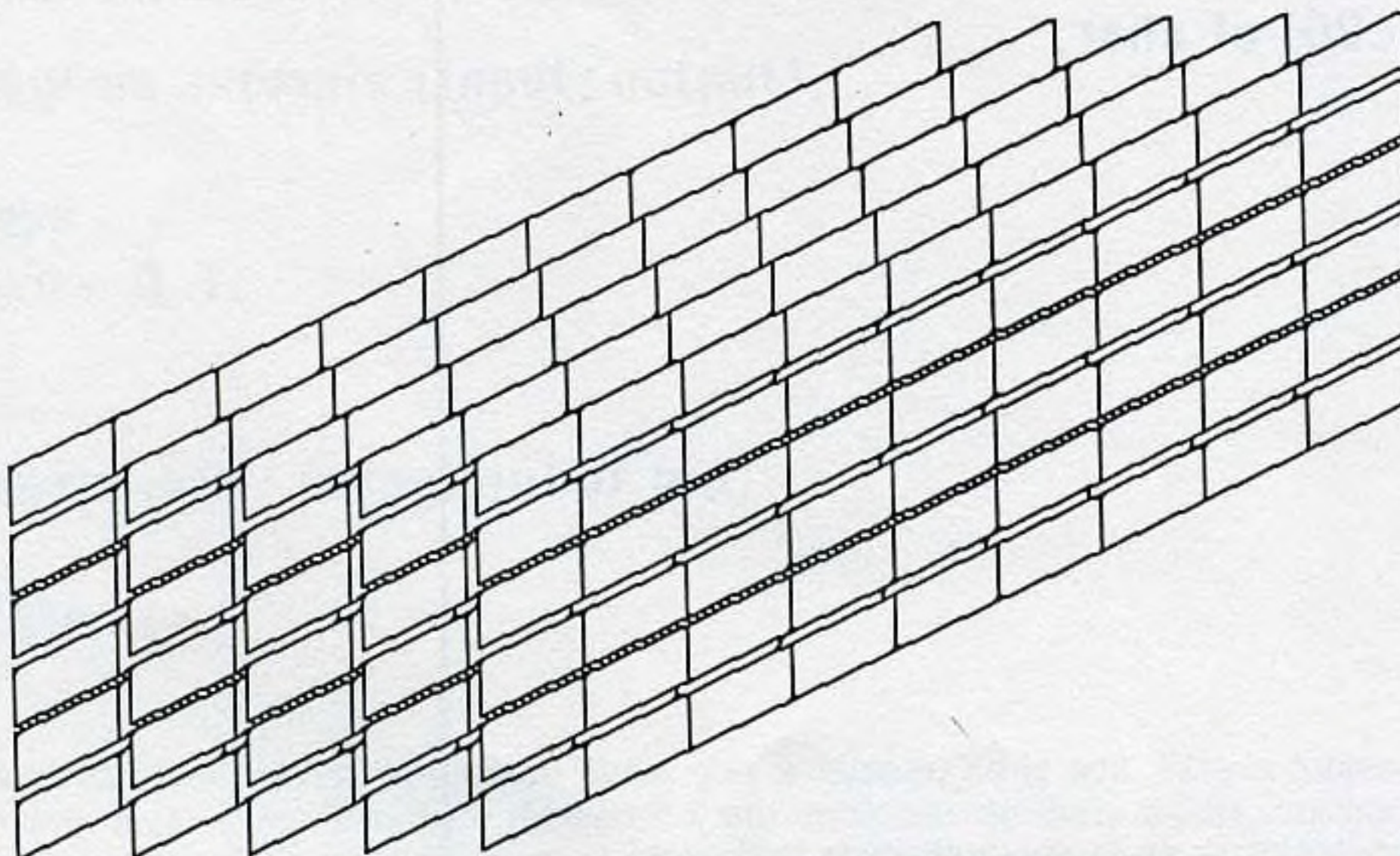
Il est possible de créer des tableaux qui possèdent plus d'une série d'index. Par exemple, en dimension deux, on peut voir un tableau comme la juxtaposition de tableaux à une dimension.



Tableau mono-dimensionnel



Et de même, un tableau en dimension trois pourra être vu comme la réunion de plusieurs tableaux en dimension deux.



La déclaration pourra se faire d'une façon classique, par exemple, en faisant, (en dimension deux) :

```
var tab_deux : array [1..100,1..100] of real;
```

qui définit un tableau possédant 100×100 cases (soit 10 000 cases, ce qui n'est pas rien). Mais si cette méthode est très naturelle, il en est une autre qui reprend au mieux la notion de formation d'un tableau à partir de tableaux de dimensions inférieures. Ainsi, il sera possible de voir, pour la déclaration précédente :

```
var tab_deux : array [1..100] of array [1..100] of real;
```

On voit là le principe du schéma présenté juste avant. Ces deux déclarations sont rigoureusement identiques. Le diagramme syntaxique du 2.1. explique les différentes possibilités de déclarations.

Mais une grosse différence apparaît au niveau de l'utilisation de ces tableaux. Si l'on utilise la déclaration :

```
var tab_deux : array [1..100,1..100] of real;
```

le tableau sera utilisé avec la notation :

```
tab_deux[i,j]
```

pour utiliser une de ses cases.
Par contre, si l'on utilise la déclaration :

```
var tab_deux : array [1..100] of array [1..100] of real;
```

il faudra utiliser les cases en les notant :

```
tab_deux[i][j]
```

qui rappelle la définition progressive (et en fait réursive, nous expliquerons bientôt ce mot) du tableau. En fait, la plupart des implémentations du Pascal autorisent les deux notations.

```
tab_deux[i,j]
```

et

```
tab_deux[i][j]
```

pour la déclaration

```
var tab_deux : array [1..100,1..100] of real;
```

par contre, il faut impérativement utiliser dans le premier cas la notation usuelle.

2.3. Les tableaux compactés

Nous avons déjà déclaré le tableau :

```
var tab_bin : array [1..100] of bin;
```

où bin était le type :

```
type bin = 0..1;
```

Comment se passe le stockage en mémoire ? L'ordinateur, au moment de la compilation, lit que le programme se servira d'un tableau de 100 cases, qui contiendront des entiers. Il sait que les entiers sont stockés, par exemple, sur 16 bits. Il réserve donc 16×100 bits, soit 1600 bits (plus de 1 kbit) mais nous, nous savons que le programme ne mettra que des nombres «binaires» 1 ou 0. Un bit suffirait à chaque fois, soit au total 100 bits. Une technique permet d'obliger le compilateur à être intelligent dans la mesure du possible. Cette technique est le compactage. En fait, le compactage est un peu plus que cela. Dans le cas de tableaux un peu plus complexes, il permet de forcer l'ordinateur à stocker chaque élément d'un tableau au mieux, c'est-à-dire que la place utilisée par cet élément soit la plus petite possible, indépendamment de celle prise par ses voisins. Autrement dit, si nous prenons la déclaration :

```
var   tab : array [1..10] of real;  
      tab_comp : packed array [1..10] of real;
```

alors en logeant dans ce tableau les nombres [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], ils prennent au total (en comptant 32 bits par réel) :

cas de tab :	$32 \times 10 = 320$ bits		
cas de tab-comp :	1se met sur	1bit	(1)
	2	2	(10)
	3	2	(11)
	4	3	(100)
	5	3	(101)
	6	3	(110)
	7	3	(111)
	8	4	(1000)
	9	4	(1001)
	10	4	(1010)
total	29 bits		

Ce gain est plus qu'appréciable.

Il y a bien sûr un inconvénient à utiliser ces tableaux compactés, sinon les tableaux classiques seraient systématiquement délaissés. Le principal problème est que pour travailler sur les éléments de tels tableaux, il faut décompacter le tableau, ou au moins l'élément. Décompacter le tableau revient à perdre une bonne partie du bénéfice, mais pas entièrement, du compactage. Décompacter les éléments ralentit singulièrement l'exécution du programme.

Un autre problème apparent sur quelques compilateurs, mais pas tous, est que les éléments d'un tableau compacté peuvent ne pas être acceptés comme arguments d'une fonction ou d'une procédure. Nous y reviendrons au moment où nous traiterons ces deux éléments de bases de tout programme structuré en Pascal.

Compactage d'un tableau et décompactage d'un tableau compacté : voici un programme simple qui montre comment réaliser ces deux opérations de base, et les précautions à prendre. Nous chargeons des valeurs dans un tableau, puis nous le compactons, puis le décompactons. Ceci est artificiel, mais permet de montrer le format des deux commandes :

```

program Comp_decomp;

var
  tab : array[1..10] of real;
  tab_comp : packed array[1..10] of real;
  i : integer;

begin

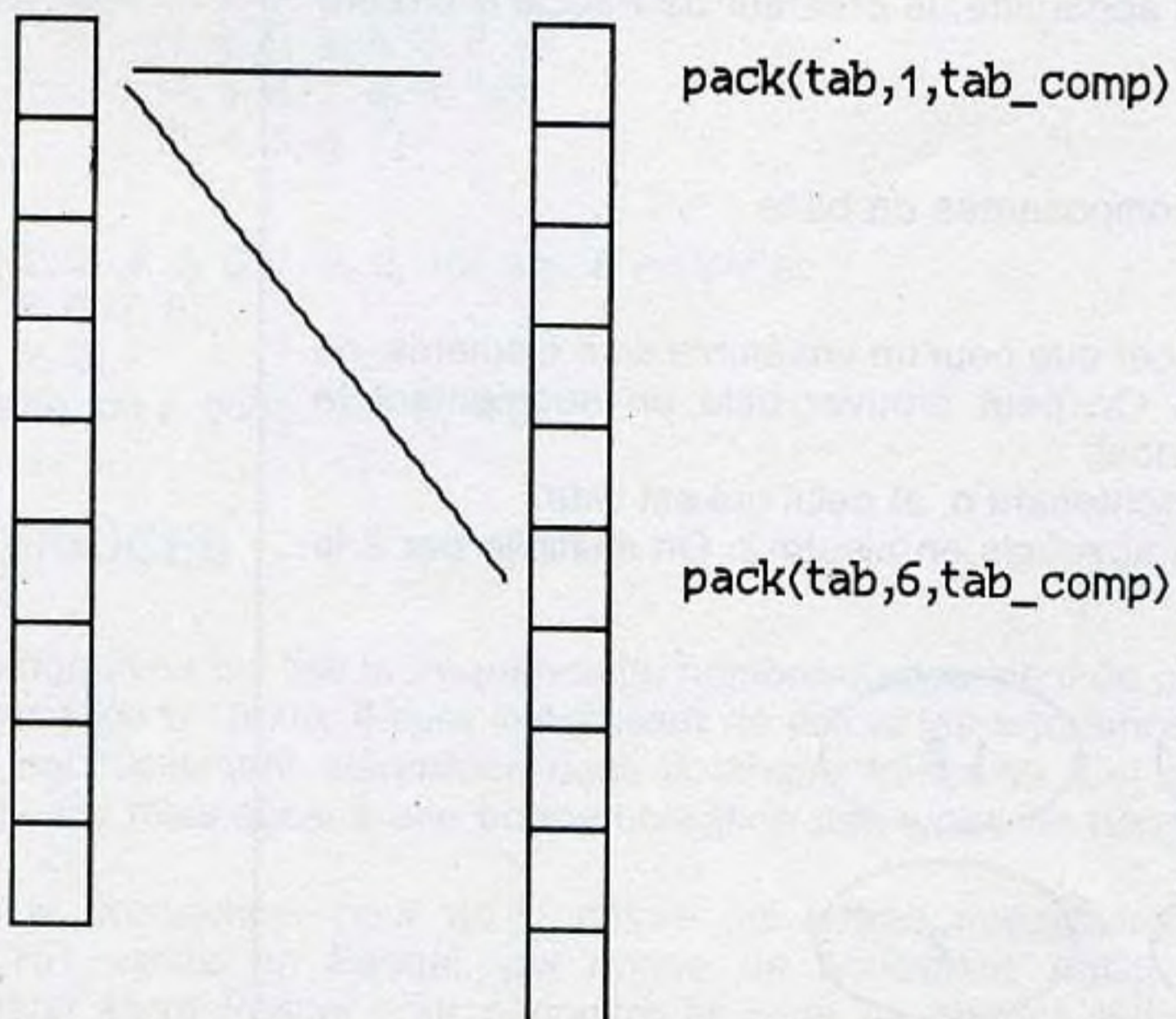
  for i := 1 to 10 do
    readln(tab[i]);
    pack(tab, 1, tab_comp);
    unpack(tab_comp, tab, 1);

  for i := 1 to 10 do
    writeln(tab[i]);

end.

```

En particulier, on voit que les deux commandes ne sont pas parfaitement symétriques. Dans la première, on met le tableau de départ, un nombre, le tableau d'arrivée. Dans la seconde, on met le tableau de départ, le tableau d'arrivée, puis le nombre. Au fait, que représente ce nombre ? Il sert à informer le programme à partir de quel index du tableau d'arrivée stocker ce qui est converti. Ici, les deux tableaux ayant la même taille, il fallait commencer à la première case.



Attention, compacter ou décompacter un tableau se fait sur tout le tableau en une seule fois. Donc, si le tableau d'arrivée est trop petit, ou si l'index que vous fournissez rend la place disponible trop restreinte, le programme plantera.

Une autre façon de voir le compactage et le décompactage :

3. LES ENSEMBLES

3.1 Introduction

Allons, vous ou vos enfants, vous avez touché aux mathématiques modernes. Donc, vous savez ce qu'est un ensemble. Mais si, vous savez bien les patates.



Description hyper-détaillée d'un ensemble en maths modernes.

En Pascal, c'est la même chose, c'est-à-dire tout ce que vous voudrez y mettre. Il est possible de définir un ensemble par énumération, par explication, par une propriété,... ou bien par rien du tout, ce qui fait un ensemble... vide. Ceci n'est pas forcément passionnant, mais chacun trouve son plaisir où il peut.

Une règle tout de même, tous les éléments d'un ensemble auront le même type. De plus, ce type ne pourra pas être réel, en raison de problèmes délicats du type de celui que nous avons évoqué, et dont un exemple est le traitement de nombres non rationnels, comme certaines racines carrées.

Pour éliminer les difficultés qui pourraient apparaître, le créateur de Pascal a préféré éliminer ce cas.

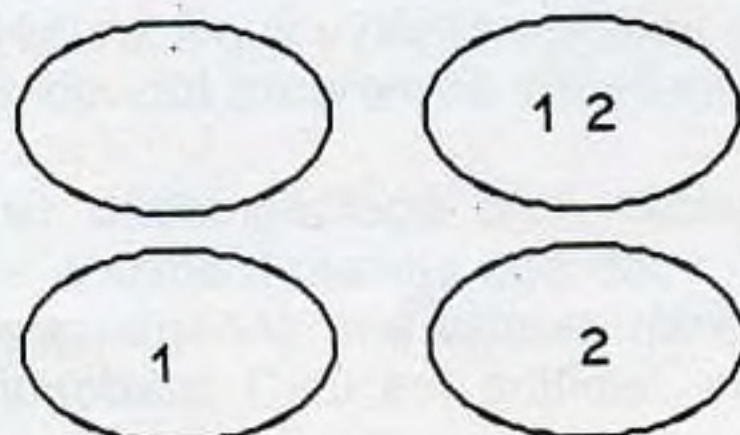
3.2. Le type d'ensemble

Un type d'ensemble est formé de deux composantes de base :

- un ensemble dit de «base»
- des sous-ensembles

Les règles de l'algèbre permettent d'énoncer que pour un ensemble de n éléments, on peut avoir 2^n sous-ensembles distincts. On peut prouver cela en augmentant le nombre n petit à petit (preuve par récurrence).

- $n = 1$, on peut avoir 2 ensembles, celui contenant n , et celui qui est vide.
- $n = 2$, on a les mêmes, plus les mêmes auxquels on ajoute 2. On multiplie par 2 le nombre d'ensembles, d'où le résultat.



3.3. Déclaration d'un ensemble

Cette déclaration est en général effectuée dans la déclaration de types, mais aussi pour les plus simples, directement au niveau des variables. De plus apparaît là la notion très forte de sous-ensemble, déclarée par :

set of

Ceci permet de ne pas être restreint par le fait qu'un ensemble ne peut être défini directement comme le sous-ensemble d'un autre ensemble. Prenons un exemple, qui aidera à la compréhension.

Nous voulons, à partir des jours de la semaine, définir un ensemble qui contiendra des jours, mais sans en savoir plus. Pour cela, nous ferons :

type

```
Jours = ( lundi, mardi, mercredi, jeudi, vendredi );  
certaines_jours : set of jours;
```

var

```
des_jours : certaines_jours;
```

Nous avons donc une variable, `des_jours`, qui contient un nombre inconnu de jours. C'est un sous-ensemble.

3.4. Les opérations disponibles

Dans la suite, `E` sera un ensemble, et `A` et `B` seront des sous-ensembles. Comme pour les ensembles en math, un certain nombre de manipulations sont possibles. Entre autres :

union : l'ensemble formé des éléments de `A` et de `B` est calculé par `A + B`.

intersection : les éléments compris dans `A` et dans `B` : `A * B`

différence : les éléments compris dans `A` mais pas dans `B` : `A - B`

égalité : par le symbole `=`. Ceci n'est pas une opération, mais une relation utilisable dans un test, par exemple.

inégalité : `<` ou `>`. Même commentaire.

inclusion : `= <`, `= >`

appartenance : par le mot réservé `in`.

Tout de suite des exemples :

`E` sera l'ensemble (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

`A` sera (1, 2, 3, 4, 5, 6, 7, 8)

`B` sera (4, 5, 6, 7, 8, 9, 10)

`C` sera (3, 4, 5, 6, 7)

On a alors :

`A + B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)`, soit `E` en entier.

`A * B = (4, 5, 6, 7, 8)`

`A - B = (1, 2, 3)`

pour l'inégalité, on a donc : `A - B < C`

4. DES EXERCICES

1) Ecrire un programme qui lise la fréquence (le nombre d'apparition) de chaque lettre dans une phrase, ou un texte. Il sera intéressant de voir si les fréquences obtenues seront celles habituellement acceptées dans la langue française. Cet exercice fait appel aux tableaux mais aussi à une bonne utilisation des types de données.

2) Compléter le programme pour qu'il compte les lettres majuscules à part. La commande `CHR` existe en Pascal, qui donne un équivalent ASCII de chaque caractère. Tester votre Pascal pour connaître le code de chaque lettre. Ecrire un programme qui le fasse.

- 3) Même problème avec les signes de ponctuations, et les blancs.
- 4) Ecrire un programme qui, à partir d'un certain nombre de données du type que vous voudrez, les ressortent en sens inverse.
- 5) Si vous aviez dix étudiants, comment feriez-vous pour calculer leur moyenne, la moyenne de la classe sur l'année, et sur chaque interrogation ?
- 6) Trier les étudiants pour imprimer leur nom par ordre de moyenne, que vous stockerez dans un tableau.
- 7) Donner un programme transformant une date au format :

10/12/86

en

10 décembre 1986.

- 8) Donner un programme qui puisse travailler sur les jours de la semaine, et permette de stocker les températures prises toutes les heures. L'en-tête du programme suffira mais libre à vous d'en proposer plus.
- 9) Faire un programme qui trie une liste de données du type de votre choix, en ordre croissant ou décroissant. Prendre tout de même des données qui soient aisément triables. Par exemple, si vous choisissez des types contenant des couleurs, il n'est pas forcément simple de choisir une façon de les trier bien que Pascal le permette.
- 10) Un petit exercice pour les matheux. Calculer le triangle de Pascal, à l'aide d'un tableau mono-dimensionné. Le triangle de Pascal est une structure permettant de faire des calculs intéressants sur les développements de certaines expressions mathématiques. Il est obtenu de la manière suivante : pour chaque élément, sa valeur est la somme du nombre au-dessus et de celui au-dessus à gauche.

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 ...

```

Tout le problème est de le calculer avec un tableau à une seule dimension. Les non matheux doivent pouvoir s'y intéresser à l'aide de la seule définition.

- 11) Un algorithme stupide pour faire des carrés magiques dit :
 Pour faire des carrés magiques de taille impaire, voici une solution :
 - le 1 est mis juste en dessous du centre
 - les éléments suivants (2, 3, 4, 5, 6, 7...) sont placés dans les cases se trouvant à l'intersection de la ligne du dessous et de la colonne de droite.
 - si l'on se heurte à une limite, on continue, comme si l'on avait affaire à un cylindre, du côté opposé, en appliquant toujours la règle.
 - si la case est déjà remplie, mettre le nombre de cases en dessous dans la même colonne.

L'algorithme est loin d'être optimal, car de très nombreuses cases sont testées plusieurs fois. Mais il marche. A vous de le programmer.

- 12) Donner un programme qui permette de calculer le nombre de termes d'un ensemble fourni.
- 13) Ecrire pour vous un noyau de programme qui demande ce qu'il doit faire par un menu, exécute ce qui lui est demandé, mais retourne au menu, si le choix fait ne correspond pas à une possibilité offerte. Nous avons vu le mois dernier comment faire.



Récemment parus

INITIATION A LA VIDÉO LÉGÈRE (THÉORIE ET PRATIQUE)

Claude Gendre.

Nous devons à Claude Gendre, l'auteur des «Magnétophones», des «Magnétoscopes et de la télévision» et des «Synthétiseurs», cet ouvrage très pratique, organisé comme un véritable guide.
- Choix d'un standard ? - Caméscopes VHS, VHS-C ou 8 mm ? - Connexion ? Compatibilité ?
- Accessoires ? Montage ? Enfin... Comment filmer.

Le nouveau livre de Claude Gendre répond à toutes ces questions. Cet ouvrage essentiellement pratique, qui n'a pas d'équivalent en librairie aujourd'hui, s'adresse (sans formules mathématiques) à tous ceux passionnés (déjà ou à venir) de vidéo ainsi qu'aux amateurs de belles images. Des illustrations en couleur donnent une excellente idée des possibilités de «filmage» et de montage.

L'avenir du cinéma d'amateur et celui de la création par l'image passeront par la vidéo légère...
Ce livre devenait urgent.

INITIATION A LA MICRO-INFORMATIQUE (COURS 1^{er} CYCLE) TOME 3

Claude Polgar

Le 3^e tome de l'Initiation à la Micro-Informatique est enfin paru. Il complète les deux premiers tomes dont la réputation n'est plus à faire ! Le talent de pédagogue de Claude Polgar une fois de plus saura conduire ses lecteurs avec le sérieux mais aussi l'humour qui lui sont propres.

Sa formule, dans les années 80, fut le contraire de celle adoptée par les innombrables cours de Basic qui poussèrent à ce moment-là comme des champignons, en promettant un peu légèrement d'apprendre à tout un chacun la micro-informatique en cinq leçons...

Il fallait avoir le courage de commencer par A pour arriver à Z.

Programmer est un loisir intelligent qui peut devenir un métier passionnant, mais l'étude de la programmation nécessite du travail et de la méthode. A ce jour, plus de 40 000 lecteurs ont suivi les deux premières parties de ce cours. Ce troisième tome avec ses 250 pages, format 21 x 27 vient boucler un ensemble de 700 pages.

INITIATION A L'ÉLECTRICITÉ ET A L'ÉLECTROTECHNIQUE

Roger Friédérich.

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

Nouvelle édition

LES MAGNÉTOSCOPES ET LA TÉLÉVISION (Revu et corrigé)

Claude Gendre

Cet ouvrage de Claude Gendre est devenu un classique des plus complets sur le sujet : de l'histoire de la télévision à la technologie de cette dernière en passant par les magnétoscopes, tout y est minutieusement expliqué et illustré. Cette deuxième édition revue et corrigée comporte deux nouveaux chapitres, l'un sur les nouveaux magnétoscopes Grundig en VHS et l'autre sur le format Vidéo 8 Sony.

EN PRÉPARATION

Collection jaune

A paraître avant la fin de l'année :

Les Montages Electroniques de Jean-Pierre Lemoine : P 30 : 250 F Sortie début novembre. Véritable encyclopédie de 276 pages Plus de 1 000 dessins. 25 montages originaux

Téléphone et Radiotéléphone de Roger-Ch. Houzé :

Les Bases de l'Electronique de Raymond Breton

Etude autour du 6809 (constructions et logiciels) de Claude Vicidomini

Collection noire

A paraître avant la fin de l'année

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 !
- 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 digitale est dû à Ph. Duquesne, chargé de cours de microprocesseurs au CNAM. L'objet de cet ouvrage est de présenter les opérateurs logiques et leurs associations. La technologie est évoquée, brièvement, elle aussi.
- 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.

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

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 E 02 E 03 E 04 E 05 E 06 L 07 P 08 L 09 L 10
- L 11 E 12 E 13 L 14 E 15 P 16 P 17 P 18 P 19 L 20
- P 21 E 22 P 23 P 24 E 25 P 26 P 27 P 28 P 29 P 30

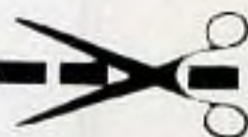
Frais de port : + 12 F par livre commandé, soit la somme totale ci-jointe, de Frs par CCP Chèque bancaire Mandat-lettre

Nom Prénom

Adresse

Ville Code Postal

LED-Micro 11



COURS DE PROGRAMMATION APPROFONDIE

Dominique Chastagnier
Jean-François Coblentz
Patrick Gueneau

Nous vous présentons aujourd'hui un petit tour dans le monde des interpréteurs, compilateurs, et autres machines infernales. Nous espérons qu'il vous aidera à maîtriser l'environnement que vous envisagez d'utiliser pour débiter la programmation en Pascal, ou pour améliorer les performances de votre Basic «standard», ou tout simplement de mieux connaître celui que vous utilisez déjà.

COURS N°14

PLAN DU COURS

Compilation et interprétation

1. Avant-propos
2. Les principes de bases
 - 2.1. Editeur
 - 2.2. Analyseur syntaxique
 - 2.3. Interpréteur
 - 2.4. Le compilateur
 - 2.5. Semi compilateur/interpréteur
 - 2.6. Assembleur
 - 2.7. Editeurs de liens
 - 2.8. Bibliothèque de procédures
3. Conclusion

COMPILATION ET INTERPRETATION

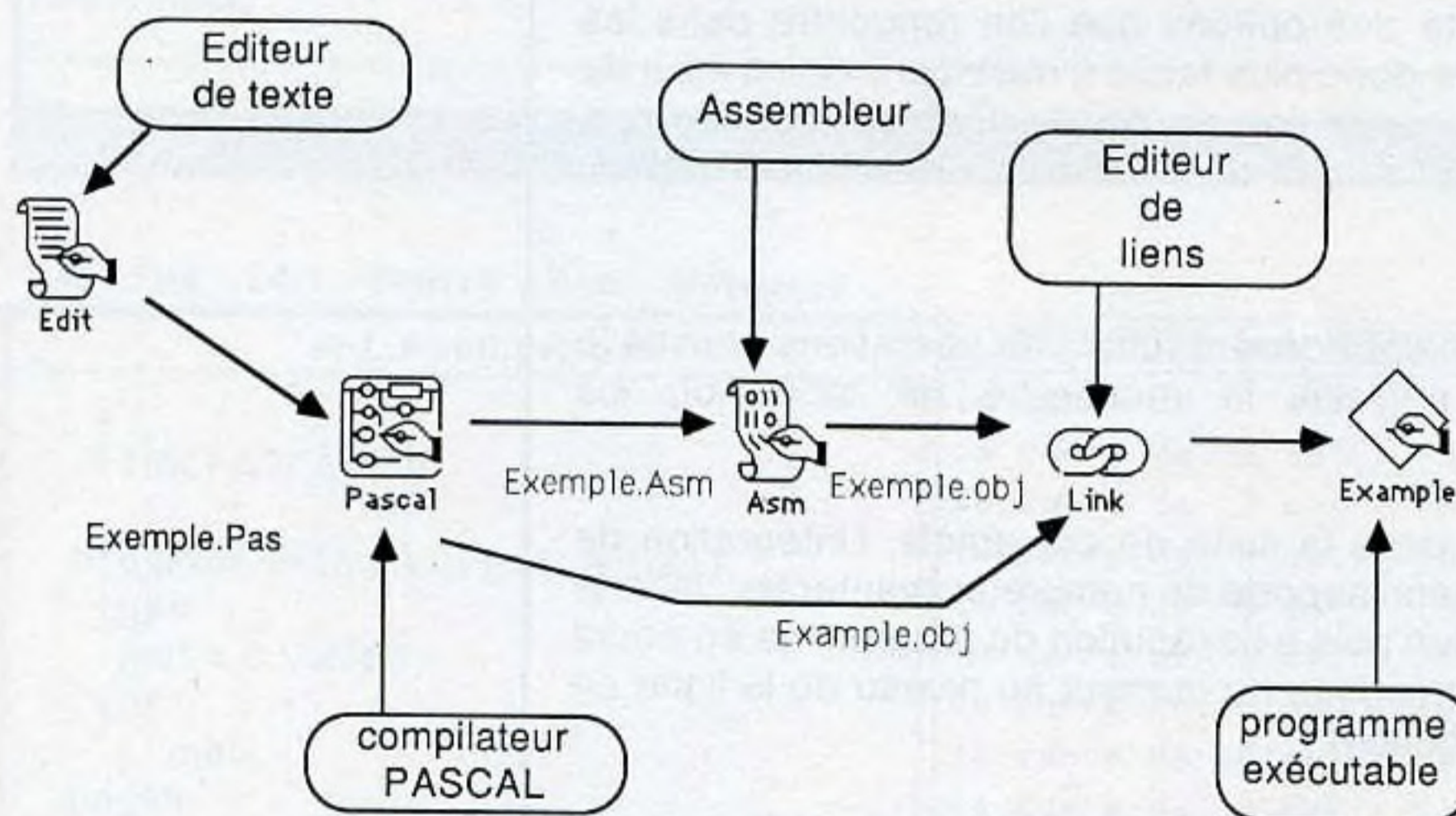
Cet article est à l'usage de ceux qui veulent en savoir un peu plus sur les environnements de programmation disponibles en PASCAL et en BASIC, mais aussi de ceux qui hésitent encore à franchir le cap, à passer du BASIC au PASCAL (par exemple), ou, plus modestement, à aller plus en avant dans la programmation BASIC. Il vient donc en complément du cours de PASCAL, mais aussi du cours de programmation approfondie.

1. AVANT PROPOS

Comme vous l'avez sans aucun doute remarqué, nous ne vous parlons dans Led-Micro que de BASIC ou de PASCAL. Pourquoi oublier ainsi tous les autres langages classiques de programmation (langage C, FORTRAN, COBOL, MODULA2, ADA...), ou encore les langages plus ciblés (FORTH, LISP, PROLOG) ? Il y a plusieurs bonnes raisons : il serait tout d'abord difficile de répertorier complètement les similitudes et différences des principaux langages de programmation sans alourdir considérablement les cours et articles que nous vous proposons chaque mois ; il faut avouer d'autre part qu'il nous est impossible de maîtriser tous ces langages ; enfin, et c'est la meilleure des raisons, le BASIC est le langage le plus répandu sur les micro-ordinateurs familiaux, et le PASCAL a été conçu au départ pour être un langage éducatif permettant de bien appréhender la programmation structurée.

2. LES PRINCIPES DE BASES

Lorsqu'on aborde la programmation en d'autres langages que le BASIC, on est forcé de changer ses habitudes. En effet, la plupart nécessite l'usage d'éditeurs de textes pour créer le programme « source » (lignes d'instructions écrites suivant la syntaxe du langage), de traduire ce texte en code exécutable, c'est le travail du compilateur, de lier ce code à un environnement standard (les fonctions prédéfinies, gestion de fichiers, etc), on invoque alors l'éditeur de liens, et enfin, dans certains cas, il est possible de créer des bibliothèques de programmes.



Exemples des étapes de création d'un programme:

- Exemple.Pas est le fichier source du programme
- Exemple.obj est le fichier contenant le code machine traduit par le compilateur.
- Exemple.asm contient la liste en instructions du code généré par le compilateur (en option).
- EXEMPLE est donc le programme directement exécutable

```
(* programme source en pascal *)
program exemple(input,output);
  type
    mot=0..65535;
  var
    i: mot;
  begin
    for i:=1 to 10 do
      writeln('le carré de ',i:2,' est :',i*i);
    end. (* fin du programme *)
```

2.1. Editeur

Rien de bien extraordinaire, si vous avez déjà vu un traitement de texte tourner. En effet, très souvent, le programme source n'est autre que du texte, entré au kilomètre, aussi l'éditeur généralement fourni avec le langage ressemble à un programme de traitement de texte. Par exemple, le TurboPascal est fourni avec un éditeur compatible au niveau des commandes avec le fameux WORDSTAR qui fut leader dans son domaine à l'époque où le système d'exploitation CP/M régnait sur la micro-informatique professionnelle. L'intérêt essentiel de ces éditeurs dédiés réside dans certaines options choisies pour faciliter l'introduction des lignes de programmes, et qu'on ne retrouve généralement pas dans un traitement de texte.

Voici les particularités les plus fréquentes des éditeurs orientés «programmation» :

– Simplicité :

Un éditeur ne dispose pas de la majorité des options que l'on rencontre dans les programmes de traitement de texte. Il sera donc plus facile à maîtriser ; moins il y a de caractères de contrôle à mémoriser et mieux l'on se porte ! Cela n'empêche pas d'avoir des accès pleine page et les fonctions de bases (insertion, modification, recherches, tabulations).

– Performances :

Pour les mêmes raisons, l'éditeur sera plus performant dans des opérations comme le défilement à l'écran («scrolling» en anglais), la recherche de mots, ou les entrées/sorties sur disque.

– Intégration :

Comme nous le verrons plus en détail, dans la suite de cet article, l'intégration de l'éditeur à l'environnement de programmation apporte de nombreux avantages : ainsi le passage rapide de l'édition à la compilation puis à l'exécution du programme en cours de mise au point, le positionnement automatique du curseur au niveau de la ligne en cas d'erreur de compilation (voire d'exécution)

– Caractéristiques particulières :

On peut noter aussi, les possibilités d'auto-indentation ; c'est-à-dire de positionnement du curseur à la verticale du premier caractère non blanc après un retour à la ligne (très utile dans un environnement PASCAL pour décaler les blocs BEGIN ... END), l'accès et la visualisation des caractères de contrôle, la conversion majuscules/minuscules, etc.

2.2. Analyseur syntaxique

Cet organe, indispensable au bon fonctionnement d'un langage, occupe une place très variable suivant le type d'implantation. En BASIC interprété, il fait partie de l'édition, repère les instructions du langage afin d'effectuer le codage des lignes en mémoire suivant les versions, une partie de l'analyse peut être reportée au cours de l'exécution

(par exemple le décodage des constantes). Pour les langages compilés, dont le BASIC, cette analyse fait partie de la compilation dans la majorité des implantations. Cependant, nous connaissons au moins un cas en PASCAL (LIGHTSPEED PASCAL pour Macintosh) où elle est effectuée au moment de l'édition, ainsi qu'un en BASIC (ZBASIC encore sur Mac) qui s'opère à l'affichage du programme. Une analyse syntaxique distincte a l'avantage d'imposer au programme source un premier filtre qui supprime les fautes de frappe, et améliore l'aspect du texte à l'écran. On trouve un exemple de ce formatage du listing sur deux situations en BASIC (compilateur ZBASIC sur Macintosh) et en PASCAL (interpréteur MAC PASCAL toujours sur Macintosh figures 2.2.1 et 2.2.2). On remarque dans ces deux copies de l'écran du Macintosh que les mots réservés du langage sont inscrits en caractères gras et que les boucles et autres blocs d'instructions sont décalés automatiquement.

File Edit Command Configure Memory

```

ZBasic™ 3.0
LIST
00100 REM programme exemple
00110 FOR I=1 TO 10
00120 PRINT "Le carre de ";I;" est :";I*I
00130 NEXT
00140 END
ZBasic Ready
COMPILE

Segment 128 ends in Stmt 1 at Line 65535 : Size is 84 bytes.
ZBasic Ready
RUN
Le carre de 1 est : 1
Le carre de 2 est : 4
Le carre de 3 est : 9
Le carre de 4 est : 16
Le carre de 5 est : 25
Le carre de 6 est : 36
Le carre de 7 est : 49
Le carre de 8 est : 64
Le carre de 9 est : 81
Le carre de 10 est : 100
ZBasic Ready
-

```

File Edit Search Run Windows

```

led_exemple.pas
(* MACPASCAL *)

program exemple (input, output);
type
  mot = 0..32767;
var
  i : mot;
begin
  for i := 1 to 10 do
    begin
      write('le carre de ', i : 2);
      writeln(' est : ', i * i : 3)
    end;
  end.

```

```

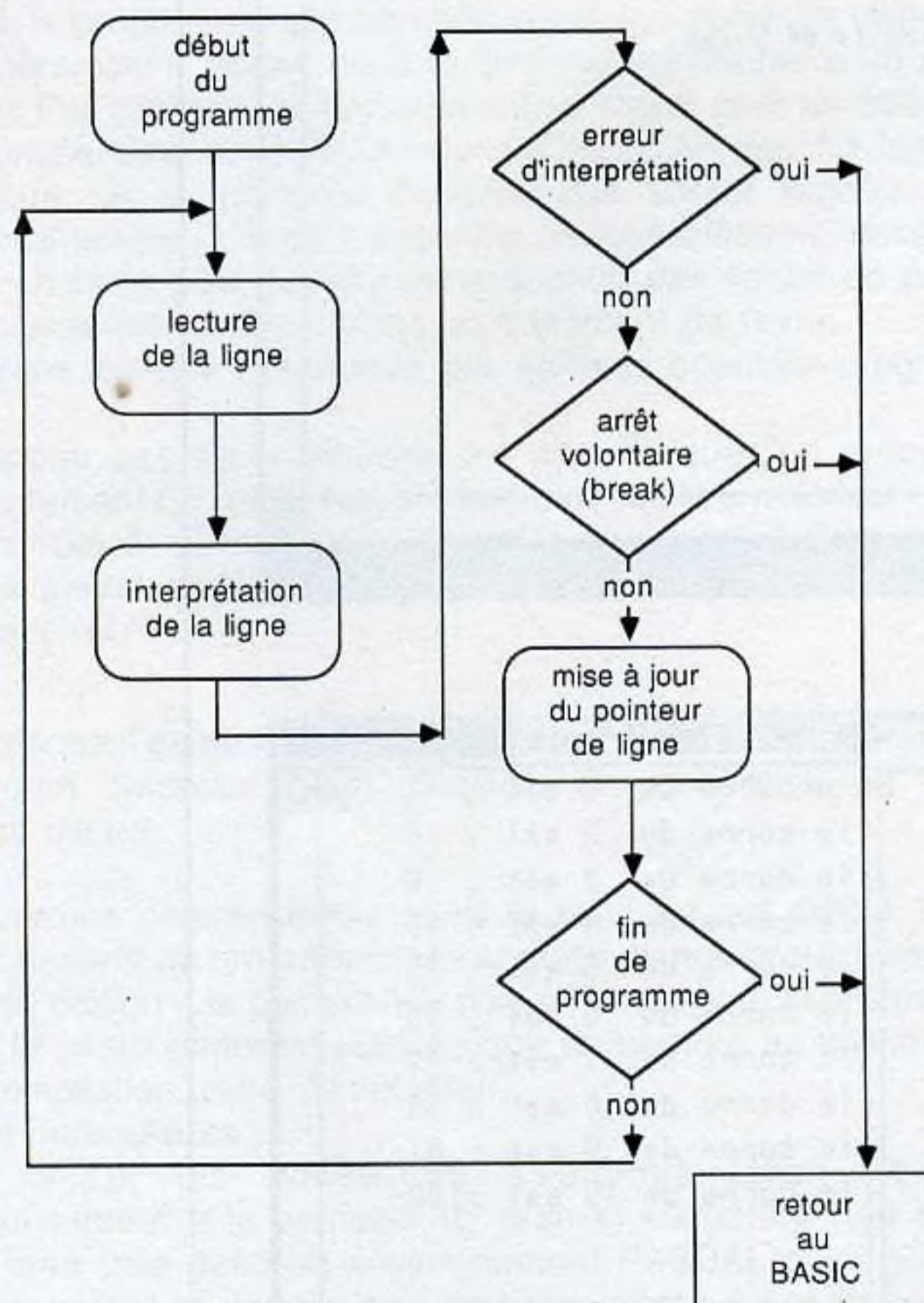
Text
le carre de 1 est : 1
le carre de 2 est : 4
le carre de 3 est : 9
le carre de 4 est : 16
le carre de 5 est : 25
le carre de 6 est : 36
le carre de 7 est : 49
le carre de 8 est : 64
le carre de 9 est : 81
le carre de 10 est : 100

```

2.3. Interpréteur

Si vous programmez en BASIC, vous connaissez le principe d'interprétation. Sans entrer dans les détails, le programme est représenté par des lignes d'instructions qui sont codées en mémoire principale. Lorsque l'on désire exécuter la séquence d'instructions, un programme particulier nommé interpréteur est lancé. Il lit au fur et à mesure les lignes d'instructions, les traduit pour qu'elles soient compréhensibles par le micro-processeur (vous savez le «cœur» de votre machine). Si, pour une raison quelconque, l'interpréteur passe plusieurs fois sur la même ligne, il devra effectuer à nouveau son travail d'analyse et de traduction. Le plus souvent, l'interpréteur fait partie d'un environnement complet intégrant édition, analyse syntaxique et exécution du programme.

On rencontre trois types de langages interprétés : les interpréteurs BASIC (pour des raisons historiques et de facilité d'apprentissage, ce langage a été longtemps exclusivement un langage interprété, tout au moins micro-ordinateurs) ; des langages comme le FORTH ou l'APL qui le sont pour des raisons de structures et de mode de fonctionnement ; enfin, les langages dits orientés intelligence artificielle comme les LISP et PROLOG. En outre, nous connaissons au moins un PASCAL interprété (MACPASCAL sur Macintosh), et il en existe sûrement d'autres.



BOUCLE PRINCIPALE D'UN INTERPRETEUR

2.4. Le compilateur

Le compilateur est aussi une sorte de moulinette comme l'interpréteur. Presque tous les langages classiques de programmation disposent de compilateurs (FORTRAN, PASCAL, BASIC, langage C, COBOL), et souvent plusieurs versions pour un même ordinateur. La différence essentielle est qu'il traduit une fois pour toutes les lignes d'instructions. Contrairement à l'interpréteur, il effectue généralement ce travail sur un fichier texte (ce que l'on a appelé plus haut le fichier source), et crée un second fichier dénommé généralement le fichier objet. Classiquement un compilateur effectue les tâches suivantes :

- analyse syntaxique (cf. plus haut)
- génération du code
- création de listes (références croisées, statistiques de compilations)
- création du source assembleur du code généré.

Très souvent, de nombreuses options permettent d'adapter le compilateur aux exigences du programmeur. Parmi ces options, les plus fréquentes sont :

- (non) vérification des indices de tableaux
- (non) optimisation du code généré
- (non) génération de code pour le « debugger »
- entier sur 2 ou 4 octets,
- etc.

```
01                               Source Listing
                               =====
00001      0 0 program exemple(input,output);
00002      0 0
00003      0 0 type    mot=0..65535;
00004      0 0 var     i:mot;
00005      0 0
00006      0 1 begin { début du programme principal}
00007      0 1     for i:=1 to 10 do
00008      0 1         writeln('Le carré de ',i:2,' est :',i*i);
00009      0 0 end.
```

```
01                               Cross Reference Listing
                               =====
EXEMPLE      PROG [PSECT($CODE)]
              1
I             VAR  MOT { IN PROGRAM EXEMPLE }
              4   7 =   8       8       8
INPUT        VAR  [EXTERNAL(PAS$FV_INPUT)] TEXT { IN PROGRAM EXEMPLE }
              *   1
MOT          TYPE SUBRANGE OF INTEGER { IN PROGRAM EXEMPLE }
              3   4
OUTPUT       VAR  [EXTERNAL(PAS$FV_OUTPUT)] TEXT { IN PROGRAM EXEMPLE }
              *   1       8
WRITELN      PROC { BUILTIN }
              *       8
```

KEY TO REFERENCE FLAGS:

```
* predeclared
= modified
A address of
F forward declared procedure or function
L label defining reference
P passed as a parameter, possibly modified or called
R READ operation done
W used in a WITH statement
```

```
01                               Pascal Compilation Statistics
                               =====
```

COMMAND QUALIFIERS

PAS/LIST/CROSS_REF LED

```
/CHECK=( BOUNDS, NOCASE_SELECTORS, NOOVERFLOW, NOPOINTERS, NOSUBRANGE)
/DEBUG=( NOSYMBOLS, TRACEBACK)
/SHOW=( DICTIONARY, INCLUDE, NOINLINE, HEADER, SOURCE, STATISTICS)
/OPTIMIZE /NOENVIRONMENT
```

```

/LIST=SYS$SYSDEVICE:LERGOMANJLED.LIS;1
/OBJECT=SYS$SYSDEVICE:LERGOMANJLED.OBJ;1
/CROSS_REFERENCE /ERROR_LIMIT=30 /NOG_FLOATING /NOMACHINE_CODE
/NOOLD_VERSION /NOSTANDARD /WARNINGS

```

COMPILER INTERNAL TIMING

Phase	Faults	CPU Time	Elapsed Time
Initialization	213	00:00.3	00:00.5
Source Analysis	231	00:00.2	00:00.3
Source Listing	57	00:00.2	00:00.3
Tree Construction	47	00:00.0	00:00.0
Flow Analysis	35	00:00.0	00:00.0
Value Propagation	13	00:00.0	00:00.0
Profit Analysis	22	00:00.0	00:00.0
Context Analysis	117	00:00.1	00:00.1
Name Packing	5	00:00.0	00:00.0
Code Selection	47	00:00.1	00:00.1
Final	71	00:00.1	00:00.1
TOTAL	860	00:01.0	00:01.5

COMPILATION STATISTICS

```

CPU Time:      00:01.0      (653 Lines/Minute)
Elapsed Time:  00:01.5
Page Faults:   860
Compilation Complete

```

2.5. Semi-compilateur / interpréteur

Sur les mini-ordinateurs, la plupart des langages classiques de programmation (FORTRAN, PASCAL, langage C, COBOL et BASIC) sont des langages compilés. Il en est de même pour les versions apparaissant peu à peu sur les micro-ordinateurs. Néanmoins, nombreux sont les faux compilateurs, que l'on pourrait plus justement qualifier de semi-compilateurs. Cette remarque est d'importance si vous recherchez un environnement de programmation qui vous autorise de bonnes performances en exécution. Le principe de ces produits particuliers est le suivant :

- 1) Un compilateur transforme le texte source en un code intermédiaire.
- 2) Un interpréteur exécute ce code intermédiaire, de la même façon que l'interpréteur BASIC analyse et exécute un programme.

L'avantage principal réside dans la simplification du compilateur, et sa standardisation en cas d'adaptation sur différentes machines ; il suffit dans ce dernier cas de redéfinir l'interpréteur. Le plus connu de ces environnements est sans nul doute le système PASCAL UCSD dont nous vous avons déjà parlé le mois dernier. Mais il existe d'autres PASCALS (Nevada PASCAL pour Apple 2 et CP/M), ainsi que des BASICs comme le CBASIC de DIGITAL RESEARCH. Bien sûr, l'inconvénient majeur est la lenteur relative des programmes développés à l'aide de ce type d'environnement. Les performances se situent en général à mi-distance entre les interpréteurs purs et les vrais compilateurs.

2.6. Assembleur

Juste quelques mots sur ce « compilateur » un peu particulier, puisque, essentiellement, il traduit des « mnémoniques » (nom des instructions de base d'un micro-processeur en leur code machine). Nous le signalons parce que de nombreux compilateurs proposent en option de générer du texte de ce format au lieu du code machine. La figure 2.6. montre un exemple de compilation avec création d'un fichier texte contenant ces instructions machines (pour les curieux ou les spécialistes ce code correspond au micro-processeur MOTOROLA 68000 du Macintosh ; les commentaires n'ont évidemment pas été faits par le compilateur... malheureusement).

EXEMPLE.ASM: extrait du résultat de la compilation de Exemple.Pas

.....

```

        jmp     exemple ; saut au début du programme
        xdef   i
i       ds.b   2
        xdef   exemple
exemple
        link  A6,#0
        jsr   PASS$InitMacEnv ; initialisation de l'environnement
        pea   il2      ; (sur Macintosh)
        pea   input(A5)
        pea   output(A5) ; création de la fenêtre par défaut
        jsr   PASS$CreatePWnd ; dont le nom sera "exemple" (cf il2)
        move.b #1,i(A5) ; initialisation de i à 1
        move.b #10,D7 ; D7 contient la valeur pour le test de fin
        cmp.b i(A5),D7 ; de boucle
        blt.w il5      ; test d'entrée (si i > 10 alors saut en il5)
i14     pea   output(A5) ; début de l'affichage par writeln;
        pea   il6      ;
        clr.w -(SP) ; on affiche d'abord la première chaine
        jsr   PASS$WPString ; " le carré de" (cf en il6)
        clr.w D6
        move.b i(A5),D6 ; on met i dans le registre D6
        pea   output(A5)
        move.w D6,-(SP)
        move.w #2,-(SP) ; 2 est ici la taille d'affichage
        jsr   PASS$WInteger ; donc écriture de i sur au plus
        pea   output(A5) ; 2 caractères
        pea   il9      ;
        clr.w -(SP) ; affichage de la deuxième chaine (" est:")
        jsr   PASS$WPString ;
        clr.w D6
        move.b i(A5),D6 ; i dans le registre D6
        clr.w D5
        move.b i(A5),D5 ; i dans le registre D5
        muls.w D6,D5 ; D5 * D6 résultat dans D5
        pea   output(A5)
        move.w D5,-(SP)
        clr.w -(SP) ; pas de longueur limite (0)
        jsr   PASS$WInteger ; écriture de i*i
        pea   output(A5)
        jsr   PASS$Weoln ; retour à la ligne (à cause de writeLN)
        cmp.b i(A5),D7 ; ( si i<=10)
        ble.w il5 ; on continue
        addq.b #1,i(A5) ; incrémentation de i de 1
        jmp   il4 ; on boucle
i15     dc.w   $A9F4 ; retour au système
i12     String_format 2
        dc.w   'exemple'
i16     dc.w   'le carré de '
i19     dc.w   ' est :'

```

.....

2.7. Editeur de liens

Il est chargé comme son nom l'indique de créer les liens qui manquaient à votre programme pour être exécutable, et relier deux parties de programmes compilées séparément. Cet «utilitaire» est réservé aux langages compilés quels qu'ils soient, mais n'est pas toujours nécessaire, donc pas toujours disponible comme nous le verrons plus loin. La plupart des compilateurs créant du code adapté à la machine, il est possible de combiner des langages pour créer une application. Ainsi, la partie calcul peut être écrite en FORTRAN, la partie gestion en PASCAL, etc. L'éditeur de liens permet de regrouper les fichiers «objets» issus des différentes compilations et de créer un unique fichier «exécutable». Attention cependant aux compatibilités dans les types de variables, leur allocation mémoire, et surtout les méthodes de passage d'arguments. Renseignez-vous bien, si vous envisagez de tels combinaisons !

Programme Principal en PASCAL

```
{ fichier exemple.pas }
program exemple(input,output);

var i,i_carre:integer;

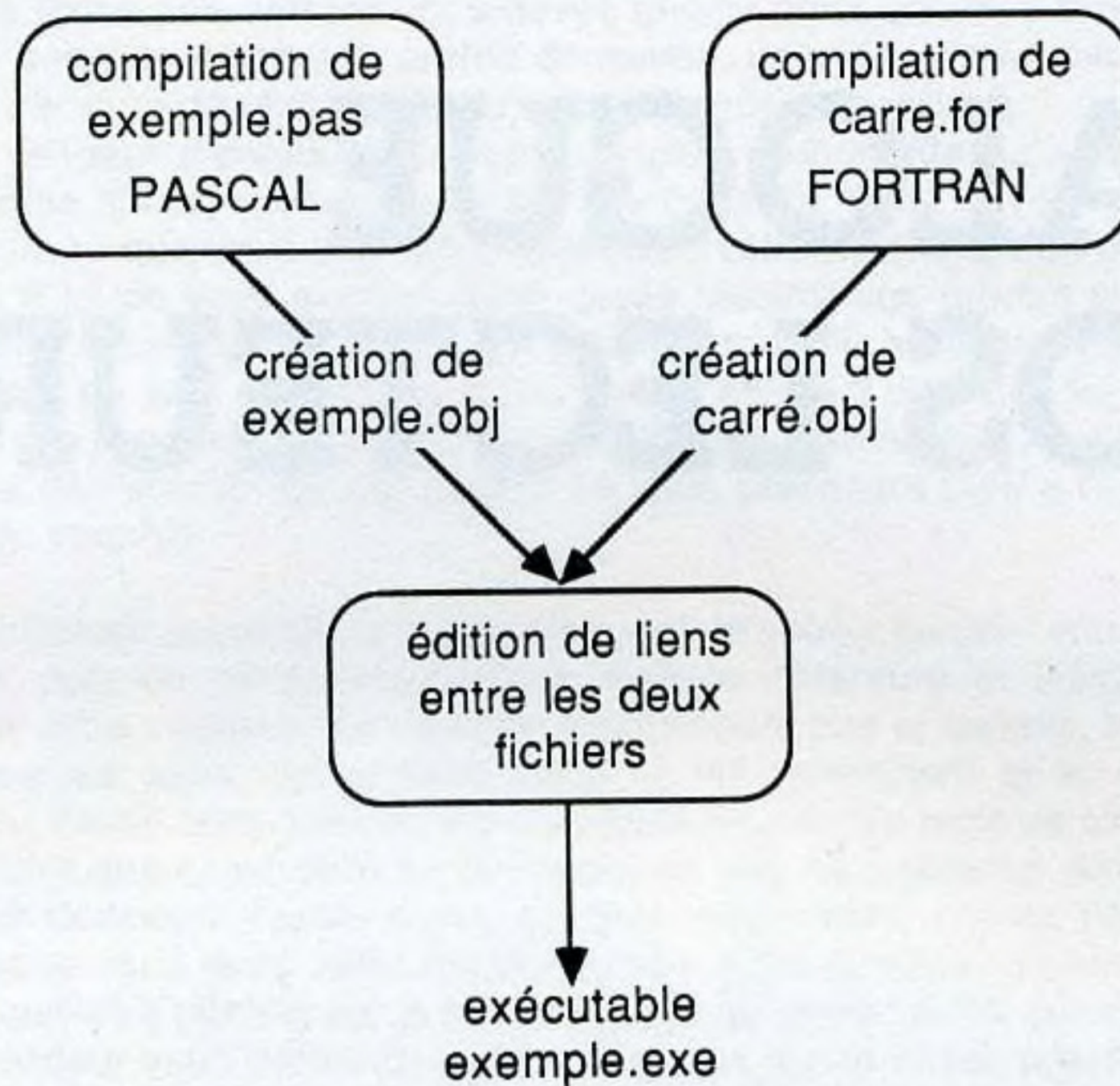
procedure CARRE(nombre:integer); external;

begin
  for i:=1 to 10 do
    begin
      CARRE(i,i_carre);
      write('Le carré de ',i:2);
      writeln(' est :',i_carre)
    end;
end.
```

précise que la procédure
sera incluse au moment
de l'édition de liens

Procédure externe en FORTRAN

```
C      Sous programme Indépendant
C      fichier CARRE.FOR
C
C      SUBROUTINE CARRE(I,N)
C      INTEGER I,N
C
C      debut de la procédure
C
C      N=I*I
C      RETURN
C      END
```



2.8. Bibliothèques de procédures

Le but est de rassembler des modules de même type dans un seul et même fichier pour en faciliter leur accès. A l'édition de liens, il suffira de préciser que tout ce qui manque dans les fichiers objets (définitions de procédures, fonctions, mais aussi constantes, variables) sera à rechercher dans une bibliothèque que l'on précisera.

Beaucoup des fonctions et procédures standard sont déjà disposées en bibliothèque(s). Le linker (c'est ainsi que les anglais appellent l'éditeur de liens), sait par défaut où se trouve cette bibliothèque standard et résout tous les appels présents dans le programme. On comprend donc mieux l'importance de la phase de «édition des liens» pour le bon fonctionnement du programme. Généralement, une bibliothèque renferme des modules objets (c'est-à-dire issus d'une compilation), mais il est fréquent d'avoir des bibliothèques de modules sources ; elles seront dans ce cas utilisées au moment de la compilation ou de l'assemblage.

3. CONCLUSION

Voici donc un premier aperçu des possibilités de développement qu'offre maintenant la plupart des micro-ordinateurs célèbres du marché. Il ne nous était pas possible, pour des raisons de place, d'aller plus en avant dans cette analyse comparative, aussi terminerons-nous le mois prochain avec quelques exemples types (nous reviendrons plus en détail sur le TURBO-PASCAL, le PASCAL UCSD, les compilateurs BASIC, etc.). Nous en profiterons pour vous fournir un glossaire des termes informatiques les plus usités dans ce domaine. Vous serez peut-être ainsi un peu moins perdu en lisant les publicités des fabricants ou les bancs d'essais de nos confrères. Nous n'en oublions pas pour autant les analyses d'applications que nous vous avons promises le mois dernier. Nous commencerons dès le mois prochain avec l'établissement du cahier des charges pour les deux exemples : mini traitement de texte et jeu interactif.

DIALOGUE AVEC NOS LECTEURS

1. REPONSE AU COURRIER

1.1. Conseils d'achat

Beaucoup d'entre vous nous posent des questions fort embarrassantes, nous demandant conseil pour l'achat de matériels. Et encore le problème se subdivise en deux : d'une part ceux qui n'ont rien (ou veulent tout changer) et ceux qui souhaitent simplement agrandir leur configuration. Notre problème est le suivant : nous ne pouvons évaluer ce que vous comptez faire avec votre ordinateur et encore moins ce que vous serez en mesure de faire une fois que l'aurez acquis. En effet, vous pouvez nous apparaître peu enclin à investir temps et argent dans ce passe-temps et simplement vouloir en posséder un uniquement pour parfaire l'éducation de vos deux charmants bambins, nous vous conseillons alors un produit modeste et surprise ! vous vous trouvez emballé par la programmation, vous résolvez tous vos exercices sur un ticket de métro et foncez chez votre revendeur pour commander la dernière nouveauté à 20 000 F, qu'avons-nous fait ? Nous vous avons fait perdre une somme non négligeable correspondant au prix d'achat de votre première machine. Mais plus grave encore, imaginez que nous vous ayons conseillé la deuxième machine et que l'informatique ne vous enthousiasme guère (si, si, cela existe, demandez à notre entourage), certes, vous pourrez le revendre au bout de six mois, mais le marché de l'occasion n'est pas des plus florissants et vous serez fort découragé par votre première approche du monde de l'informatique. Alors que faire ? Nous avons coutume de dire autour de nous qu'il y a analogie entre les ordinateurs et les voitures.

On achète une voiture en fonction de ce que l'on veut en faire mais aussi en fonction de ses capacités à la conduire ; si Ferrari offre des cours de pilotage à ses clients, c'est que cela ne se conduit peut-être pas comme une 2CV avec tout le respect que nous avons pour cette voiture. Autre analogie, tout comme pour les voitures, il y a des inconditionnels d'une marque que rien ne fera changer d'avis et ici aussi les revendeurs ne font pas toutes les marques, ainsi ont-ils toujours intérêt à favoriser les produits qu'ils distribuent, c'est humain.

Pour conserver notre parallèle, nous vous conseillons de faire un essai des produits que vous souhaitez mettre en concurrence. Pour les comparer, quelques critères sont utiles ; la consommation (en électricité bien sûr), sera probablement très proche, mais ce qui peut l'être moins ce sont les performances (voir notre article sur les benchmarks), le coût de l'extension du système (imprimante et autres périphériques) ou encore la diversité des programmes existants (un ordinateur qui aura

prochainement des milliers de logiciels, peut encore attendre ses développeurs deux ans plus tard). Parmi ces critères vous devez choisir ceux qui vous importent le plus, avec comme dans de nombreux autres domaines, une idée de la valeur du service après-vente, car quoi de plus inutile qu'un ordinateur en panne ?

Toutefois, le vendeur d'ordinateurs n'est pas plus malhonnête qu'un autre, et vous pouvez, de prime abord, vous appuyer sur ses conseils. Ainsi, si vous avez une idée bien arrêtée de ce que vous souhaitez faire avec votre machine, rien de plus simple, dites-le lui et à lui de vous montrer dans quelle mesure son produit est capable de répondre à votre attente et à quel prix. Le suivi des produits et les nouveautés à même de les remplacer ne sont annoncées qu'au dernier moment pour que les fabricants ne se retrouvent pas avec des tonnes d'invendus, mais la lecture de «C'est arrivé demain» vous préserve des avatars de cet acabit, en vous prévenant bien à l'avance sur les fluctuations du marché.

Abordons maintenant le problème le plus fréquent de votre courrier : vous avez acheté une machine, puis un périphérique d'une marque différente et malgré toutes les promesses de votre vendeur, les deux ne fonctionnent pas ensemble. Ne croyez pas que vous êtes les seuls, l'un d'entre nous l'a fait sciemment et le «petit» logiciel assembleur qui devait faire l'interface a nécessité plus de six mois de mise au point et personne d'autre que lui ne peut s'y aventurer en cas de problème. Alors, si vous ne souhaitez pas découvrir l'assembleur, ou plus simplement, n'avez pas six mois à perdre, déplacez-vous avec votre machine chez votre détaillant et faites l'essai sur place, c'est peut-être pénible sur le moment, mais au moins on est sûr de son fait, car même si le vendeur vous présente la démonstration sur le «même» ordinateur que le votre, comment savoir que vous êtes exactement dans les mêmes conditions ? Il peut fort bien provenir d'une série ultérieure ou le bogue qui existe sur votre machine dans ce cas-là a été éliminé, ou un composant dont la version de base ne fournit pas les performances requises a été changé, sachant qu'une fois le matériel acheté, vous serez bien obligé d'acquiescer le reste ; aussi, nous ne sommes que trop enclins à vous inciter à la méfiance.

Néanmoins, la plupart du temps votre revendeur ne cherchera nullement à vous bernier, et vous pourrez lui faire confiance. Si vous avez un doute, essayez de savoir de quelle machine il se sert pour gérer vos factures ; il doit en être satisfait.

1.2. Nous avons reçu des lettres nous demandant si nous enseignions quelque part. Hélas pour vous, mais peut-être heureusement pour les élèves éventuels, ce n'est pas le cas.

1.3. Le sujet qui passionne certains d'entre vous est la Téléinformatique, que ce soient les modems eux-mêmes, ou bien l'utilisation de Minitel (y aurait-il une mode ?). Nous vous promettons, dans un avenir proche, un cours sur le fonctionnement des Modulateurs-Démodulateurs et sur les protocoles qu'ils emploient. Pour M. Tegar, de Basse-Terre, notamment, les ordinateurs se «parlent» les uns aux autres en échangeant des 1 et des 0, la plupart du temps par des lignes téléphoniques spécialisées, le modem transformant ses 0 en un premier bruit et ses 1 en un deuxième bruit, et le second modem du second ordinateur entendant le premier bruit le traduit en 0 et l'autre en 1 et le tour est joué. Mais c'est plus complexe à l'emploi et surtout ça va plus vite !

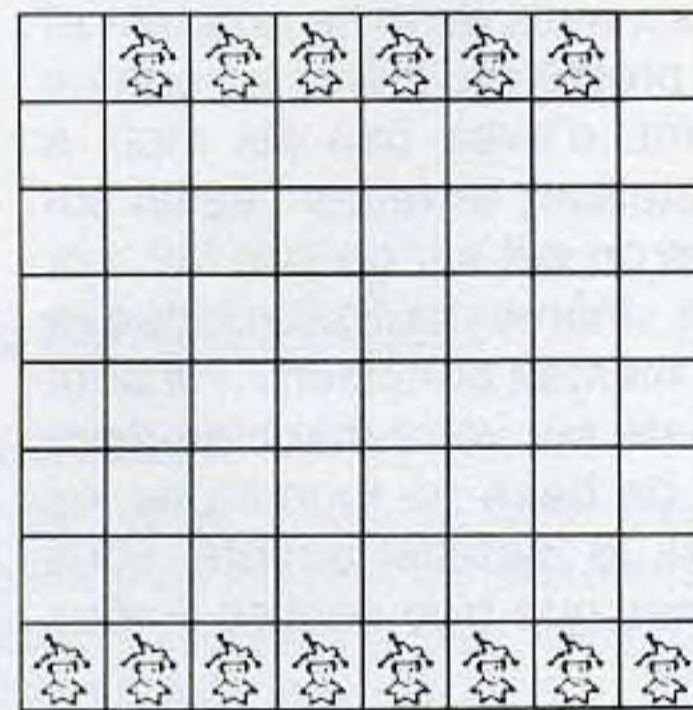
2. SOLUTIONS D'EXERCICES

Ce mois-ci, nous vous donnons enfin la solution du problème du rangement maximal de pièces sur un échiquier, et les résultats s'avèrent pour chacune des pièces envisagées, assez triviaux :

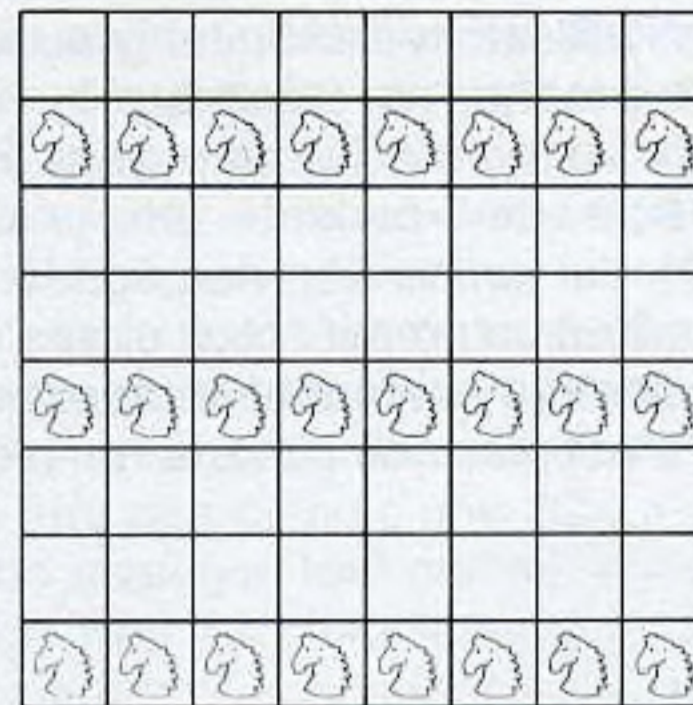
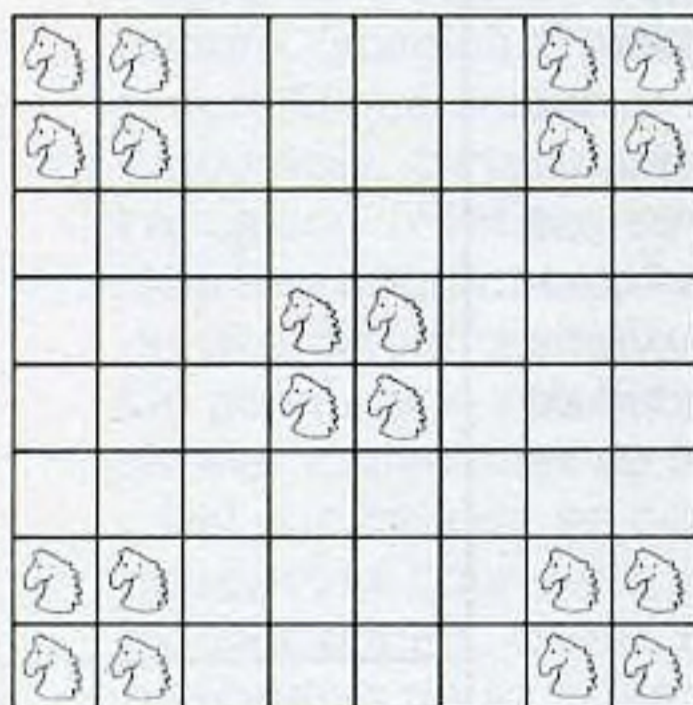
– pour les pions, 4 lignes droites nous fournissent 32 pièces, ou comme nous l'a fait remarquer M. Rhelimi de Ville d'Avray, seulement 28 si l'on considère qu'une fois arrivés sur la ligne adverse, ils se transforment immédiatement en dames. Si l'on élimine les symétries, il n'y qu'une solution.



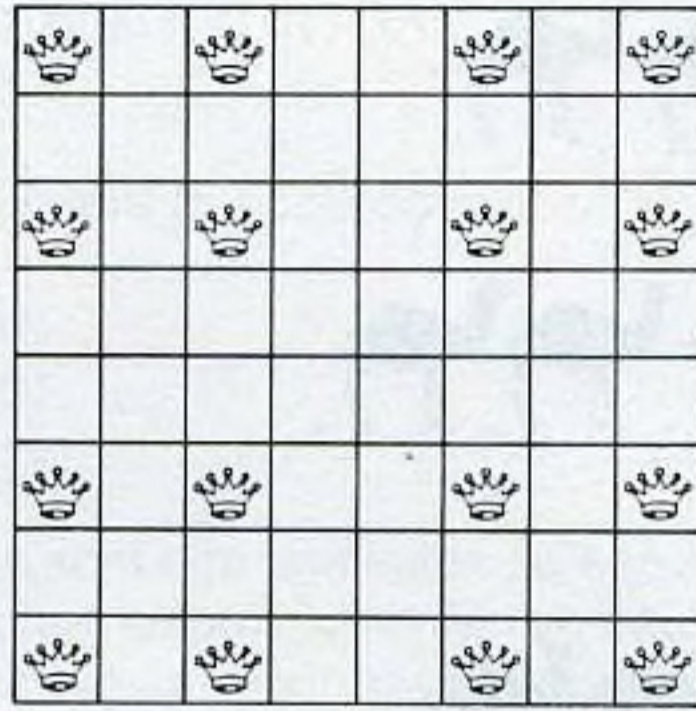
– pour les fous, il suffit de remplir presque entièrement les deux bords, ce qui fait 14 pièces. Il y a deux solutions.



– pour les cavaliers, un programme a donné une mauvaise solution, certes fort jolie puisque n'admettant aucune symétrie étant symétrique elle-même, mais non valable quant au résultat n'obtenant que 20 cavaliers alors que 24 étaient possibles.



– pour les rois, le résultat est également très simple, donnant 16, mais le nombre de solutions semble plus complexe : aucun lecteur ne les a toutes recensées ! Vous vous êtes contentés de nous donner une solution et nous aussi !



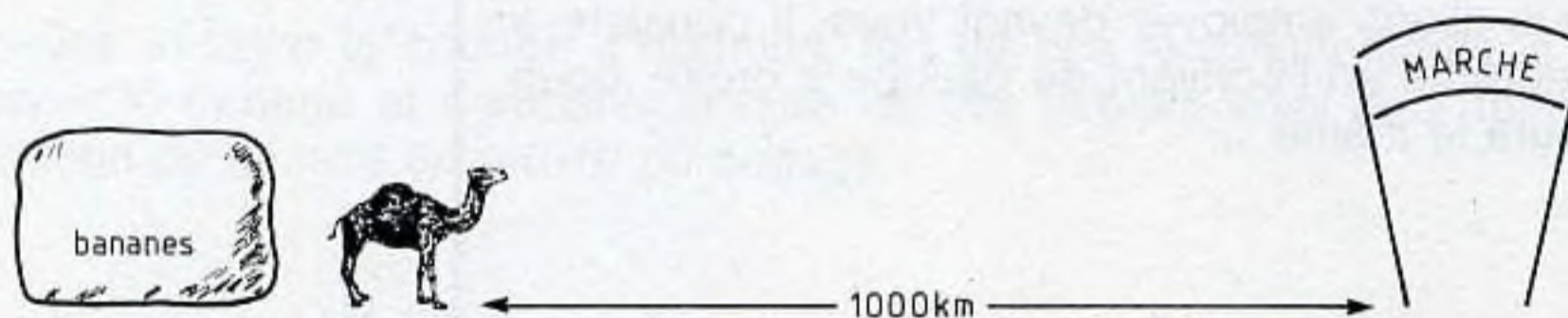
3. EXERCICES

3.1. Exercices avec des chameaux

Comme la plupart des exercices que nous vous avons déjà présentés (tours de Hanoï, carrés magiques ou CEE), ce sont encore des exercices qu'il convient de faire deux fois, la première de façon théorique, la seconde en programmant. Ils ont tous les deux la particularité de mettre en œuvre des chameaux !

Le chameau et les bananes : vous comptez aller vendre vos bananes à un marché qui se trouve à une distance de 1 000 km ; à cet effet, vous avez fait l'acquisition d'un chameau. Malheureusement, il y a quelques contraintes :

- votre chameau n'est capable de supporter que 1000 bananes à la fois,
- votre chameau ne se déplace qu'en consommant une banane par kilomètre,
- vous avez au départ une cargaison de 3000 bananes.

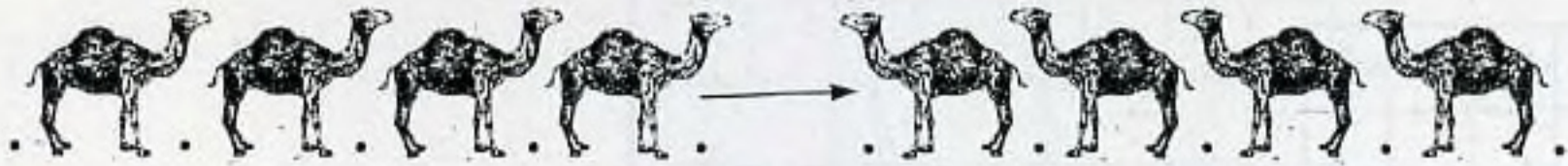


alors se pose la question principale : quel est le nombre maximal de bananes qui parviendra au marché ? Pour ceux qui rétorqueraient immédiatement que c'est impossible, pensez que l'on peut revenir en arrière après avoir déposé des bananes, mais le chameau mange sa banane à chaque kilomètre, qu'il soit chargé ou non. Une fois résolu ce problème, généralisez-le en modifiant, le kilométrage, la charge maximale et la cargaison initiale.

Les chameaux face à face : vous avez en présence deux groupes de 4 chameaux qui se font face et qui désirent se croiser, toutefois leurs déplacements sont réglementés :

- ils ne peuvent avancer que si la place immédiatement devant eux est libre,
- ils ne sont autorisés à sauter par dessus un autre chameau (si !, si !) uniquement lorsque celui-ci se trouve immédiatement devant eux et que la place suivante est libre pour leur permettre d'atterrir.

Il s'agit de permettre aux deux caravanes de se croiser sans provoquer d'embouteillage.



3.2. Le codage d'Huffman

Le but de ce codage est de réduire la taille des fichiers en tenant compte de la probabilité d'occurrence des signes qui le composent. Mais nous pensons qu'un exemple vaut mieux qu'un long discours. Imaginez que vous soyez en présence d'une langue n'utilisant que les six premiers caractères de l'alphabet français avec les probabilités suivantes :

- A : 5 % - B : 5 %
- C : 10 % - D : 20 %
- E : 30 % - F : 30 %

l'idée qui vient tout de suite, c'est de coder à chaque caractère sur trois bits puisque avec deux vous ne pouvez posséder que quatre signes différents (2 à la puissance 2 contre 2 à la puissance 3), et encore il vous reste deux codes libres.

Au contraire, Huffman a pensé qu'il serait plus subtil de donner une longueur différente de codage aux signes selon leur fréquence, les plus répétés étant les plus courts, aussi a-t-il écrit un algorithme que nous allons employer devant vous. Il consiste à construire le code de chacun des caractères en l'écrivant de gauche à droite, nous verrons à la fin qu'aucun caractère n'aura le même :

- A : 5 % : 0 - B : 5 % : 1
- C : 10 % - D : 20 %
- E : 30 % - F : 30 %

maintenant que l'on sait différencier le A du B, on les regroupe

- A&B : 10 % - (A : 0, B : 1)
- C : 10 % - D : 20 %
- E : 30 % - F : 30 %

On regroupe encore les deux ensembles les plus faibles en fréquence, d'une part A & B, d'autre part C, en donnant 0 au premier et 1 au second :

- A&B&C : 20 % - (A : 00 B : 01 C : 1)
- D : 20 % - E : 30 %
- F : 30 %

on fait ensuite l'association avec D :

- A&B&C&D : 40 % (A:000, B:001, C:01, D:1)
- E : 30 % - F : 30 %

Ici les deux groupes les plus faibles sont E et F, on les associe donc :

- A&B&C&D : 40 % (A:000, B:001, C:01, D:1)
- E&F : 60 % (E:0, F:1)

enfin les deux derniers qui donnent le code définitif :

- A : 0000
- B : 0001
- C : 001
- D : 01
- E : 10
- F : 11

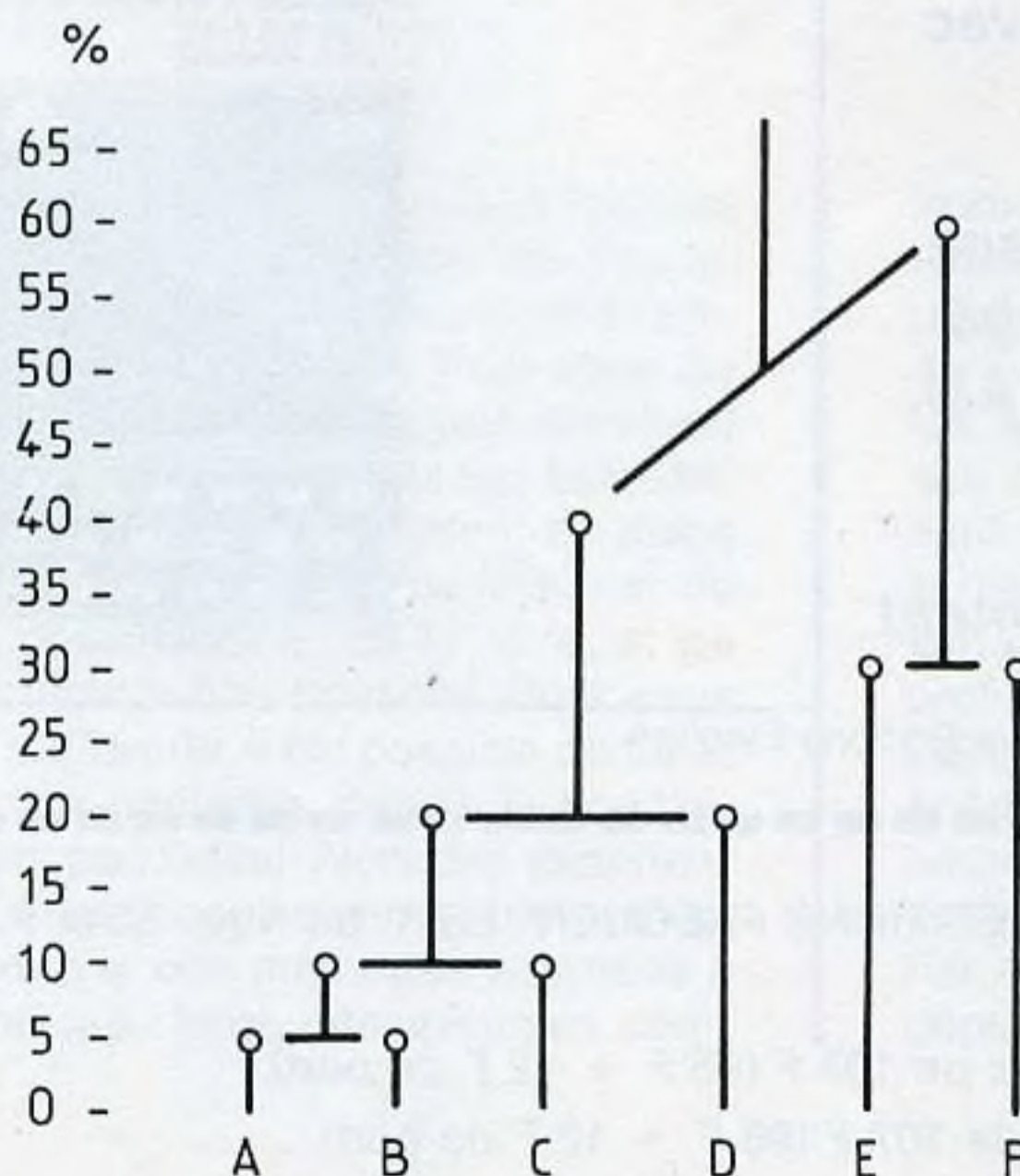
vous remarquez que les codes sont dits terminaux : c'est-à-dire que l'on ne peut avoir deux codes totalement identiques jusqu'à la fin de l'un deux : il n'y a pas de code 0 ou 00 ou 000 ou 1, il faut lire le bit suivant pour savoir à qui on a affaire et une fois atteint un signe complet, il n'en n'existe pas de plus long commençant pareil. Etudions maintenant les performances :

- nous codons sur 4 bits dans 10 % des cas (A ou B)
- sur 3 bits dans 10 % (C)
- sur 2 bits dans 80 % (D ou E ou F)

ce qui donne en moyenne un codage sur 2,3 bits entraînant un gain de 23 % de la taille des fichiers, avouez que cela en vaut la peine. Le problème réside dans le fait qu'il faut savoir à l'avance la probabilité de rencontrer chacun des signes, pour la langue française, elles sont les suivantes selon un traité fort ancien :

- | | | | |
|-------------|------------|-------------|------------|
| - A : 7,68 | - B : 0,80 | - M : 2,72 | - N : 7,61 |
| - C : 3,33 | - D : 3,60 | - O : 5,34 | - P : 3,24 |
| - E : 17,76 | - F : 1,06 | - Q : 1,34 | - R : 6,81 |
| - G : 1,10 | - H : 0,64 | - S : 8,23 | - T : 7,30 |
| - I : 7,23 | - J : 0,19 | - U : 6,05 | - V : 1,27 |
| - K : n. s. | - L : 5,89 | - W : n. s. | - W : 0,54 |
| | | - Y : 0,21 | - Z : 0,07 |

vous pouvez essayer le codage d'Huffman, fort de ces renseignements, à vous de programmer le codage et comparer la taille de vos fichiers voici pour terminer une symbolisation de la mise en œuvre du codage :



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

C'EST ARRIVE DEMAIN



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

Vous le savez, l'ordinateur prend une place toujours plus importante dans la vie, sinon quotidienne, tout au moins professionnelle. Mais, avez-vous pensé à l'attitude d'handicapés devant un clavier ? Pour ceux qui sont incapables de se servir de ce moyen, comment utiliser un ordinateur ? Nous en sommes aux balbutiements, mais diverses méthodes se mettent en place pour permettre à ceux qui n'auraient pu imaginer de communiquer avec une machine, de le faire, et ce d'une façon aussi efficace que possible. Pour ceux qui ont une habileté suffisante, il est possible de taper sur le clavier avec une baguette, depuis la bouche, mais ce n'est vraiment pas l'idéal. Alors des expériences sont en cours d'études dans diverses universités, dont le but est de trouver des méthodes adaptées à chaque cas de handicap. Ainsi, des aveugles com-

munique par ordinateur, via des modems et le réseau des 200 ordinateurs mis à la disposition du public. Cela leur permet d'éviter en partie l'isolement, d'étudier, de faire leurs courses, de payer leurs factures sans écritures. Mais cela n'est que la partie connue des possibilités de l'informatique. Il est maintenant raisonnable de parler de véritables études pour le bien-être des seuls handicapés, et non de l'utilisation par eux de systèmes ouverts à tous, et dont ils profitent. En effet, il y a 13 millions de handicapés aux USA, et il faut bien le dire, cela représente un marché énorme. Ce n'est pas toujours comme cela qu'ils sont perçus, mais pourquoi se voiler la face, au niveau du développement de matériel, cela est important aussi. Par exemple, un capteur à infrarouge, qui permet de déplacer un curseur sur l'écran, comme si l'on dispo-

sait d'une souris, et dont la validation se fait, non par frappe d'une touche ou d'un bouton, mais en clignant des yeux. Ce système a permis à plusieurs infirmes de communiquer avec leurs familles, car ils étaient paralysés des membres et muets, souvent après un accident de la voie publique. Un ordinateur particulier, appelé VersaBraille, permet de sortir des textes en Braille et dispose d'un clavier en Braille, etc. Ainsi, il est possible aux aveugles de disposer également d'ordinateurs pour leurs besoins, sans se déplacer. Pour beaucoup, cette possibilité de communiquer représente la différence entre le handicap et la vie des autres.

L'aide aux handicapés représente une somme de petites modifications de la vie courante, certaines très simples, d'autres qui s'avèrent être des merveilles technologiques coûteuses. L'informatique n'échappe pas à ce principe et commence à sortir les handicapés du ghetto. Comme toujours, il faut dire que les progrès n'en sont qu'aux balbutiements, mais beaucoup de gens trouvent que cela transforme leur vie. Mais il est certain qu'il ne faut pas considérer que tout est fait.



Un prêtre, écrivain de surcroît, vient de se voir soumettre une requête très surprenante. Un étudiant (28 ans) en informatique du MIT, le fameux Massachusetts Institute of Technology, lui demande une bourse pour prouver l'existence de Dieu à travers l'immense quantité de données que l'on peut trouver sur la nature du monde. Il déclare en bref : «Je débusquerai Dieu de son refuge». Selon lui, les informations dont nous disposons aujourd'hui sont révélatrices de l'existence de Dieu, mais en traitant scientifiquement, il sera possible de prouver, ce qui est différent, son existence. Roger Lambert, le prêtre, a repoussé cette demande, devant son manque de rigueur esthétique et éthique. L'esthétique car cela laisserait Dieu prisonnier de données, l'éthique car cela nuirait à la foi. Selon Lambert, un Dieu dont l'existence serait prouvée serait juste ennuyeux. Mais, l'étudiant a obtenu ailleurs de quoi commencer ses recherches et de l'épouse du prêtre une attention très particulière. Le résultat de cette étude : rien de probant.

Au fait, il s'agit d'un livre, qui est le best-seller du moment.

Il était temps que certaines langues se délient. Des cadres, utilisateurs de micro-ordinateurs, viennent de déclarer que le service après-vente de quelques compagnies est lamentable et coûteux. Cela vise en particulier Ashton-Tate, auteur et distributeur de la série des D-Base. Parmi les plaignants, des dirigeants de Mac-Donnell-Douglas, le fabricant d'avions. Ce type de clients est d'ordinaire pourtant très soigné. Mais ils trouvent que payer près de 300\$ par an pour obtenir un service déficient, lent et incapable est totalement ridicule. Ils viennent donc de se réunir, pour imposer au fabricant de programmes de modifier leur service. Résultat : Ashton-Tate vient d'annoncer une nouvelle politique de son service clients. Cela va de la suppression pure et simple des protections à la livraison gratuite des releases, ces modifications permanentes des logiciels. Il semble que le mouvement soit bien amorcé, et donc que les autres devront suivre pour survivre.

IBM a bien des malheurs. Alors que toute la communauté informatique a eu bruit de ses replis successifs sur le marché de la micro, chacune des initiatives du fabricant semble tomber à plat ou donner lieu à des critiques. On croirait entendre les commentaires sur Apple d'il y a deux ans. Par exemple, IBM vient d'annoncer et de montrer un nouveau XT, destiné à remplacer celui qui se voit menacé par la concurrence. Les essais des confrères locaux sont catastrophiques. Rien ne trouve grâce à leurs yeux. Ils considèrent que ce nouveau XT est en fait un AT, et que le prix plus bas est juste une manœuvre commerciale. Tous les gros utilisateurs, comme les banques, les assurances et autres grosses sociétés, estiment qu'il y a là un abus de la part de IBM qui leur avait suggéré de s'équiper en AT voilà quelques mois. Par exemple, Home Insurance Co a acheté voilà trois mois 450 AT. Voilà un joli parc qui devient obsolète. D'autant plus, que même au prix du nouveau XT, soit 4000\$, les compatibles sont toujours plus performants à prix égal.



Le vrai problème est que ce type d'entreprise représente le gros des clients d'IBM, et que ceux-ci, mécontents de l'attitude du géant informatique, se tournent de plus en plus vers les compatibles, venus de Corée ou simplement du Texas. Et ces parts de marchés sont alors très difficiles à récupérer, voire impossible si le fabricant de compatibles fait son travail, ce qui est le cas pour le moment. En effet, imaginez ces clients à qui IBM avait déclaré que le AT était un standard pour longtemps, et qui voient que le nouveau XT est une sorte de AT en moins cher. D'autant plus que certains ont acheté... 500 AT à IBM. Ils peuvent être mécontents.

Savez-vous que le scanner est très à la mode en micro-informatique lourde ? Ce scanner là est un lecteur optique qui permet de lire une page de texte, et de le restituer sous un traitement de texte. Car, s'il est simple de lire du texte comme un dessin, c'est-à-dire de le considérer comme du point par point, il est beaucoup plus difficile de le comprendre comme une série de lettres, car reconnaître une lettre est un travail difficile, sur lequel les experts en reconnaissance des formes travaillent d'arrache-pied. En effet, les polices de caractères ne sont pas normalisées, certaines imprimantes simples, de faible qualité, sont irrégulières, etc. En plus, il faut comprendre l'italique, le souligné, et autres arrangements de présentation. Aussi, les scanners sont-ils coûteux, de 3000\$ pour les plus simples, à des sommets aussi impressionnants que 100 000\$. Les moins chers permettent de reconnaître une police personnelle, mais en général sans les soulignés, italiques et caractères spéciaux. Les plus chers mémorisent des centaines de polices de caractères, avec tous les artefacts de présentation. AST vient de proposer un scanner simple à 2300\$, ce qui constitue une prouesse. Sa résolution est bonne, de 300 points par inch (soit 120 points au centimètre), sa vitesse correcte. Il représente un progrès certain dans le cadre des utilitaires accessibles pour une petite entreprise, car il permet de ne pas retaper du texte déjà dactylographié. Avec un scanner une journée de frappe se transforme en 20 minutes de lecture pour l'appareil. Cela s'appelle le progrès.

Le monde des grandes expositions est inquiet. Il devient évident que les petites expositions, plus locales, plus limitées, mais plus accessibles au commun des utilisateurs, sont plus suivies. Il est plus facile d'obtenir des informations lors des petites expositions, car les exposants sont moins débordés, plus disponibles. De plus, ces salons sont souvent très spécialisés, et permettent de rencontrer des techniciens à même de répondre à des questions précises, au contraire de ceux qui sont employés pour les grandes expositions, comme le Comdex, où la personne que vous interrogerez est entraînée à connaître moins de détails sur plus de produits. Moralité, les grands salons ne sont plus rentables, les petits gagnent de l'argent et les exposants deviennent rares aux salons géants.

Une barrière psychologique vient de tomber pour Apple. Le plus gros fabricant de produit pour IBM PC et compatibles, AST, vient d'annoncer la création d'un département spécialisé en produits pour la gamme Macintosh et Apple II GS, le nouveau 16-Bits d'Apple. Ce qui est important est que cette société marque par là que Apple est devenue crédible auprès des grosses sociétés, gourmandes en améliorations adaptées à leurs besoins spécifiques. C'est une grande première depuis l'Apple II, car le Macintosh en particulier n'avait jusqu'alors pas bénéficié de ce type d'attention, son architecture fermée étant aussi coupable. Mais le Mac + , et le futur Mac ouvert sont des arguments qui n'ont plus de raisons d'être ignorés. C'est là un bon point à mettre au crédit d'Apple. Au fait, le nouveau Mac ouvert sera en fait des nouveaux Mac ouverts, car toute une gamme devrait être dévoilée, allant de 4000\$ à 8000\$. UNIX sera proposé en système d'exploitation. L'envol des prix chez Apple est impressionnant, mais parallèle à l'intérêt de leurs produits.

Au sujet de la lutte entre le standard IBM et le standard MAC, il est intéressant de se reporter à une série de conférences données, il y a maintenant un an. A cette époque, alors que John Sculley arrivait chez Apple, il avait déclaré que IBM perdrait très vite des parts de marché, car imposer un standard médiocre ne pouvait avoir de vrai avenir. Tout le monde avait ri. Les rieurs se sont calmés.

Amstrad, dont vous pouvez voir tourner les excellents produits en France, commence sa percée aux USA. Son compatible PC est vraiment extraordinaire, proposant pour un prix 50 % inférieur à un PC d'IBM, des possibilités totalement absentes chez IBM. Il n'y pas de raison que le succès ne vienne rapidement. Notons que la plus grosse firme de diffusion par correspondance, Conroy-La-Pointe, propose un compatible XT pour 850\$. La concurrence de plus en plus vive dans ce domaine est certainement un signe de bonne santé du standard IBM-PC, il est simplement dommage que IBM n'en tire aucun profit.

Des anciens membres de l'équipe de développement de Visicalc, le premier tableau de l'histoire, ont annoncé un programme de type Lotus 1-2-3, aussi complet, pour un prix de 30\$. Sachant que l'original coûte près de 500\$, Ontio 259 le nouveau produit, semble promis à un bel avenir. D'autant plus que de nombreuses protections, non disponibles sur 1-2-3- sont proposées en série.

Atari vient de rendre public une information qui prouve la bonne santé de cette dynamique entreprise. Le comité d'exploitation a décidé de mettre Atari sur le marché boursier. 4,5 millions d'actions vont donc être proposées au public. Leur prix initial sera d'environ 12\$. Il semble que la somme réunie permettra de payer les dettes les plus criantes de la société californienne, en particulier vis-à-vis de l'ancien proprié-

taire, Warner Communications. Peut-être ceci ralentira-t-il l'intérêt des investisseurs, mais ce serait dommage car les Atari 520 et 1040 ST sont de très bons produits. Il leur reste néanmoins à prouver leur possibilité de diffusion sur les différents marchés offerts, USA et Europe.

Et maintenant, la belle histoire du mois, mais qui, contrairement aux contes de fées, peut mal finir.

En 1977, un coréen sans un sou, émigré lors de la guerre de Corée, débute dans son garage (cela devient d'un banal, il faudrait penser aux salles de bain, aux caves, aux greniers...), la fabrication de terminaux vidéo à bas prix pour ordinateurs. Le succès est rapide, et lors de l'introduction en Bourse, la valeur de la société, Télévidéo, passe de 163 000 000\$ à 1 100 000 000\$ en quelques jours. Cela s'appelle, en français dans le texte, une «succes story».

Un petit air de violon s'il vous plaît...

En 1984, Hwang le créateur de Télévidéo, décide de s'attaquer au marché des compatibles IBM. Echec complet dû à la baisse venant des compatibles et à laquelle IBM se joint. Télévidéo n'est plus performant. La chute est aussi rapide que l'a été le succès. Maintenant, Télévidéo végète, mais survit. Il semble en fait que les perspectives ne soient pas trop mauvaises, mais le plein succès sera long à revenir.

Dans notre partie essai, nous allons parler ce mois-ci, des programmes de traitements graphiques, au sens large. Cela va du traitement de données financières à l'analyse de résultats, et au calcul scientifique. Les programmes capables de ce type de performances apparaissent à un rythme effréné, sur tous les types de matériels. Nous allons ici en décrire les grandes catégories, avec les principaux atouts, et surtout les limitations importantes qui souvent font toute la différence entre un programme que l'on utilise avec plaisir et un programme qui incite à taper sur son ordinateur car il semble parfois scandaleux de ne pas disposer de certaines possibilités. Par exemple, certains programmes d'analyses de données, avec sorties sous forme de camemberts, de graphiques et autres barres, ne permettent pas de relire des fichiers de données provenant d'autres programmes, comme des tableurs. Il faut alors retaper toutes les données pour disposer des sorties graphiques. C'est simplement ridicule, mais cela existe encore.

Il semble que l'on se rende enfin compte que 99 % des gens ne peuvent dessiner correctement des courbes, des graphes et autres. Donc un programme qui le fasse est une nécessité pour présenter correctement des résultats. Parmi ces programmes, les plus

récents sont Freelance, Chart-Master, Harvard Presentation Graphics, ou FullPaint.

Enfin, il faut noter que le standard MS-DOS a repris récemment du poids dans ce domaine, grâce à la sortie du standard EGA, qui est accepté par la plus grande part des développeurs. Ainsi, le MS-DOS qui pêchait sur le plan graphique peut-il maintenant n'être plus absent de la scène. De plus, ce standard est de bonne qualité et propose des sorties couleurs, ce qui est un excellent point. Parmi les programmes qui tirent le mieux parti de cette évolution, Harvard Presentation Graphics est un des pionniers de l'utilisation de l'EGA. Il est possible de créer tous les types de graphiques habituels, avec ou sans texte de présentation et de légende. Il est très simple d'utilisation, avec des menus intégrés, et il suffit de rentrer les données, ou de les lire ailleurs, depuis quasiment tout programme de traitement de nombre. Enfin, une palette permet d'améliorer les résultats, comme sur un MAC. Un beau compliment pour un programme un peu cher, 400\$. Il existe des programmes presque aussi bons, depuis 39\$. Un autre excellent programme pour PC compatibles est celui de HP, du nom de Graphics Gallery. Comme le précédent, la récupération de données venant d'ailleurs, est simple, et dont les sorties sont aisément transférables sous des traitements de textes. Son prix est prohibitif, comme souvent chez Hewlett-Packard, 700\$. Plus abordable, Cricket Graphic, sur le Mac qui coûte 200\$ et propose bien plus de possibilités que Chart, qui vaut lui 120\$. Freelance pose un problème majeur. Il est très performant et complet, mais il ne permet pas d'entrer les données directement au clavier, mais demande un fichier prédéfini de données. Ceci peut être une vraie limitation dans le cas d'une utilisation personnelle, mais tombe dès que vous disposez d'un tableau ou autre programme de traitement.

En conclusion, il faut dire que ces programmes, qui sont de bien beaux joujous, sont en général assez coûteux, et que l'investissement ne se justifie pas, en général, dans le cadre d'une utilisation personnelle. Par contre, si vous en avez un vrai besoin, n'hésitez plus, faites vous plaisir. Il faut dire que peu de programmes récents se comportent mal, mais que certains plus anciens, tels Chart, ou Graph-In-The-Box, sont assez limités, le premier par ses possibilités en général, larges mais jamais très développées, le second par ses performances générales, sa documentation, sa présentation et son prix, 100\$ pour finalement très peu.

Au mois prochain.

COURS D'INITIATION AU PROGICIEL MULTIPLAN

Charles-Henry Delaleu

3^e PARTIE

Nous voici arrivés à notre troisième cours sur Multiplan. Grâce aux deux premiers cours, nous avons acquis les bases de ce progiciel. Aujourd'hui, nous abordons le cœur de ce tableur. Aussi, afin de nous faciliter la tâche, une large partie est consacrée aux exemples. Deux exercices sont à réaliser par vos soins. Leurs résultats seront donnés le mois prochain.

Bien que Multiplan soit un progiciel à part entière, et qu'il ne s'agisse pas d'un module d'un progiciel multifonctions du style FRAMEWORK, OPEN ACCESS, etc., il est possible de l'interfacer, dans une certaine mesure, avec d'autres programmes. Ainsi, la plupart des progiciels de comptabilité autorisent le passage des chiffres compris dans Multiplan dans leurs propres données. Certains programmes de traitement de texte (ex. : TEXTOR) permettent, eux aussi, le passage des données grâce à un utilitaire approprié. Ainsi, Multiplan peut servir de base de données (numériques) à de nombreux progiciels. Bien que la tendance soit aux progiciels intégrés, il est bon de savoir que de nombreux progiciels sont interfaçables entre eux. L'avantage des progiciels intégrés concerne les possibilités de leurs fonctions. Les progiciels spécialisés sont généralement plus puissants que les outils compris dans les intégrés. Ainsi, sans passage direct des informations, il est possible d'obtenir, avec plusieurs progiciels spécialisés, une base de travail plus puissante. Ce que l'on perd en simplicité d'utilisation, on le regagne en puissance. Conclusion, si vous êtes en possession de Multiplan, lors de l'achat d'autres progiciels, vérifiez toujours qu'il existe des «ponts».

FONCTION SOMME

Nous avons étudié la manière avec laquelle il était possible de réaliser une addition sous Multiplan. (Par exemple $L7C5 = L7C2 + L7C3 + L7C4$).

Cette méthode simple ne peut s'appliquer qu'à une série de cellules limitée. En effet, dans les autres cas, l'écriture s'avère délicate. Afin de simplifier le travail de l'utilisateur, il existe une fonction appelée SOMME permettant une programmation simplifiée des grandes additions.

Exemple :

	2	3	4	5	6	7	8
2							TOTAL
3		200	300	75	22		?
4		12	27	9	1 024		?
5							
6				TOTAL	?		

a) Nous désirons connaître la somme des cellules L3C3 à L3C6. Avec SOMME, cela s'écrit :

SOMME (L3C3 : 6) pour la cellule L3C8.

b) Nous désirons connaître la somme des cellules L3C6 à L4C6. Avec SOMME, nous programmons :

SOMME (L3 : 4C6) pour la cellule L6C6.

c) Nous désirons connaître la somme des cellules comprises de L3C3 à L4C6, soit toutes les cellules contenant des données. Nous écrivons :

SOMME (L3 : 4 C3 : 6) pour la cellule L4C8.

Cette opération se réalise comme pour toutes les autres opérations en MODE CALCUL. Les cellules à additionner se trouvent placées entre les parenthèses qui suivent le MOT somme. Ce dernier doit être marqué en toutes lettres. Aucun espace blanc ne sera placé entre le S et la dernière parenthèse. L'extension : est bien entendu autorisée, et même préférable.

FONCTION NOM

Afin de simplifier la programmation d'une feuille de calculs, certaines cellules contenant des valeurs fixes – comme des taux de T.V.A. par exemple – peuvent être nommées. Ce nom peut être utilisé en mode CALCUL.

Exemple de trois taux de T.V.A. : 7 %, 18,6 %, 33,33 %.

Ces taux placés en L2C3, L3C3, L4C3 et nommés

7 % placé en L2C3 sera appelé TAUX1	} Afin de se rappeler les noms utilisés, il est préférable de les écrire sur la cellule précédente en mode ALPHA
18,6 % placé en L3C3 sera appelé TAUX2	
33,33 % placé en L4C3 sera appelé TAUX3	

Nous pourrions donc écrire les formules suivantes :

	2	3	4	5	6	7
2	TAUX1	7				
3	TAUX2	18,60				
4	TAUX3	33,33				
5						
6		REF	Prix HT	Prix TTC		
7						
8		livre	102	?		
9		peinture	67	?		
10		voiture	65 000	?		

$L8C5 \Rightarrow L8C4 \times TAUX1$

$L9C5 \Rightarrow L9C4 \times TAUX2$

$L10C5 \Rightarrow L10C4 \times TAUX3$

LES REFERENCES RELATIVES ET LES REFERENCES ABSOLUES

Jusqu'ici nous avons étudié la méthode des références absolues. Chaque cellule utilisée est nommée par son adresse exacte. Exemple : L4C2 (ligne 4 colonne 2).

Il existe une seconde méthode dans Multiplan qui est désignée par la méthode dite des références relatives. Cette possibilité bien que plus difficile à assimiler dans un premier temps, autorise en réalité une bien meilleure souplesse d'utilisation d'un tableur. De plus, cette option permet toute une série de recopies impossibles à réaliser en calcul avec les références absolues.

Soit l'addition simple suivante :

$$10 \text{ voitures bleues} + 15 \text{ voitures blanches} + 27 \text{ voitures rouges.}$$

Sur un tableau, nous avons :

	colonne 2	colonne 3	colonne 4
ligne 3	voit. bleues		10
ligne 4	voit. blanches		15
ligne 5	voit. rouges		27
ligne 6			
ligne 7	TOTAL		52

a) L'addition effectuée L7C4 est égale, en mode absolu, à :

$$L7C4 \Rightarrow L3C4 + L4C4 + L5C4$$

b) En mode relatif, cette équation s'écrit :

$$L7C4 \Rightarrow L(-4)C + L(-3)C + L(-2)C$$

Explication : Nous plaçons notre cellule active en L7C4. Il s'agit d'additionner le contenu des cellules L3C4, L4C4 et L5C4.

La cellule L3C4 se trouve sur la même colonne que L7C4 mais 4 lignes au-dessus en mode relatif, elle s'écrit L(-4)C. L'indication comprise entre parenthèses indique la position relative de la ligne et/ou de la colonne désignée.

Soit le tableau suivant :

	2	3
4	A	C
5	B	cellule active

Il convient d'indiquer, par rapport à la cellule active, la position en relatif des trois autres cellules comprises dans le tableau

A) L4C2 \Rightarrow L(-1)C(-1)

B) L5C2 \Rightarrow L C(-1)

C) L4C3 \Rightarrow L(-1)C

EXERCICE : Donner la position des cellules comprises dans le tableau suivant en fonction de la position de la cellule active :

Nota : Ne donner que les cellules contenant une étoile.

	2	3	4	5
2	★	★	★	
3	★	★	★	
4	★	★	cellule active	
5				

LES EXEMPLES

EXEMPLE N° 1

Soit le calcul des achats des matières premières d'un atelier de mécanique. Le but est de connaître le montant des achats hors taxes et TTC du mois. Afin de faciliter le calcul du reversement de la TVA au fisc, les différents taux de TVA sont ventilés. Les calculs sont réalisés, dans un premier temps, par référence absolue des cellules, puis dans un second temps par référence relative.

Les achats se répartissent de la manière suivante :

- 100 vis de 5 x 20
- 200 écrous de 5
- 10 m² d'acier de 5 mm d'épaisseur
- 5 m² d'aluminium de 2 mm d'épaisseur
- 20 m de tube acier de 20 mm de diamètre
- 30 m de tube inox de 20 mm de diamètre
- 1 kg de graisse
- 150 g de poudre d'or pour dorure
- 30 rouleaux d'abrasif
- 12 litres de peinture noire
- 8 litres de peinture rouge
- 4 litres de peinture blanche
- 2 fraises de 50 mm
- 4 lames à découper

EXEMPLE N° 2

Soit le calcul des versements d'un emprunt de 100 000 F sur une durée de 6 mois au taux de 17,50 % l'an.

- 1) Calcul du remboursement
- 2) Calcul du coût total.

Nota : Réaliser l'exercice en mode absolu et en mode relatif.

EXEMPLE N° 3

Soit le suivi des ventes réalisées par quatre attachés commerciaux sur quatre mois. La société concernée vend des aspirateurs à 1 000 F sur lesquels elle réalise une marge nette de 30 %. Ses charges sont de 15 000 F par mois et par vendeur. Les frais généraux sont en augmentation de 10 % tous les mois en raison du lancement d'une campagne publicitaire. Il convient de connaître l'état de la trésorerie de l'entreprise pour chaque mois, en fonction du nombre d'aspirateurs vendus. Le premier mois, les frais généraux sont de 10 000 F.

Nota : Réaliser l'exercice en mode relatif.

EXERCICE N° 2

Cet exercice n'est pas résolu dans ce cours, les résultats seront donnés le mois prochain. Soit la vente sur 12 mois de fers à repasser. Un fer coûte, hors taxes, 200 F, la TVA est de 18,60 %. Il se vend 20 fers au mois de janvier. A partir de février, les ventes sont en progression de 15 % chaque mois par rapport au mois précédent. Calculer les chiffres mensuel et annuel des ventes TTC, hors taxes et du montant de la TVA. Cet exercice est à réaliser en mode relatif.

EXEMPLE N° 1, LES DONNEES

REF	QTE	PU.HT	PT.HT	TVA	P.TTC
VIS 5*20	100	0,02 F	2,00 F	0,37 F	2,37 F
EC 5	200	0,01 F	2,00 F	0,37 F	2,37 F
ACIER 5	10	150,00 F	1500,00 F	279,00 F	1779,00 F
ALU 2	5	125,00 F	625,00 F	116,25 F	741,25 F
TUBE AC20	20	57,00 F	1140,00 F	212,04 F	1352,04 F
TUBE IN20	30	65,00 F	1950,00 F	362,70 F	2312,70 F
GRAISSE	1	12,00 F	12,00 F	2,23 F	14,23 F
P.OR	0,15	98000,00 F	14700,00 F	4899,51 F	19599,51 F
RO.AB	30	12,45 F	373,50 F	69,47 F	442,97 F
P.NOIRE	12	57,00 F	684,00 F	127,22 F	811,22 F
P.ROUGE	8	59,00 F	472,00 F	87,79 F	559,79 F
P.BLANCHE	4	58,00 F	232,00 F	43,15 F	275,15 F
F.50	2	658,89 F	1317,78 F	245,11 F	1562,89 F
LAM.DEC	4	63,12 F	252,48 F	46,96 F	299,44 F
TOTAL			23262,76 F	6492,18 F	29754,94 F
TVA 18,60 :		1592,67 F			
TVA 33,33 :		4899,51 F			

EXEMPLE N° 1, PROGRAMMATION EN MODE ABSOLU

"REF"	"QTE"	"PU.HT"	"PT.HT"	"TVA"	"P.TTC"
"VIS 5*20"	100	0,02	L4C3*L4C4	L4C5*0,186	SOMME (L4C5:6)
"EC 5"	200	0,01	L5C3*L5C4	L5C5*0,186	SOMME (L5C5:6)
"ACIER 5"	10	150	L6C3*L6C4	L6C5*0,186	SOMME (L6C5:6)
"ALU 2"	5	125	L7C3*L7C4	L7C5*0,186	SOMME (L7C5:6)
"TUBE AC20"	20	57	L8C3*L8C4	L8C5*0,186	SOMME (L8C5:6)
"TUBE IN20"	30	65	L9C3*L9C4	L9C5*0,186	SOMME (L9C5:6)
"GRAISSE"	1	12	L10C3*L10C4	L10C5*0,186	SOMME (L10C5:6)
"P.OR"	0,15	98000	L11C3*L11C4	L11C5*0,3333	SOMME (L11C5:6)
"RO.AB"	30	12,45	L12C3*L12C4	L12C5*0,186	SOMME (L12C5:6)
"P.NOIRE"	12	57	L13C3*L13C4	L13C5*0,186	SOMME (L13C5:6)
"P.ROUGE"	8	59	L14C3*L14C4	L14C5*0,186	SOMME (L14C5:6)
"P.BLANCHE"	4	58	L15C3*L15C4	L15C5*0,186	SOMME (L15C5:6)
"F.50"	2	658,89	L16C3*L16C4	L16C5*0,186	SOMME (L16C5:6)
"LAM.DEC"	4	63,12	L17C3*L17C4	L17C5*0,186	SOMME (L17C5:6)
"TOTAL"			SOMME (L4:17C5)	SOMME (L4:17C6)	SOMME (L4:17C7)
"TVA 18,60 :"					
"TVA 33,33 :"					
			SOMME (L4:17C6)-L11C4		
			L11C6		

EXEMPLE N° 1, PROGRAMMATION EN MODE RELATIF

"REF"	"QTE"	"PU. HT"	"PT. HT"	"TVA"	"P. TTC"
"VIS 5*20"	100	0,02	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"EC 5"	200	0,01	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"ACIER 5"	10	150	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"ALU 2"	5	125	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"TUBE AC20"	20	57	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"TUBE IN20"	30	65	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"GRAISSE"	1	12	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"P. OR"	0,15	98000	LC(-2)*LC(-1)	LC(-1)*0,3333	LC(-2)+LC(-1)
"RO. AB"	30	12,45	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"P. NOIRE"	12	57	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"P. ROUGE"	8	59	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"P. BLANCHE"	4	58	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"F. 50"	2	658,89	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"LAM. DEC"	4	63,12	LC(-2)*LC(-1)	LC(-1)*0,186	LC(-2)+LC(-1)
"TOTAL"			SOMME (L(-15)C:L(-2)C)	SOMME (L(-15)C:L(-2)C)	SOMME (L(-15)C:L(-2)C)
"TVA 18,60 :"					
"TVA 33,33 :"					

SOMME (L(-17)C(+3):L(-4)
C(+3))-L11C6
L(-11)C(+3)

EXEMPLE N° 2, LES DONNEES

TAUX	MONTANT	JAN	FEB	MAR	AVR	MAI	JUI
TAUX : 17,5	100000						
TAUX MENS : 1,458333333							
		18125,00 F	18125,00 F	18125,00 F	18125,00 F	18125,00 F	18125,00 F
TOTAL FAYER:	108750,00 F						
COUT	: 8750,00 F						

EXEMPLE N° 2, PROGRAMMATION EN MODE ABSOLU

```

"TAUX      : "
"TAUX MENS:"
17,5
L1C2/12
"JAN"
(L1C5/6)+(L1C5*L2C2/100 (L1C5/6)+(L1C5*L2C2/100 (L1C5/6)+(L1C5*L2C2/100 (L1C5/6)+(L1C5*L2C2/100
)
)
)
"TOTAL PAYER:"
"COUT      : "
SOMME(L6C2:7)
LBC3-L1C5
"MAR"
"AVR"
"MAI"
"JUI"
"JUI"
MONTANT : "
100000

```

EXEMPLE N° 2, PROGRAMMATION EN MODE RELATIF

```

"TAUX      : "
"TAUX MENS:"
17,5
L(-1)C/12
"JAN"
(L(-5)C(+3)/6)+(L(-5)C( LC(-1)
+3)*L(-4)C/100)
"TOTAL PAYER:"
"COUT      : "
SOMME(L(-2)C(-1):L(-2)C
(+4)
L(-1)C-L(-8)C(+2)
"MAR"
"AVR"
"MAI"
"JUI"
MONTANT : "
100000
LC(-1)
LC(-1)
LC(-1)
LC(-1)

```

EXEMPLE N° 3, LES DONNEES

VENTES	JANVIER	FEVRIER	MARS	AVRIL
VENDEUR1	56	45	45	63
VENDEUR2	12	60	78	78
VENDEUR3	89	45	95	102
VENDEUR4	41	61	56	80
TOTAL	198	211	274	323

CHIFFRE				
VENDEUR1	56000	45000	45000	63000
VENDEUR2	12000	60000	78000	78000
VENDEUR3	89000	45000	95000	102000
VENDEUR4	41000	61000	56000	80000
TOTAL	198000	211000	274000	323000

CHARGES				
SALAIRES	60000	60000	60000	60000
FRAIS	10000	11000	12100	13310
TOTAL	70000	71000	72100	73310

ACHATS				
	138600	147700	191800	226100

RESULTAT				
	-10600	-7700	10100	23590

EXEMPLE N° 3, PROGRAMMATION EN MODE RELATIF

"VENTES"	"JANVIER"	"FEVRIER"	"MARS"	"AVRIL"
"VENDEUR1"	56	45	45	63
"VENDEUR2"	12	60	78	78
"VENDEUR3"	89	45	95	102
"VENDEUR4"	41	61	56	80
"TOTAL"	SOMME (L (-5) C : L (-2) C)	SOMME (L (-5) C : L (-2) C)	SOMME (L (-5) C : L (-2) C)	SOMME (L (-5) C : L (-2) C)
"-----"	"-----"	"-----"	"-----"	"-----"
"CHIFFRE"				
"VENDEUR1"	L (-11) C * 1000	L (-11) C * 1000	L (-11) C * 1000	L (-11) C * 1000
"VENDEUR2"	L (-11) C * 1000	L (-11) C * 1000	L (-11) C * 1000	L (-11) C * 1000
"VENDEUR3"	L (-11) C * 1000	L (-11) C * 1000	L (-11) C * 1000	L (-11) C * 1000
"VENDEUR4"	L (-11) C * 1000	L (-11) C * 1000	L (-11) C * 1000	L (-11) C * 1000
"TOTAL"	SOMME (L (-5) C : L (-2) C)	SOMME (L (-5) C : L (-2) C)	SOMME (L (-5) C : L (-2) C)	SOMME (L (-5) C : L (-2) C)
"-----"	"-----"	"-----"	"-----"	"-----"
"CHARGES"				
"SALAIRES"	15000*4	15000*4	15000*4	15000*4
"FRAIS"	10000	LC (-1) * 1,1	LC (-1) * 1,1	LC (-1) * 1,1
"TOTAL"	SOMME (L (-3) C : L (-2) C)	SOMME (L (-3) C : L (-2) C)	SOMME (L (-3) C : L (-2) C)	SOMME (L (-3) C : L (-2) C)
"-----"	"-----"	"-----"	"-----"	"-----"
"ACHATS"				
	L (-15) C * 0,7	L (-15) C * 0,7	L (-15) C * 0,7	L (-15) C * 0,7
"-----"	"-----"	"-----"	"-----"	"-----"
"RESULTAT"				
	L (-21) C - (L (-12) C + L (-6) C)	L (-21) C - (L (-12) C + L (-6) C)	L (-21) C - (L (-12) C + L (-6) C)	L (-21) C - (L (-12) C + L (-6) C)

ABONNEZ-VOUS A

hors série
LED
MICRO

Je désire m'abonner à **LED-MICRO** France : 160 F - Etranger* : 240 F.

NOM.....

PRENOM.....

N°..... RUE.....

CODE POSTAL..... VILLE.....

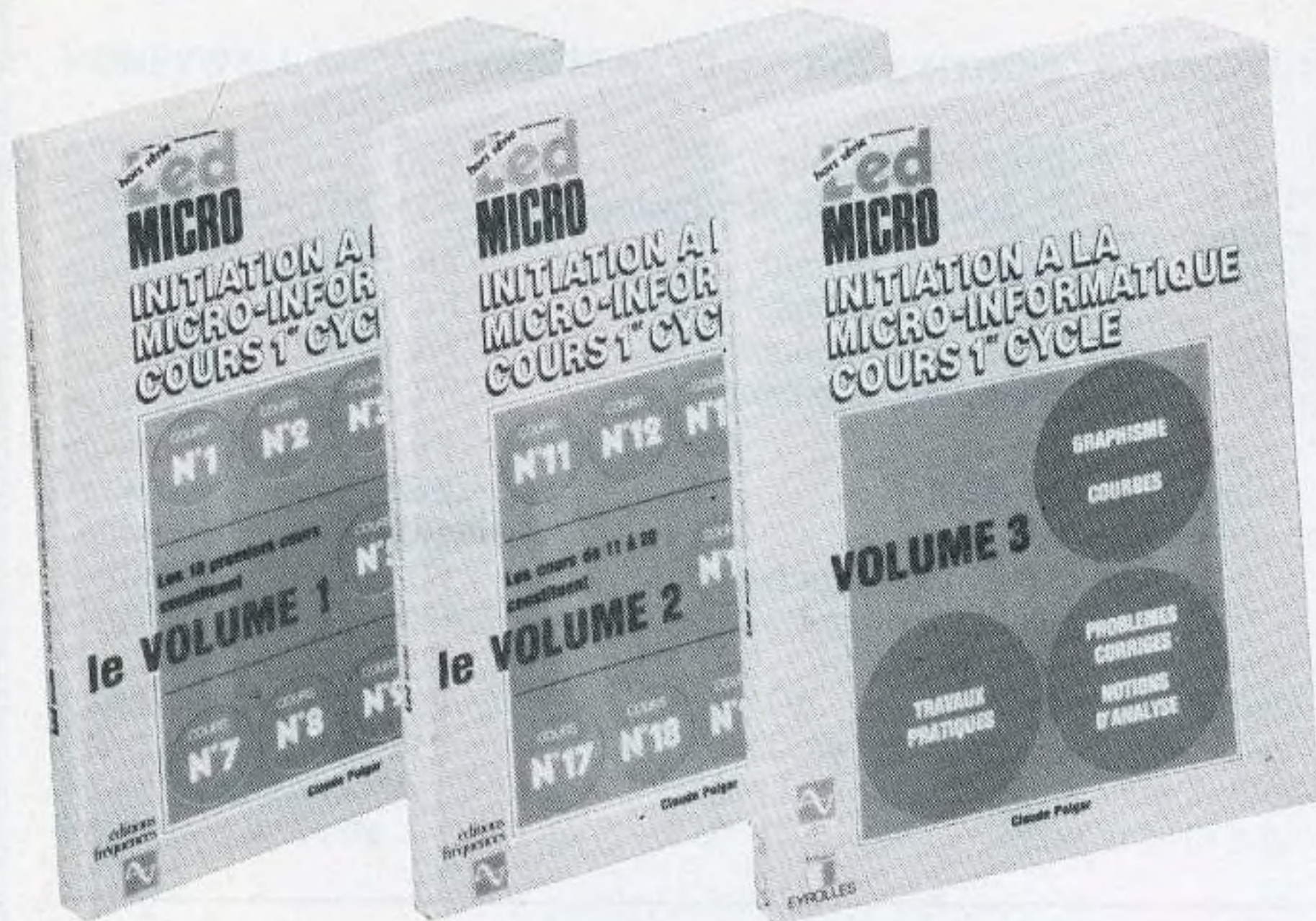
* Pour les expéditions « par avion » à l'étranger, ajoutez 60 F au montant de votre abonnement.

Ci-joint mon règlement par : chèque bancaire C.C.P. Mandat

Le premier numéro que je désire recevoir est : N°.....

EDITIONS FRÉQUENCES 1, boulevard Ney 75018 PARIS - Tél. : 46.07.01.97





**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 !).

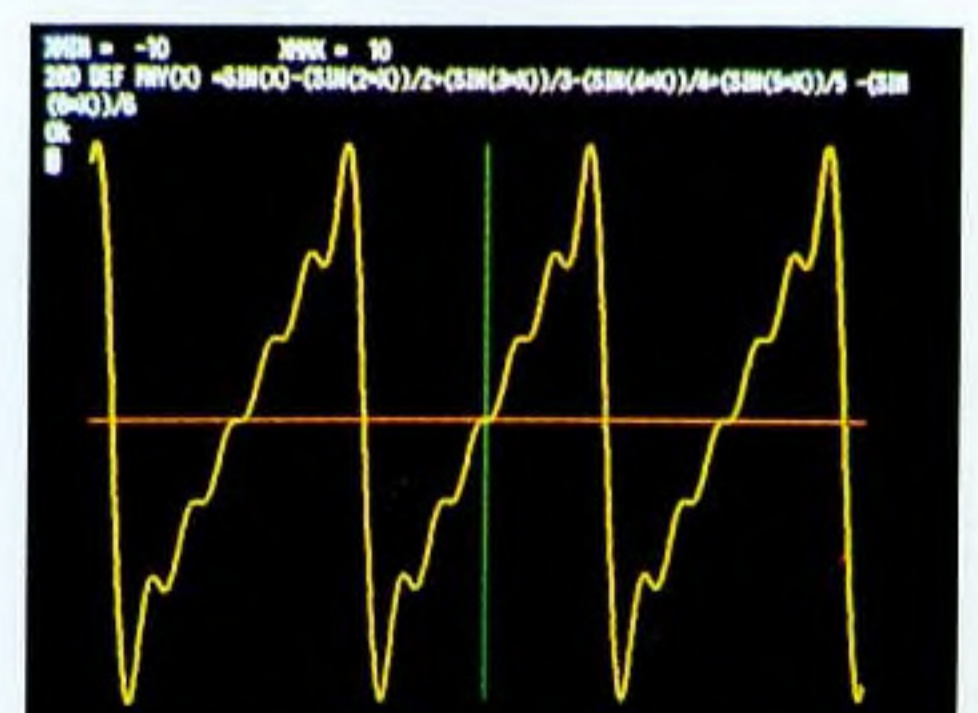
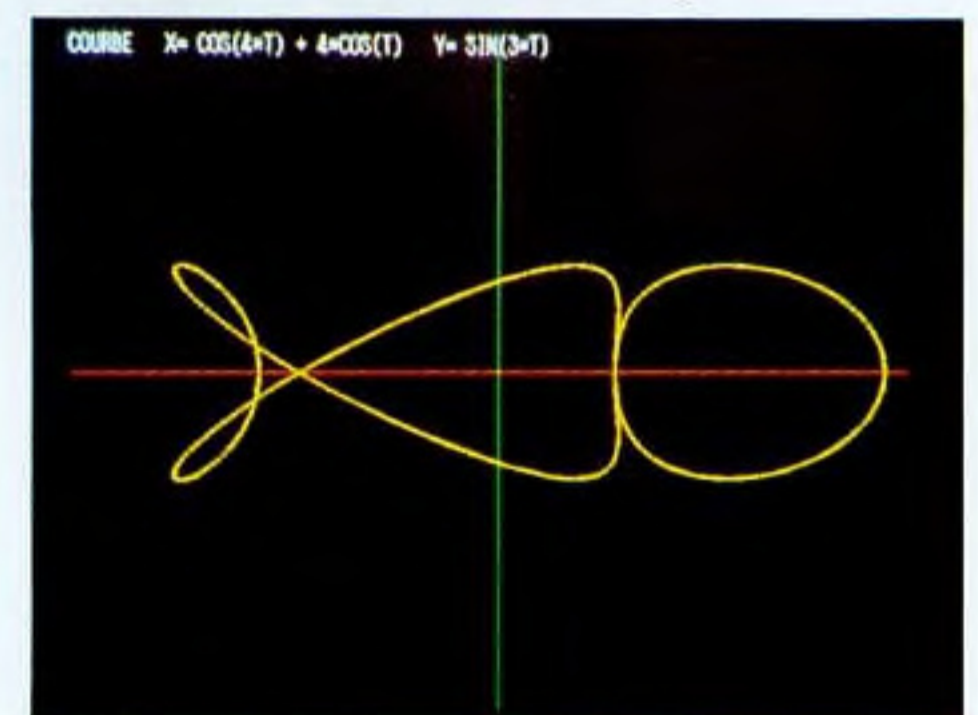
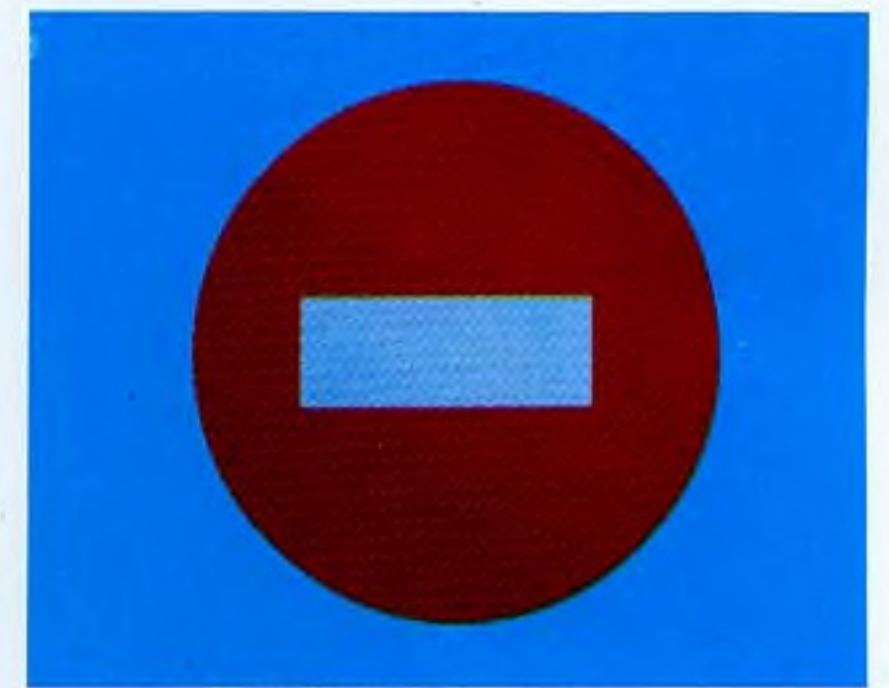
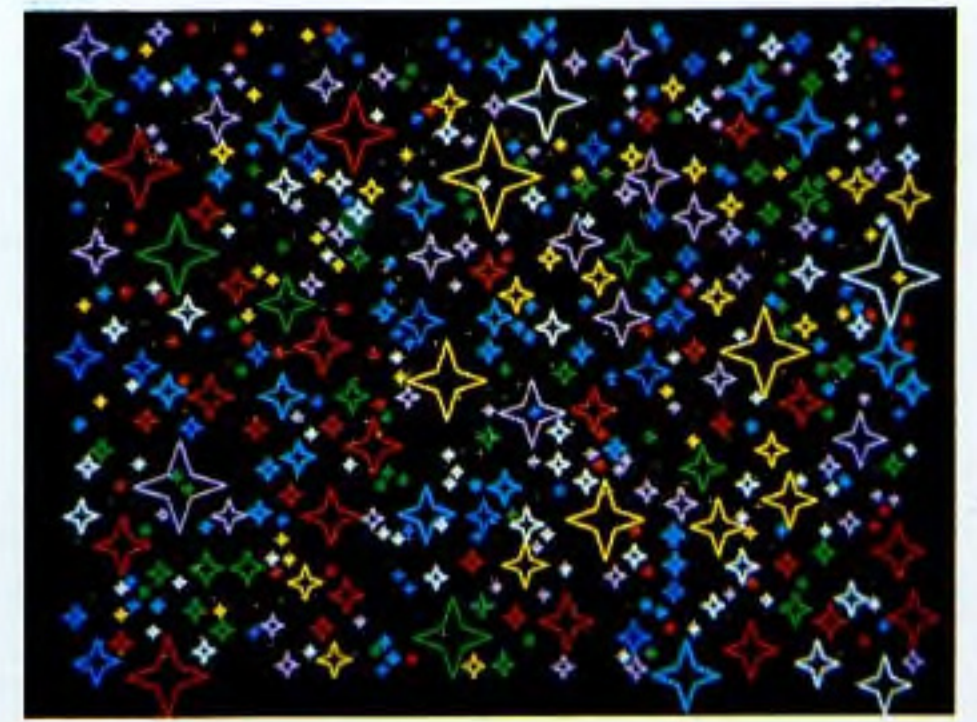
3.21. Techniques de mise au point

- ★ 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