

LOISIRS TECHNIQUES D'AUJOURD'HUI

hors série

Leed

MICRO

PROGRAMMATION

COURS 2^{ème} CYCLE

COURS

N°35

Suite
2^e cycle

N°15

COURS DE
PASCAL
les Record

COURS DE
PROGRAM-
MATION
APPROFONDIE :
zones graphiques

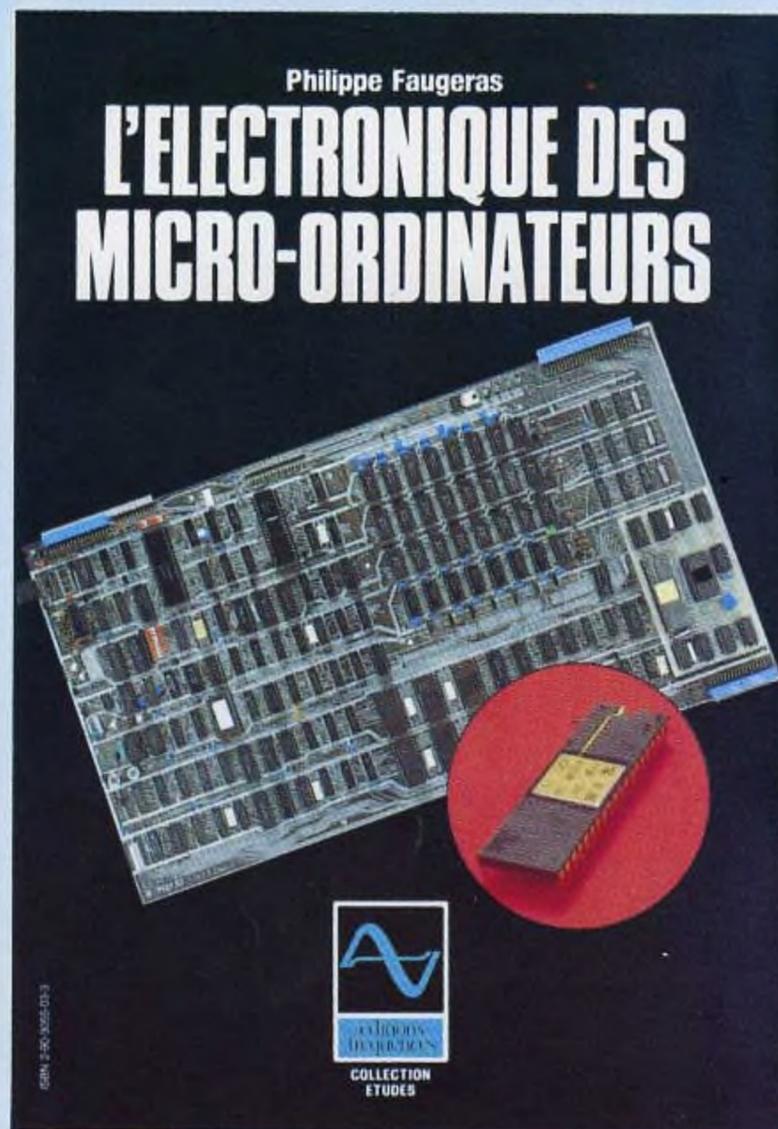
COURS
D'INITIATION AU
PROGICIEL
MULTIPLAN



ISSN 0757-6889

VOYAGE AU COEUR DES MICRO-ORDINATEURS

dans la
COLLECTION
«ETUDES»
aux
éditions
fréquences



une véritable schémathèque

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

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

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

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

En vente chez votre libraire et aux Editions Fréquences

BON DE COMMANDE

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

Nom

Adresse

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

Règlement ci-joint :

Par chèque bancaire par chèque postal par mandat

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

LO... MOUES D'AUJOURD'HUI

hors série

LED

MICRO

PROGRAMMATION COURS 2^e CYCLE

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

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

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

**Services Rédaction-Publicité-
 Abonnements :**

1, bd Ney, 75018 Paris
 Tél. : (1) 46.07.01.97
 Lignes groupées

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

Secrétaire de Rédaction
 Chantal Cauchois

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

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

Réalisation
 Composition-Photogravure
 Ed' Systèmes
 Impression
 Berger-Levrault - Nancy



DECEMBRE 86

COURS DE PASCAL de la page 4 à la page 17

- Les structures de base p. 5
 - Résumé du cours précédent
 - Les Record
 - Introduction
 - Comment cela peut-il bien marcher ?
 - Utilisation dans un programme
 - Exercices pour le mois prochain
- Corrigé de certains exercices ... p. 8

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

NOTRE COUVERTURE : Il y a longtemps que la bureautique est à la mode, mais aujourd'hui les applications de gestion de production arrivent sur micro-ordinateur. Avec l'avènement des machines équipées du nouveau 80386 de chez Intel, il n'y a vraiment plus beaucoup de différence entre un mini et un micro !...

COURS DE PROGRAMMATION APPROFONDIE

de la page 18 à la page 25

- Environnement
 de développement p. 19
 - Caractéristiques de l'écran
 - Utilitaires disponibles
 - Fenêtres et clipping

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

DIALOGUE AVEC NOS LECTEURS de la page 26 à la page 30

C'EST ARRIVÉ DEMAIN
 de la page 34 à la page 36

COURS D'INITIATION AU PROGICIEL MULTIPLAN de la page 37 à la page 49

- La commande fenêtre
 partage désignation p. 38
- La commande fenêtre couplage p. 39
- La commande fenêtre encadré p. 40
- La commande
 protège expression p. 41
- La commande sortie p. 42
- Liaison entre feuilles de calcul . p. 42
- La commande lit-écran
 efface-écran p. 43
- Construction d'une
 feuille de détails p. 43
- Comment passer les données
 sur la feuille récapitulative p. 44
- Corrigé d'exercices p. 47

Charles-Henry Delaleu

COURS DE PASCAL

Dominique Chastagnier
Jean-François Coblentz
Patrick Gueneau

Ce cours est le dernier volet de la série sur les structures de base en Pascal et présente la notion de «Record». Plutôt que d'aborder d'autres aspects de ce riche langage, nous avons préféré revenir sur les exercices précédents, et donc effectuer un tour d'horizon des notions acquises. Enfin, bien entendu, nous vous proposons, comme chaque mois, d'autres exercices.

COURS N° 6

PLAN DU COURS

1. Les structures de base
 - 1.1. Résumé du cours précédent
 - 1.2. Les Record
 - 1.2.1. Introduction
 - 1.2.2. Comment cela peut-il bien marcher ?
 - 1.2.3. Utilisation dans un programme
 - 1.3. Exercices pour le mois prochain
2. Corrigé d'exercices

1. LES STRUCTURES DE BASES (FIN)

1.1. Résumé du cours précédent

Lors du précédent cours, nous vous avons présenté un complément sur les types généraux de données, puis la structure de tableaux. Enfin, les ensembles ont été traités.

Pour bien comprendre toutes ces notions, il nous a semblé préférable de revenir plus longuement sur les exercices. En conséquence, ce mois-ci nous traiterons le corrigé de quelques-uns des exercices du cours précédent.

Mais avant tout, voyons un peu la notion de Record.

1.2. Les Record

1.2.1. Introduction

Vous le savez, il n'est pas possible, jusqu'à présent, de jongler avec des types différents dans un même tableau. Un tableau peut accueillir des données d'un seul type. Il est pourtant commode de pouvoir stocker simultanément des mots, des nombres, des booléens se rattachant aux mots... En BASIC, le standard ne le permet pas, mais tout au long du cours de Programmation Approfondie, nous avons montré que rien n'est impossible.

En Pascal, une merveille permet de faire la même chose, mais elle est définie en standard. Cette «chose» est un RECORD ou enregistrement, en français.

1.2.2. Comment cela peut-il bien marcher ?

Tout simplement, vous donnez un nom à une structure complexe qui peut contenir tout ce que vous voulez. Par exemple, si vous stockez, pour vous en souvenir, le nombre de jours de la semaine et leurs noms (peut-être pouvez-vous vous en souvenir sans cela), il est possible de définir :

```
type
    descr_jours = record
        nb_jours : integer;

        nom_jours : array [1..7] of string;
    end;
```

Un autre, tout aussi trivial qui permet de définir un point par ses coordonnées :

```
type
    point = record
        abscisse : real;
        ordonnée : real;
    end;
```

Enfin, un enregistrement typique des bases de données les plus simples :

```

type
  Info_personne = record
    nom:string;
    prenom : string;
    age : integer;
    adresse : string;
    marie : boolean;
    enfants : integer;
  end;

```

On le voit, un enregistrement est défini comme un type, un peu particulier, en ce sens que ce type n'est pas simple, mais composé. Chaque composante est appelée un champ. Chaque champ peut avoir n'importe quelle structure déjà définie. Exemple, pour reprendre celui qui vient d'être défini :

```

Type
  employé = record
    inf_personnelle : info_personne;
    emploi : string;
    emploi_précédent : string;
  end;

```

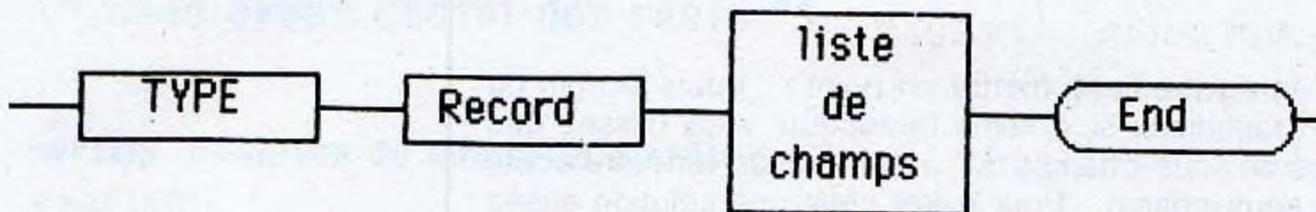
On voit donc qu'il est possible de compliquer les champs, avec des sous-champs, etc. Il faut néanmoins rester clair. Il est aussi possible de définir des champs avec des alternatives, par exemple :

```

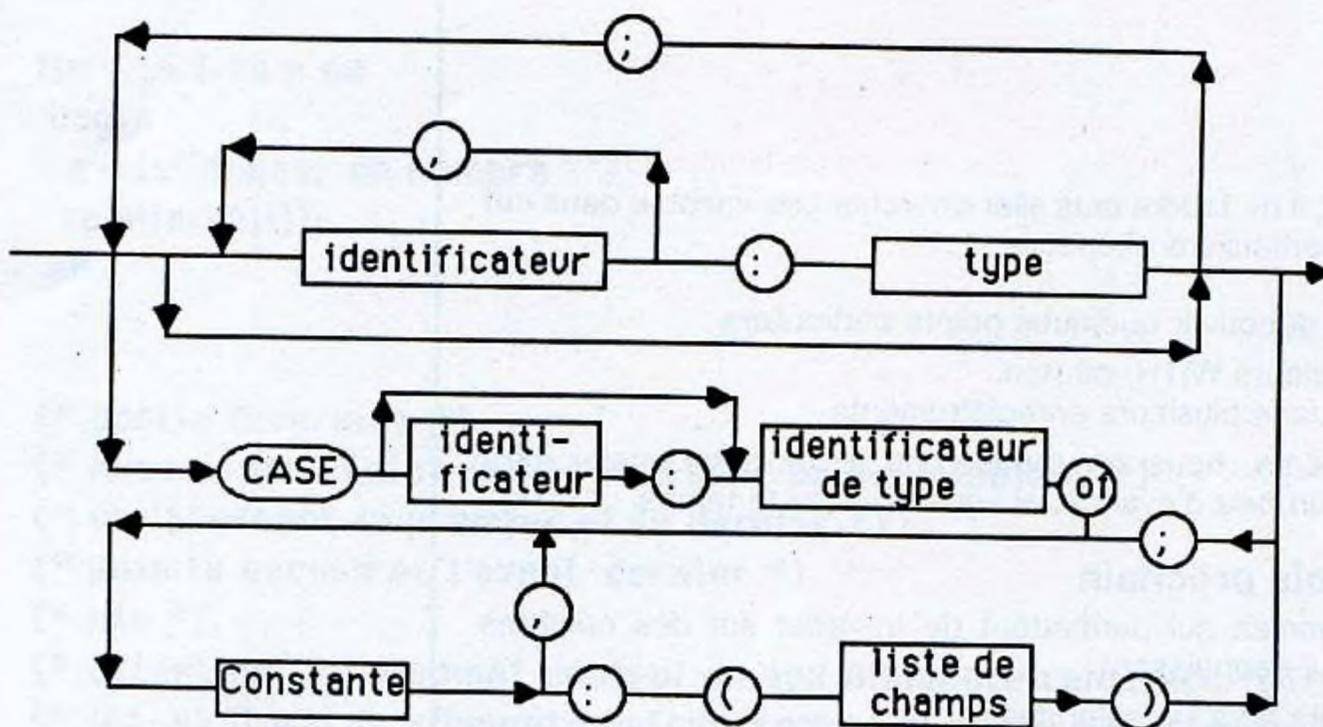
type
  Fig_geo = record
    x,y : real;
    case f : (cercle : carré) of
    cercle : (diamètre : real);
    carré : (z,t : real);
  end;

```

Ici, bien sûr, les différentes figures auront été définies auparavant. Les champs dont la nature varie en fonction de critères programmés et non fixes, sont appelés variants. Voyons maintenant le diagramme syntaxique d'un enregistrement :



ou la liste de champs à la structure suivante :



Remarquez la présence d'une «liste de champs», à l'intérieur d'une liste de champs. Il y a là une imbrication fort intéressante.

1.2.3. Utilisation dans un programme

Il est clair que l'utilisation d'un enregistrement dans un programme doit être précisée, car il faut savoir de quelle structure de l'enregistrement on parle à chaque instant. Pour cela, la notation de base est la suivante, en reprenant le record déjà vu :

```

type
  point = record
    abscisse : real;
    ordonnée : real;
end;
  
```

On utilisera les deux champs par un nom composé. Par exemple :

```
point.abscisse := 12;
```

Il faut donc rappeler le nom de l'enregistrement, mettre un point (.) puis le nom du champ. Cela peut devenir long et fastidieux si, comme beaucoup, vous utilisez des noms longs, ou des champs formés de sous-champs qui, à leur tour, doivent être écrits avec le nom, le point, le nom du sous-champ... Pour éviter cela, une solution assez élégante, le AVEC. En effet, il suffit de prévenir le programme que l'on utilise un certain enregistrement pour que, par la suite, il suffise d'appeler les champs uniquement, sans rappeler le nom de l'enregistrement. Par exemple, en reprenant toujours le même enregistrement :

```
with point do  
  abscisse := 12;  
  ordonnée := 0;  
end;
```

Le END prévient que dorénavant, il ne faudra plus aller chercher une variable dans cet enregistrement, sauf précision particulière ultérieure.

Nous vous laissons le plaisir de découvrir quelques points particuliers :

- Possibilité d'imbriquer plusieurs WITH, ou non.
- Noms de champs égaux dans plusieurs enregistrements.

Il y a d'autres points intéressants, nous en reparlerons le mois prochain, mais rapidement. A vous de prendre un peu d'avance, si vous en avez le temps.

1.3. Exercices pour le mois prochain

- 1.3.1. Ecrire des petits programmes qui permettent de travailler sur des nombres complexes, pour ceux qui connaissent.
- 1.3.2. Créer des enregistrements pour un calendrier.
- 1.3.3. Créer des enregistrements pour un cahier de notes, qui pourront être intégrés au programme demandé le mois dernier.
- 1.3.4. Ecrire des en-têtes de programmes utilisant la structure de Record emboîtés, ou avec des conditions.

2. CORRIGE DE CERTAINS EXERCICES

2.1. Exercice d'inversion de données, numéro 4 du cours précédent.
Voici une solution, simple mais qui ne demande qu'à grandir.

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

```
  const
```

```
    max = 1000;
```

```
  var
```

```
    nb : array[1..max] of real;
```

```
    i, n : integer;
```

```
    temp : real;
```

```
begin
```

```
(* Partie de lecture des donnees *)
```

```
(* nous avons choisi des reels *)
```

```
write(' combien de nombres a entrer : ');
```

```
readln(n);
```

```
if n > max then
```

```
begin
```

```
  writeln(' n est trop grand, et le programme le limite a max');
```

```
  n := max;
```

```
end;
```

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

```
begin
```

```
  write('donner un nombre : ');
```

```
  readln(nb[i]);
```

```
end;
```

```
(* partie inversion *)
```

```
(* nous avons choisi de le faire avec un seul tableau *)
```

```
(* en inversant le premier et le dernier *)
```

```
(* puis le second et l'avant-dernier *)
```

```
(* etc *)
```

```
(* attention a l'element central ou aux elements centraux *)
```

```
(* ici, si il y a un element central, comme il reste a sa place *)
```

```
(* rien n'est fait dessus *)
```

```
for i := 1 to trunc(n / 2) do
```

```
begin
```

```
  temp := nb[i];
```

```
  nb[i] := nb[n + 1 - i];
```

```
  nb[n + 1 - i] := temp;
```

```
end;
```

```
(* impression de la modification *)
```

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

```
  writeln(' element ', i, ' : ', nb[i]);
```

```
end.
```

Description du programme :

Pour tous les exercices de ce cours, les sous-programmes n'étant pas encore décrits, il a fallu programmer de manière la plus linéaire possible. Cela se voit bien sur la plupart, où le Pascal n'est pas employé dans son acception la plus intéressante.

Un premier bloc sert à lire les données, le second à les inverser, le troisième à imprimer le résultat.

Chaque bloc est court et ne doit pas poser de problème. Vous pouvez vérifier à la main que l'inversion fonctionne bien, elle est due au fait que si la liste a un nombre impair de données, la donnée centrale ne bouge pas.

Exemples d'exécution :

Nous en avons mis deux, un pour un nombre pair de données, le second pour un nombre impair, pour persuader les derniers sceptiques que cela marche.

Combien de nombres à entrer : 9

donner un nombre : 1
 donner un nombre : 2
 donner un nombre : 3
 donner un nombre : 4
 donner un nombre : 5
 donner un nombre : 6
 donner un nombre : 7
 donner un nombre : 8
 donner un nombre : 9

élément 1 : 9.0e + 0
 élément 2 : 8.0e + 0
 élément 3 : 7.0e + 0
 élément 4 : 6.0e + 0
 élément 5 : 5.0e + 0
 élément 6 : 4.0e + 0
 élément 7 : 3.0e + 0
 élément 8 : 2.0e + 0
 élément 9 : 1.0e + 0

Combien de nombres à entrer : 10

Donner un nombre : 0

donner un nombre : 1
 donner un nombre : 2
 donner un nombre : 3
 donner un nombre : 4
 donner un nombre : 5
 donner un nombre : 6
 donner un nombre : 7
 donner un nombre : 8
 donner un nombre : 9
 donner un nombre : 10

élément 1 : 9.0e + 0
 élément 2 : 8.0e + 0
 élément 3 : 7.0e + 0
 élément 4 : 6.0e + 0
 élément 5 : 5.0e + 0
 élément 6 : 4.0e + 0
 élément 7 : 3.0e + 0
 élément 8 : 2.0e + 0
 élément 9 : 1.0e + 0
 élément 10 : 0.0e + 0

2.2. Exercices 5 et 6

Ces exercices avaient pour but de créer un programme permettant de calculer la moyenne d'élèves sur plusieurs mois, de les classer, par devoirs, et sur le classement final. Voici un programme qui fait cela, proposé par M. P.G. de Puteaux. Il ne fait que le minimum et, en particulier, ne fait pas de stockage, ce qui n'était pas demandé. Enfin, la structuration s'améliorera lorsque nous aurons défini les sous-programmes.

program etudiants (input, output);

const

```
nb_etud = 10;
note_max = 20;
note_min = 0;
nb_result = 20;
long_nom = 30;
```

type

```
nom = string[30];
tab_note = array[1..nb_etud, 1..nb_result] of real;
tab_nom = array[1..nb_etud] of nom;
tab_moy = array[1..nb_etud] of real;
```

var

```
t_moy : tab_moy;
fin : boolean;
etud, n_note : integer;
i, j, k : integer;
t_note : tab_note;
t_nom : tab_nom;
moyenne_generale_classe, moy, moy_max : double;
n_et : string[30];
```

begin

(* saisie des noms *)

```
Write(' Nombre total des etudiants : ');
readln(etud);
if etud > nb_etud then
begin
write(' mauvais dimensionnement des parametres ');
readln;
end
```

```

else
  for i := 1 to etud do
    begin
      writeln;
      write(' nom de l etudiant : ');
      readln(t_nom[i]);
    end;

(* saisie des notes *)
writeln;

writeln;
writeln(' saisie des notes ');
write(' nombre de notes : ');
readln(n_note);
writeln;
for i := 1 to etud do
  begin
    writeln(' pour l etudiant ', t_nom[i]);
    for j := 1 to n_note do
      begin
        write(' saisie de la note', j, ' : ');
        readln(t_note[i, j]);
      end;
    end;

(* resultats *)
writeln;
writeln;
for i := 1 to etud do
  begin
    for j := 1 to n_note do
      begin
        moyenne_generale_classe := moyenne_generale_classe +
t_note[i, j];
      end;
    end;
    moyenne_generale_classe := moyenne_generale_classe / (etud *
n_note);
    writeln('moyenne generale de la classe :
moyenne_generale_classe);

(* pour un etudiant *)
repeat

```

```

begin
  moy := 0;
  writeln;
  writeln;
  write('nom de l etudiant : ');
  readln(n_et);
  for i := 1 to etud do
    if n_et = t_nom[i] then
      j := i;
  for i := 1 to n_note do
    moy := moy + t_note[j, i];
  moy := moy / n_note;
  writeln(' moyenne de ', n_et, ' = ', moy);
  writeln;
  writeln(' fin du traitement : ');
  write('          oui / non : ');
  readln(n_et);
  if n_et = 'oui' then
    fin := true;
end;
until fin = true;

(* pour une interrogation *)
repeat
  begin
    fin := false;
    moy := 0;
    writeln;
    writeln;
    write('indice de l interrogation : ');
    readln(i);
    for j := 1 to etud do
      moy := moy + t_note[j, i];
    moy := moy / etud;
    writeln(' moyenne de l interrogation ', i, ' = ', moy);
    writeln;
    writeln(' fin du traitement : ');
    write('          oui / non : ');
    readln(n_et);
  end;
until fin = true;

```

```

if n_et = 'oui' then
  fin := true;
end;
until fin = true;
(* tri par moyenne des etudiants *)
writeln;
for i := 1 to etud do
  begin
    t_moy[i] := 0;
  end;
for i := 1 to etud do
  begin
    for j := 1 to n_note do
      begin
        t_moy[i] := t_moy[i] + t_note[i, j];
      end;
    t_moy[i] := t_moy[i] / n_note;
  end;
for i := 1 to etud do
  begin
    moy_max := 0;
    for j := 1 to etud do
      if t_moy[j] > moy_max then
        begin
          moy_max := t_moy[j];
          k := j;
        end;
    t_moy[k] := -1;
    writeln(t_nom[k], ' moyenne : ', moy_max);
  end;
end.

```

Commentaires :

- Le programme commence par demander un certain nombre de paramètres, nombre d'étudiants, de notes, les noms et les notes. Ceci représente trois blocs d'instructions bien séparés, car il est ainsi plus aisé de les reprendre par la suite, afin d'améliorer le programme.
- Il est fait un calcul sur les moyennes, sur la moyenne de la classe pour un devoir, enfin un classement général.
- Nous verrons très bientôt comment améliorer facilement un tel programme, mais avant tout, bravo à M. P.G.

2.3. Exercice 7

Il s'agissait de changer de format de dates. Depuis un format de type xx/yy/zz, donner la date xx mois 19zz. Voici une solution :

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

```
const
```

```
  nb_mois = 12;
```

```
type
```

```
  t_mois = array[1..nb_mois] of string;
```

```
  lg_mois = array[1..nb_mois] of integer;
```

```
var
```

```
  tab_mois : t_mois;
```

```
  long_mois : lg_mois;
```

```
  jour, annee, j, m, a : integer;
```

```
  mois : string;
```

```
begin
```

```
  tab_mois[1] := 'Janvier';
```

```
  tab_mois[2] := 'Fevrier';
```

```
  tab_mois[3] := 'Mars';
```

```
  tab_mois[4] := 'Avril';
```

```
  tab_mois[5] := 'Mai';
```

```
  tab_mois[6] := 'Juin';
```

```
  tab_mois[7] := 'Juillet';
```

```
  tab_mois[8] := 'Aout';
```

```
  tab_mois[9] := 'Septembre';
```

```
  tab_mois[10] := 'Octobre';
```

```
  tab_mois[11] := 'Novembre';
```

```
  tab_mois[12] := 'Decembre';
```

```
  long_mois[1] := 31;
```

```
  long_mois[2] := 28;
```

```
  long_mois[3] := 31;
```

```
  long_mois[4] := 30;
```

```
  long_mois[5] := 31;
```

```
  long_mois[6] := 30;
```

```
  long_mois[7] := 31;
```

```
  long_mois[8] := 31;
```

```

long_mois[9] := 30;
long_mois[10] := 31;
long_mois[11] := 30;
long_mois[12] := 31;

write('date à transformer : ');
readln(j, m, a);

jour := j;
mois := tab_mois[m];
annee := 1900 + a;

writeln(' date : ', j, ' ', tab_mois[m], ' ', 1900+a);

end.

```

Notons que le type STRING n'est pas standard. Il est possible de le créer par :

```

type
string = packed array [1..max] of char;

```

où max est n'importe quel entier que vous aurez créé auparavant en constante.

Un exemple d'exécution :

```

date à transformer : 17 10 76
17 octobre 1976

```

2.4. Programme de lecture de températures sur une semaine (ou plus)

```

program temp (input, output);

```

```

const
nb_semaines = 1;
nb_jours = 7;
nb_heures = 24;
nb_minutes = 60;

```

```

type
  tab_temp = array[1..nb_jours, 1..nb_heures] of real;
  tab_jour = array[1..nb_jours] of string;

var
  i, j : integer;
  nb_prises_jours : integer;
  t_temp : tab_temp;
  jour : tab_jour;
  temp : real;

begin

  jour[1] := 'lundi';
  jour[2] := 'mardi';
  jour[3] := 'mercredi';
  jour[4] := 'jeudi';
  jour[5] := 'vendredi';
  jour[6] := 'samedi';
  jour[7] := 'dimanche';

  for j := 1 to nb_jours do
    begin
      for i := 1 to nb_heures do
        begin
          write('temperature relevee le ', jour[j], ' a ', i, ' h : ');
          readln(t_temp[j, i]);
        end;
      end;
    end;

  (* voila le programme pret pour la suite *)

end.

```

Commentaires :

- Nous ne proposons que le début de ce programme qui est prêt pour tous les ajouts possibles pour des statistiques, des courbes, et tout ce que votre imagination vous autorisera.
- Les types sont définis pour la simplicité de la suite du programme. Il est clair que pour si peu, ils auraient pu être évités.

COURS DE PROGRAMMATION APPROFONDIE

Dominique Chastagnier
Jean-François Coblentz
Patrick Gueneau

Nous voici face à une nouvelle «saga» qui va nous emmener cette fois-ci dans les zones graphiques. Comme vous vous en doutez, cela nous prendra plusieurs numéros : ce mois-ci nous passerons en revue l'environnement de développement des programmes de dessin.

Dans un premier temps, nous présenterons les faiblesses des programmes graphiques notamment dans le domaine du rendu du dessin, puis nous étudierons les moyens d'adaptation des outils au dessin que l'on souhaite créer quelle que soit sa taille. Alors, à vos crayons !

COURS N° 15

PLAN DU COURS

1. Environnement de développement
 - 1.1. Caractéristiques de l'écran
 - 1.2. Utilitaires disponibles
 - 1.3. Fenêtres et clipping

1. ENVIRONNEMENT DE DEVELOPPEMENT

1.1. Caractéristiques de l'écran

On appelle résolution graphique le nombre de pixels (c'est dans le nouveau Petit Larousse Illustré) que contient votre écran quand il est employé en mode graphique. On présente la résolution sous la forme du produit de deux nombres entiers (ex. : 1 280 × 1 024) indiquant le nombre de pixels sur une ligne, puis sur une colonne. Il convient ensuite de connaître la taille de l'image, vous vous doutez qu'à résolution égale un écran plus petit donnera une image que l'œil considèrera de meilleure qualité, c'est ce que les photographes appelleraient le « grain » de la photo.

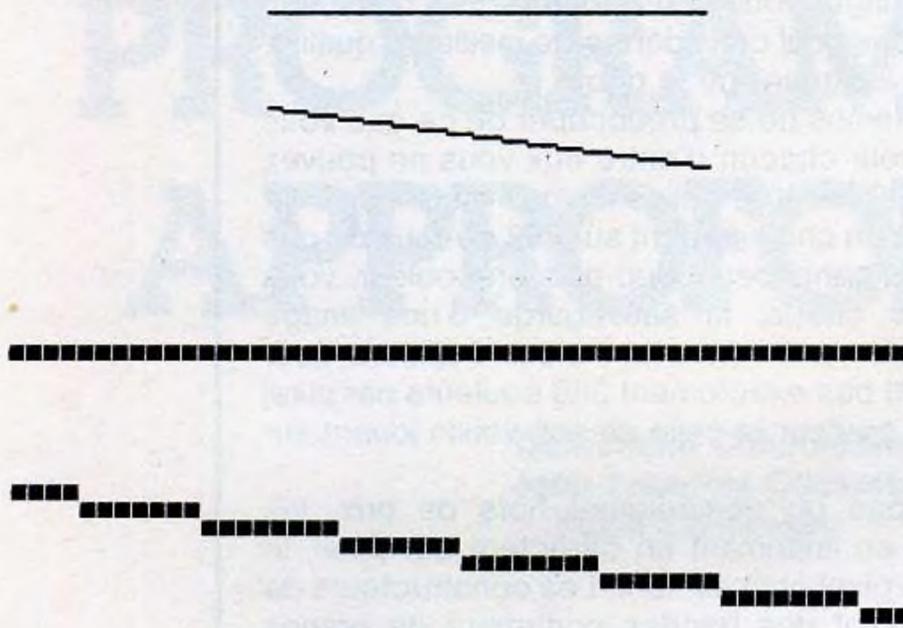
Une fois la taille de votre image établie, il est temps de se préoccuper de ce que vous pouvez faire avec chacun de vos pixels, si pour chacun d'entre eux vous ne pouvez choisir qu'entre noir et blanc, vous n'aurez pas de très belles images mais leur taille mémoire sera faible. Au contraire, si vous avez un choix portant sur 256 niveaux de gris en noir et blanc, ou un choix de 256 couleurs parmi beaucoup plus en couleur, vous employez un octet par pixel, et dans ce cas-là, la sauvegarde d'une image 1 024 × 1 024 vous coûte purement et simplement 1 Moctet : c'est d'ailleurs pour réduire cette taille que les images n'acceptent pas exactement 256 couleurs par pixel mais plutôt une sorte de compromis entre sa couleur et celle de son voisin jouant sur les faiblesses de l'œil humain.

Lorsque les tables traçantes n'existaient pas ou demeuraient hors de prix, les programmeurs remédiaient à cette situation en imprimant un caractère par pixel, le caractère étant d'autant plus important que le pixel était sombre. Les constructeurs de grosses machines envoyaient d'ailleurs souvent des bandes contenant de grands dessins (Marilyn Monroe, Lucky Luke, etc.) comme vœux de Nouvel An.

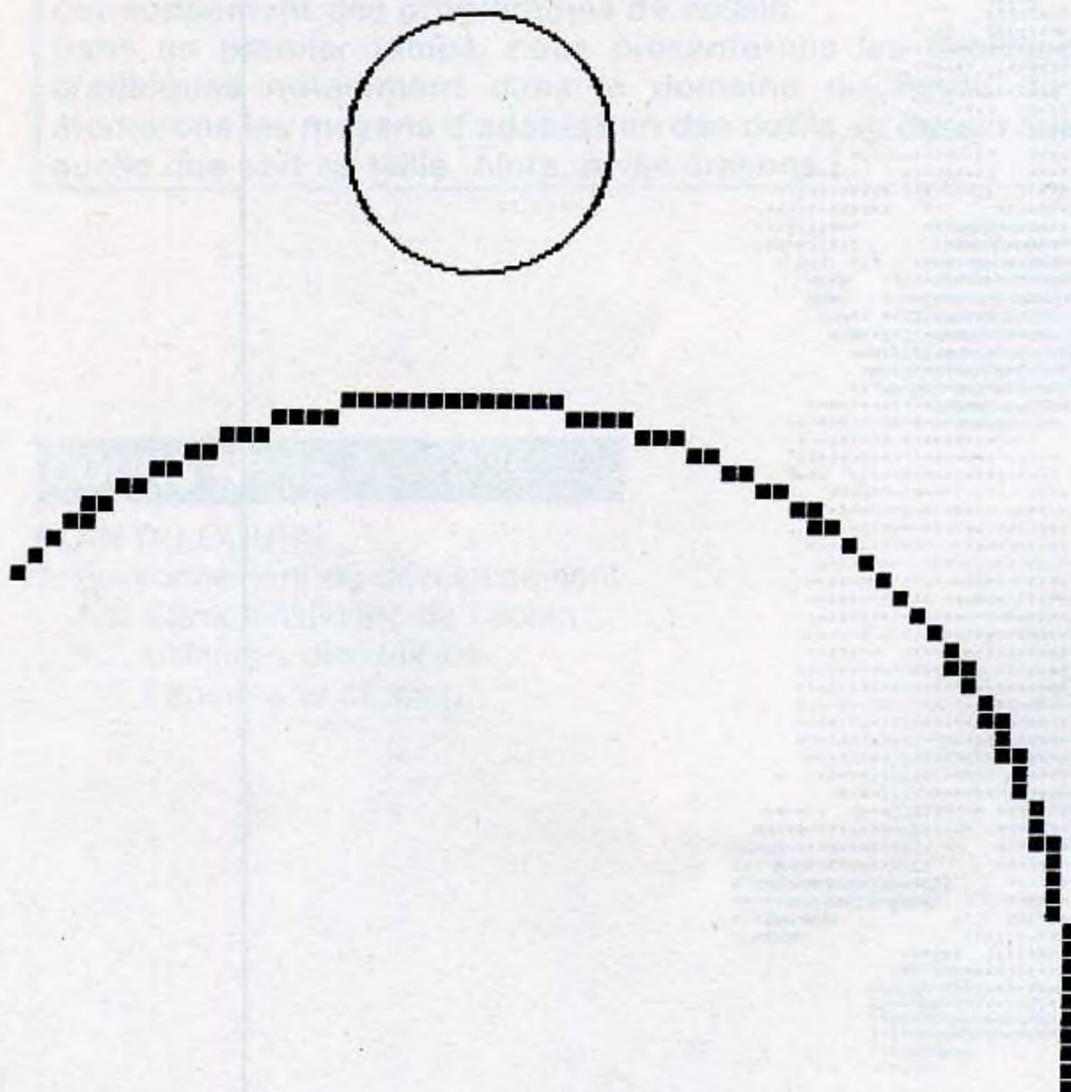


1.2. Utilitaires disponibles

La plupart des ordinateurs du marché disposent d'outils graphiques plus ou moins performants, néanmoins il convient d'apprécier les limites de ces différents produits. Si l'on trace parallèlement à l'un des axes, le produit est de très bonne qualité, en diagonale, il n'en est pas toujours de même et apprécier l'angle d'incidence relève parfois de l'exploit.



Pour les cercles, pour peu qu'ils soient de taille modeste, cela devient rapidement critique.

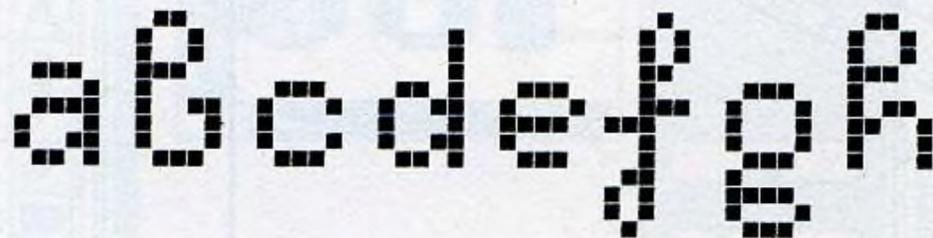
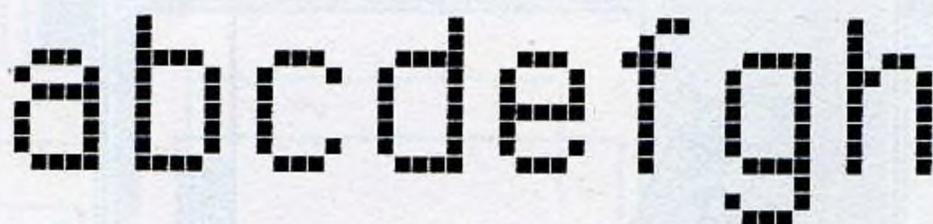


Dans le cas de la construction d'objet, tout comme dans celle de caractères, le grossissement peut à l'inverse permettre de rectifier les erreurs initiales.

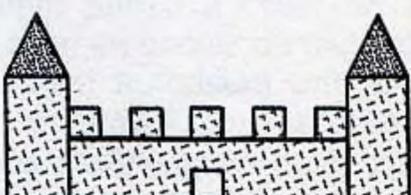
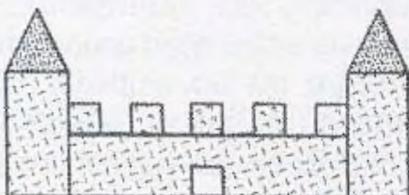
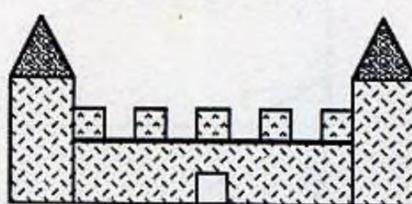
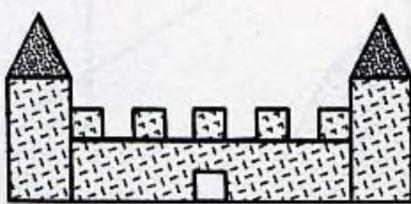
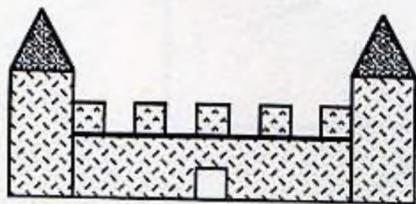


abcdefgh

aBcdefgH

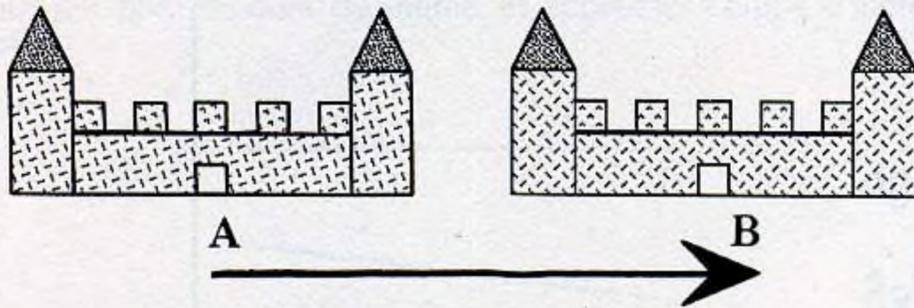


A partir d'une entité (segment, cercle, objet ou caractère) quelconque, il doit être possible de procéder à deux opérations : déplacement et duplication. Opérations somme toute analogues, la seule question étant la persistance ou non du dessin originel. Ces deux transformations requièrent la conservation des caractéristiques propres à chaque entité afin de les déplacer ou de les dupliquer, et s'il est obligatoire de conserver tous les points d'un caractère, la vision précédente des droites et autres cercles laisse supposer qu'une définition paramétrique de cette figure sera largement suffisante.

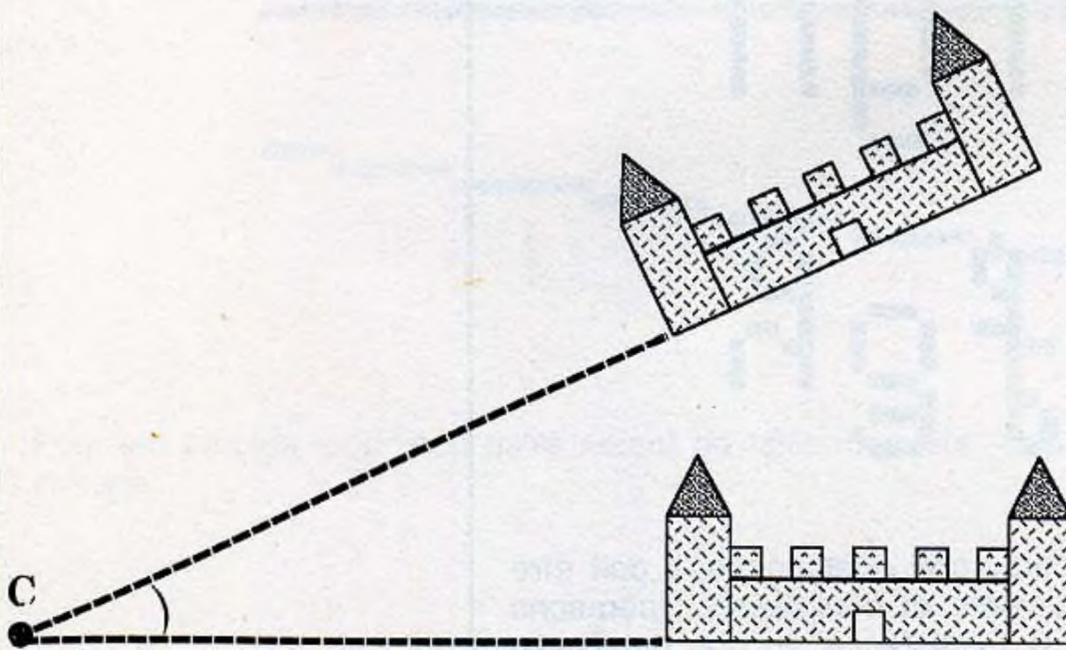


On ne considère comme possibles que les transformations laissant invariantes les caractéristiques intrinsèques de la pièce, elles sont au nombre de trois :

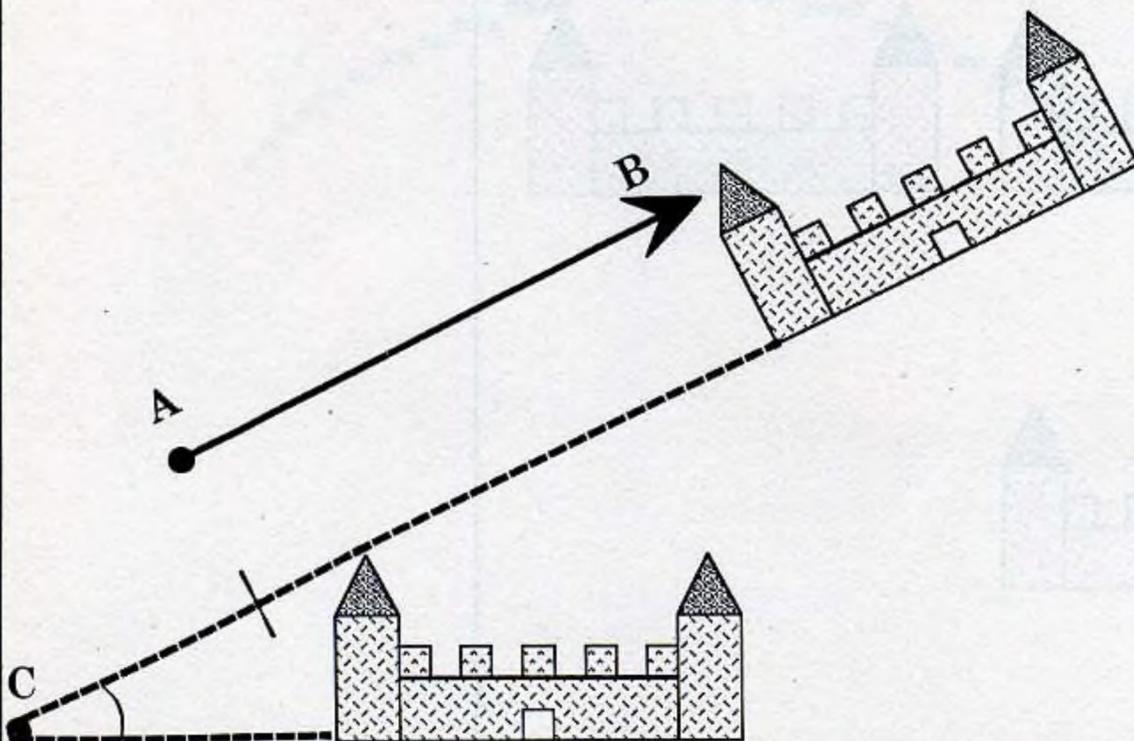
- la translation : on déplace la pièce suivant un vecteur préalablement défini,



- la rotation : on fait tourner la pièce autour d'un point préalablement défini d'un angle également choisi auparavant,

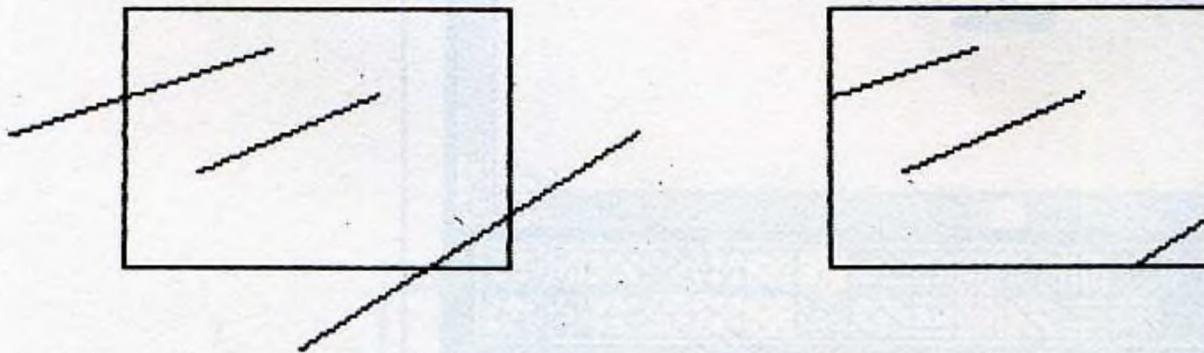
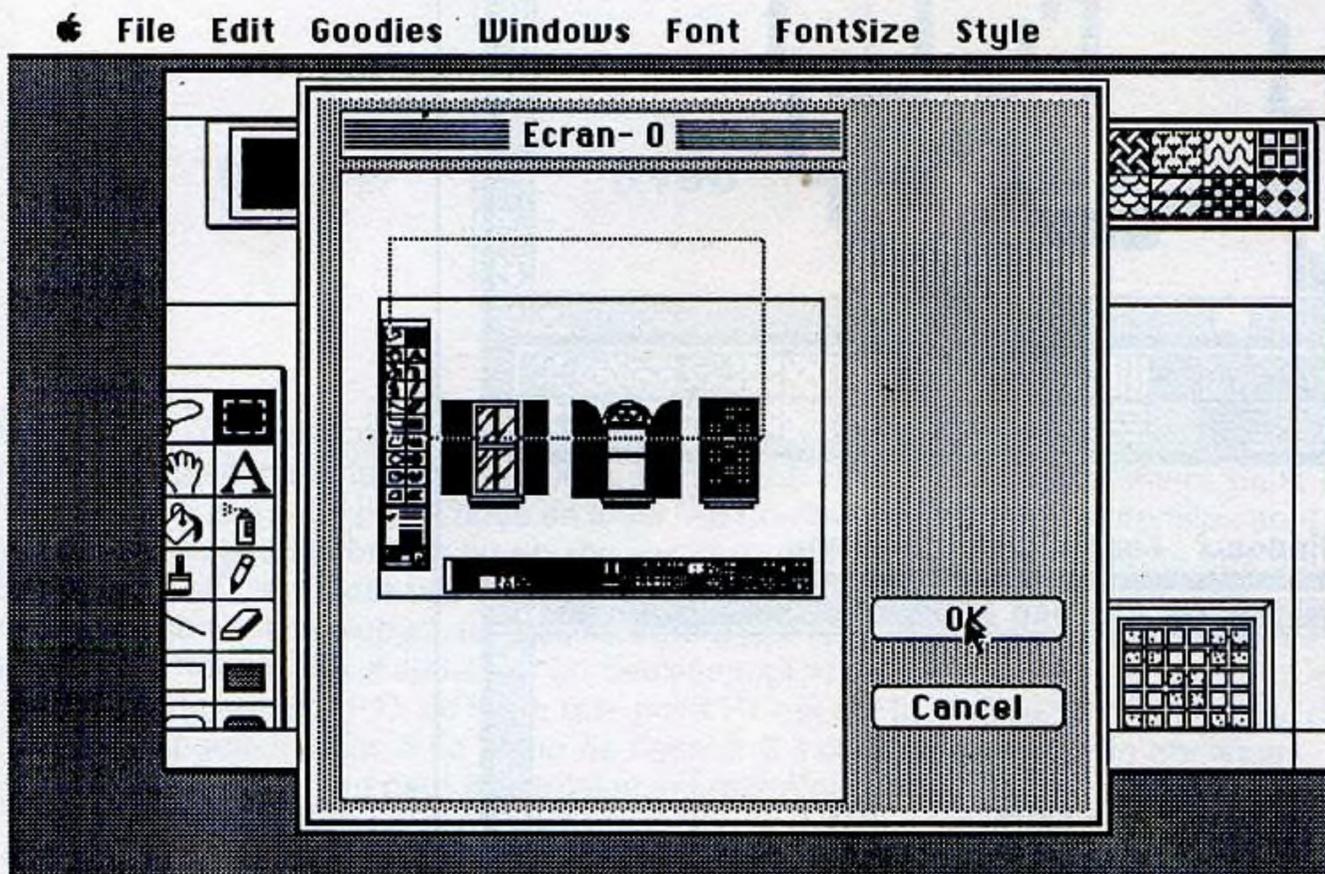


- la rotation-translation : combinaison des deux dans l'ordre indiqué, la translation consécutive ayant pour direction la droite d'arrivée de la rotation passant par l'axe de celle-ci.

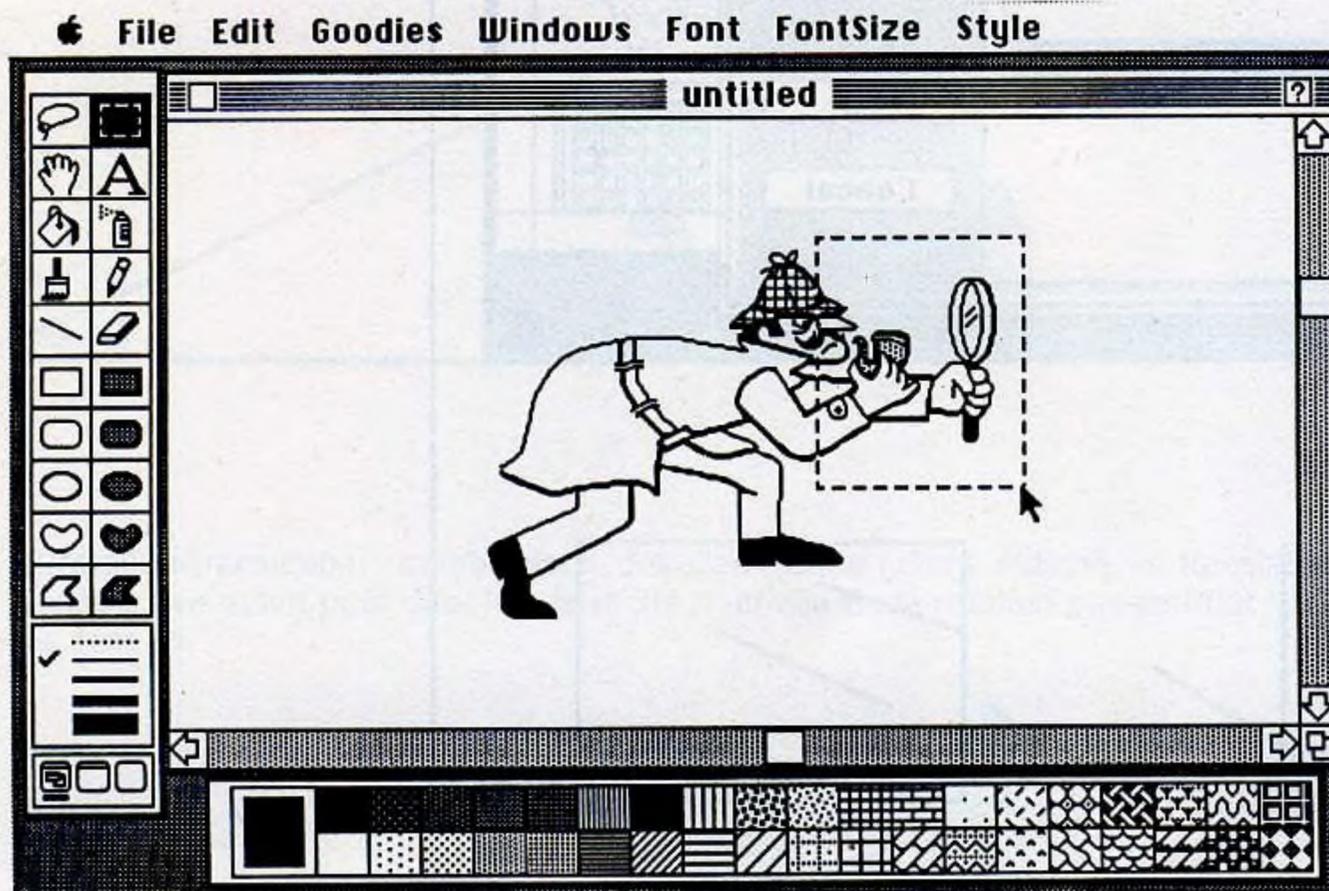
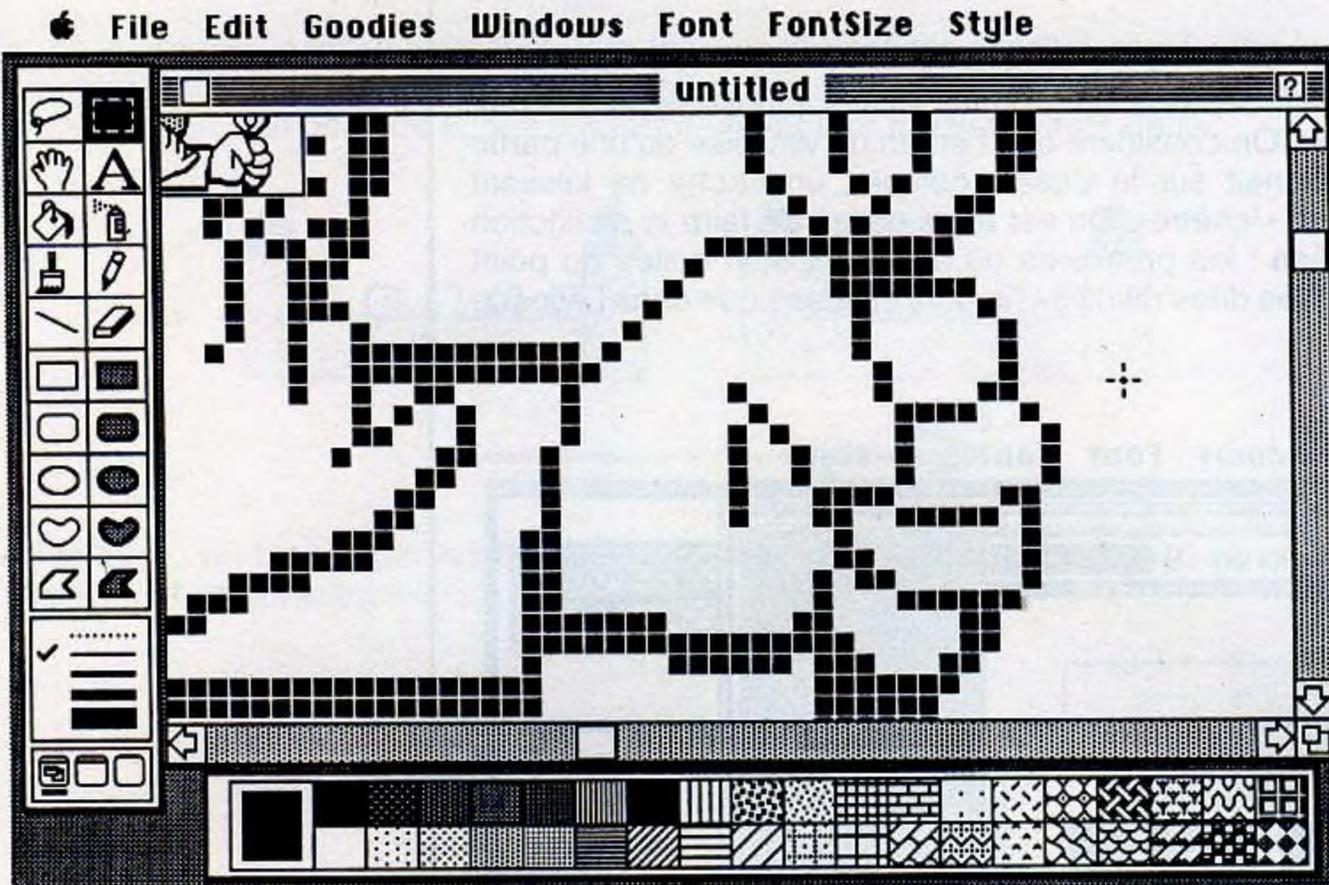


1.3. Fenêtres et Clipping

Il est assez fréquent qu'une image dépasse en taille les limites imposées par l'écran, apparaît alors la notion de fenêtre. On considère que l'écran ne visualise qu'une partie du dessin, comme si l'on promenait sur le dessin complet un cache ne laissant découvert qu'un rectangle appelé «fenêtre». On est alors obligé de faire la distinction entre deux types de coordonnées : les premières ou absolues sont celles du point dans le dessin global ; les secondes dites relatives ne sont valables que dans l'espace de la fenêtre.



Une fois envisagé le fait de ne pas représenter tout le dessin dans l'écran, les concepteurs ont immédiatement pensé à créer un zoom ne respectant plus les mesures originelles du dessin pour en choisir de plus appropriées à telle ou telle partie du croquis ou un agrandissement autorisera une précision fort appréciable. Cela entraîne toutefois un surcroît de travail pour le programme qui se devra, dans un premier temps, de diviser les caractéristiques de ses pixels en un nombre plus grand puis, lors du zoom arrière, de condenser les informations portant sur plusieurs pixels en un seul perdant par là même de l'information. Un des dangers les plus pernicieux est que le simple fait d'un zoom avant suivi d'un zoom arrière n'altère le dessin.



La création de fenêtres a soulevé un problème générateur de nombreux algorithmes : comment savoir si un segment ou une figure est situé dans une fenêtre ou non, en partie ou totalement ; ce problème porte le nom de «clipping». Voici l'algorithme mis au point par Cohen et Sutherland :

– dans un premier temps, on partage l'espace absolu en neuf parties dont la centrale symbolise la fenêtre ; ces neuf régions recevant un code sur quatre bits comme suit :

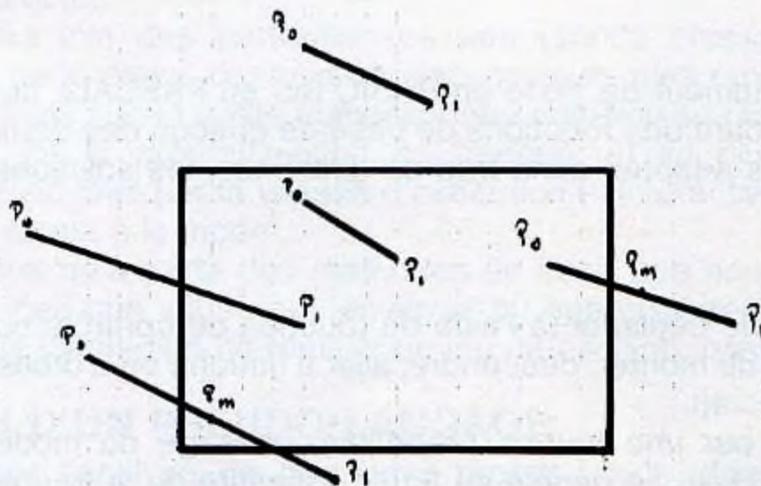
- le premier indique que les points dans cette région sont à gauche de la fenêtre ;
- le second, les points sont à droite ;
- le troisième, les points sont en dessous de la fenêtre ;
- le quatrième, ils sont au-dessus.

1001	1000	1010
0001	0000	0000
0101	0100	0110

Maintenant, considérons le segment que nous souhaitons clipper : si ses deux extrémités sont dans la région 0000, pas de problème, il est intégralement dans la fenêtre et nous n'avons qu'à nous en louer. Au contraire, si les deux extrémités sont à l'extérieur, pratiquons bit par bit un «ou exclusif» entre les codes inhérents aux deux points, si le résultat n'est pas nul, alors le segment est totalement à l'extérieur.

Malheureusement, la plupart des cas ne seront pas résolus à ce stade de l'algorithme, aussi est-il nécessaire d'appliquer un deuxième processus :

- Soit le segment (P0, P1), on teste si le point P1 est sur la fenêtre. Si c'est le cas, on trace le segment jusqu'à sa sortie de l'écran. S'il n'est pas visible, on continue.
- On teste si le segment peut être totalement rejeté par la méthode du «ou exclusif». Si cela ne l'élimine pas, on continue.
- On divise le segment en deux par le point Pm, milieu de (P0, P1). On teste si le segment (Pm, P1) peut être totalement rejeté. Si c'est le cas, on recommence au début avec (P0, P1). Dans l'éventualité contraire, on repasse en 1 avec le segment (Pm, P1) puis (P0, Pm).



DIALOGUE AVEC NOS LECTEURS

REALISATION DE DEUX APPLICATIONS TYPES

INTRODUCTION

Nous allons mettre en pratique les bons conseils prodigués depuis plus d'un an sur les deux exemples prévus, à savoir, le mini-éditeur et le jeu interactif. Nombreux sont les lecteurs qui ne savent pas par quel bout commencer, et comme rien ne vaut la pratique, la programmation de ces deux exemples sera une occasion de participer plus efficacement à la rubrique «dialogue avec nos lecteurs», mais aussi, pour beaucoup d'entre vous, d'apprendre à programmer avec méthode et réflexion. Nous essaierons de rendre cette «série» accessible à la majorité d'entre vous, c'est-à-dire, d'une part de ne pas aller trop vite (ce qui ne doit pas empêcher les courageux de nous envoyer leurs solutions à l'avance), et d'autre part de rester le plus longtemps possible indépendant du langage de programmation retenu.

Le cahier des charges

Cette contrainte est essentielle dans le succès ou l'échec aussi bien de l'analyse que de la programmation. Elle l'est d'autant plus dans notre cas que tout ce qui n'est pas explicité est sujet à diverses interprétations, les nôtres ne correspondant pas toujours aux vôtres. Nous commencerons donc par une description plus ou moins détaillée de ce que fait – pardon fera – chaque programme, mais aussi de ce qu'il ne fera pas. Les extensions ne seront envisagées qu'après réussite de l'ensemble du projet.

EDITEUR

Le principe est de créer un mini-traitement de texte en BASIC (ou en PASCAL), qui reste le plus proche possible du standard des fonctions de base de chacun des deux langages afin que vous puissiez tous adapter, sans trop de difficultés, les solutions que nous proposerons.

Fonctions de bases

- Accès plein écran : il est possible de déplacer (à l'aide de touches de contrôle ou des touches prédéfinies du clavier) de monter, descendre, aller à gauche ou à droite sans affecter le texte présent à l'écran.
- Insertion/surimpression : toujours par une touche prédéfinie, passage du mode insertion (où le texte à droite du curseur se décale au fur et à mesure de la frappe des caractères), au mode surimpression où les caractères entrés remplacent ceux déjà présents.
- Dans les limites définies ci-après, il sera possible d'effectuer un défilement à l'écran de l'ensemble du texte de haut en bas ou de bas en haut, insérer des lignes vierges ou en détruire (encore par des touches de contrôle).

Limitations

- La largeur maximale d'une ligne sera fixée une fois pour toutes dans l'application (40 à 80 colonnes suivant l'ordinateur), et l'insertion du texte sera limitée à cette caractéristique (il n'y aura notamment pas de retour automatique à la ligne).
- Les limitations de votre machine détermineront la taille de texte que vous pourrez traiter (on voit déjà poindre un tableau de chaîne de caractères derrière tout cela).
- Les fonctions (BASIC ou PASCAL) utilisées pour l'accès à l'écran seront celles les plus standard possibles, de même que les manipulations des chaînes de caractères (pour faciliter l'adaptation).

Contraintes

Bien que nous ne voulions pas réaliser un logiciel susceptible de rivaliser avec les meilleurs traitements de texte du marché, il faudra faire attention à prévoir dans la conception l'ajout de nouvelles touches de fonctions, ce qui nous permettra d'ajouter toutes sortes de modules supplémentaires :

- dans un premier temps, aucune fonction de sauvegarde ou d'impression n'est prévue, mais il faut avoir à l'esprit qu'elles seront à terme nécessaires,
- la gestion et le contrôle du clavier devront permettre cette évolution sans une réécriture complète.

JEU INTERACTIF

Ce type de programme impose d'autres contraintes pour être le plus performant donc le plus attractif possible. Il faut orienter l'analyse et la programmation de ce jeu sur la vitesse d'exécution de l'ensemble des fonctions indispensables. L'évolution n'est possible que dans la limite des performances de la machine. Vous aurez bien entendu beaucoup plus d'aisance avec un compilateur (BASIC ou PASCAL) parce que sans effort supplémentaire votre application tournera 5 à 100 fois plus vite qu'avec un interpréteur. Le cahier des charges se résume donc à ces fonctions essentielles en espérant obtenir dans l'environnement utilisé des performances suffisantes. La phase de mise au point et d'optimisation sera donc prépondérante dans la qualité finale du jeu.

Le cahier des charges

- Le but est d'atteindre le plus de cibles possibles en un certain temps (non pas un temps absolu, mais un temps lié au nombre de passages dans la boucle de gestion du jeu). Le jeu est prévu en mode texte pour ceux qui ne disposent pas de mode graphique, mais aussi pour éviter la surcharge d'affichage en mode haute résolution. Nous étudierons parmi les extensions le passage en mode graphique, et l'ajout d'options de jeux. Comme vous pouvez le voir sur l'exemple d'affichage, le score devra être indiqué en bas de l'écran et modifié bien évidemment après chaque destruction d'objet.
- Nous sommes loin des performances des grands classiques comme «SPACE INVADERS», sans parler des plus récents toujours plus rapides et réalistes. Mais, vous l'aurez sans aucun doute compris, notre ambition se résume à comprendre les mécanismes de la programmation en général, et ici plus particulièrement les contraintes imposées par la vitesse d'exécution («l'interactivité du programme pour employer un terme à la mode).
- Nous espérons qu'à partir des méthodes de base que nous allons définir, et des idées et astuces que vous nous enverrez ou que nous réaliserons, tous ceux qui parlent depuis longtemps de réaliser un jeu de ce type, passeront à l'acte.

DEFINITION D'UN PSEUDO-LANGAGE

Pour développer l'analyse de ces deux projets, nous utiliserons un langage mi-français, mi-informatique, qui aura le mérite de ne pas orienter l'analyse sur un langage quelconque. Nous pourrons ainsi à loisir définir des mots signifiant soit des actions élémentaires, soit des opérations complexes. Le plus simple est de vous exposer un exemple.

Nous voulons afficher les valeurs d'un tableau T à l'aide d'une boucle, taille étant le nombre de valeurs du tableau.

```

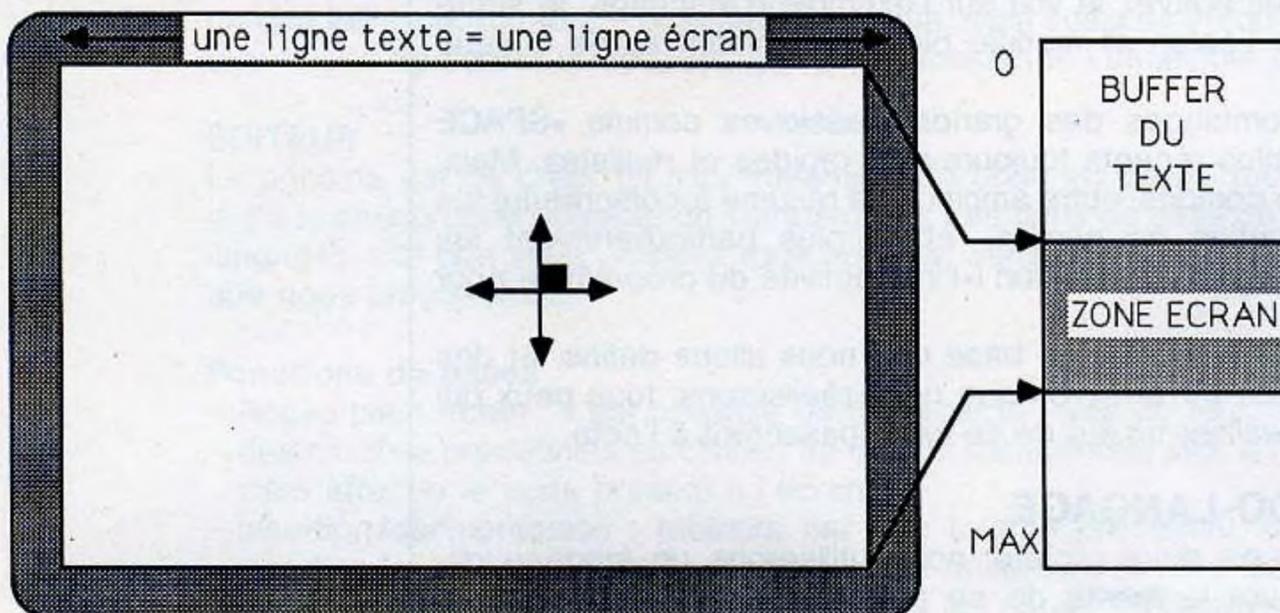
POUR i allant de 1 à taille
REPETER
  affiche(T(i));
FIN_REPETER.

```

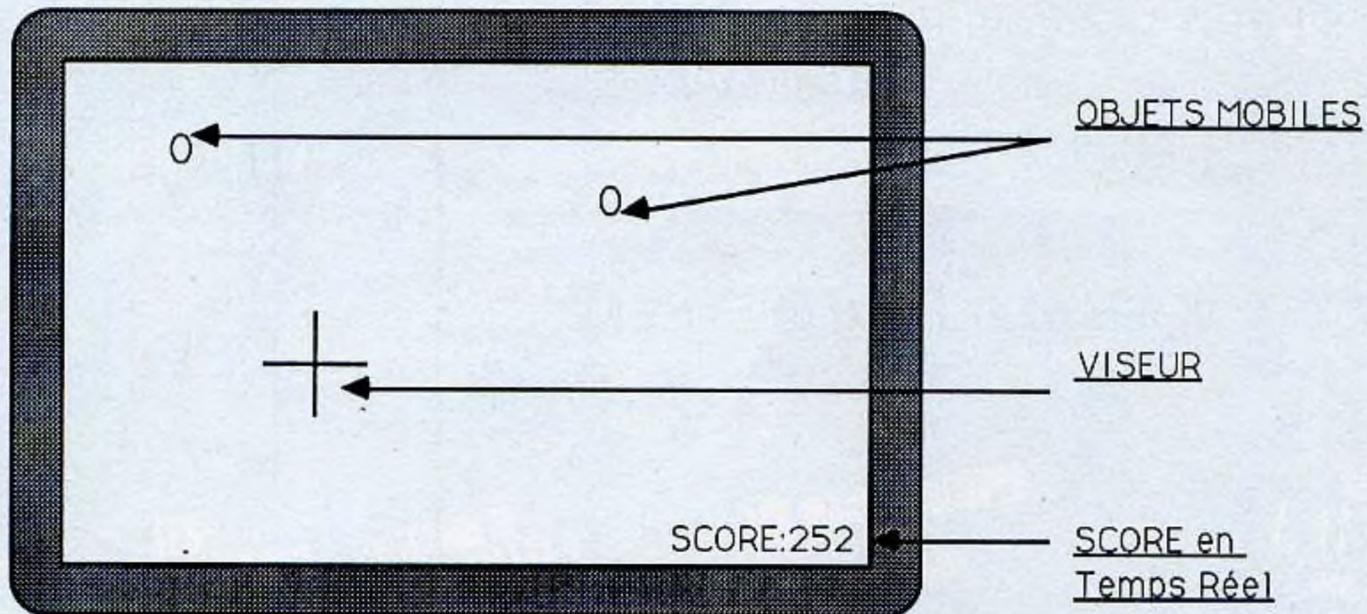
Evidemment, la fonction **affiche** peut être réduite à un simple PRINT BASIC ou WRITE PASCAL ; mais elle pourra tout aussi bien réaliser un formatage complexe, une conversion ou autres triturations délicates de la valeur du tableau. L'avantage d'une telle représentation réside dans la facilité de lecture. Elle permet en outre d'affiner petit à petit l'analyse du programme : par exemple, on pourra plus tard définir de manière analogue (c'est-à-dire à l'aide du même langage) la routine **affiche**. En fait, il faut bien l'avouer, la structure de ce langage ressemble à la traduction près à celle du langage PASCAL. Cependant, nous ne nous poserons pas de question sur sa syntaxe exacte ni sur l'existence réelle des fonctions que nous introduirons au fur et à mesure de l'analyse. L'essentiel est de connaître leur action. Ce n'est que lorsque la programmation réelle sera nécessaire que l'on résoudra ce type d'ambiguïté. Dans la pratique, ce langage est propre à chaque programmeur, l'essentiel étant qu'il le comprenne, et qu'il puisse sans difficulté le traduire dans son langage de prédilection (PASCAL, BASIC ou autre).

Du pain sur la planche

C'est tout pour commencer. Le premier effort est un effort de réflexion et d'analyse globale. Il faut déterminer une structure générale du programme, afin de découper le travail en parties indépendantes, ce qu'on appellera plus tard des modules, et de connaître les liens à créer entre ces parties pour que l'application fonctionne. Il n'est pas important de savoir pour le moment quelle touche servira pour l'insertion d'une ligne dans le texte, ou pour le tir sur les cibles. Le mois prochain sera donc consacré à la définition des grandes lignes des deux projets.



PRINCIPE DE STOCKAGE DU TEXTE



ECRAN DU JEU INTERACTIF

COURRIER DES LECTEURS

M. ARA nous a gâté ce mois-ci avec de nombreux programmes BASICs dont un sur le triangle de Pascal que nous avons adapté et que nous vous présentons ci-après. Nous répondrons à sa question au sujet du PASCAL sur Amstrad PCW 256 dans un prochain numéro, car comme vous avez pu le constater, nous n'avons pas encore donné suite au comparatif des compilateurs, interpréteurs en BASIC et PASCAL. Il est en effet difficile de trouver des précisions sur certains des produits que nous voulions vous présenter. Peut-être le mois prochain aurons-nous plus de renseignements. En tout cas, nous attendons toujours des réponses sur les précédents exercices proposés. Eh oui, ce n'est pas facile !

```

10 REM triangle de PASCAL
20 REM =====
30 REM initialisation de l'écran
40 CLS
80 PRINT "triangle de PASCAL           (T)"
90 PRINT "Développement de (a+b)^m    (D)"
100 PRINT "STOP/FIN                   (S)"
110 PRINT:PRINT "votre choix:";R$="**"
120 WHILE R$<>"S" AND R$<>"D" AND R$<>"T"
125 R$=INKEY$:R$=UCASE$(R$)
126 WEND

```

```

130 IF R$="S" THEN CLS:PRINT "Au revoir":END
140 CLS:PRINT "mémorisation du triangle... patientez"
150 REM definition des dimensions du tableau
160 EE=10
170 DIM C(EE,EE+1)
180 FOR k=1 TO EE:C(k,1)=1:C(k,2)=k:NEXT k:REM calcule C1 et C2
190 REM calcul de C3 à Cee
200 FOR i=2 TO EE
210 FOR j=3 TO EE+1
220 C(i,j)=c(i-1,j-1)+c(i-1,j)
230 NEXT j,i
310 IF R$="T" THEN 500
320 CLS
330 PRINT "Developpement de (a+b)^m avec le triangle de PASCAL"
340 PRINT
350 INPUT "Donnez la valeur de a:":a:
360 INPUT "Donnez la valeur de b:":b:
370 INPUT "Donnez la valeur de m:":m%:
380 IF m%<0 OR m%>10 THEN 370
410 CLS
430 PRINT "Developpement de (a+b)^m avec le triangle de PASCAL"
440 PRINT " a=";a;" b=";b;" (";a;"+";b;" )";m%;" = ";(a+b)^m%
445 PRINT:PRINT "Developpement:"
447 PRINT " a";m%;" +";TAB(24);" = ";USING "#####.##";a^m%:
450 FOR r%=1 TO m%-1
460 PRINT C(m%,r%+1);"* a";m%-r%;" * b";r%;" +";TAB(24);" = ";
470 PRINT USING "#####.##";C(m%,r%+1)*a^(m%-r%)*b^r%:
480 NEXT
490 PRINT "+ b";m%;TAB(24);" = ";USING "#####.##";b^m%:
495 PRINT :PRINT
497 Z$=INKEY$:IF Z$="" THEN 497 ELSE GOTO 40
500 REM affichage du triangle
510 CLS
520 PRINT "TRIANGLE arithmétique de PASCAL":PRINT
540 FOR j=1 TO 10
550 PRINT "(a+b)";j;TAB(13);"->";
560 FOR k=1 TO j+1 : PRINT TAB(15+6*k) " |";c(j,k);:NEXT:PRINT
570 NEXT j
580 z$=INKEY$:IF z$="" THEN 580 ELSE 40

```


LES MONTAGES ELECTRONIQUES

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

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

La finalité de cet ouvrage est de permettre à chacun, passionné d'électronique, de concevoir ses montages lui-même. Or, à notre époque où la technologie évolue sans cesse, il importe, principalement dans ce domaine, de bien connaître la majorité des composants mis en œuvre et les différentes façons de les utiliser. Devant l'ampleur du sujet, peu d'alternatives possibles, le savoir passant par la prise de conscience nette et délibérée de stockage, voire de mémorisation d'un maximum d'informations relatives aux différents matériels et schémas types d'exploitation. Par quelques 1 000 dessins et représentations divers, répartis sur plus de 380 figures, ainsi que par la description de 25 montages originaux, cet ouvrage représente un véritable outil de travail, permettant de familiariser l'électronicien avec les études électroniques, afin de l'amener, grâce à la somme de renseignements fournis, à concevoir et élaborer lui-même des réalisations personnalisées.

Le sommaire :

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

LE TELEPHONE ET LES RADIOTELEPHONES

Roger-Ch. Houzé

Le sommaire :

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

SORTIE PRÉVUE COURANT DÉCEMBRE

**Collection
jaune**

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

**Collection
noire**

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

Collection noire (format 165 x 240)

LES SYNTHETISEURS, UNE NOUVELLE LUTHERIE de Claude Gendre - 184 p. - Face au développement spectaculaire des synthétiseurs, grâce à l'électronique numérique, le besoin d'un ouvrage complet accessible et surtout bien informé des dernières ou futures techniques, se faisait ressentir. Le vœu est comblé, en 180 pages.

LES HAUT-PARLEURS de Jean Hiraga - 320 p. - Un gros volume qui connaît un succès constant : bien plus qu'un traité, il s'agit d'une véritable encyclopédie, alliant théorie et pratique, histoire en une mine inépuisable d'informations, reconnue dans le monde entier.

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

L'OPTIMISATION DES HAUT-PARLEURS ET ENCEINTES ACOUSTIQUES de Charles-Henry Delaleu - 240 p. - Seconde édition améliorée d'un ouvrage fort attendu des passionnés d'électroacoustique. Ce livre permet aux amateurs et aux professionnels de se familiariser avec les rigoureuses techniques de modélisation des haut-parleurs et enceintes acoustiques et d'en mener à bien la réalisation.

LES MAGNETOPHONES de Claude Gendre - 160 p. - Pour tout savoir sur le magnétophone depuis l'avènement de cette mémoire des temps modernes, jusqu'aux enregistreurs numériques, en passant par la cassette «Les magnétophones» est un ouvrage pratique, complet, indispensable à l'amateur d'enregistrement magnétique.

LES MAGNETOSCOPES ET LA TELEVISION de Claude Gendre - 256 p. - Complément direct des «Magnétophones» «Les magnétoscopes et la télévision» débute par un bel historique de la télévision et la description des premiers magnétoscopes. La théorie et la pratique de la capture et de l'enregistrement moderne des images vidéo en sont la teneur essentielle.

L'ELECTRONIQUE DES MICRO-ORDINATEURS de Philippe Faugeras - 128 p. - Cet ouvrage est destiné aux électroniciens désireux d'aborder l'étude du «hard» des micro-ordinateurs. Cette étude s'articule autour du microprocesseur Z-80 très répandu, et en décrit les éléments périphériques : mémoire, clavier, écran, interfaces de toutes sortes.

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

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

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

Introuvable aujourd'hui, une sélection des meilleurs articles de la célèbre revue «L'Audiophile». Le tome 1 traite de l'électronique audio à tubes et transistors. Dans un esprit identique, le tome 2 traite du domaine passionnant que constituent les transducteurs en audio.

LE MINI STUDIO de Denis Fortier - 160 p. - Le monde de l'audio évolue... Un secteur d'activité entièrement neuf vient d'apparaître : les mini-studios. L'ouvrage de Denis Fortier, ingénieur du son, aborde le sujet de la manière la plus globale. Après les données physiques indispensables, le choix des maillons, la manière d'installer et d'exploiter.

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

Collection rouge (format 135 x 210)

CONSEILS ET TOURS DE MAIN EN ELECTRONIQUE de Jean Hiraga 160 p. - Le «dernier coup de patte» apporté à un montage, celui qui fait la différence entre la réalisation approximative et le kit bien fini, ce savoir-faire s'acquiert au fil des ans... ou en parcourant «Conseils et tours de main en électronique».

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

LEXIQUE DE L'ELECTRONIQUE ANGLAIS-FRANÇAIS de Jean Hiraga - 72 p. - Pour la première fois en électronique, un lexique anglais-français est présenté sous une forme pratique avec en plus des explications techniques, succinctes mais précises. Ce sont plus de 1 500 mots ou termes anglais qui n'auront plus de secret pour vous.

FILTRES ACTIFS ET PASSIFS POUR ENCEINTES ACOUSTIQUES de Charles-Henry Delaleu - 160 p. - Finis les calculs fastidieux et erronés ! Grâce à cet ouvrage, les concepteurs d'enceintes acoustiques gagneront un temps appréciable durant la phase d'étude et de mise au point : 120 abaques et tableaux pour tous types de filtres et d'impédances de HP !

17 MONTAGES ELECTRONIQUES de Bernard Duval - 128 p. Voici enfin réunies dans un même ouvrage, dix-sept descriptions complètes et précises de montages électroniques simples. Il s'agit de réalisations à la portée de tous, dont bon nombre d'exemplaires fonctionnent régulièrement. Les schémas d'implantation et de circuits imprimés sont systématiquement publiés.

WEEK-END PHOTO de Philippe Folie-Dupart - 208 p. - Accessible à tous, «Week-end photo» permet de découvrir de façon simple les différents aspects de la photographie actuelle. Vous y trouverez les bases indispensables pour vous perfectionner, un guide de choix des appareils 24 x 36 et des illustrations abondamment commentées.

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

Collection jaune (format 210 x 270)

INITIATION A LA ROBOTIQUE 96 p. - Cet ouvrage eut un succès retentissant dès sa sortie. Bien plus qu'un cours d'initiation, il s'agit aussi du premier recueil d'informations données par les concepteurs, les utilisateurs et les fans de cybernétique enfin réunis !

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

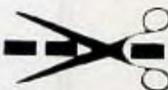
Frais de port : + 12 F par livre commandé, soit la somme totale ci-jointe, de Frs par CCP Chèque bancaire Mandat-lettre

Nom Prénom

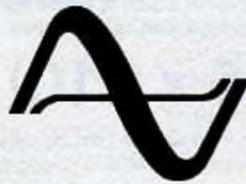
Adresse

Ville Code Postal

LED-Micro 11



C'EST ARRIVE DEMAIN



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

Il est étonnant de voir que la suprématie d'un standard tient à peu de choses. Prenons par exemple Unix. Voilà un système d'exploitation qui ne présente aucun intérêt particulier, qui est lourd, lent et volumineux. Mais, après avoir pris d'assaut le marché des minis, voilà qu'il fait de même avec les micros. Il existe depuis des années des émulations plus ou moins efficaces sous PC compatibles, et de nouvelles sortent qui collent mieux au standard tel qu'il est en train de se définir. Apple, dont on se demanderait volontiers ce qu'elle vient faire de ce côté, fait partie du groupe de sociétés, qui est en train de faire ce standard. Il est maintenant acquis que le prochain Mac tournera sous Unix, via un émulateur, ce qui le rendra maître du terrain de la micro, car il semble difficile que les petites sociétés (entendez Commodore et Atari) puissent

résister à la pression et au fumet de la compatibilité. Car, le mot est lancé, le problème est cette fameuse compatibilité, légendaire et jamais au rendez-vous. Unix permettra-t-il de rendre compatibles un PC et un Mac ? Il ne faut tout de même pas rêver ! Déjà, si les communications sont simplifiées, ce sera très satisfaisant ! Mais il faut bien se souvenir que se réfugier dans un standard, c'est la mort du génie inventif à court terme. Croyez-vous que MM. Jobs et Wozniack auraient la moindre chance si Unix régnait en maître et que tous les ordinateurs soient compatibles, s'ils débarquaient avec un micro révolutionnaire, peu coûteux, mais non compatible ? Alors, la compatibilité, oui, mais que cela ne signifie pas l'uniformité, la grisaille et la monotonie. Oui à une industrie ambitieuse et inventive.

Il semble que les systèmes multi-utilisateurs basés sur un micro unique soient un désastre, au sens le plus strict du terme. Un seul ordinateur permet de s'en sortir, au prix d'un processeur par utilisateur, et d'une zone mémoire protégée pour chaque. En clair, plusieurs micros en un seul. Un réseau en une seule boîte. Compupro, le fabricant, a réalisé une étude de marché qui a montré que tout système multi-utilisateurs sur micro fonctionne de la manière suivante : quelques utilisateurs, au plus quatre, utilisent des fichiers communs, et n'accèdent que rarement à d'autres zones de mémoire. Le Compupro est donc une machine pour au plus quatre utilisateurs simultanés et on peut lier deux, ou plusieurs machines ensemble. C'est parfait, ou presque, pour des petites structures de travail, bureaux... mais dès que la structure augmente d'elle-même, les problèmes apparaissent. Délais d'attente peu ou pas raisonnables, liaisons complexes entre machines (certaines sociétés en sont à lier une trentaine de machines), rien ne va plus pour les extensions un peu lourdes. Le problème est tel que l'image de Compupro en prend un sérieux coup. Compupro envisage d'ailleurs, dit-on, de retirer cette machine du marché. Cela signifierait qu'une tentative louable est abandonnée et rendrait alors de la vigueur à l'adage qui veut que pour bien fonctionner, un micro doit avoir un processeur par utilisateur, ce qui est une cruelle limitation.

Les systèmes experts sont censés simuler l'intelligence et la gestion des connaissances humaines. Il est maintenant accepté que dans certains domaines bien définis (acceptation de l'usage de cartes de crédit, recherche de pièces défectueuses sur un système mécanique complexe, gestion des réseaux routiers en fonction de l'heure, de la date, du temps et de la circulation), un système expert peut accroître l'efficacité de l'équipe chargée de ces tâches, au point de l'amener à être plus efficace qu'un expert humain du domaine. Le principal bénéfice est que l'ordinateur peut donc devenir un maillon de la chaîne de décision, voire le seul, donner des indications utiles et directement exploitables, proposer des solutions de rechanges si un problème particulier se pose. Cela devient donc un outil intéressant, même s'il ne faut pas se cacher qu'un système expert est une structure lourde, souvent difficile à contrôler, et non exempte d'erreurs de fonctionnement, un théorème mathématique prouvant même qu'il n'est pas possible d'éliminer toute erreur résiduelle d'un tel programme. Ceci dit, la technologie des systèmes experts est au stade de la maturité et il est réconfortant de voir que la science-fiction devient parfois réalité, et même très vite.

Ce mois-ci, ouvrant le mensuel *Byte*, je découvre une annonce élogieuse pour un nouveau produit, un traitement de texte. Après avoir lu toutes les possibilités

de ce merveilleux produit, à les en croire, je regarde les coordonnées du fabricant. Et là, la vraie, grande surprise. L'adresse est Paris (et pas Paris Texas, croyez-moi...). Alors, il n'est plus possible de résister, il faut dire de qui il s'agit. La société qui a réussi à avoir un article élogieux dans une revue américaine est... : Anatex 18, rue Troyon, Paris 17^e, leur traitement de texte tourne sur Macintosh.

Stanford et une société de recherche ont développé un produit permettant de simuler les effets d'un tremblement de terre, et d'en étudier les effets. Il s'agit de pouvoir analyser les données recueillies lors d'un tremblement de terre, et de les traiter pour en tirer des conclusions sur le plan habitat, industrie... pour cela, un micro, portable de surcroît, avec Lotus 1-2-3 suffit. Prenons un cas concret simple. Vous souhaitez savoir quelle est la probabilité pour que l'immeuble dans lequel vous travaillez bien tranquillement depuis des années s'effondre au moindre tremblement de terre, il suffit d'entrer dans le programme les données du problème, avec le poids de l'immeuble, du toit, les matériaux, etc., et vous saurez jusqu'à quel point ce bâtiment est sûr. Il y a du déménagement dans l'air.

Tout le monde le sait, le plus complexe des réseaux artificiels n'est rien à côté de celui formé par le cerveau du plus simple des animaux. Un exemple, la possibilité de faire de la reconnaissance de formes que possèdent tous les animaux ou presque, et qui reste à faire tourner efficacement sur une machine. Un laboratoire chez ATT Bell tente de développer des programmes de reconnaissance des formes à partir de schémas de réseaux neuronaux simples intégrés sur des composants. A l'heure actuelle, trois réseaux de ce type fonctionnent, allant de l'équivalent électronique de 22 neurones, de 54, et de 300. Le premier est suffisamment simple pour qu'il soit simple à tester. Par contre, dès le second, tout se complique. 54 neurones, et plus de 3 000 synapses, les liaisons inter-neurones, cela devient déjà presque trop complexe. Alors, imaginez le troisième, qui n'en est qu'à la phase expérimentale, ces composants ont été testés de la manière suivante. Une liste de noms leur est « apprise », puis un nom différent leur est soumis, et ils doivent dire lequel parmi ceux qu'ils connaissent est le plus proche. L'architecture parallèle du processeur est plus efficace que celle de processeurs traditionnels, environ 100 à 1 000 fois, en termes de vitesse de traitement. Voilà qui est intéressant, mais il faut sans doute ne pas se leurrer, cette étape est importante, mais cela ne règle rien à l'incapacité humaine à reproduire le cerveau, et sans doute faut-il ne pas trop s'en plaindre.

Ce mois-ci, les sujets abordés sont très orientés vers les débouchés de l'intelligence artificielle. Or, celle-ci est dépendante d'un domaine connexe proche, la

représentation des connaissances, dont certains pensent qu'il s'agit de la pierre angulaire de l'I.A. Les progrès des années passés sont tels que maintenant on pense à l'I.A. en termes de commerce, ce qui était risible voilà encore deux ans. Mais il reste le problème de stocker la connaissance, qui est vaste et non structurée, par définition. Vient ensuite le problème du raisonnement, basé actuellement sur des règles du type SI... ALORS... SINON, ce qui n'est pas d'une grande souplesse pour exprimer un sentiment de type normand (p'têt ben..., vous devez connaître). Il devient indispensable, pour conserver l'approche, de rajouter des coefficients, dits de vraisemblance, qui permettent de moduler la réponse à une règle, qui doit être propagée par la suite, pour se souvenir de ladite vraisemblance. Il s'ensuit des lourdeurs dans le stockage des enchaînements. Difficile de s'en sortir. Alors, diverses méthodes sont à l'étude, certaines sont intéressantes. Par exemple, des algorithmes qui dépisent les problèmes éventuels se manifestant en cours de route. L'avenir de ces systèmes est basé sur une recherche passionnante, mais ardue.

Digital Research Inc. est la société à qui l'on doit des merveilles pour leur époque, CP/M, GEM, et d'autres produits pleins d'intérêt. Il n'empêche que la société a des problèmes de plus en plus visibles. Pour la troisième fois depuis sa création, DRI va débaucher du personnel, 22 % cette fois-ci. La société va se replier sur les produits existants, GEM et Concurrent DOS, en deux équipes distinctes. C'est donc la mort pour la branche développement, tout au moins à terme. Et c'est sans doute également la mort pour DRI en tant que telle, car il n'est plus possible de survivre avec des produits figés et sans projets à terme. Alors il est dommage de choisir la mort lente, car les ingénieurs de DRI ont montré qu'ils sont capables de bien belles choses. Mais, selon l'analyse d'un spécialiste des sociétés de technologie, DRI «dans les deux dernières années, a produit un effort sans précédent chez les sociétés de sa catégorie, pour introduire des produits excellents sur un mauvais créneau.» Cela semble une belle épitaphe, mais le président de DRI se

déclare persuadé que, dès la fin de l'année, les profits reviendront. Oui, mais dans deux ans ?

Il y a quelques grands réseaux de particuliers aux U.S.A. Parmi ceux-ci Comuserve est l'un des grands. Oui, mais pour s'y retrouver quand vous êtes le client lambda, il faut une patience indéfectible. Or, la patience n'est pas la qualité première du client, c'est bien connu. Il faudrait d'ailleurs savoir si cela doit être sa qualité première, lui qui est soit-disant roi. Toujours est-il que les clients boudent Comuserve, et certains autres. Alors, que reste-t-il à faire ? A simplifier. En particulier, au niveau de la gestion des protections qui permettent à certains de se protéger de l'accès d'autres dans leurs discussions. Mais celles-ci, d'après les statuts, sont publiques. Il semble que la solution soit délicate à trouver entre ceux qui échangent de l'information et ceux qui souhaitent s'insérer dans la discussion. Les cadres de Comuserve n'ont pas encore la solution, mais une protection plus rigide hors de question, car alors 90 % des clients ne trouveraient plus de raison de payer un abonnement fort coûteux si l'information n'est plus au rendez-vous.

Il paraît que l'expérience des autres ne sert jamais. «Erreur» répond le président de Sytek, qui fabrique des cartes d'interface pour un réseau sur IBM PC. Sytek est une des nombreuses compagnies à perdre du jour au lendemain son contrat avec IBM, pour la fourniture de produits directement implantés sur les PC. Mais, au vu des autres, elle a eu le temps de réaliser un produit complet personnel, de bonne qualité et performant. Si bien que là où une dizaine de sociétés ont déjà bu la tasse, Sytek a réussi sa reconversion, bien qu'IBM lui ait signifié l'arrêt complet et immédiat de leurs relations. IBM est coutumier du fait, semble-t-il, ce qui fait montre d'une délicatesse et d'une correction à toute épreuve. Il paraît que les businessmen sont durs ici, mais peut-être ne saviez-vous pas qu'ici, cela veut dire grossier.

Au mois prochain.

COURS D'INITIATION AU PROGICIEL MULTIPLAN

Charles-Henry Delaleu

4^e PARTIE

Commencé il y a maintenant trois mois, notre cours sur Multiplan s'achève dans ce présent numéro de LED-MICRO. Il ne nous reste plus qu'à étudier les fonctions avancées de ce progiciel.

Le mois prochain, nous ferons un bref survol des progiciels SAARI.

- comptabilité
- gestion commerciale
- paie.

Il ne s'agira pas d'un cours tel que pour Multiplan. En effet, il est impossible d'assimiler ces trois programmes en un seul numéro. Inversement, un cours approfondi risquerait de laisser bon nombre de lecteurs. Nous nous attacherons donc à décrire les concepts essentiels de ces progiciels. Il convient de noter que la gestion d'entreprise peut être actuellement fortement informatisée. Les machines de type «IBM PC et compatible», équipées d'un disque dur de 10 à 20 Mo peuvent, sans aucun problème, résoudre la majorité des difficultés posées en entreprises de petites et moyennes tailles.

Aussi, nous en sommes sûr, la description des programmes SAARI vous fera découvrir des solutions toutes prêtes, bien rodées, et fiables.

Dès le mois de février, nous reprendrons notre cours d'initiation avec un nouveau progiciel. Ce cours finira au mois de juin, il durera donc cinq mois. Nous avons actuellement le choix entre deux programmes :

- dBase III (gestionnaire de bases de données)
- Textor (traitement de texte).

Ces deux progiciels seront étudiés en cours, à vous de choisir par lequel il convient de commencer !

LA COMMANDE FENETRE PARTAGE DESIGNATION

PARTAGE

Le nombre de lignes et de colonnes est important dans Multiplan. Lors du défilement, il est possible d'oublier les titres des lignes et/ou des colonnes. Il est souhaitable de partager sa feuille de calcul en une partie fixe, et l'autre non.

FENETRE: **Partage** Encadre Supprime Couplage

Choisissez une option ou frappez le caractère de commande

L3C2 20000 97% Libre Multiplan: MARTIN

Pour partager une feuille, il suffit d'appuyer sur la commande ZONE-FENETRE. La fonction partage est accessible à la première commande.

*ce nombre depend de
l'emplacement du
pointeur de cellule*

FENETRE PARTAGE DESIGNATIONS nb de lignes: 2 nb de colonnes: 1

Entrez un nombre

L3C2 20000 97% Libre Multiplan: MARTIN

Après avoir validé «PARTAGE», il ne reste plus qu'à choisir le nombre de lignes et de colonnes que vous désirez inclure dans votre fenêtre.

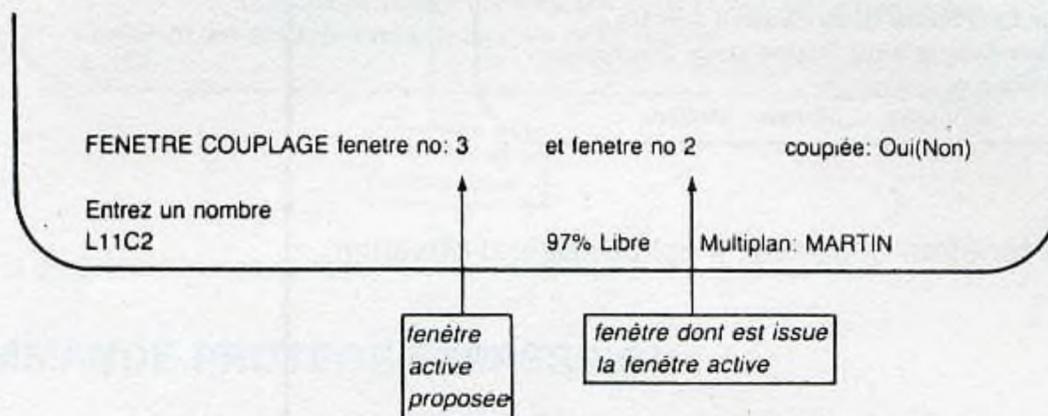
#1	1	#2	2	3	4	5
1			Janvier	Février	Mars	Avril
2						
3	Ventes		20000,00 F	20000,00 F	20000,00 F	20000,00 F
4						
5	Coûts					
6	Matériaux		4000,00 F	4000,00 F	4000,00 F	4000,00 F
7	Main d'oeuvre		7000,00 F	7000,00 F	7000,00 F	7000,00 F
8	Frais divers		4000,00 F	4000,00 F	4000,00 F	4000,00 F
9						
10	Coût total		15000,00 F	15000,00 F	15000,00 F	15000,00 F
11						
12						
13						
14						
15	Marge Brute		5000,00 F	5000,00 F	5000,00 F	5000,00 F
16						
17						
18						
19						

COMMANDE: Alpha Blanc Calcul Detruit Edite Format Guide Insere Lit_Ecrit Mouv.
Nom Options Protège Quitte Recopie Sortie Tri Vers Xterne ZoneFenêtre
Choisissez une option ou frappez le caractère de commande
L3C2 20000 97% Libre Multiplan: MARTIN

Dans cet exemple, nous avons deux fenêtres # 1 et # 2. Une seule fenêtre est active à un instant donné. Il est possible de valider une fenêtre grâce à la commande «VERS». Ici nous pouvons faire défiler les variables de la fenêtre n°2 tout en gardant les titres des lignes grâce à la fenêtre n°1. Il est donc possible de lire toutes les variables de janvier à décembre tout en gardant sous les yeux un écran parfaitement lisible.

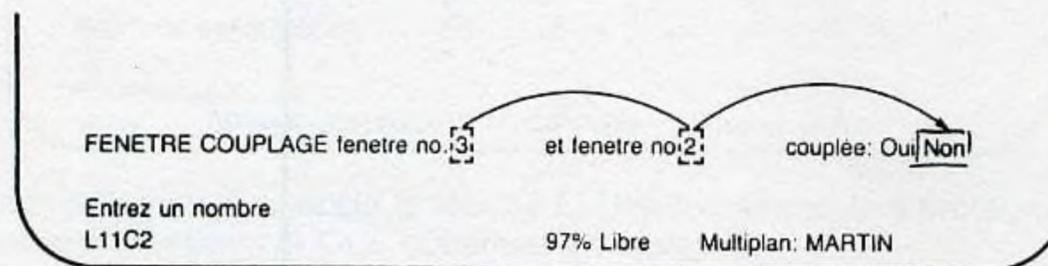
LA COMMANDE FENETRE COUPLAGE

Si vous utilisez une feuille de calcul très vaste, il vous est possible d'y créer plusieurs fenêtres pour réaliser des rapprochements.



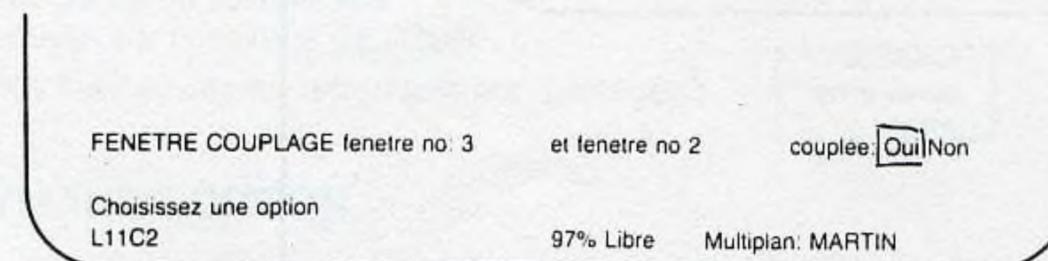
Pour coupler deux fenêtres, il convient d'appeler ZONE-FENETRE, puis la commande COUPLAGE.

En deuxième lieu, il faut indiquer les numéros des fenêtres.



Pour valider la commande, déplacer le pointeur sur «OUI».

Appuyer sur la touche TAB deux fois.



Enfin appuyer sur la touche O ou espace pour sélectionner «OUI».

#1	1	#2	2	3	4	5
1			Janvier	Fevrier	Mars	Avril
2						
3	Ventes		20000,00 F	20000,00 F	20000,00 F	20000,00 F
4						
5	Couts					
6	Matériaux		4000,00 F	4000,00 F	4000,00 F	4000,00 F
7	Main d'oeuvre		7000,00 F	7000,00 F	7000,00 F	7000,00 F
8	Frais divers		4000,00 F	4000,00 F	4000,00 F	4000,00 F
9						
10	Cout total		15000,00 F	15000,00 F	15000,00 F	15000,00 F
11		#3				
12		11				
13		12				
14		13				
15	Marge Brute	14				
16		15	5000,00 F	5000,00 F	5000,00 F	5000,00 F
17		16				
18		17				
19		18				

COMMANDE: Alpha Blanc Calcul Détruit Edite Format Guide Insere Lit_Ecrit Mouv.
 m Options Protege Quitte Recopie Sortie Tri Vers Xterne ZoneFenêtre
 Choisissez une option ou frappez le caractere de commande
 L11C2 97% Libre Multiplan: MARTIN

les numeros de
 colonne disparaissent
 de la fenetre #3

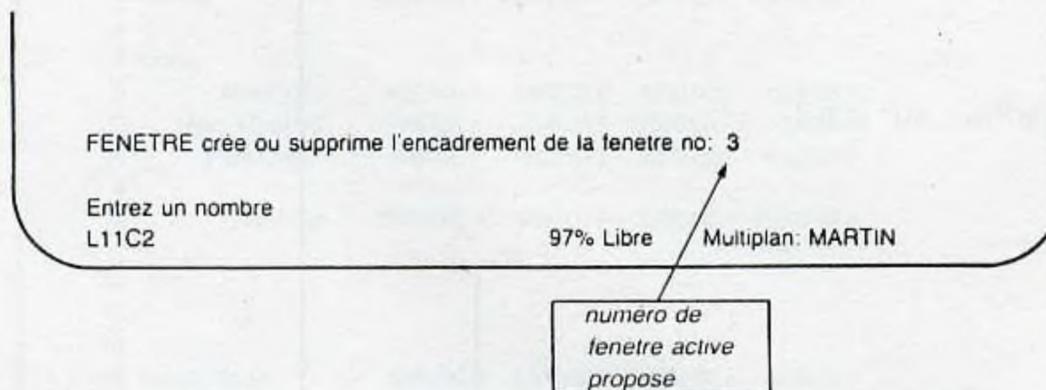
Exemple d'une feuille de calcul avec des fenêtres couplées. Il est possible d'observer plusieurs résultats en même temps.

LA COMMANDE FENETRE ENCADRE

La commande ENCADRE permet d'obtenir une meilleure présentation d'une feuille de calcul.

Cette commande est accessible par :

1. ZONE-FENETRE
2. ENCADRE.

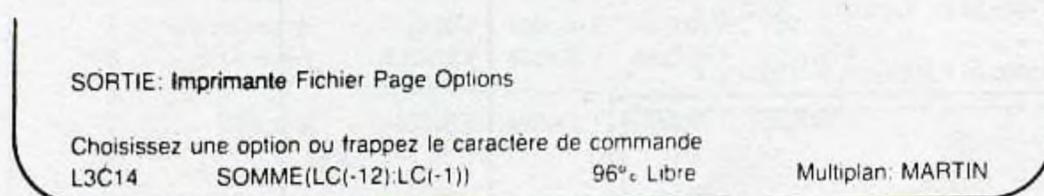


Pour encadrer une fenêtre, il suffit de donner le numéro de cette dernière.

LA COMMANDE SORTIE

La commande SORTIE permet de sortir des données sur l'imprimante :

1. Appuyer sur SORTIE
2. Valider IMPRIMANTE.



IMPRIMANTE

Lors d'une demande d'impression, vérifier toujours que l'imprimante est bien sous tension, qu'elle est reliée au calculateur, qu'elle est «ON LINE».

Après avoir choisi la commande IMPRIMANTE, appuyer sur RETOUR pour faire démarrer l'impression.

FICHER

La sortie fichier autorise deux possibilités :

1. Si vous ne disposez pas d'une imprimante, vous pouvez créer un fichier sur disquette correspondant au contenu de votre feuille pour une impression ultérieure.
2. Vous souhaitez intégrer vos résultats de Multiplan dans un traitement de texte. Cette commande vous permet de traduire vos données en un format standard accessible par tout progiciel de traitement de texte (voir introduction du cours n° 3).

PAGE

Cette sous-commande vous permet de spécifier des marges pour imprimer votre feuille de calcul.

OPTIONS

Cette sous-commande vous permet de spécifier ce que vous voulez imprimer :

- les résultats,
- les équations,
- les expressions,
- les numéros de lignes et de colonnes.

LIAISON ENTRE FEUILLES DE CALCUL

LE PRINCIPE

Vous manipulez un nombre très important de données. Dans ce cas, vous obtenez une feuille de calcul vaste, lourde, et peu maniable. Afin d'éviter ce genre de situation, Multiplan vous offre une solution.

Il faudra créer plusieurs feuilles contenant chacune une application simple, une simulation de base. Plusieurs feuilles de calculs primaires donneront naissance à une feuille de calcul de synthèse regroupant les résultats des feuilles primaires. Bien entendu, ces différentes feuilles de calcul doivent être liées afin d'autoriser une mise à jour automatique des résultats sur la feuille de synthèse. La feuille de synthèse est appelée : feuille RECAPITULATIVE. Elle dépend des données et des calculs effectués sur les feuilles «DETAIL».

Exemple : Gestion d'un chantier.

FEUILLE DETAIL : MATERIAUX**MATERIAUX**

	JAN	FEV	MAR	AVR
CIMENT	100	110	110	100
BRIQUES	200	300	250	500
DIVERS	500	50	10	20
TOTAL	800	460	370	620

FEUILLE DETAIL : FRAIS DIVERS**FRAIS DIVERS**

	JAN	FEV	MAR	AVR
SALAIRE	100	100	100	100
CHARGE	500	0	0	150
ASSURANCE	600	0	0	0
TOTAL	1 200	100	100	250

FEUILLE RECAPITULATIVE : DEPENSES**DEPENSES**

	JAN	FEV	MAR	AVR
FRAIS	1 200	100	100	250 (1)
MATERIAUX	800	460	370	620 (2)
TOTAL	2 000	560	470	870

(1) Report de la ligne Total FRAIS

(2) Report de la ligne Total MATERIAUX

LA COMMANDE LIT-ECRIT EFFACE-ECRAN

La commande EFFACE-ECRAN est utilisée pour effacer l'écran.

- Elle vide la feuille de toutes informations.
- Elle prépare une feuille complètement vierge.
- Toute information protégée est gardée.

CONSTRUCTION D'UNE FEUILLE DE DETAILS

Si vous désirez réaliser un feuille de synthèse (RECAPITULATIVE), il est absolument nécessaire de bien préparer son travail. Toutes les feuilles de détails seront réalisées suivant le même modèle. La feuille récapitulative prendra la même architecture. Ce genre de méthode facilite grandement la présentation et le passage des chiffres vers la feuille de synthèse.

PROCESSUS D'UTILISATION

1. Dans un premier temps, il convient de réaliser ses différentes feuilles de détails.
2. Il sera nécessaire de nommer les groupes de cellules qui contiennent les valeurs (données) à transférer dans la feuille récapitulative.
3. Chaque feuille de détails est sauvegardée séparément.
4. Charger en mémoire vive la feuille de calculs récapitulative.
5. Grâce à la commande EXTERNE, passer les informations des feuilles de détails sur la feuille récapitulative.

LA FEUILLE DE DETAILS

Avant de sauvegarder chaque feuille de détails, il est nécessaire de nommer chaque groupe de cellules à transférer sur la feuille récapitulative.

Exemple :

a)

NOM: donner le nom: Cout_Materiaux de référence à: L8C2:13

Entrez une référence de cellule ou de groupe de cellules

L8C13 LC(-1)*100,8% 98% Libre Multiplan: TEMP

b)

NOM: donner le nom: Cout_Frais_Divers de référence à: L15C2:13

Entrez un nom

L15C2 4000 98% Libre Multiplan: TEMP

*l'option
proposée
est correcte*

Pour nommer un groupe de cellules, utilisez la commande NOM.

COMMENT PASSER LES DONNEES SUR LA FEUILLE RECAPITULATIVE

1. Utiliser la commande EXTERNE.

EXTERNE: Recopie Liaisons Substitue

Choisissez une option ou irapez le caractère de commande

L6C2 4000 96% Libre Multiplan: MARTIN

2. Valider la sous-commande RECOPIE.

EXTERNE RECOPIE depuis feuille:	nom de zone:
destination: L6C2	couplée:(Oui)Non
Entrez un nom de fichier	
L6C2 4000	96% Libre Multiplan: MARTIN

3. Indiquer le nom de la feuille DETAIL d'où viennent les informations.

EXTERNE RECOPIE depuis feuille: MARCOUT	nom de zone:
destination: L6C2	couplée:(Oui)Non
Entrez un nom de fichier	
L6C2 4000	96% Libre Multiplan: MARTIN

4. Préciser le nom des cellules contenant les données.

EXTERNE RECOPIE depuis feuille: MARCOUT	nom de zone: Cout_Matériaux
destination: L6C2	couplée:(Oui)Non
Nommez la zone sur la feuille externe	
L6C2 4000	96% Libre Multiplan: MARTIN

nom défini dans la feuille detail

NOTA : Cette opération est effectuée pour chaque feuille de détails, et chaque groupe de cellule à transférer en feuille récapitulative.

IMPORTANT

La présentation de la feuille récapitulative doit être réalisée avec attention. La destination de la copie doit être vierge de toute information préalable.

Le format et le nombre des données doivent être les mêmes sur la feuille de détails et sur la feuille récapitulative.

#1	1	#2	2	3	4	5
1			Janvier	Février	Mars	Avril
2						
3	Ventes		20000,00 F	20200,00 F	20402,00 F	20606,02 F
4						
5	Coûts					
6	Matériaux					
7	Main d'oeuvre		7000,00 F	7000,00 F	7000,00 F	7000,00 F
8	Frais divers		4000,00 F	4000,00 F	4000,00 F	4000,00 F
9			-----	-----	-----	-----
10	Cout total		11000,00 F	11000,00 F	11000,00 F	11000,00 F
11						
12						
13						
14						
15	Marge Brute		9000,00 F	9200,00 F	9402,00 F	9606,02 F
16						
17						
18						
19						

la destination de la copie externe doit être vierge

COMMANDE: Alpha Blanc Calcul Détruit Edite Format Guide Insere Lit_Ecrit Mouv.
 Nom Options Protège Quitte Recopie Sortie Tri Vers Xterne ZoneFenêtre
 Choisissez une option ou frappez le caractère de commande
 L6C2 96% Libre Multiplan: MARTIN

Feuille récapitulative avant transfert.

#1	1	#2	2	3	4	5
1			Janvier	Février	Mars	Avril
2						
3	Ventes		20000,00 F	20200,00 F	20402,00 F	20606,02 F
4						
5	Coûts					
6	Matériaux		4000,00 F	4032,00 F	4064,26 F	4096,77 F
7	Main d'oeuvre		7000,00 F	7000,00 F	7000,00 F	7000,00 F
8	Frais divers		4000,00 F	4000,00 F	4000,00 F	4000,00 F
9			-----	-----	-----	-----
10	Cout total		15000,00 F	15032,00 F	15064,26 F	15096,77 F
11						
12						
13						
14						
15	Marge Brute		5000,00 F	5168,00 F	5337,74 F	5509,25 F
16						
17						
18						
19						

COMMANDE: Alpha Blanc Calcul Détruit Edite Format Guide Insere Lit_Ecrit Mouv.
 Nom Options Protège Quitte Recopie Sortie Tri Vers Xterne ZoneFenêtre
 Choisissez une option ou frappez le caractère de commande
 L6C2 (Marcout Cout_Matériaux) 96% Libre Multiplan: MARTIN

la cellule active indique le nom de la feuille et le nom des cellules copiées

Feuille récapitulative après transfert.

NOTA : Le coût des matériaux a été transféré en ligne 6. Ces informations étaient comprises sur la feuille de détails appelée MARCOUT.

CORRECTION EXERCICE N° 1

	2	3	4	5
2	L(-2)C(-2)	L(-2)C(-1)	L(-2)C	
3	L(-1)C(-2)	L(-1)C(-1)	L(-1)C	
4	LC(-2)	LC(-1)	cellule active	

CORRECTION EXERCICE N° 2

	1	2	3
1	"TVA :"	18,6	
2	"FER :"	200	
3			
4			
5			"JAN"
6			
7		"NBRE FER"	20
8		"VENTES H.T."	L(-1)C*PFER
9		"T.V.A"	L(-1)C*TVA*0,01
10		"VENTES T.T.C."	L(-2)C+L(-1)C
11			
12		"TOTAL ANNUEL HT :"	SOMME(L(-4)C:L(-4)C(+11))
13		"TOTAL ANNUEL TVA :"	SOMME(L(-4)C:L(-4)C(+11))
14		"TOTAL ANNUEL TTC :"	22
	4	5	6
	"FEV"	"MAR"	"AVR"
	LC(-1)*1,15	LC(-1)*1,15	LC(-1)*1,15
	L(-1)C*PFER	L(-1)C*PFER	L(-1)C*PFER
	L(-1)C*TVA*0,01	L(-1)C*TVA*0,01	L(-1)C*TVA*0,01
	L(-2)C+L(-1)C	L(-2)C+L(-1)C	L(-2)C+L(-1)C
	7	8	9
	"MAI"	"JUI"	"JUL"
	LC(-1)*1,15	LC(-1)*1,15	LC(-1)*1,15
	L(-1)C*PFER	L(-1)C*PFER	L(-1)C*PFER
	L(-1)C*TVA*0,01	L(-1)C*TVA*0,01	L(-1)C*TVA*0,01
	L(-2)C+L(-1)C	L(-2)C+L(-1)C	L(-2)C+L(-1)C

10

11

12

"AOU"

"SEP"

"OCT"

LC(-1)*1,15
L(-1)C*PFER
L(-1)C*TVA*0,01
L(-2)C+L(-1)C

LC(-1)*1,15
L(-1)C*PFER
L(-1)C*TVA*0,01
L(-2)C+L(-1)C

LC(-1)*1,15
L(-1)C*PFER
L(-1)C*TVA*0,01
L(-2)C+L(-1)C

13

14

"NOV"

"DEC"

LC(-1)*1,15
L(-1)C*PFER
L(-1)C*TVA*0,01
L(-2)C+L(-1)C

LC(-1)*1,15
L(-1)C*PFER
L(-1)C*TVA*0,01
L(-2)C+L(-1)C

TVA : 18,6
FER : 200

JAN

FEV

MAR

NBRE FER	20	23	26
VENTES H.T.	4000,00	4600,00	5290,00
T.V.A	744,00	855,60	983,94
VENTES T.T.C.	4744,00	5455,60	6273,94

TOTAL ANNUEL HT : 116006,67
TOTAL ANNUEL TVA : 21577,24
TOTAL ANNUEL TTC : 22,00

AVR

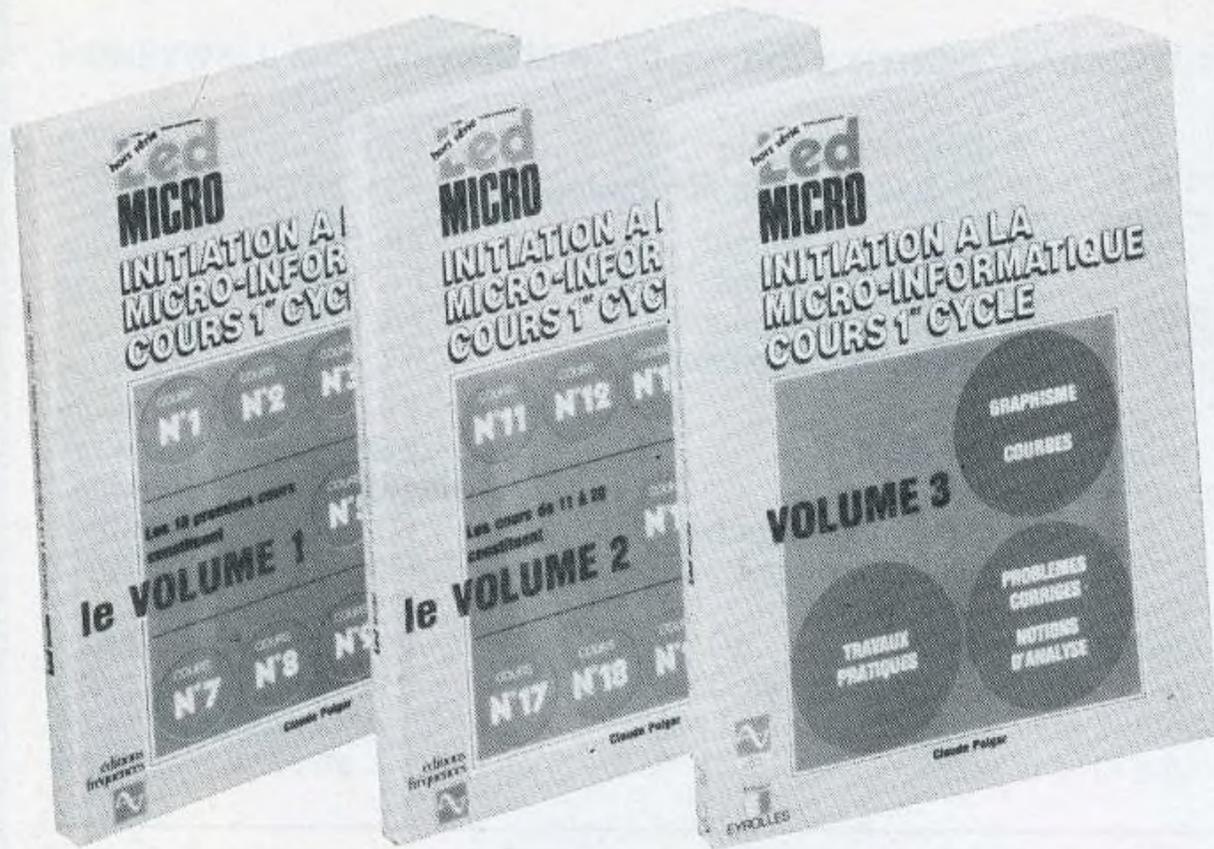
MAI

JUI

JUL

AOU

30	35	40	46	53
6083,50	6996,03	8045,43	9252,24	10640,08
1131,53	1301,26	1496,45	1720,92	1979,05
7215,03	8297,29	9541,88	10973,16	12619,13



**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 !).

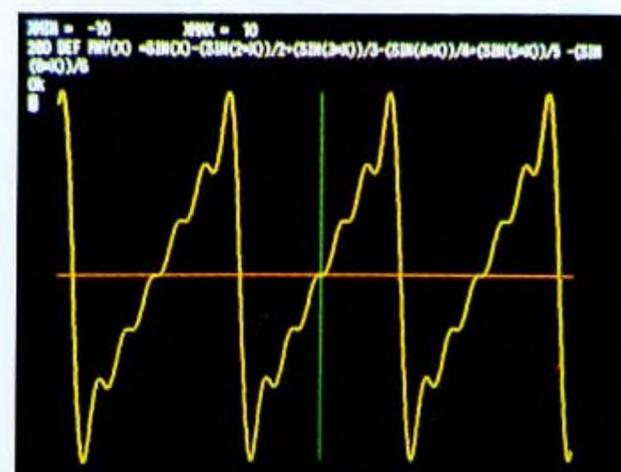
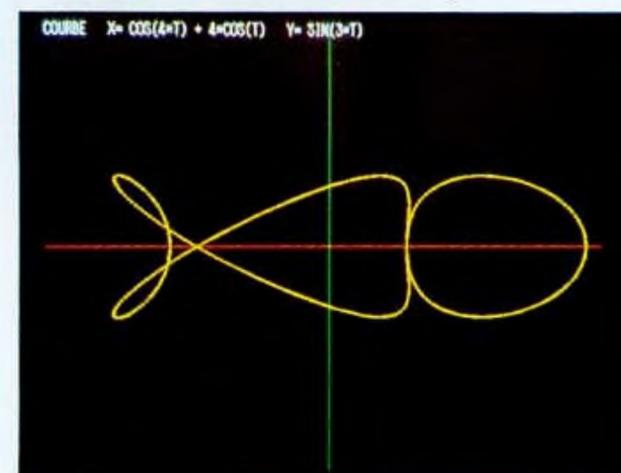
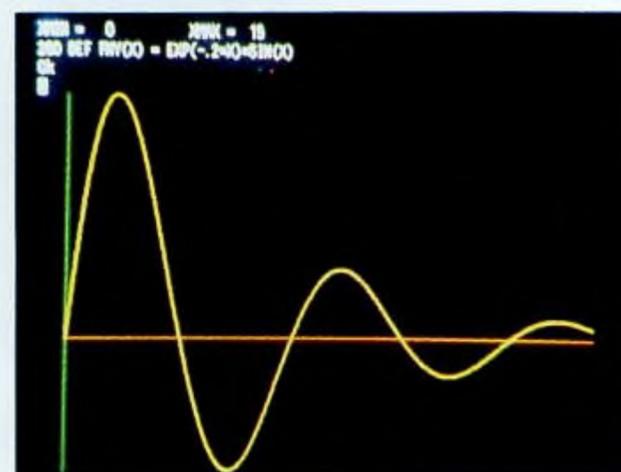
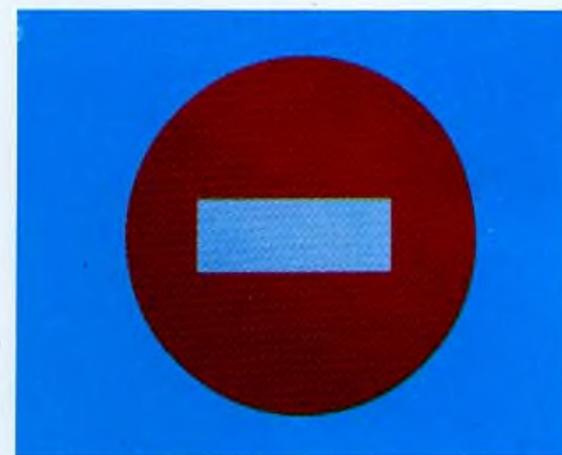
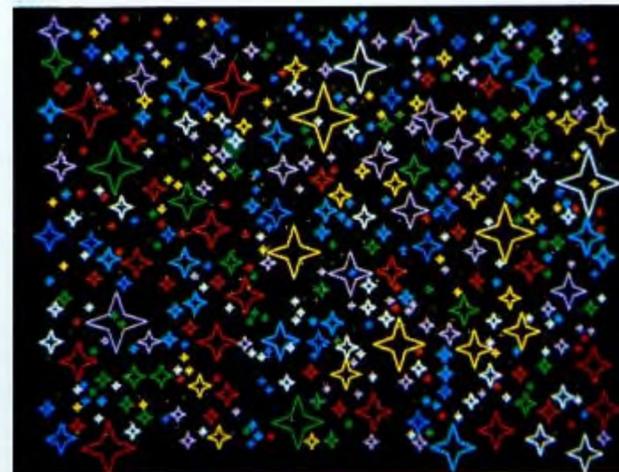
3.21. Techniques de mise au point

- ★ 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.

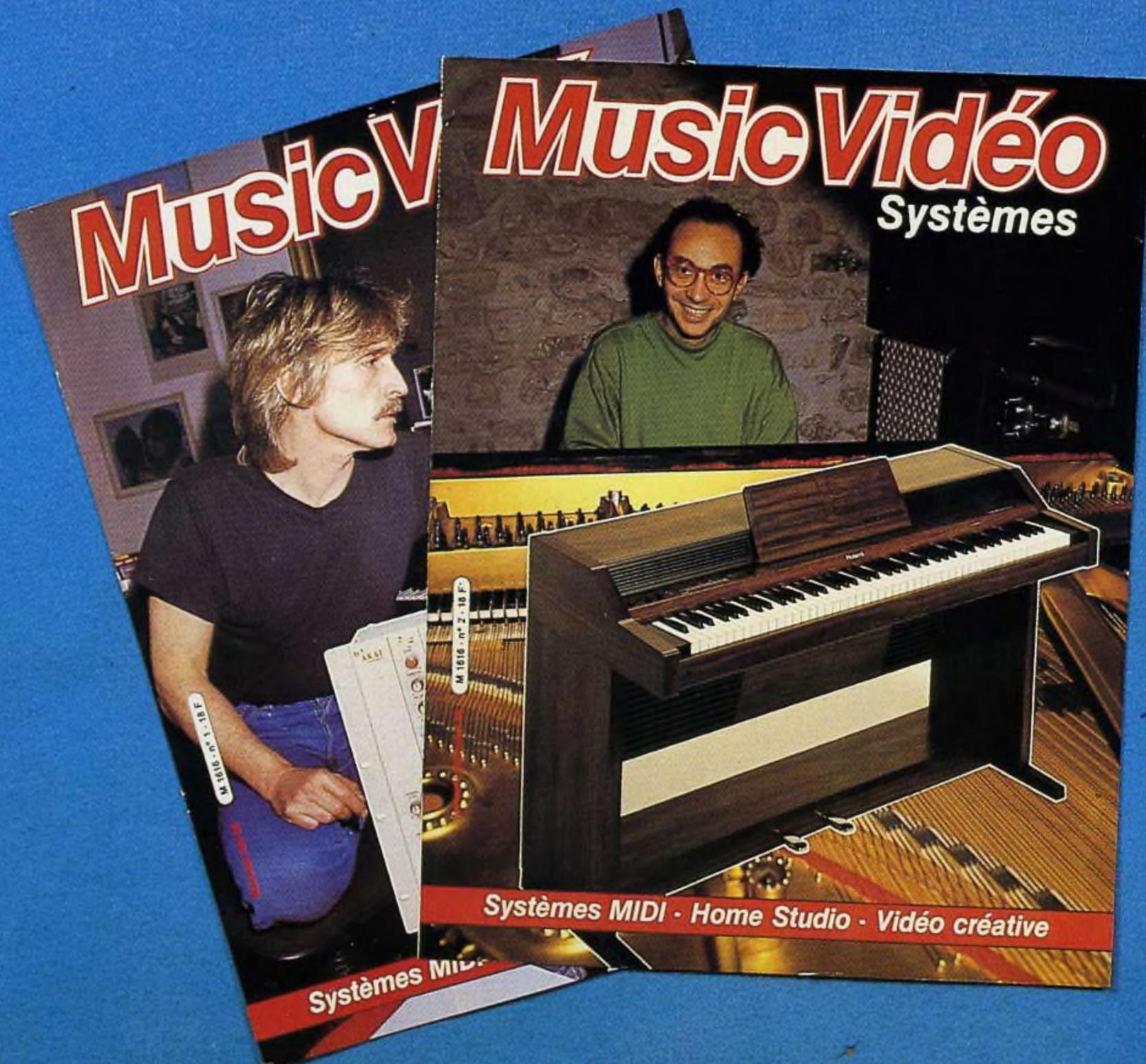
3.22. Problèmes de synthèse - Notions d'analyse

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