

LOISIRS TECHNIQUES D'AUJOURD'HUI

hors série

Leed

MICRO

PROGRAMMATION

COURS 2^{ème} CYCLE

ISSN 0757-6889

COURS
N°27

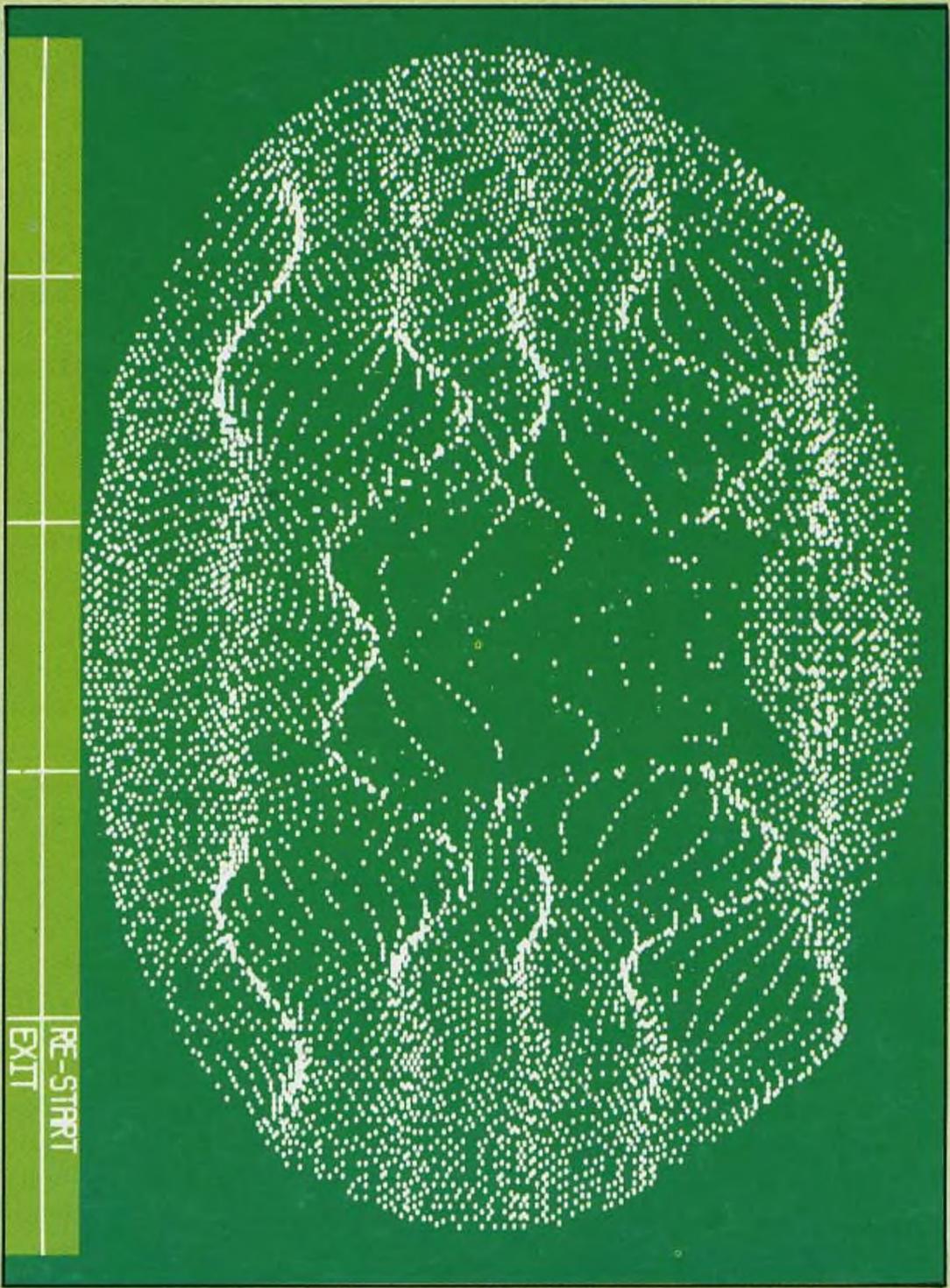
Suite
2^o cycle

N°7

**COURS DE
BASIC :**
révision générale

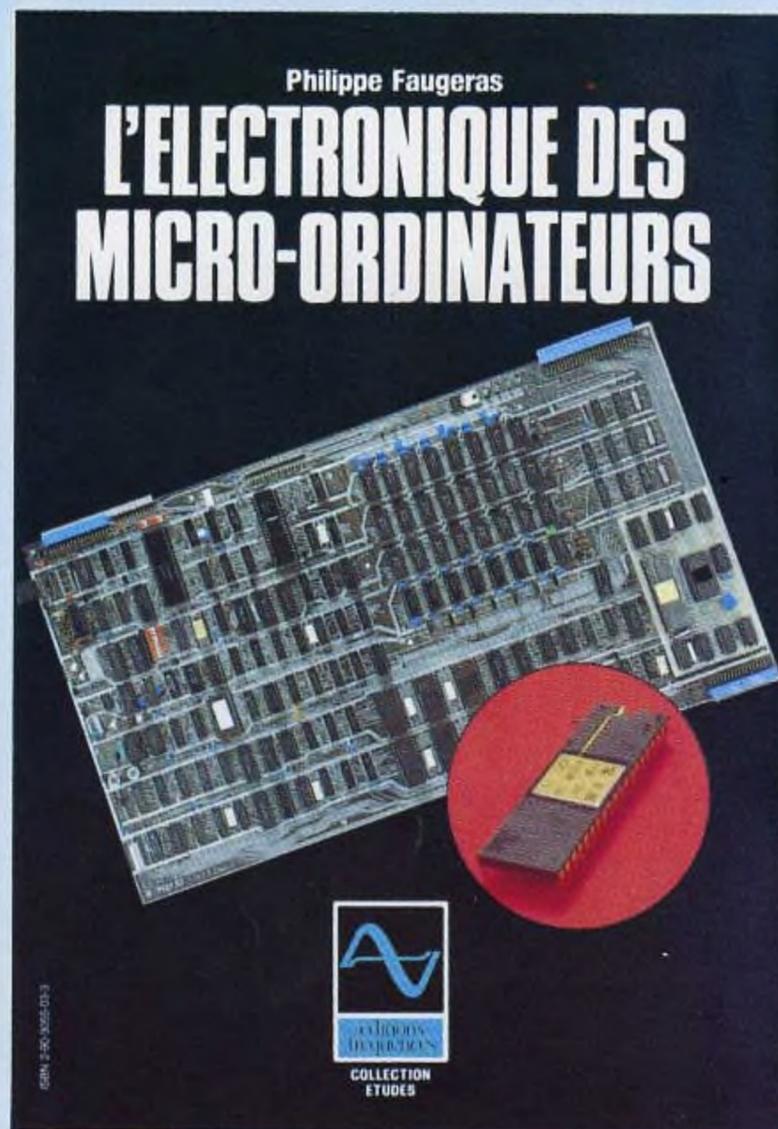
**COURS DE
PROGRAM-
MATION
APPROFONDIE :**
utilisation des
arbres binaires

**COURS DE
GENIE LOGICIEL :**
de la théorie
à la pratique



VOYAGE AU COEUR DES MICRO-ORDINATEURS

dans la
COLLECTION
«ETUDES»
aux
éditions
fréquences



une véritable schémathèque

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

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

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

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

En vente chez votre libraire et aux Editions Fréquences

BON DE COMMANDE

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

Nom

Adresse

.....

A adresser aux EDITIONS FREQUENCES 1 boulevard Ney, 75018 Paris

Règlement ci-joint :

Par chèque bancaire par chèque postal par mandat

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

LOM...QUES D'AUJOURD'HUI

hors série

LED

MICRO

PROGRAMMATION COURS 2^e CYCLE

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

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

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

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

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

Secrétaire de Rédaction
 Chantal Cauchois

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

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

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

FEVRIER 86



COURS DE BASIC
 de la page 4 à la page 11
**Dominique Chastagnier Jean-François
 Coblentz Patrick Gueneau**

**COURS DE PROGRAMMATION
 APPROFONDIE**
 Utilisation des arbres binaires
 de la page 14 à la page 27

- A quoi sert une pile ? p. 15
 - La pile «opérationnelle» des calculatrices
 - La pile du micro-processeur
 - La pile, structure dynamique pour l'allocation de mémoire
- La récursivité p. 17
 - Un premier exemple : calcul de N!
 - Présentation
 - Fonctionnement
 - Application à un programme Basic.
- Tri par arbres binaires p. 20
 - Génération de l'arbre
 - Relecture de l'arbre
 - Détails de fonctionnement
 - Création de l'arbre
 - Parcours de l'arbre
 - Le programme
- Exercices d'application p. 24

- Le slalom
 - Elaboration des structures à tous les niveaux
- Les tours de Hanoï
 - Présentation
 - Programmation
- Les dames en prises
- Les nombres parfaits
- Point sur les exercices précédents.

**Dominique Chastagnier Jean-François
 Coblentz Patrick Gueneau**
C'EST ARRIVÉ DEMAIN
 de la page 30 à la page 31
COURS DE GENIE LOGICIEL
 De la théorie à la pratique
 de la page 33 à la page 48

- Les outils graphiques p. 33
- Les opérations graphiques p. 34
 - les dimensions
 - les surfaces
 - les opérations mathématiques
 - les langages graphiques
 - le matériel graphique
 - les ordres graphiques simples
- Rotation, translation, échelle... p. 35
- Les outils graphiques p. 36
 - les écrans
 - les tables traçantes
 - les imprimantes graphiques
 - les light-pens
 - les tables à digitaliser
 - les souris
 - les clés de fonctions
 - les écrans tactiles
 - le clavier
- Le dessin en trois dimensions... p. 38
- Graphique 3D
 - Rotation axe des X p. 39
 - Rotation axe des Y p. 41
 - Rotation axe des Z p. 42
 - Chargement d'échelle p. 43
 - Translation p. 44
 - Perspective p. 45
- Le graphisme et la mécanique... p. 47
- Déroulement du programme p. 48

Charles-Henry Delaleu

NOTRE COUVERTURE : Grâce aux fonctions trigonométriques, il est possible en graphique de créer toutes sortes de dessins.

COURS DE BASIC

Dominique Chastagnier
Jean-François Coblenz
Patrick Gueneau

1. INTRODUCTION

Aujourd'hui commence la dernière partie du cours de Basic niveau 1. Il s'agit d'une révision de toutes les commandes, avec des exemples concis, destinée à vous fournir une référence courte où vous saurez trouver tous les renseignements concernant les commandes Basic. Encore une fois, le manque de standardisation nous oblige à rester à un niveau général, mais il vous sera très facile de compléter par les commandes propres à votre système chéri.

2. LES COMMANDES CONCERNANT LES VARIABLES

Les variables peuvent être de deux grands types en Basic. Il y a les variables numériques et les variables alpha-numériques qui sont les variables contenant un mot ou des mots. Les variables numériques peuvent être :

- entières. On les distingue par le symbole % terminant le nom de la variable : A% est une variable entière. Une variable entière est une variable qui ne peut contenir que des nombres ne comportant pas de chiffre après la virgule. 1 234 est un nombre entier, et 1 234,56 est un nombre décimal car il présente des chiffres après la virgule ;
- réelles. Ce sont les variables les plus générales, ce qui explique qu'il ne soit pas nécessaire de mettre un symbole particulier pour les reconnaître. Lorsqu'une variable ne présente pas de suffixe particulier, c'est une variable réelle, c'est-à-dire décimale (on peut aussi ajouter le suffixe !)
- double précision. Caractérisées par le symbole # en général, ces variables sont stockées sur plus de place mémoire, pour permettre de ne pas perdre trop de précision. Pour des détails sur les pertes de précision, voir les cours de niveau 2. Sachez tout de même que le travail avec des variables simples précisions peut rapidement amener des erreurs importantes en manipulant des calculs, mais que les variables doubles précisions prennent deux fois plus de place en mémoire.

Les variables alpha-numériques :

Elles se terminent par le symbole \$ et permettent de stocker des mots, des phrases, des lettres et même des nombres, qui sont alors considérés comme des mots. Vous pouvez donc y mettre tout. Attention à la place mémoire.

Il faut bien se souvenir que des variables se terminant par un suffixe différent sont différentes. Par exemple, A et A\$ sont deux variables qui n'ont rien en commun. Mais il

faut aussi se souvenir qu'en général seules les premières lettres de la variable sont retenues par le système (au minimum deux lettres). Donc, AT et ATCHOUM représentent la même variable, même si vous ne le voulez pas. Par là même, AT\$ et ATCHOUM\$ seront la même variable, car le suffixe est pris en compte à part. Un résumé suit dans le tableau ci-dessous :

A%	nombre entier	A% = 3425
A	nombre réel	A = 3425.1231
A#	double précision	A# = 1000000.99999999
A\$	alpha-numérique	A\$ = " IL EST 12 H 15 "

Types de variables en Basic.

Certains Basic acceptent en plus les variables en octal, mais elles sont immédiatement transformées :

```
10 A = &247 : PRINT A
```

donne :

```
167
```

Il en va de même pour l'héxadécimal, avec :

```
10 A = &H9A : PRINT A
```

donne :

```
169.
```

Un autre domaine important de la manipulation des variables est la priorité des opérations les unes par rapport aux autres. Les opérateurs sont :

- () les parenthèses sont les délimiteurs des expressions. Un exemple :

$$A = (2 * 8 + 3 \wedge 8) * 6B + 3 * C$$

- **Exponentielles** : ceci correspond à la mise à la puissance. Un exemple :

```
10 A = 3
```

```
20 X = 6
```

```
30 A = A \wedge X
```

Certains systèmes utilisent la flèche \uparrow au lieu du chapeau \wedge pour indiquer la mise à la puissance. Enfin, il faut savoir que la plupart des petites imprimantes ne reconnaissent pas le symbole spécial que représente l'exponentiation et donne un résultat bizarre sur papier, différent de ce que vous auriez pu attendre.

- **Les opérations de base** +, -, *, /.

- **La division entière** \, existe sur quelques systèmes, et donne la partie entière de quotient d'une division. Un exemple :

```
10 A = 7
20 B = 3
30 PRINT A \ B
```

donnera la sortie :

2.

- **Le modulo** : c'est l'opération complémentaire de la précédente, donnant le reste de la division entière. Avec les variables précédentes, vous aurez :

```
40 PRINT A MOD B
```

donnant

1.

$$A = B * Q + R$$

→ Reste obtenu par A MOD B

→ Quotient obtenu par A \ B

Fonctionnement de la division entière.

- **Les opérateurs logiques** : ce sont les opérateurs du type <, >, =, <=, >=, <>, NOT, AND, OR, XOR, EQV, IMP... Ces opérateurs donnent un résultat de type booléen, c'est-à-dire correspondant à oui ou non, vrai ou faux. La traduction est 0 pour faux, et un autre nombre pour vrai. Selon les systèmes, ce second nombre est 1, 2 ou -1. Ils peuvent être utilisés dans des tests, mais aussi dans des formules. Ainsi,

$$A = 3 * (5 < 3)$$

est égal à 0, car $5 < 3$ est faux. Cette écriture est équivalent à :

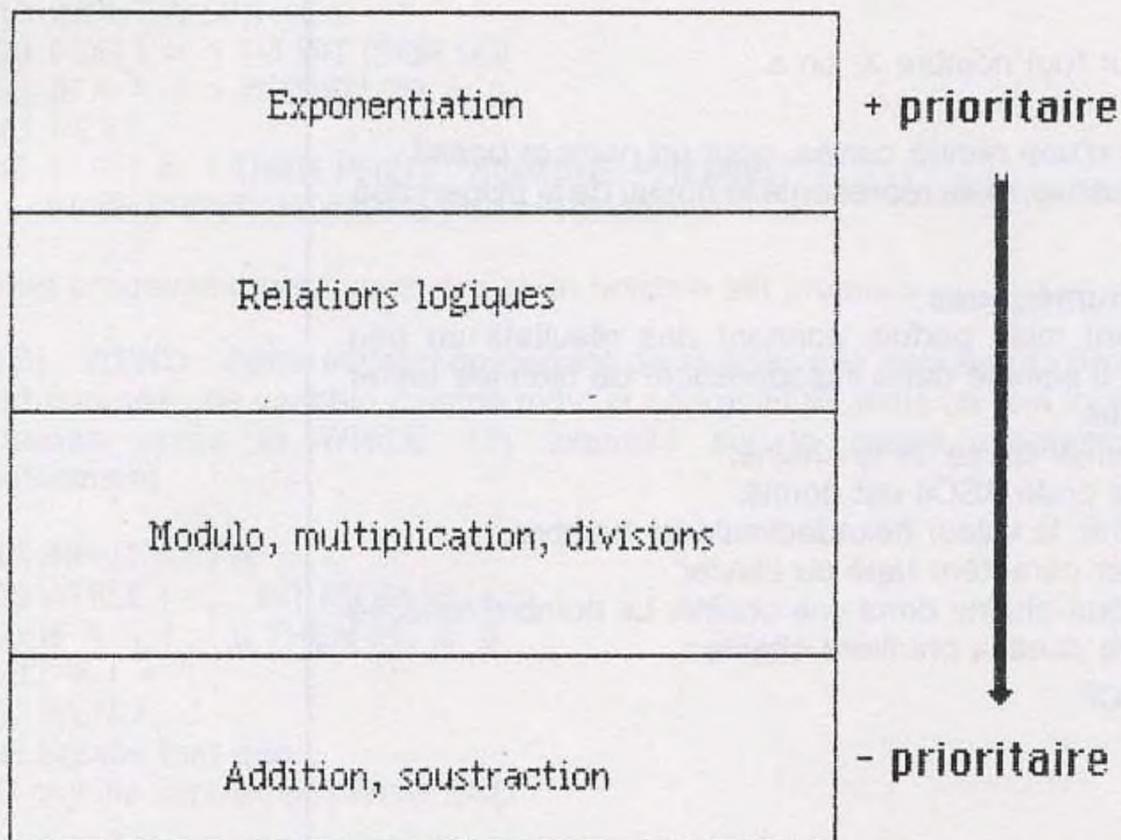
```
10 IF 5 < 3 THEN A = 0
20 IF 5 > 3 THEN A = 3 (ou 6 ou -3 selon les distinctions faites avant).
```

Etudions maintenant la priorité des opérateurs. Cette priorité a pour but de permettre à l'ordinateur de décider comment effectuer les opérations si vous ne mettez pas toutes les parenthèses. Ainsi, les deux expressions suivantes sont équivalentes, mais la seconde est plus complète :

```
10 A = 10 * 3 + 4 \ 2 * (4 > X) - 1/3 + 16
10 A = (10 * 3) + (4 \ 2) * (4 > X) - (1/3) + 16
```

Ceci est possible grâce aux priorités.

Le tableau suivant vous donne l'ordre utilisé par le Basic.



Les fonctions intrinsèques arithmétiques

Les plus courantes sont :

- **ABS** : donne la valeur absolue d'un nombre x , c'est-à-dire le plus grand nombre de l'ensemble $(x, -x)$.
- **ATN** : donne la valeur de l'arctangente d'un nombre.
- **CINT, CSNG, CDBL** : transforment un nombre en entier, en simple précision ou en double précision respectivement. Cette transformation est locale et momentanée. Attention aux erreurs dues à la conversion. Un exemple fameux :

```
10 A = 0.1
20 B# = CDBL(A)
30 PRINT A,B#
```

donne :

```
0.1 0.999999991853027
```

En conclusion, évitez les mélanges.

- **COS, SIN, TAN**, donnent les valeurs des fonctions trigonométriques usuelles.
- **EXP** : donne la valeur de $E \wedge X$, où E est le nombre de Neper (E vaut approximativement 2.71828).
- **FIX** et **INT** : donnent respectivement le nombre entier le plus proche et le nombre entier **inférieur** le plus proche :

```
10 A = -3.01
20 B = 3.99
30 PRINT FIX(A), FIX(B), INT(A), INT(B)
```

donne :

```
-3 4 -4 3
```

- **LOG** : renvoie la valeur du logarithme népérien du nombre.
- **RND** : donne un nombre aléatoire. Attention à l'exactitude de cette fonction qui laisse souvent à désirer.
- **SGN** : renvoie le signe du nombre, sous la forme :
 - 1 si le nombre est négatif,
 - 1 si le nombre est positif,

0 si le nombre est positif.

on peut résumer en disant que pour tout nombre X, on a :

$$X = \text{SGN}(X) * \text{ABS}(X)$$

- **SQR** : permet de calculer la valeur d'une racine carrée, pour un nombre positif.
- Cette liste de fonctions est non-exhaustive, mais représente le noyau de la plupart des systèmes actuels.

Les fonctions intrinsèques alpha-numériques :

Elles sont présentes le plus souvent mais parfois donnent des résultats un peu différents dans des cas particuliers. Il semble donc indispensable de bien les tester pour connaître leurs réactions exactes.

- **ASC** : donne le code ASCII du premier terme de la chaîne.
- **CHR\$** : donne le caractère dont le code ASCII est donné.
- **HEX\$** : donne sous forme de chaîne, la valeur hexadécimale du nombre.
- **INKEY\$** : prend la valeur du dernier caractère tapé au clavier.
- **INSTR** : donne la position d'une sous-chaîne dans une chaîne. Le nombre renvoyé est la position du premier caractère dans la première chaîne :

```
10 X$ = "COUCOU, ME VOICI"
```

```
20 Y$ = "VOI"
```

```
30 PRINT INSTR(X$, Y$)
```

donne

```
12      position du V dans X$.
```

- **LEFT\$, RIGHT\$, MID\$** : permettent de ne travailler qu'avec une partie de la chaîne, débutant respectivement à gauche, à droite, n'importe où, dans la chaîne originale.
- **LEN** : donne le nombre de caractères de la chaîne.
- **SPACES\$** : donne une chaîne de X espaces, X fourni en paramètres.
- **STR\$** : traduit un nombre en une chaîne formée par le nombre.
- **TAB** : permet de décaler de X caractères vers la gauche.
- **VAL** : inverse de STR\$, donnant un nombre à partir d'une chaîne contenant ce nombre.
- **PRINT USING** : permet de définir un format pour les sorties écran et papier de vos chaînes de caractères et de nombres. Il s'agit de la commande la plus complète du BASIC, et le détail de ses possibilités a déjà été vu. Son utilisation est simple et performante pour des présentations propres.

3. LES BRANCHEMENTS ET LES BOUCLES EN BASIC

Les boucles et branchements sont les commandes permettant de se promener dans un programme sans être contraint à une exécution linéaire du programme et permettant de réaliser des exécutions réfléchies de ce programme. Les commandes sont en petit nombre. Elles sont :

- **GOTO** : permet de se rendre à la ligne indiquée. Cette commande est commode mais est un véritable piège pour programmeur. Il est en effet très facile de se perdre dans une série de GOTO, donc de réaliser des programmes « fous », car incontrôlables.
- **GOSUB** : équivalent au GOTO, mais avec une différence notable. Par un GOSUB, vous entrez dans une structure nouvelle, le sous-programme, dont vous ressortirez par un RETURN. Ceci oblige à de la rigueur, contrairement à GOTO, donc nous ne saurions trop vous la recommander, et de fuir les GOTO comme la peste.
- **IF** : réalise le test de la relation qui le suit immédiatement et, si le test est réussi, exécute l'instruction suivant le THEN qui y est attaché. Certains systèmes possèdent de plus l'instruction ELSE qui permet d'indiquer quelle instruction ou série d'instructions réaliser si le test échoue.
- **FOR... NEXT** : permet d'exécuter une série d'instructions un certain nombre de fois, ce nombre étant fixé par l'utilisateur, ou calculé par le programme directement. Un exemple :

```

10 INPUT A : PR = 1
20 FOR I = 1 TO INT (SQR (A))
30 IF A \ I = 0 THEN PR = 0
40 NEXT
50 IF PR = 1 THEN PRINT "NOMBRE PREMIER"
   ELSE PRINT "NOMBRE NON PREMIER"

```

Ce court programme peut vous dire si un nombre est premier.

- **WHILE... WEND** : cette instruction permet de réaliser une séquence d'instruction suivant non pas une variable comme avec la boucle FOR, mais un test logique, qui est réalisé après le WHILE. Un exemple sur le même programme que précédemment :

```

10 INPUT A : PR = 1
20 WHILE I < INT (SQR(A)) + 1
30 IF A \ I = 0 THEN PR = 0
40 I = I + 1
50 WEND

```

WHILE signifie tant que.

WEND signifie simplement While END.

- **REPEAT... UNTIL** : Cette instruction est la même que WHILE WEND, mais le test n'est pas en début de boucle, mais en fin. Cette nuance est très souvent utile, car elle permet d'exécuter la série d'instructions une fois au moins, ou une fois de plus, puisque le test n'est réalisé qu'après exécution. Le programme précédent devient :

```

10 INPUT A : PR = 1
20 REPEAT
30 IF A \ I = 0 THEN PR = 0
40 I = I + 1
50 UNTIL I = INT (SQR(A))

```

REPEAT signifie refaire. UNTIL signifie jusqu'à ce que.

4. LES TABLEAUX

Un tableau est une structure formée d'un ensemble de variables de même type, réunies sous un même nom, et repérées par un indice dans la structure. Un tableau peut être à une ou plusieurs dimensions (souvent le nombre maximal de dimension est supérieur à 80). Les dimensions sont proposées dans un ordre qui doit être conservé tout au long de l'utilisation. Ainsi, pour un tableau TAB(3,3), la case 2,3 est différente de la case 3,2. Pour le reste, un tableau suit toutes les propriétés des variables simples. Un tableau est déclaré par la commande :

```
10 DIM TAB (2,3,4,2,3,4)
```

DIM est l'instruction de déclaration. Il est généralement impossible de déclarer deux fois un tableau.

OPTION BASE permet de faire débiter la numérotation des cases à 1 et non à 0 comme cela se fait autrement. Le but est de ne pas charger la mémoire avec des cases qui, le plus souvent, ne sont pas utilisées, la logique voulant que la première valeur d'une série soit stockée en case 1 et non 0. Le gain dû à OPTION BASE peut être important, comme dans le cas ci-dessous. Il s'agit d'un tableau 20 x 20, allant de 0 à 20 dans chaque sens. Avec OPTION BASE 1, 41 cases économisées, soit plus de 200 octets pour un tableau de réels.

6. LES COMMANDES D'INTERFAÇAGE AVEC LE SYSTEME

Elles sont de deux types : les commandes d'accès direct à la mémoire ou à des routines externes et celles qui vous permettent de gérer vos programmes.

Gestion de la mémoire :

CALL : autorise l'accès à un programme en langage machine (la syntaxe est très variable d'un BASIC à l'autre).

PEEK : pour lire le contenu d'une case mémoire.

POKE : pour écrire une valeur (1 octet) dans une case mémoire.

VARPTR : permet d'obtenir la position en mémoire du début d'une variable, c'est-à-dire du premier octet où se trouve stockée cette variable.

Gestion des programmes BASIC :

CHAIN : chaînage de programmes avec conservation des valeurs des variables.

FILES : pour lister le catalogue de la disquette.

LOAD : chargement d'un programme en mémoire.

MERGE : fusion du programme avec un autre sur disque.

SAVE : sauvegarde d'un programme sur cassette, disquette voire disque dur.

SYSTEM : pour revenir au système d'exploitation (ss MS-DOS, etc.),

et sûrement beaucoup d'autres qui sont intimement liées à votre machine.

CONCLUSION

Avec cette révision, vous voilà à même de faire face à tout type de programme. Entraînez-vous sur ceux donnés le mois dernier, et vous vous rendrez compte qu'il n'est pas du tout impossible de programmer avec rigueur et plaisir, conjointement. Voici un rappel des programmes proposés lors du précédent numéro.

1. Proposer un fichier FICHMOT, dont les mots puissent avoir plus de huit lettres, et dans ce cas, huit lettres de ces mots au hasard.
2. Séparer les voyelles des consonnes, pour la création de mots suivant la règle originelle du jeu.
3. Réaliser un jeu de pendu, à partir du fichier FICHMOT.
4. Proposer des programmes de tri. Plusieurs méthodes ont été décrites, mais toutes autres seraient appréciées.
5. Créer un programme de traduction de l'alphabet Morse, et qui fonctionne dans les deux sens.
6. Réaliser un programme qui permette de visualiser une série de nombres par un graphique de type colonnes proportionnelles, ou camembert, ou autre.

Nous commencerons à les réaliser la fois prochaine, avec vos propres solutions ou au moins une analyse de chaque situation.



Claude Polgar est né en 1926 à Paris. Ingénieur de l'Ecole Centrale de Paris, il fut ingénieur d'études chez Kodak-Pathé, chez Renault-Machine-Outils et aux machines Bull puis chef de département aux engins Matra. Parallèlement à cette carrière classique d'ingénieur, Claude Polgar a poursuivi des recherches personnelles en créant en 1954 le matériel Prototypia (qui fut le premier «Meccano» de micro-robotique) et en 1982 le logiciel d'habillement Alamod (qui permet de réaliser des patrons personnalisés). Claude Polgar se consacre actuellement à l'enseignement des techniques modernes. Les Editions Fréquences ont publié son cours de programmation dans la revue Led-Micro.

**2 volumes (près de 500 pages - format 21 x 27)
représentant le récapitulatif de 2 ans des cours progressifs
de Claude Polgar**

DE NOMBREUX ADDITIFS

Que de changements depuis la sortie
du numéro 1 de LED-MICRO !

Il n'est plus possible d'ignorer :

- le MS-DOS (le système d'exploitation de l'IBM PC)
- les Mémoires à Bulles
- le Compact-Disc
- le développement du Minitel et des réseaux de télématique amateur
- les notions de base de l'Intelligence artificielle (ce qu'est PROLOG etc...)
- l'emploi des calculettes aux examens.

J'ai profité de cette réédition pour ajouter des exercices, mieux présenter certains thèmes, donner aux professeurs le moyen de préparer des disquettes autochargeables.

Que voulez-vous ? C'est ma nature !

C. POLGAR

le cours d'initiation à la micro-informatique le plus complet

non, on ne s'initie pas à la micro-informatique et au basic en 5 leçons ou en 3 semaines !

Le mythe de l'informatique loisir facile s'est envolé, accéder à la programmation relève d'une pédagogie sérieuse et progressive, c'est le pari gagné que fit Led-Micro à une époque où fleurissait chaque jour un nouvel ouvrage-miracle.

Parmi les centaines de lettres reçues, nous nous permettons de citer 3 d'entre elles, elles permettent de situer comment, en général, a été perçu et apprécié ce cours.

J'enseigne les mathématiques dans une Université de Sciences Humaines et j'ai été amenée, alors que je 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 nos 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

J'ai déjà essayé, à deux reprises au moins, antérieurement, de me familiariser vraiment avec le BASIC sans grand résultat, je l'avoue. La méthode que vous mettez en œuvre dans «Led-Micro» — me conduira-t-elle au but recherché, je n'en sais rien encore — a du moins le mérite d'être sympathique et agréable à suivre. Ma seule ambition étant d'utiliser les micros comme distrac-

tion intellectuelle (je suis retraité), j'espère ainsi y parvenir.

Merci, donc, de votre aide et continuez à nous faire avancer progressivement et sûrement.

Docteur Y.C. Sees, le 19/2/84

Je viens de découvrir votre magazine ce matin dans un kiosque, cet après-midi je vous commande les 18 premiers numéros.

Je suis très emballé par vos cours, que je trouve très bien faits.

Je suis un «vrai» débutant, je possède un ZX81 que j'ai du mal à faire tourner, par manque d'information, grâce à vos cours je pense que j'y arriverais. Je possède pas mal de bouquins sur la question mais aucun n'explique aussi clairement que vous.
A.A. Marseille, le 17/4/85

en vente chez votre libraire ou aux Editions Fréquences (collection pédagogique).

Initiation à la micro-informatique C. Polgar

En vente chez votre libraire ou aux Editions Fréquences 1, bd 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)
les deux tomes 280 F (260 F + 20 F de frais de port)

Je joins mon règlement à la commande :

chèque bancaire

mandat

C.C.P.

Nom

Prénom

Adresse

Code postal

Localité



COURS DE PROGRAMMATION APPROFONDIE

Dominique Chastagnier
Jean-François Coblentz
Patrick Gueneau

COURS N° 7

Utilisation des arbres binaires

PLAN DU COURS

1. A quoi sert une pile ?
 - 1.1. La pile « opérationnelle » des calculatrices
 - 1.2. La pile du micro-processeur
 - 1.3. La pile, structure dynamique pour l'allocation de mémoire

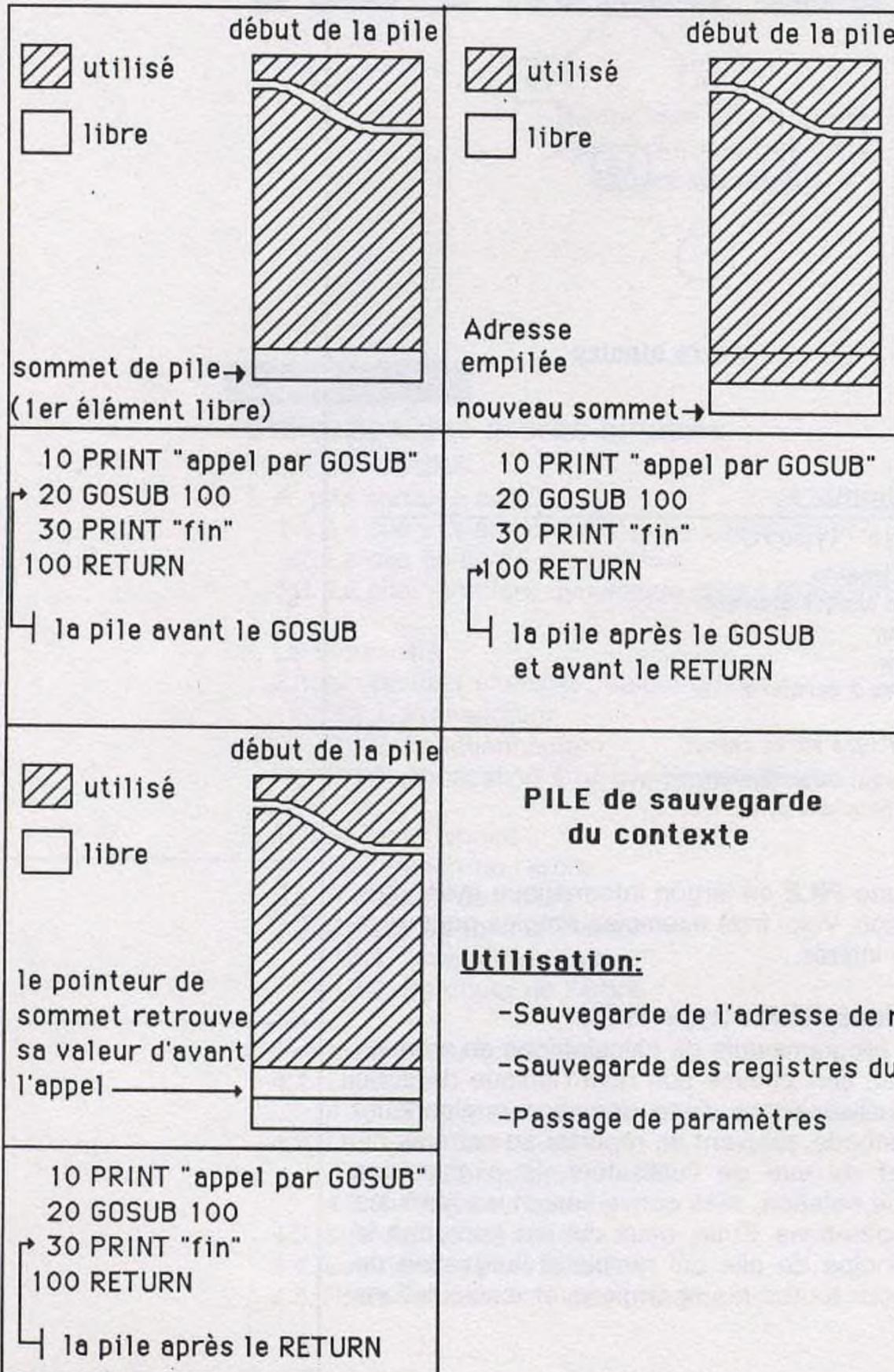
2. La récursivité
 - 2.1. Un premier exemple : calcul de $N!$
 - 2.1.1. Présentation
 - 2.1.2. Fonctionnement
 - 2.1.3. Application à un programme Basic

3. Tri par arbre binaire
 - 3.1. Génération de l'arbre
 - 3.2. Relecture de l'arbre
 - 3.3. Détails de fonctionnement
 - 3.3.1. Création de l'arbre
 - 3.3.2. Parcours de l'arbre

4. Exercices d'applications
 - 4.1. Le slalom
 - 4.1.1. Elaboration des structures à tous les niveaux
 - 4.2. Les tours de Hanoi
 - 4.2.1. Présentation
 - 4.2.3. Programmation
 - 4.3. Les dames en prises
 - 4.4. Les nombres parfaits
 - 4.5. Point sur les exercices précédents

<table border="1"> <tr><td>T</td><td>10.1</td></tr> <tr><td>Z</td><td>0.0</td></tr> <tr><td>Y</td><td>0.0</td></tr> <tr><td>X</td><td>0.0</td></tr> </table> <p>au départ la valeur de T sera perdu au prochain ENTER</p>	T	10.1	Z	0.0	Y	0.0	X	0.0	<table border="1"> <tr><td>T</td><td>0.0</td></tr> <tr><td>Z</td><td>0.0</td></tr> <tr><td>Y</td><td>0.0</td></tr> <tr><td>X</td><td>2</td></tr> </table> <p>ENTER empilage d'une valeur (au moment de l'empilage, il y a duplication de X)</p>	T	0.0	Z	0.0	Y	0.0	X	2	<table border="1"> <tr><td>T</td><td>0.0</td></tr> <tr><td>Z</td><td>0.0</td></tr> <tr><td>Y</td><td>2.0</td></tr> <tr><td>X</td><td>27.1</td></tr> </table> <p>27.1 + (l'opération provoque un empilage de la valeur entrée, ici 27.1)</p>	T	0.0	Z	0.0	Y	2.0	X	27.1	<table border="1"> <tr><td>T</td><td>0.0</td></tr> <tr><td>Z</td><td>0.0</td></tr> <tr><td>Y</td><td>0.0</td></tr> <tr><td>X</td><td>29.1</td></tr> </table>	T	0.0	Z	0.0	Y	0.0	X	29.1
T	10.1																																		
Z	0.0																																		
Y	0.0																																		
X	0.0																																		
T	0.0																																		
Z	0.0																																		
Y	0.0																																		
X	2																																		
T	0.0																																		
Z	0.0																																		
Y	2.0																																		
X	27.1																																		
T	0.0																																		
Z	0.0																																		
Y	0.0																																		
X	29.1																																		

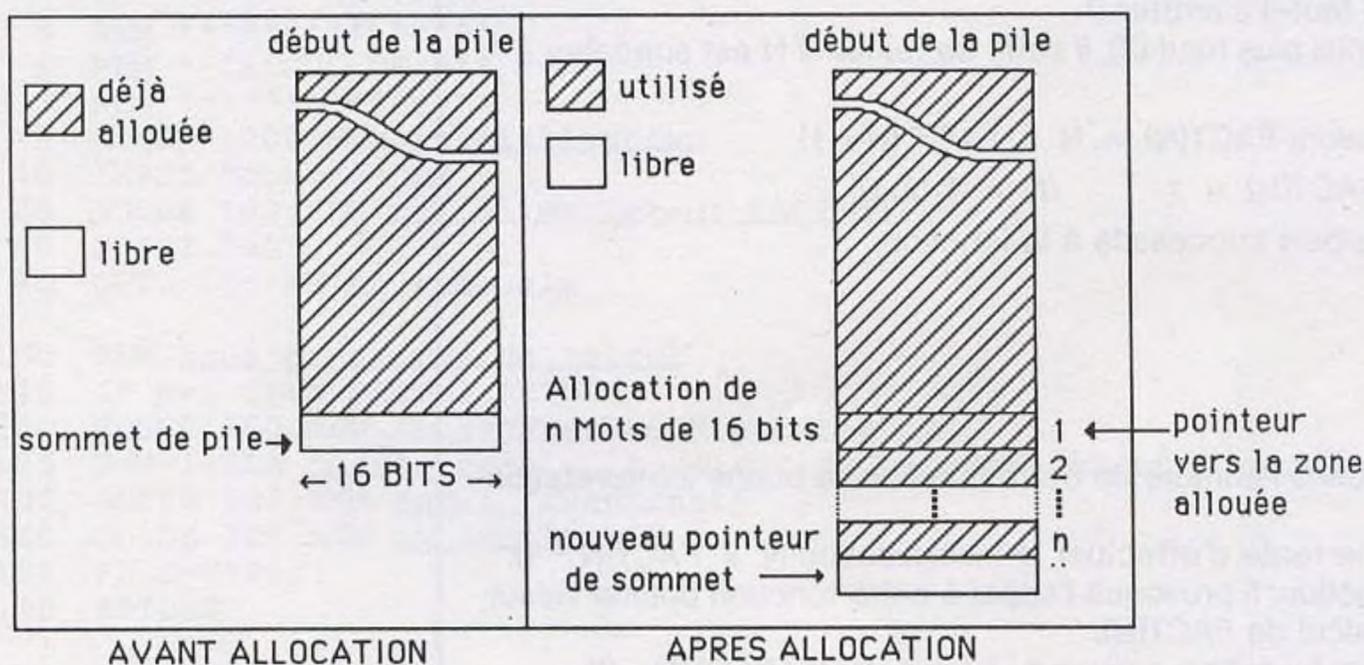
1.2. La pile du micro-processeur



Dans ce cas, la pile sert à mémoriser l'état du micro-processeur avant l'appel à un sous-programme afin de le restituer au retour de ce sous-programme. Cette fonction est fondamentale pour le bon fonctionnement du micro-processeur et la pile est l'unique moyen de traiter les branchements avec retour. Tout micro-processeur a une pile qui se situe dans l'espace mémoire de l'ordinateur. On retrouve ce principe sous tous les langages lors de l'appel à un sous-programme, que la pile utilisée soit celle du micro-processeur ou bien une pile dédiée à cette tâche (et donc gérée par programme). En plus de la sauvegarde de l'adresse de retour, la pile sert très souvent à passer des paramètres aux sous-programmes ou fonctions ou encore à sauver le contenu des registres utilisés par ce sous-programme.

1.3. La pile, structure dynamique pour l'allocation de mémoire

C'est une généralisation du principe précédent, avec l'allocation de zone mémoire sur la pile en fonction des besoins. Le principal avantage de cette gestion réside dans la plus grande souplesse de l'utilisation de l'espace mémoire. Ainsi en BASIC, le programme que nous vous présentons aujourd'hui en est un exemple, il faut déclarer au départ la taille de chaque tableau ce qui impose une limite stricte dans l'utilisation du programme, à savoir ne pas aller plus loin que la déclaration ne le prévoit. L'inconvénient est double puisque la plupart du temps seule une partie du tableau sera utilisée. Dans le cas de l'utilisation d'une pile et lorsque l'environnement l'autorise (c'est-à-dire en PASCAL, C, etc.), il suffit d'allouer la place correspondant à la quantité de données à traiter et il est de plus possible, en fin de traitement mais au sein du même programme, de récupérer pour d'autres usages la place libérée. Nous aborderons ces notions plus en détail lorsque nous débuterons le cours de PASCAL.



2. LA RECURSIVITE

Pour mieux comprendre l'utilisation de cette pile, voici une introduction simple à une notion nouvelle pour la plupart d'entre vous mais si importante au niveau de la programmation et de la gestion de structures de données complexes (en PASCAL, C, etc.) : la **récurtivité**.

2.1. Un premier exemple : calcul de N!

2.1.1. Présentation

Si l'exemple que nous vous présentons ci-dessous est, à n'en pas douter, la programmation la plus maladroite, il constitue toutefois le meilleur moyen de bien assimiler le principe. La formule,

$$N! = N * (N - 1) * (N - 2) * \dots * 2 * 1 \quad (1)$$

peut s'écrire aussi :

$$N! = N * (N - 1)! \quad (2)$$

De cette équation, on déduit qu'il suffit, dans le cas général ($N \geq 2$), de connaître le résultat à l'étape $N - 1$ pour en déduire par la formule (2) le résultat pour N . Il ne reste plus qu'à poser $0! = 1! = 1$ et nous avons le résultat suivant :

$$\begin{aligned} \text{si } N > 1 \text{ alors } N! &= N * (N - 1)! & (3) \\ \text{sinon } N! &= 1. \end{aligned}$$

Supposons maintenant que nous voulions définir une fonction FACT qui, pour le paramètre N renvoie $N!$, en reprenant l'équation, on aurait pour calculer FACT(N) :

$$\text{FACT}(N) = N * \text{FACT}(N - 1).$$

Aussi bizarre que cela puisse paraître, il est possible de définir une telle fonction en PASCAL par exemple et même quelques fois avec certains BASICs. Explorons en détail le cheminement pour une valeur $N = 5$ par exemple :

$$N = 5 \rightarrow \text{FACT}(5) = 5 * \text{FACT}(4)$$

$$N = 4 \rightarrow \text{FACT}(4) = 4 * \text{FACT}(3)$$

on appelle à nouveau la fonction FACT
mais cette fois pour 4
nouvel appel pour $N = 3$

$$\text{(d'où } \text{FACT}(5) = 5 * 4 * \text{FACT}(3))$$

ainsi de suite...

Un problème demeure, quand faut-il s'arrêter ?

Si l'on respecte le principe choisi plus haut (3), il suffit de tester si N est supérieur à 1 ou non. On obtient alors :

$$\begin{aligned} \text{Si } N > 1 \text{ alors } \text{FACT}(N) &= N * \text{FACT}(N - 1) \\ \text{sinon } \text{FACT}(N) &= 1 \quad (N = 1 \text{ ou } 0) \end{aligned}$$

Ce test permet d'arrêter les appels successifs à la fonction.

2.1.2. Fonctionnement

Attention, il y a une ambiguïté dans l'écriture de cette fonction, la bonne interprétation est la suivante :

- Lors de l'appel, le programme tente d'effectuer la multiplication $N * \text{FACT}(N - 1)$.
 - Il reconnaît en FACT une fonction. Il provoque l'appel à cette fonction pour la valeur $(N - 1)$, ce qui suspend le calcul de FACT(N).
 - Le calcul de FACT($N - 1$) sera lui-même suspendu pour calculer FACT($N - 2$).
 - C'est seulement lorsque le programme arrive à 1 qu'il répond directement FACT(1) = 1.
 - Enfin il remonte au calcul de FACT(2) puis de FACT(3), etc. jusqu'à FACT(N).
- Toute cette imbrication implicite n'est possible que grâce au recours d'une pile. Elle est chargée non seulement de sauvegarder l'adresse de retour en cas d'appel à un sous-programme (ici une fonction) mais aussi, c'est une condition nécessaire, tout ce que l'on peut qualifier de contexte du programme au moment de l'appel. En effet, la fonction s'appelant elle-même, le contenu des variables utilisées est changé à chaque appel (il suffit d'observer les différentes valeurs prises par le paramètre N), sans «empilage» des valeurs de ces variables, on perdrait à chaque fois les anciennes valeurs. Voici un exemple :

- Au retour du calcul de FACT(4), il faut récupérer pour N la valeur avant l'appel, c'est-à-dire 5, pour pouvoir effectuer le calcul $5 * \text{FACT}(4)$, sachant que le calcul de FACT(4) affecte à N la valeur 4. Dans cet exemple, seule la valeur de N est à

conserver. Nous utiliserons donc la pile pour sauvegarder N avant l'appel «récuratif» puis pour restituer sa valeur afin d'effectuer le calcul.

2.1.3. Application à un programme BASIC

Nous avons écrit deux versions du programme :

- La première utilise une pile classique représentée sous forme d'un tableau PILE, que l'on suppose suffisamment grand pour ne pas avoir de problème de taille. Le sommet de la pile est pointé par la variable PPILE. Nous avons besoin de deux routines de gestion de cette pile :
 - 1) empilage d'une donnée (un entier).
 - 2) désempilage d'une donnée.
- La deuxième optimise la sauvegarde de la valeur N. Il suffit, au retour de l'appel au sous-programme pour (N - 1), d'ajouter 1 à N. Le contenu de N est donc préservé après l'exécution du sous-programme de la même façon qu'il le serait avec l'empilage-désempilage de la pile. Par contre, cette deuxième version simplifie beaucoup le programme.

Une dernière précision sur l'astuce de l'appel par GOSUB du sous-programme lui-même. Elle n'est efficace que parce que le BASIC gère lui-même une pile de sauvegarde de l'adresse de retour au moment du GOSUB.

programme de calcul de factorielle N: première version

```

5  REM *****
6  REM *  CALCUL DE N!*
7  REM *****
8  GOSUB 1000:REM initialisation
10 INPUT "nombre:";N
20 GOSUB 100:REM calcul, en retour FACT=N!
30 PRINT FACT
40 GOTO 10:REM on reboucle

100 REM sous programme de calcul
110 IF N=1 THEN FACT=1:RETURN
120 GOSUB 200:REM sauvegarde de N avant appel
125 N=N-1:REM on décrémente la valeur du paramtre avant l'appel
130 GOSUB 100:REM appel "réentrant"
140 GOSUB 300:REM on restitue N
150 FACT=N*FACT
160 RETURN

200 REM EMPILAGE
210 PPILE=PPILE+1
220 PILE(PPILE)=N
230 RETURN

300 REM DESEMPILAGE
310 N=PILE(PPILE)
320 PPILE=PPILE-1
330 RETURN

1000 REM init de la pile
1010 DIM PILE(200)
1020 PPILE=0:REM au départ la pile est vide
1030 RETURN

```

programme de calcul de factorielle N: version améliorée

```

5  REM *****
6  REM * CALCUL DE N!*
7  REM *****
10 INPUT "nombre:";N
20 GOSUB 100:REM calcul, en retour FACT=N!
30 PRINT FACT
40 GOTO 10:REM on reboucle

100 REM sous programme de calcul
110 IF N<=1 THEN FACT=1:RETURN
120 N=N-1:REM on décrémente la valeur du paramètre avant l'appel
130 GOSUB 100
140 N=N+1:REM on restitue N, ici on a optimisé car on connaît
145 REM la valeur de N au retour
150 FACT=N*FACT
160 RETURN

```

3. TRI PAR ARBRE BINAIRE

Il nous reste à étudier comment créer l'arbre de manière à trier les données en relecture. Le programme se compose donc essentiellement de ces deux parties.

3.1. Génération de l'arbre

La méthode employée pour ajouter un élément N est la suivante :

1. On part de la racine de l'arbre (dans le programme le début de l'arbre est pointé par ENTREE).
2. S'il n'y a pas de nœud on en crée un, on met à jour le pointeur (la variable LIBRE indique le premier élément du tableau inutilisé. Il faut ensuite mettre à jour le pointeur en amont d'où la sauvegarde de RAC par RACP) et retour.
3. Sinon, suivant la valeur du nœud, on explore soit la partie gauche ($N < \text{nœud}$) soit la partie droite ($N > \text{nœud}$) de l'arbre en recommençant en (2).

On obtient donc un arbre où tous les éléments sont classés d'une telle façon que le sous-arbre de **chaque nœud** ne contient que des valeurs inférieures à ce nœud et le sous-arbre droit que des valeurs supérieures.

3.2. Relecture de l'arbre

C'est dans cette partie que l'on va mettre à profit le principe de récursivité étudié plus haut. Le parcours suit les règles suivantes :

1. s'il n'y a pas de nœud, retour immédiat ;
2. on parcourt le sous-arbre gauche ;
3. on écrit le nœud ;
4. on parcourt le sous-arbre droit ;
5. retour du sous-programme.

Contrairement à la création de l'arbre, il faut mémoriser chaque position du pointeur avant appel (parcours gauche ou droit) afin de pouvoir afficher le contenu du nœud et ensuite passer au sous-arbre droit.

3.3. Détails de fonctionnement

3.3.1. Création de l'arbre

Le principe d'ajout d'un élément à l'arbre respecte la règle suivante :

- il suffit dans la recherche de la position du nouvel élément de mémoriser la dernière valeur du pointeur RAC. On peut ainsi, lorsqu'on arrive au bout d'une branche, mettre à jour le pointeur en amont (soit à gauche soit à droite), à l'aide de RACP.

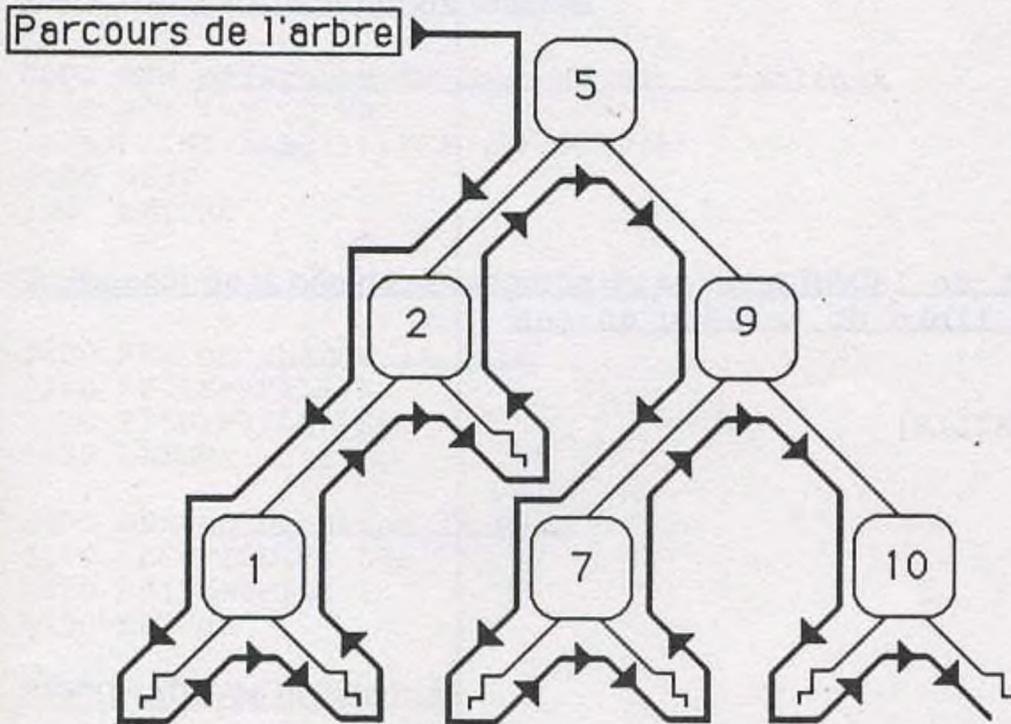
Cette solution est en fait une optimisation analogue à celle utilisée pour la gestion de la récursivité dans l'exemple du calcul de factorielle, car on tient compte de l'information qu'on désire conserver. Cette remarque ne s'appliquerait pas à la programmation en PASCAL du même exercice, la gestion de la pile étant implicite et non modulable.

3.3.2. Parcours de l'arbre

L'utilisation de la pile est, dans ce cas, indispensable car le programme doit pouvoir récupérer à tout moment la situation du pointeur afin de progresser dans l'arbre. De plus, on ne connaît pas la taille des branches.

Par contre, il est tout à fait possible d'optimiser la sauvegarde de ce pointeur, pour limiter les mouvements sur la pile qui sont coûteux en temps d'exécution, et c'est ce que nous avons fait :

- La sauvegarde s'effectue au début de l'appel si le pointeur courant (RAC) n'est pas nul (c'est-à-dire si le nœud existe).
- Si le pointeur est nul, il y a au contraire restitution de la précédente valeur de RAC: En effet, lorsque le sous-arbre gauche d'un nœud a été exploré, le dernier pointeur est nul, il faut donc écrire la valeur du nœud et pour cela retrouver son adresse.
- Contrairement au parcours du sous-arbre gauche, il est sans intérêt de sauver la valeur du pointeur pour le parcours du sous-arbre droit : il ne sert à rien pour le parcours du reste de l'arbre.



TRI PAR ARBRE (Parcours GRD)

Remarques

- Un conseil pour vous permettre de mieux comprendre le cheminement du programme : ajoutez des PRINT à chaque empilage-démpilage pour suivre la trace du parcours de l'arbre. Si besoin, utilisez une sortie brute des valeurs telles qu'elles sont stockées en mémoire à l'aide du sous-programme commençant en 3000. Elle vous permettra de retrouver qui pointe vers quoi.
- Attention, il est fort possible que votre BASIC limite le nombre d'appels imbriqués à des sous-programmes (la pile qu'il utilise étant limitée). Cette limitation a pour simple conséquence de vous interdire le parcours d'un arbre d'une profondeur supérieure à ce nombre maximum. D'où l'avantage de parcours des arbres équilibrés qui optimisent la profondeur pour un nombre de nœuds donné.

3.4. Le programme

PROGRAMME PRINCIPAL:

```

10 REM *****
12 REM *TRI à l'aide d'un arbre binaire de type GRD *
13 REM *****
15 DEFINT A-Z:REM il n'est plus nécessaire d'ajouter le %
17 REM grâce à cette déclaration,toutes les variables sont
18 REM par défaut entières.
20 GOSUB 1000 :REM initialisation
25 NB=0
30 REM debut de la boucle principale
40 INPUT "nouvelle valeur ( 0 pour arrêter)",N$
45 N=VAL(N$)
50 IF N=0 THEN GOTO 80:REM fin de la saisie
60 GOSUB 2000:REM insertion dans l'arbre
65 NB=NB+1
70 GOTO 40:REM fin de la boucle
75 REM affichages de la liste:
80 GOSUB 3000:REM sortie non triée
90 GOSUB 4000:PRINT:REM sortie triée
100 END

```

INITIALISATION:

```

1000 REM initialisation de l'arbre
1005 REM PPILE : est le pointeur du sommet de la pile
1007 REM LIBRE : indique la première case libre du tableau en cas
1008 REM d'ajout d'élément
1010 TAILLE=100:LIBRE=1:PPILE=0
1020 DIM ARBR(TAILLE),SGAU(TAILLE),SDRO(TAILLE)
1025 DIM PILE(200)
1030 FOR i=1 TO TAILLE
1040 ARBR(i)=0:SGAU(i)=0:SDRO(i)=0
1050 NEXT i
1060 ENTREE=0
1070 RETURN

```

Deux sous programmes de parcours de l'arbre:

```

1200 REM descente à droite
1205 REM on sauvegarde l'ancien pointeur RAC
1210 RACP=RAC
1220 RAC=SDRO(RAC)
1230 RETURN

1300 REM descente à gauche
1305 REM 1310 comme la ligne 1210
1310 RACP=RAC
1320 RAC=SGAU(RAC)
1330 RETURN

```

RECHERCHE de la place d'un nouvel élément:

```

2000 REM ajout d'un élément
2005 RACP=0:REM au debut il n'y a pas de valeur sauvegardée
2010 RAC=ENTREE:REM on part du sommet de l'arbre
2020 IF RAC=0 THEN GOTO 2100
2030 IF ARBR(RAC)<N THEN GOSUB 1200:GOTO 2020:REM descente à droite
2040 IF ARBR(RAC)>N THEN GOSUB 1300:GOTO 2020:REM descente à gauche
2050 REM si c'est égal on ne recompte pas l'élément
2060 RETURN
2100 REM ajout d'une feuille, on arrive au bout d'une branche
2110 ARBR(LIBRE)=N:SDRO(LIBRE)=0:SGAU(LIBRE)=0
2120 IF RACP=0 THEN ENTREE=1:GOTO 2140:REM cas particulier du 1er
    NOEUD
2125 REM en fonction du noeud précédant l'insertion, on met à jour à
2126 REM gauche ou à droite
2130 IF ARBR(RACP) > N THEN SGAU(RACP)=LIBRE ELSE SDRO(RACP)=LIBRE
2140 LIBRE=LIBRE+1
2150 RETURN

```

AFFICHAGE de la structure interne:

```

3000 REM affichage du contenu des 3 tableaux
3010 FOR i=1 TO NB
3020 PRINT ARBR(i),SGAU(i),SDRO(i)
3030 NEXT
3040 RETURN

```

Traitement de la pile de sauvegarde du pointeur RAC:

```

3200 REM on charge la pile
3210 PPILE=PPILE+1
3220 PILE(PPILE)=RAC
3230 RETURN

3300 REM on décharge la pile
3310 RAC=PILE(PPILE)
3320 PPILE=PPILE-1
3330 RETURN

```

AFFICHAGE de la liste triée

```

4000 REM tri par parcours GRD
4005 RAC=ENTREE
4010 IF RAC=0 THEN GOSUB 3300:RETURN:REM on dépile
4020 GOSUB 3200:REM on empile
4025 REM on parcourt le sous-arbre gauche
4030 RAC=SGAU(RAC):GOSUB 4010
4040 REM on écrit le noeud
4050 PRINT ARBR(RAC); ", ";
4055 REM enfin, on parcourt le sous-arbre droit
4060 RAC=SDRO(RAC):GOSUB 4010
4070 RETURN

```

4. EXERCICES D'APPLICATION

4.1. Le Slalom

Dans cette période hivernale, le ski fait partie des sports populaires, aussi nous vous proposons de faire un programme gérant le déroulement d'une compétition de ski.

4.1.1. Elaboration des structures à tous les niveaux

Il convient de faire les choix de structures à employer pour représenter les concurrents en fonction de ce que l'on souhaite traiter comme informations les concernant. Dans le cas qui nous intéresse, les opérations suivantes semblent nécessaires :

- l'enregistrement de l'heure de départ,
- l'enregistrement de l'heure d'arrivée,
- la différence des deux, consécutivement, une fonction d'horloge,
- une élimination pour chute,
- une élimination pour dépassement de temps,
- un classement.

Il convient, maintenant de déterminer quel type de structure vous allez employer pour gérer votre ensemble de concurrents. Pour cela, nous vous conseillons de revoir l'ensemble du cours avant de vous prononcer, la spécification des fonctions nécessaires précédemment établie se révélant très utile. Vous notez que le protocole de manipulation de vos données est de type Premier Entré-Premier Sorti (en effet, il est exceptionnel qu'un skieur en dépasse un autre (!) et l'on peut, de toutes façons, considérer cela comme éliminatoire). D'autre part, une fois les slalomeurs arrivés, il devient impératif d'établir un classement et de conserver les temps en vue d'une éventuelle deuxième manche. Rappelons à cette occasion que lors de la seconde manche, les cinq premiers repartent dans l'ordre décroissant (5, 4, 3, 2, 1) puis tous les suivants dans l'ordre de leur performance (6, 7, etc.).

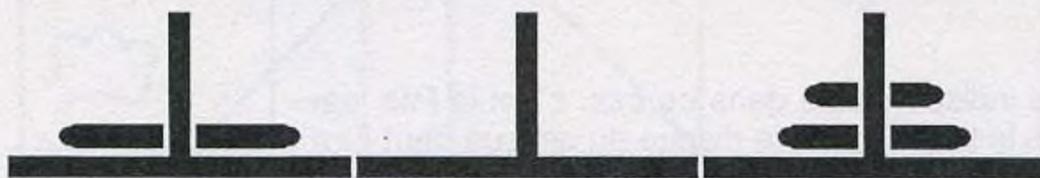
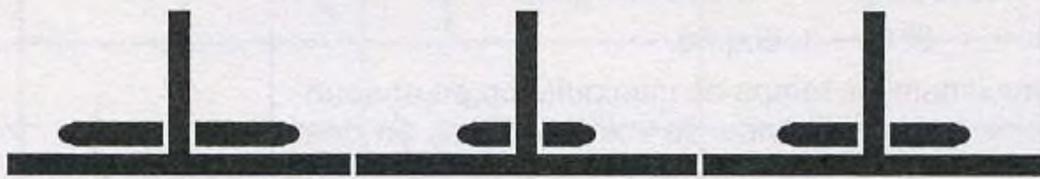
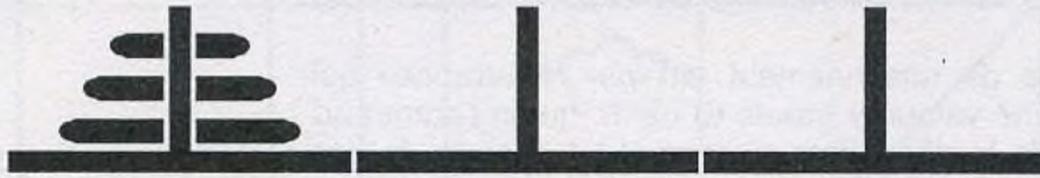
Maintenant nous espérons que vous avez fait votre choix en matière de structure pour pouvoir comparer avec le nôtre que nous allons argumenter. Cependant notre solution n'est qu'indicative. Si vous avez une bonne argumentation à développer concernant votre décision, ne remettez pas tout en cause. Si au contraire vous n'avez procédé que par élimination des autres possibilités, voyez avec nous ce que nous proposons. La règle Premier Entré-Premier Sorti vous laisse présager une File, cette idée a ses adeptes mais la nécessité d'éliminer des concurrents avant l'arrivée exige une fonction d'accès à l'intérieur de la File et cela entre nettement plus dans la philosophie de la Liste Linéaire. En effet, l'insertion et l'élimination d'un élément quelconque de la Liste Linéaire sont préalablement construites et les procédures d'emploi existent.

4.2. Les Tours de Hanoi

4.2.1. Présentation

Ce jeu nous provient du Viet-Nam comme son nom l'indique. il consiste en trois piquets autour desquels sont empilés des disques de diamètres inégaux depuis le plus grand jusqu'au plus petit. On ne peut manipuler qu'un seul disque à la fois : celui placé le plus haut et uniquement dans le but de le déplacer vers un des deux autres piquets où il ne pourra être empilé que si le disque placé immédiatement sous lui est de diamètre supérieur. Initialement, les disques sont tous situés sur le piquet de gauche. Le but est de tous les déplacer sur le piquet de droite.

POSITION DE DEPART (3 pièces)



POSITION FINALE

4.2.2. Cette description apporte deux commentaires : l'un légendaire, l'autre théorique. Si ce programme porte le nom de tours de Hanoï, c'est qu'il existerait à Hanoï un temple où se trouveraient les trois piquets avec 64 disques. Plusieurs bonzes auraient pour unique «charge» de déplacer les disques, la fin de leur mission devant coïncider avec celle de notre monde. Passons maintenant au second aspect pour rassurer certains.

On connaît en mathématiques un type de raisonnement dit «par récurrence» qui annonce que si une loi est vraie pour une valeur N triviale (0 ou 1), qu'on l'admet au rang N , il suffit pour qu'elle soit juste de la démontrer au rang $N + 1$ à partir de ces hypothèses. Du point de vue mathématique, cet outil est très puissant. Utilisons-le ici :

La formule est : pour N disques, il faut $2^{N+1} - 1$ déplacements. Vérifions-la :

Si $N = 1$, il suffit bien de $2^{0+1} - 1 = 1$ opération.

Admettons-le au rang N , et stabilisons au rang $N + 1$:

Pour déplacer les N premiers, il faut $2^{N+1} - 1$: pour manipuler le $N + 1^{\text{ème}}$, il suffit d'une, puis pour redéplacer les N premiers, de nouveau $2^{N+1} - 1$, soit au total :

$$2 * (2^{N+1} - 1) + 1 = 2^{N+2} - 1. \text{ C.q.f.d.}$$

Le simple calcul, même en réduisant au maximum les temps de manipulation de chacun des 64 disques, ne nous laisse néanmoins aucune chance de voir là-dedans un des grands périls qui menacent notre planète.

4.2.3. Programmation

Vous l'avez tous reconnue : la structure indispensable dans ce cas, c'est la Pile, car dans l'énoncé même du problème. Nous le lisons : seul le disque du dessus peut être déplacé. Une fois ceci établi, à vous de faire le programme en y ajoutant les images qui vous plaisent pour représenter les tours.

4.3. Les Dames en prises

Le but est ici encore plus simple : il s'agit de disposer sur un échiquier le maximum de dames sans qu'aucune ne se retrouve en situation (d'ailleurs réciproque) de prise. La solution est, bien évidemment, de huit, reste à connaître la géographie des reines sur le carré 8×8 .

Une fois ce problème résolu, deux types de corollaires : d'une part, la détermination de l'ensemble des solutions en éliminant toutes les formes de symétrie et de rotation possibles, d'autre part, le problème identique avec des pions, des cavaliers, des fous et des rois (nous n'exigerons pas les tours). Nous attendons vos solutions programmées pour cette dernière partie.

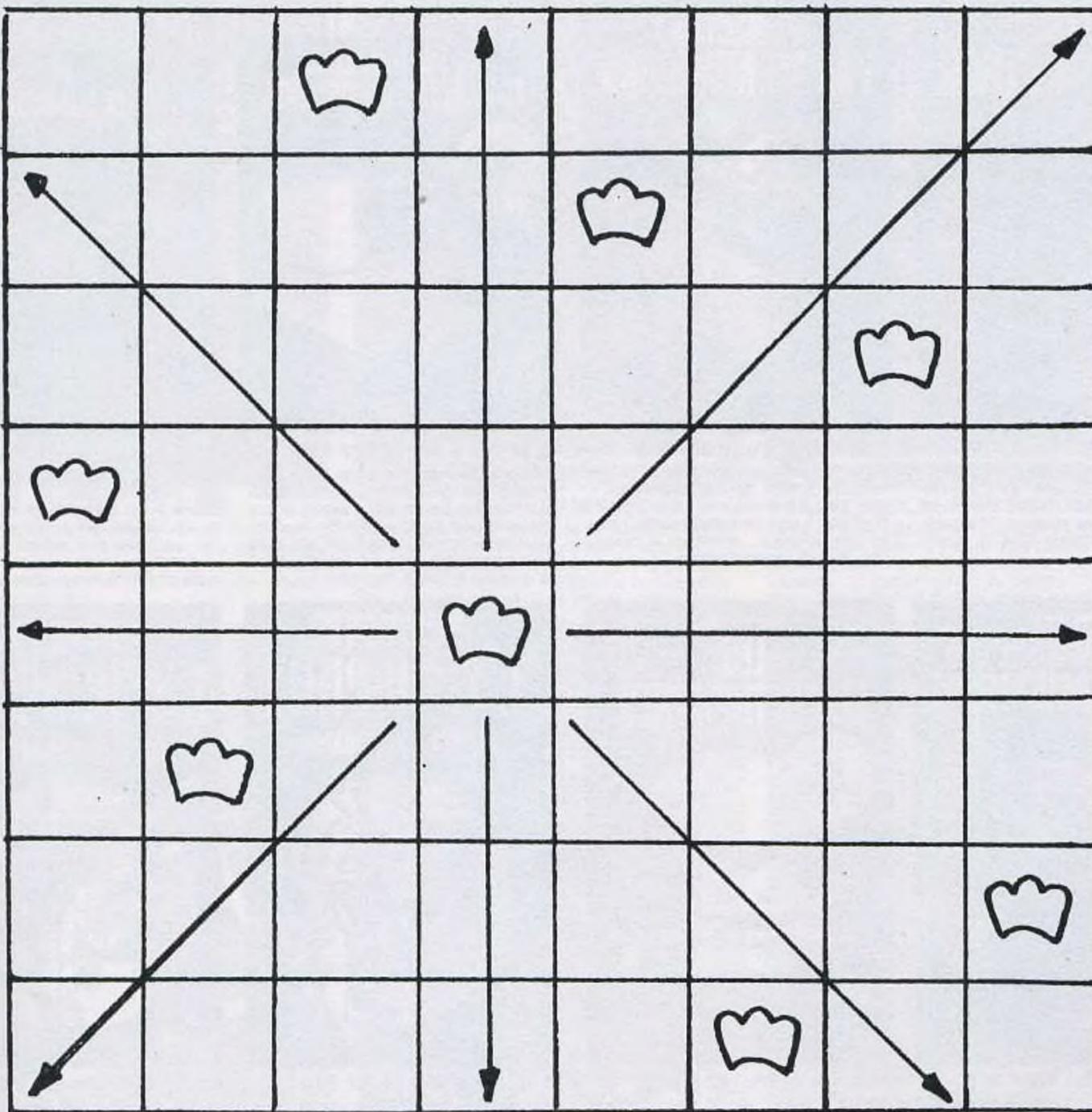
Du strict point de vue de la programmation, il est agréable de manipuler des tableaux creux pour le programme concernant les reines. Afin de résoudre les autres questions, le choix s'impose beaucoup moins.

4.4. Les Nombres Parfaits

Un de nos lecteurs qui a tenu à garder l'anonymat en omettant de nous fournir ses coordonnées, nous a posé le problème des nombres parfaits et ne pouvant matériellement lui répondre, nous vous suggérons de le faire à notre place en soumettant le sujet à votre sagacité :

Un nombre parfait a la particularité d'être la somme de tous ses diviseurs autres que lui-même. Les deux premiers étant 6 et 28 :

$$\begin{aligned} 6 &= 1 + 2 + 3 \\ 28 &= 1 + 2 + 4 + 7 + 14 \end{aligned}$$



les trois suivants sont encore dans des valeurs supportables.

Ce programme n'entre pas directement en prise avec le cours mais l'emploi de la pile pour stocker l'ensemble des diviseurs d'un nombre paraît être une idée judicieuse.

4.5. Point sur les exercices précédents

Chose curieuse : personne n'a jugé utile de nous envoyer une solution de carrés magiques, pourtant ce problème nous avait paru amusant tant sur le plan de la programmation que sur le plan de l'originalité, aussi nous aimerions qu'à l'instar de notre lecteur anonyme, vous nous exposiez les problèmes qui vous «interpellent». En attendant, bonne programmation !

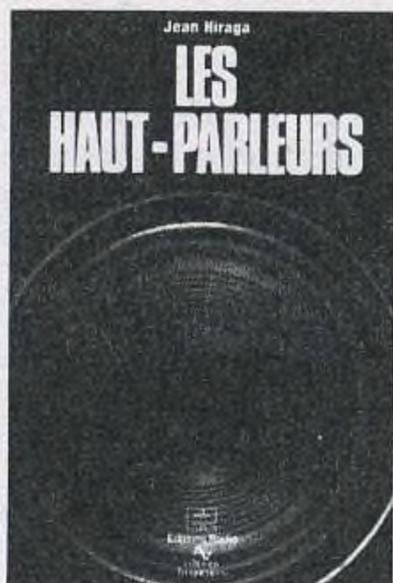


BIBLIOTHEQUE TECHNIQUE

Collection études (format 165 x 240)



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



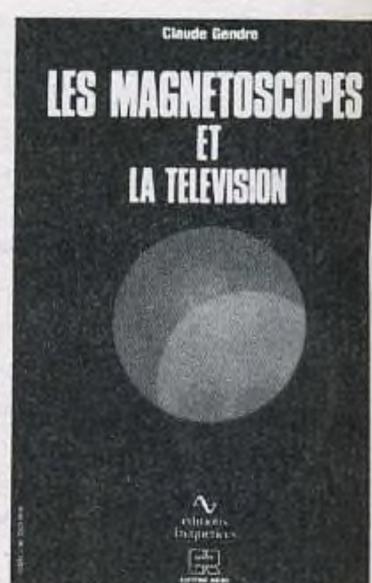
E 01. 320 p. Prix : 165 F TTC
Un gros volume qui connaît un succès constant : bien plus qu'un traité, il s'agit d'une véritable encyclopédie, alliant théorie et pratique, histoire, en une mine inépuisable d'informations, reconnue dans le monde entier !



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



E 04. 240 p. Prix : 154 F TTC
Seconde édition 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.



E 03. 256 p. Prix : 145 F TTC
Complément direct des «Magnétophones», les «Magnétoscopes et la Télévision» débute par un bel historique de la télévision et la description des premiers magnétoscopes. La théorie et la pratique de la capture et de l'enregistrement moderne des images vidéo en sont la teneur essentielle.



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



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



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



E 13. 256 p. Prix : 165 F TTC
Une sélection des meilleurs articles de la célèbre revue «l'Audiophile» choisis parmi les plus significatifs des quinze premiers numéros, introuvables aujourd'hui. Le tome 1 traite de l'électronique audio, à tubes et à transistors.



E 12. 256 p. Prix : 155 F TTC
Dans un esprit identique, le tome 2 traite du domaine passionnant que constituent les transducteurs en audio : on y aborde la modélisation théorique des enceintes, la conception géométrique des tables de lecture, le réglage des cellules et des bras.

Collection loisirs (format 135 x 210)



L 07. 160 p. Prix : 68 F TTC
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».



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



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



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



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



L 20. 208 p. Prix : 130 F TTC
Accessible à tous, «Week-end photo» permet de découvrir de façon simple les différents aspects de la photographie actuelle. Vous y trouverez les bases indispensables pour vous perfectionner, un guide de choix des appareils 24x36 et des illustrations abondamment commentées.

DES EDITIONS FREQUENCES

Collection pédagogique (format 210 x 270)



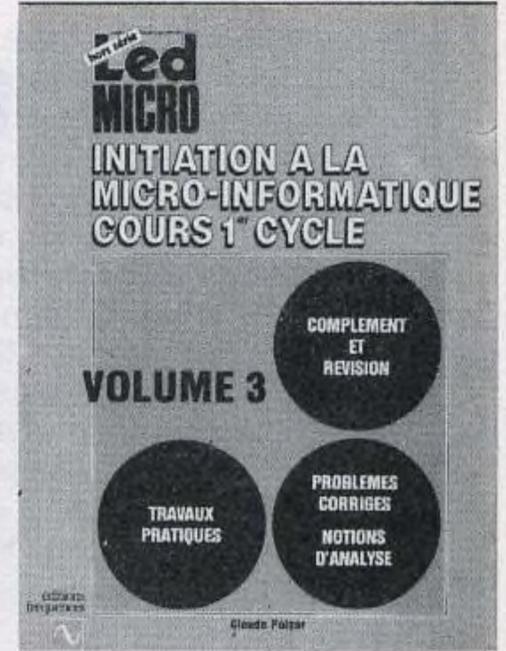
P 08. 96 pages. Prix : 115 F TTC
Cet ouvrage eut un succès retentissant dès sa sortie. Bien plus qu'un cours d'initiation, il s'agit aussi du premier recueil d'informations données par les concepteurs, les utilisateurs de robots et les fans de cybernétique, enfin réunis !



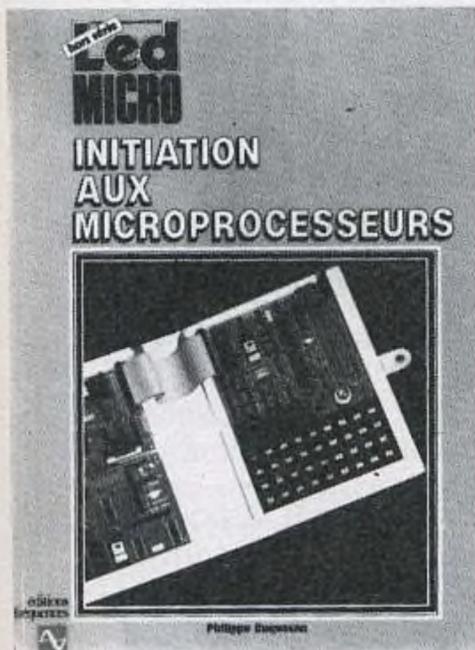
P 16. 272 pages. Prix : 130 F TTC
Passé les premiers remous de la révolution que fut l'avènement de la micro-informatique, il fallut bien tenter d'en réunir les enseignements. Une lacune apparut : celle d'un ouvrage d'initiation à la programmation, universel et complet. En voici le premier tome.



P 17. 208 pages. Prix : 130 F TTC
Le tome 2 est la suite du tome 1 : l'esprit puissamment didactique de l'auteur s'y retrouve, le contenu du livre permettra d'acquérir un niveau suffisant pour exercer l'analyse, la programmation, la gestion, l'automatisme, la simulation et d'autres choses encore !



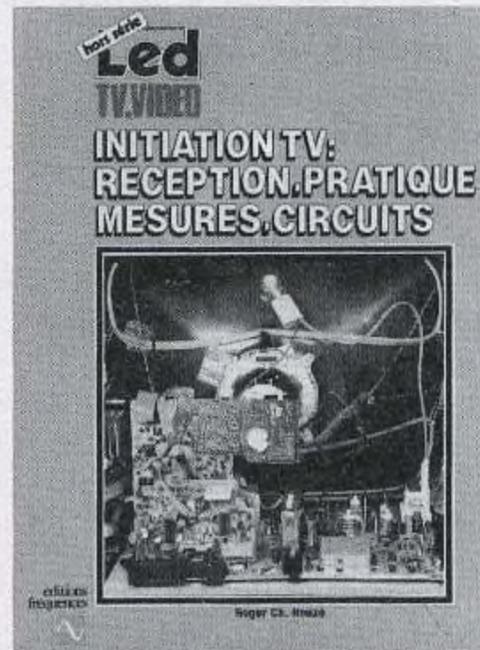
A Paraître
Le troisième volume du cours de Programmation, dû à Cl. Polgar, pédagogue apprécié de tous. Il continue dans la lignée d'un réel souci didactique, de haut niveau, maintenant, mais en conservant l'aspect progressif qui fit son succès initial.



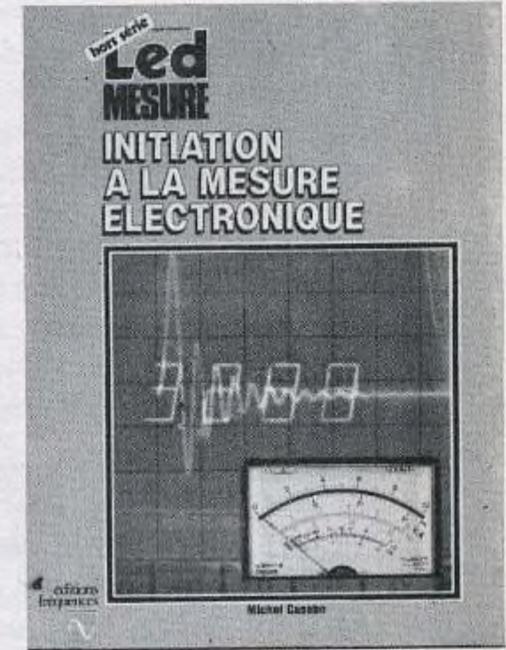
P 18. 136 pages. Prix : 95 F TTC
Du même auteur, Ph. Duquesne, on nous propose cette fois-ci, de pénétrer au cœur même de l'ordinateur, de comprendre le fonctionnement de l'élément vital qu'est le microprocesseur et enfin de maîtriser l'assembleur, langage du microprocesseur.



P 19. 104 pages. Prix : 95 F TTC
Ce cours d'Initiation à l'Electronique Digitale est dû à Ph. Duquesne, chargé de cours de microprocesseurs au CNAM. L'objet de cet ouvrage est de présenter les opérateurs logiques et leurs associations. La technologie est évoquée, brièvement, elle aussi.



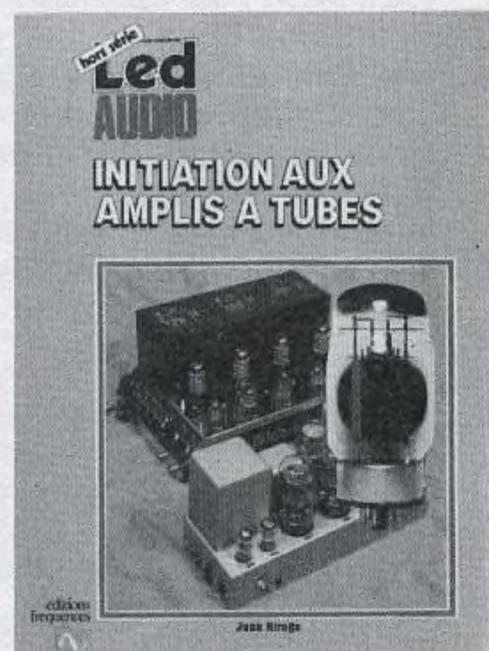
P 21. 136 pages. Prix : 135 F TTC
Issu d'un cours régulièrement remis à jour, ce livre permet à l'amateur comme au professionnel de se tenir au courant de l'état actuel de la technologie en télévision. De nombreux schémas explicatifs illustrent le contenu du livre.



P 23. 120 pages. Prix : 140 F TTC
Il n'existait pas, jusqu'à présent, un ouvrage couvrant de manière générale mais précise, l'ensemble des problèmes relatifs à l'instrumentation et à la méthodologie du laboratoire électronique. C'est chose faite aujourd'hui avec ce volume récemment paru.



P 24. 96 pages. Prix : 130 F TTC
Après un bref historique du transistor, cet ouvrage traite essentiellement de la conception des amplificateurs modernes à transistors. La théorie est décrite de manière simple et abordable, illustrée d'exemples de réalisations commerciales. Le but du livre est de donner à chacun la possibilité de réaliser soi-même son amplificateur...



à paraître
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.

En vente chez votre libraire ou aux Editions Fréquences
Bon de commande à retourner aux Editions Fréquences
1, boulevard Ney 75018 Paris

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

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

Frais de port : + 10 F par livre commandé, soit la somme totale ci-jointe, de Frs _____ par

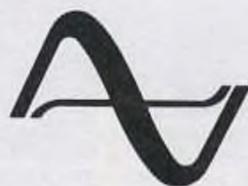
CCP Chèque bancaire Mandat-lettre

Nom Prénom

Adresse

Ville Code postal

C'EST ARRIVE DEMAIN



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

Comme je vous le disais le mois dernier, IBM a décidé et annoncé de rendre impossible la modification du PC-AT consistant à changer le quartz pour disposer d'un appareil tournant plus vite que les modestes 6 MHz de la machine au sortir de l'usine. Les raisons techniques invoquées sont toutes réfutées par des entreprises aussi sérieuses que Microsoft, General Motors, General Electric,... utilisatrices des systèmes concernés. Excusez du peu !!! Des hypothèses, toutes plus extravagantes les unes que les autres circulent, mais aussi, les revendeurs tentent une nouvelle fois de se mobiliser pour dissuader la société IBM de réaliser cette modification interne. Gageons qu'une nouvelle fois, IBM les ignorera superbement.

Le COMDEX a donc eu lieu vers la mi-novembre, et beaucoup de petites nouveautés ont été exhibées par les fabricants de hard comme de soft. Par contre, peu ou pas de grandes nouveautés ou de surprises. Pourtant, près de 90 000 personnes avaient fait le déplacement, dont beaucoup de journalistes de revues françaises!!!

Enfin, tout le monde ne peut y aller en voisin!!!

Le nombre d'exposants est tel que certaines grandes compagnies refusent de s'y rendre alléguant l'impossibilité de travailler dans de telles conditions. Parmi elles, Lotus, qui est toujours en tête des ventes avec

1-2-3 et Symphony.

Les seules nouveautés importantes sont la nouvelle version de D-BASE III, et l'environnement convivial (comprenez plus ou moins copié sur celui du MacIntosh) de Microsoft, appelé Window, dédié aux compatibles IBM-PC.

C'est peu, très peu, d'autant que jamais autant de sociétés n'avaient annoncé des produits à cette exposition, pour plus tard, sans même faire un semblant d'effort pour les sortir pour l'exposition. Ceci est une nouveauté dans la philosophie et l'attitude des producteurs, qui pourraient remettre en cause, à terme, le principe de telles expositions mammoth. D'ailleurs, jamais autant de désistements de spectateurs, mais aussi d'exposants, n'avaient eu lieu, si bien que de petites sociétés de logiciel se retrouvèrent à des places de choix le premier jour, et non au fond du dernier niveau.

Apple a montré le MacIntosh Jonathan, nom de la version couleur de cet ordinateur, et annoncé diverses améliorations du produit et de la gamme existants. Le seul à avoir réalisé un sans faute commercial est la société Atari, qui au lieu d'un stand traditionnel, avait fractionné l'espace loué en une cinquantaine de petites zones où trônait un 520 ST, et un développeur

logiciel maison ou non. Ainsi, ce programmeur pouvait expliquer au mieux le produit qu'il avait lui-même créé, le but de la manœuvre étant de prouver que cet ordinateur dispose d'ores-et-déjà de logiciels variés et de grande qualité. Le succès de cette méthode fut total, le nombre de distributeurs à signer un contrat de vente dépassant 50, chiffres Atari. Des programmes parmi ceux montrés étaient d'une qualité exceptionnelle, que ce soit des programmes de comptabilité, de traitement de textes, de graphique, ou des jeux. Cet ordinateur, ainsi que l'Amiga sont au prix de l'Apple II, pour les performances du niveau, sinon meilleures, de celles du MacIntosh. Et ceci est renforcé par le dernier point très intéressant du Comdex, les écrans couleurs. Ils sont arrivés à une qualité telle que pour des ordinateurs simples comme ceux dont je vous parlais, le fait d'avoir un tel écran couleur pour du texte n'est plus un handicap et une gêne. Les couleurs sont parfaites et l'utilisation texte se fait sans bavures, les utilisateurs d'Apple II comprenant qu'il n'y a là nul jeu de mot.

Contrairement à ce que fait Apple, Atari, reparti du fond du gouffre, prépare l'avenir avec dynamisme, et s'intéresse à la puce la plus performante du moment, le 3232 de National Semiconductor. Cette dernière société en est enchantée, car la baisse du prix d'un tel chip n'est possible que par une diffusion en masse (Atari en est capable), et cela renforce la crédibilité du fabricant, qui faisait jusqu'alors figure de parent pauvre face à Motorola, Intel, Zilog et autres.

Halleluja, le prix des programmes atteint enfin des niveaux raisonnables. De plus en plus de programmes de très grande qualité et remarquablement documentés, sont maintenant proposés à moins de 100\$ (ce qui le mettra en France dans les 2000 F, l'appétit des intermédiaires n'ayant d'égal que leur mépris du client). Ceci a même obligé Microsoft à baisser ses prix sur de nombreux produits devenus non compétitifs. Espérons que la sympathique équipe de Borland (société créée et dirigée par le français Kahn) parviendra à survivre à cette évolution, car il faudra se souvenir qu'ils furent les premiers à vendre des logiciels extraordinaires pour moins de 40\$, et parmi eux Turbo Pascal, qui reste un des meilleurs, sinon le meilleur des compilateurs Pascal.

IBM victime du succès qu'elle a imposé sans gloire ? On pourrait se le demander à entendre les tergiversations des cadres de la société concernant la mise à jour du système d'exploitation PC-DOS, et du MS-DOS. Voilà les faits. IBM souhaite actualiser les PC-DOS, qui équipent les PC de la marque, pour les rendre plus compatibles avec les exigences du moment, et surtout pour tenter de les rendre proches du système des plus gros IBM, appelés VM. Ceci en raison du fait que les PC deviennent de plus en plus performants, presque autant que les IBM de cette gamme moyenne. Mieux vaudrait les rendre les plus

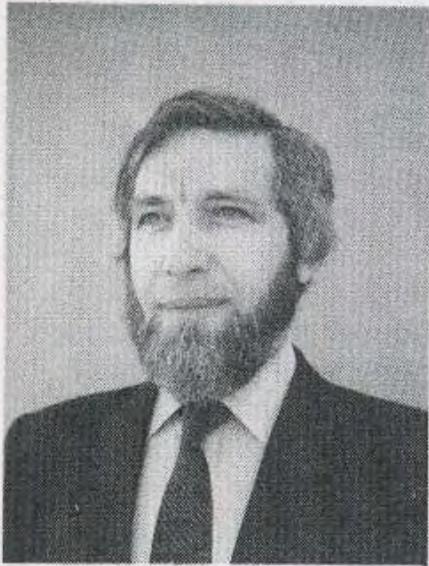
compatibles possibles, ou tout au moins susceptibles de dialoguer le plus aisément possible. Le problème, grave s'il en est, est que cette modification rendrait les PC existants, disposants de l'ancien système et les nouveaux non compatibles à 100%. Or, une compatibilité à 99% est une non compatibilité de fait, et IBM veut l'éviter. En effet, après avoir imposé une norme, il serait délicat d'en proposer une autre à des gens qui n'avaient acheté IBM que pour être sûr de disposer d'un système ne risquant pas de disparaître rapidement, l'image d'IBM étant dans ce cas précis celle d'une firme solide et non celle d'une maison risquant de disparaître chaque année. Alors PC-DOS monstre monolithique mourant sur place de vieillesse, ou starlette (comme les autres) et évoluant de temps en temps ?

Un développement intéressant de la miniaturisation des composants est la sortie depuis quelques temps de modems internes. Cela signifie que vous n'avez plus nécessairement une boîte énorme à côté de votre petit ordinateur pour communiquer, mais qu'une simple carte dans cet ordinateur peut en faire tout autant. Votre table ou bureau ne ressemble plus à une enclave du salon des arts ménagers 1935, et, avantage au moins aussi intéressant, tout est plus rapide, depuis le protocole de communication jusqu'au temps de travail de votre petite bestiole préférée. La note des PTT en sera d'autant diminuée, ce qui n'est pas un mal par les temps qui courent, que vous utilisiez votre ordinateur à titre professionnel, ou qu'il ne s'agisse que de recevoir le jeu préféré de votre ami qui est à l'autre bout du pays.

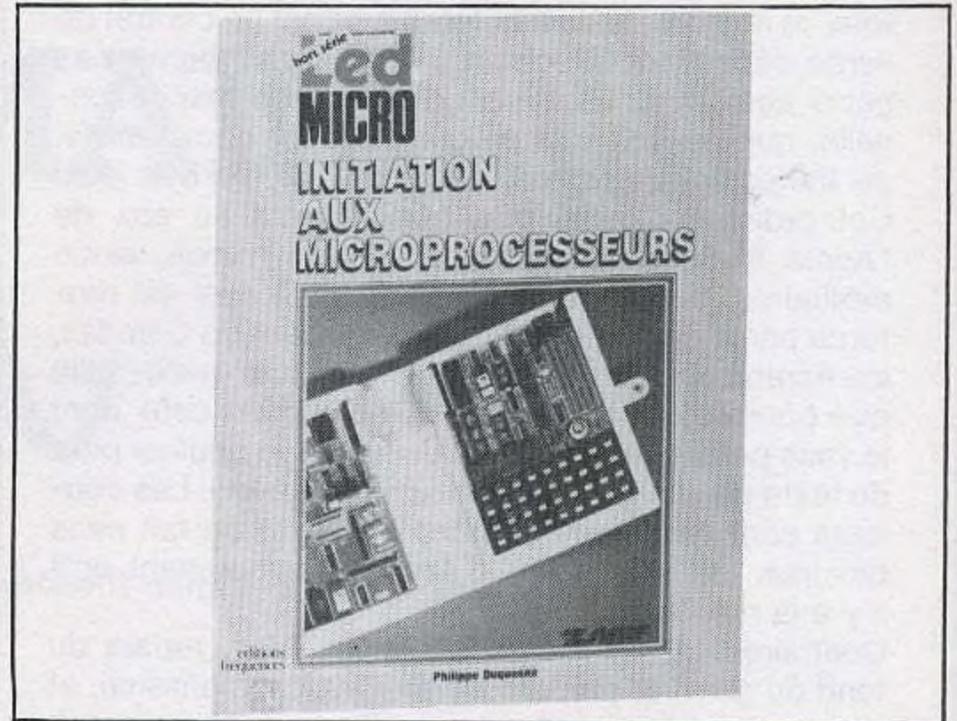
Les premiers essais complets de l'Amiga, l'ordinateur de Commodore, sont plutôt décevants. En effet, la puissance potentielle de l'unité Centrale est contrebalancée par les multiples défauts qui sont décelés au fur et à mesure de la diffusion grand public. Le système d'exploitation (Release 1.0) est considérablement buggé, par exemple le contrôle des tâches pour ce système qui se veut multi-tâches. Tout va bien tant que vous ne lancez pas une tâche demandant trop de mémoire. Dans ce dernier cas, tout plante, sans possibilité de sauvegarde. Autre point à noter, le multi-tâche est géré de telle sorte qu'une fenêtre de plus ou de moins occasionne des variations de vitesses importantes en exécution. Ceci, plus d'autres détails font de l'Amiga un produit non fini, mais son potentiel est tel que lorsque les petits problèmes auront disparu, il sera un concurrent réel du MacIntosh, pour un prix très très inférieur, et des performances souvent supérieures de beaucoup.

Dernière minute. Les raisons d'IBM pour limiter la vitesse de l'actuel AT sont enfin officiellement connues. Une des hypothèses était justifiée. Un PC-AT plus rapide va être proposé par la société, à un prix très supérieur à celui de la version existante.

Microprocesseurs un cours essentiellement pratique !



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



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

Electronique digitale ?

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



En vente chez votre libraire et aux Editions Fréquences

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

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

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

Ci-joint mon règlement par : CCP Chèque bancaire Mandat

Nom Prénom

Adresse

Code postal Ville

COURS DE GENIE LOGICIEL

De la théorie à la pratique

Charles-Henry Delaleu

LES OUTILS GRAPHIQUES

Grâce aux possibilités d'intégration des composants électroniques, le graphisme est devenu de nos jours, un outil largement diffusé dans la plupart des calculateurs. Du plus petit des micro-ordinateurs à la plus puissante des machines, les options dans le domaine du dessin sont plus que nombreuses. Bien que les prix de telles extensions aient beaucoup baissé, il reste encore des possibilités afin d'assouvir tous les budgets : de quelques centaines de francs aux millions de francs, tout est possible. Les outils graphiques sont aujourd'hui présents dans l'industrie, les domaines médical et militaire, l'audiovisuel, l'enseignement, etc.

Grâce à cela, la conception assistée par ordinateur a pu obtenir un essor très important. De nombreux constructeurs de composants électroniques proposent désormais des processeurs graphiques spécialisés et les constructeurs de matériel informatique disposent dans leur catalogue de toute une panoplie de périphériques en tous genres.

Si le dessin en informatique a d'abord fait son entrée dans les domaines techniques et scientifiques, les milieux financiers ont récemment découvert qu'un joli dessin était beaucoup plus parlant et plus vite assimilé qu'un long listing de chiffres. C'est pourquoi tous les bons tableaux et autres outils de prévision sont aujourd'hui dotés d'options graphiques. Si, dans les domaines de l'animation visuelle, l'ordinateur prend de jour en jour une place plus importante, les constructeurs automobiles étudient déjà les répercussions de l'installation de moniteur graphique dans le tableau de bord des véhicules. Ces aides à la navigation pourraient servir à toutes sortes d'applications, telles que le contrôle de la voiture, mais aussi de cartes routières, etc.

Dans les domaines militaires, les possibilités graphiques de l'informatique sont très

largement utilisées en études et recherches, mais aussi en fabrication. Qui n'a jamais entendu parler de simulation de vol en aéronautique civile et militaire. En instrumentation, le dessin a permis de transformer des calculateurs en appareils de mesure.

Mais attention, si dans le jargon informatique, les termes de mémoire vive ou mémoire de masse sont connus, les subtilités graphiques le sont beaucoup moins et les excès de langage sont très courants. Ainsi, contrairement aux idées reçues, un système évolué en informatique graphique est encore loin des possibilités financières de monsieur tout le monde.

LES OPERATIONS GRAPHIQUES

LES DIMENSIONS : Il existe deux grandes familles de dessin :

- les dessins en deux dimensions ;
- les dessins en trois dimensions.

LES SURFACES : Les dessins peuvent être exécutés soit sur écran vidéo, soit sur table traçante. Généralement dans tout système évolué, il est possible de choisir :

- les surfaces de dessin ;
- les échelles ;
- les localisations de trace.

LES OPERATIONS MATHÉMATIQUES : Elles autorisent dans le cadre du calcul matriciel appliqué au graphique, toutes sortes de possibilités telles que :

- la réalisation de véritables dessins en trois dimensions ;
- les changements d'échelles, le zoom ;
- les rotations ;
- les translations ;
- les perspectives.

LES LANGAGES GRAPHIQUES : Les petits systèmes disposent dans leur interpréteur de langages de quelques extensions qui autorisent des opérations graphiques simples. Les systèmes plus évolués disposent de véritables langages graphiques qui sont, soit complètement indépendants, soit de grandes extensions, soit des modules ou utilitaires spécialisés.

LE MATÉRIEL GRAPHIQUE : Le plus connu des matériels graphiques est, sans aucun doute, l'écran vidéo mais il convient de noter :

- les tables traçantes ;
- les imprimantes graphiques ;
- les light-pens ;
- les tables à digitaliser ;
- les souris ;
- les clés de fonction pré-programmées ;
- les écrans tactiles ;
- le clavier, etc.

LES ORDRES GRAPHIQUES SIMPLES : Les ordres graphiques simples sont assez nombreux, mais il convient d'énumérer les principaux :

1^{er} groupe :

- la localisation de la surface de trace ;
- la définition de l'échelle
 - a) échelle fixée par la machine,
 - b) échelle fixée par l'utilisateur.
- l'utilisation de fenêtres ;
- l'utilisation de pages graphiques.

2^e groupe :

- le déplacement de plume avec ou sans trace ;
- les contrôles de plume ;
- les opérations sur les textes à dessiner ;
- les remplissages de polygones irréguliers ;
- les opérations couleurs.

3^e groupe :

- le stockage de pages vidéo graphiques ;
- le stockage de matrice de points ;
- les fichiers divers en mémoire vive ou mémoire de masse.

ROTATION, TRANSLATION, ECHELLE :

Dans le cas d'un dessin réalisé à base de matrices, il est possible de faire toutes sortes de manipulations. En effet, l'utilisation de matrices autorise le calcul matriciel et ouvre ainsi de larges possibilités en graphisme.

LA ROTATION :

La rotation consiste à faire tourner l'objet dessiné dans différents axes.

A) en graphique 2 dimensions, il est possible de réaliser :

- une rotation dans l'axe des X ;
- une rotation dans l'axe des Y.

B) en graphique 3 dimensions, il est possible de réaliser :

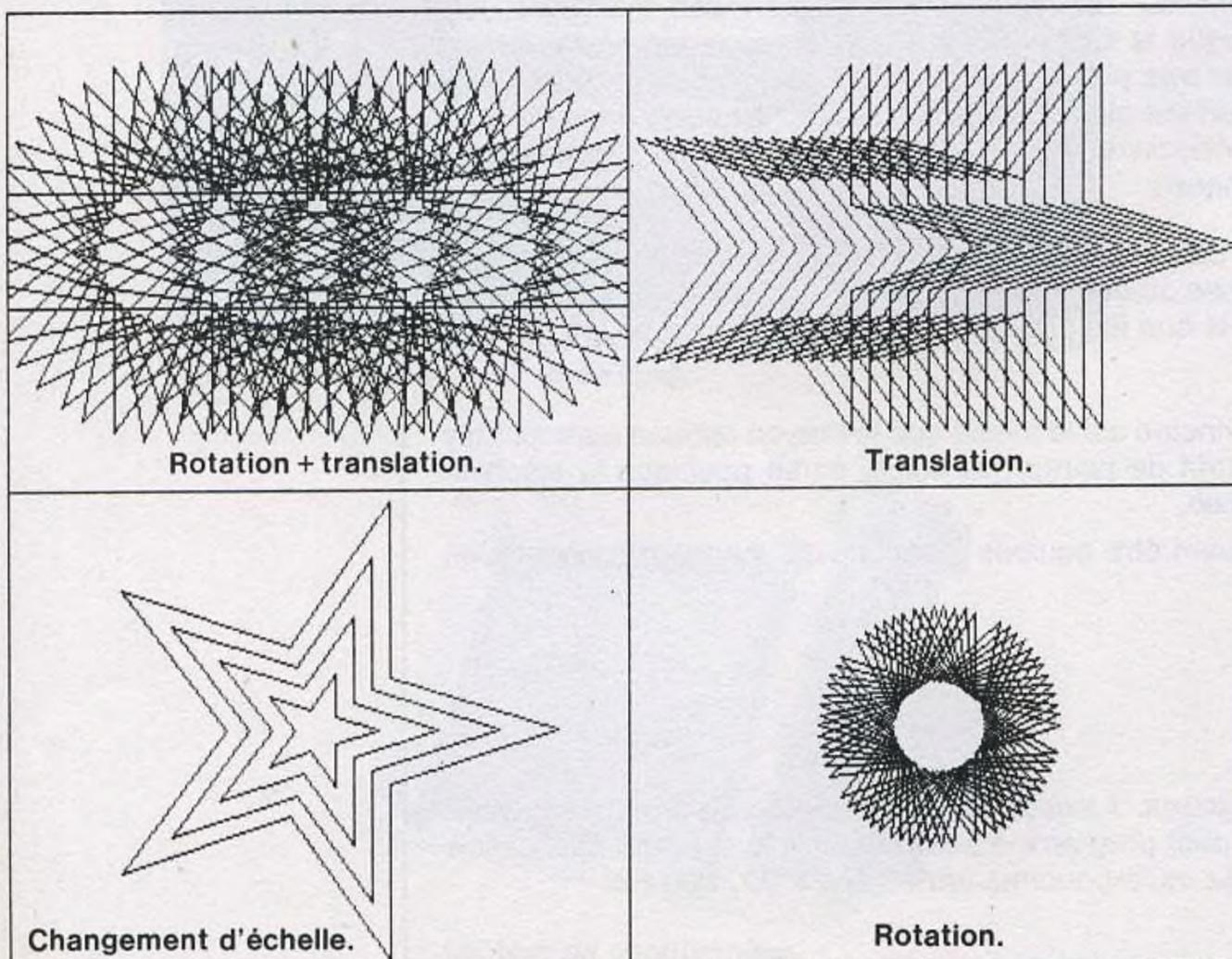
- une rotation dans l'axe des X ;
- une rotation dans l'axe des Y ;
- une rotation dans l'axe des Z.

TRANSLATION :

La translation consiste à déplacer un dessin dans l'écran.

ECHELLE :

Le changement d'échelle est en fait équivalent au zoom en photographie. Il permet d'obtenir des agrandissements ou des réductions d'un dessin à l'écran. Dans le cas de dessins complexes, il sera possible grâce aux changements d'échelle de localiser une partie d'un dessin sur lequel on désire travailler. Dans le cas d'un dessin d'immeuble, il sera possible de partir du dessin complet de l'immeuble pour corriger les détails d'une canalisation par exemple.

ETOILE DESSINEE EN DEUX DIMENSIONS :

LES OUTILS GRAPHIQUES

LES ECRANS : Il existe, au niveau des écrans vidéo, plusieurs possibilités :

- le noir et blanc ;
- la couleur.

La résolution graphique sera fonction :

- du processeur graphique ;
- de l'espace mémoire ;
- de la précision de l'écran.

LES TABLES TRAÇANTES : Le tracé du dessin peut être réalisé soit en déplaçant le stylet, soit le papier, soit une version hybride. Le traceur peut avoir une ou plusieurs plumes (couleur). Le papier peut être soit déposé sur une table, soit enroulé sur un tambour. Dans ce dernier cas, il sera possible de réaliser des dessins allant jusqu'à 17 m de longueur sur certains modèles. Enfin, il existe parfois la possibilité de déplacer une plume à l'aide d'un curseur afin de lire des coordonnées. Le traceur devient alors digitaliseur.

LES IMPRIMANTES GRAPHIQUES. De nombreuses imprimantes autorisent désormais une copie d'écran vidéo. Ceci est possible sur les modèles à matrices, à jet d'encre, laser et thermiques.

LES LIGHT-PENS. Les light-pens ont été un des premiers outils graphiques. Ce processus consiste à placer un crayon optique sur l'écran vidéo et à en lire les coordonnées X, Y. Ce dernier peut servir à pointer l'écran mais aussi à dessiner.

LES TABLES A DIGITALISER : Il s'agit d'une table sur laquelle il est possible de déplacer un crayon relié électriquement. Le fait d'appuyer le crayon sur la table provoque une lecture de la position du crayon sur la table. Ces tables permettent une meilleure précision que les light-pens.

LES SOURIS : Les souris font leur apparition sur la plupart des machines, leur utilisation est très facile et lorsque le logiciel qui les accompagne est correctement réalisé, elles deviennent un outil très puissant. Une souris est réalisée à l'aide d'une bille que l'on déplace sur une surface plane, la position relative de cette dernière étant représentée à l'écran par l'intermédiaire d'une flèche. Il existe plusieurs variantes de souris plus ou moins perfectionnées.

LES CLES DE FONCTIONS. Les clés de fonctions pré-programmées servent en général soit à appeler des routines ou des sous-programmes spécialisés, mais aussi à configurer des périphériques tels que les crayons optiques, les souris ou les tables à digitaliser.

LES ECRANS TACTILES : Le principe est le même que le crayon optique mais ici il n'y a pas de capteur (crayon). Il suffit de pointer du doigt l'écran pour que la machine puisse détecter la position choisie.

LE CLAVIER : Les claviers peuvent être équipés d'options qui autorisent une aide au dessin :

- les flèches ;
- les clés de fonction ;
- les roues codées ;
- les mini-digitaliseurs.

LES LOGICIELS. Outre les langages, il existe de nombreux logiciels qui permettent une aide au dessin. Cela va du petit programme pour enfant à la superbe application pour réaliser des circuits intégrés multi-couches (de 70 F à 1 000 000 F).

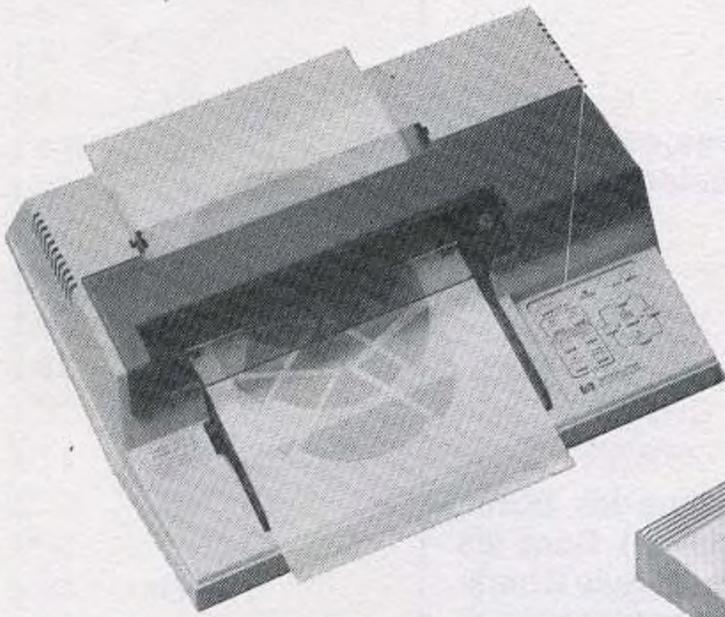
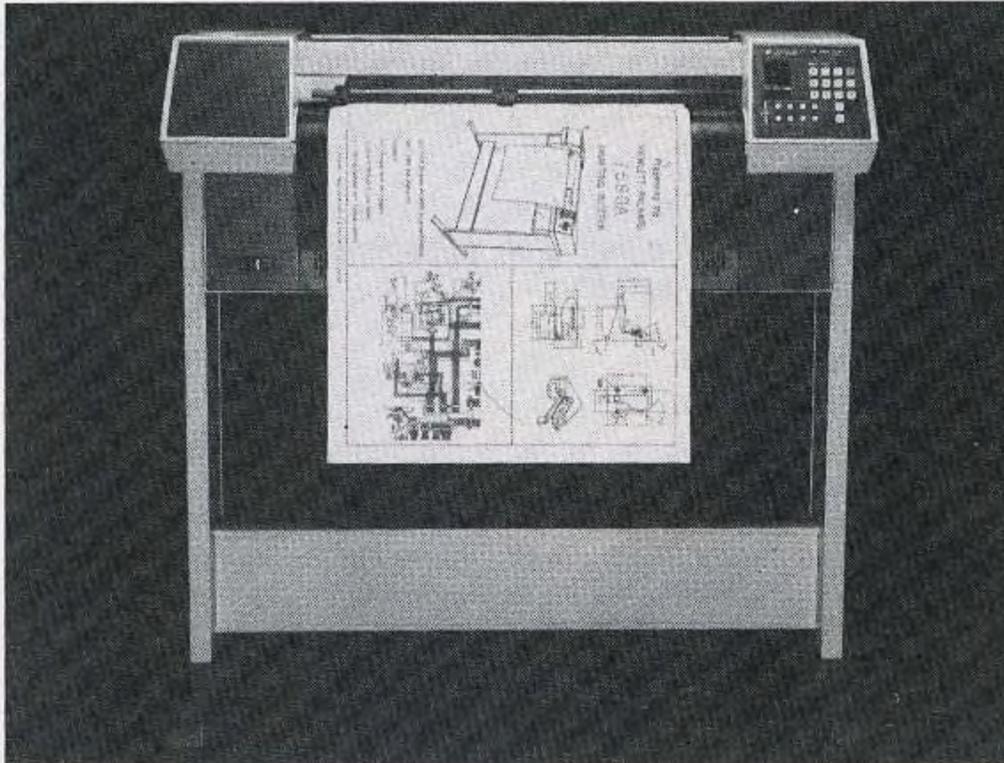


Table traçante, digitaliseur, à déplacement de papier bi-couleur.



Table à digitaliser.



Traceur à rouleau, ce type de traceur peut dessiner des documents allant jusqu'à 17 m de long.



Lecteur de coordonnées.

LE DESSIN EN TROIS DIMENSIONS

Contrairement aux dessins classiques, un dessin en trois dimensions est réalisé à base d'une matrice. Ceci permet de faire sur cette matrice différentes opérations mathématiques autorisant :

- la rotation dans l'axe des X ;
- la rotation dans l'axe des Y ;
- la rotation dans l'axe des Z ;
- la translation ;
- le changement d'échelle ;
- la perspective.

Une matrice de base comprend les coordonnées X, Y et Z de tous les points principaux du dessin ainsi que la position de plume (trait ou plume levée). Dans les exemples suivants est représentée une ébauche de voiture automobile réalisée à partir de 70 points principaux. On appelle **points principaux** les différentes coordonnées X, Y, Z entre lesquels seront tracés des traits.

Dans un système graphique trois dimensions plus évolué, il sera possible d'ajouter différentes informations :

- choix de la couleur de trace et du remplissage ;
- possibilité d'utiliser des faces cachées ;
- possibilité de réaliser des ombres en simulant des éclairages.

LES EXEMPLES :

Dans les pages qui suivent sont donnés quelques exemples d'opérations effectuées sur matrice dans le cas d'un dessin en trois dimensions. Il s'agit d'un programme d'étude pour dessin en trois dimensions. Il se divise en trois parties :

- 1^{re} partie** : lignes 30 à 154, appels des fonctions
- 2^e partie** : lignes 390 à 640, trace du dessin
- 3^e partie** : opérations sur matrice pages 7 à 12.

1^{re} partie :

```

30 Setup:OPTION BASE 1
31     GINIT
40     DIM Object(80,4),Points(80,4)
50     REAL Angle,X,Y,Z,Array_size,Size
60     ASSIGN @F TO "CAR_3D"
70     ENTER @F;Object(*)
80     MAT Points= (0)
90     DEG
100    Angle=0
101    X=0
102    Y=0
103    Z=0
110    Array_size=80
120    PLOTTER IS 3,"INTERNAL"
130    GRAPHICS ON
140    SHOW -200,200,-200,200
141    ON KEY 0 LABEL "ROT/X d" GOSUB Rotxd

```

```

142  ON KEY 5 LABEL "ROT/X g" GOSUB Rotxg
143  ON KEY 1 LABEL "ROT/Y d" GOSUB Rotyd
144  ON KEY 6 LABEL "ROT/Y g" GOSUB Rotyg
145  ON KEY 2 LABEL "ROT/Z d" GOSUB Rotzd
146  ON KEY 7 LABEL "ROT/Z g" GOSUB Rotzg
147  ON KEY 3 LABEL "SCALE " GOSUB Scale
148  ON KEY 13 GOSUB Scale_
150  ON KEY 4 LABEL "TRANS " GOSUB Trans
151  ON KEY 8 LABEL "PERSP " GOSUB Persp
152  ON KEY 18 GOSUB Persp_
153  GOTO 153
154  Debut: !

```

2^e partie :

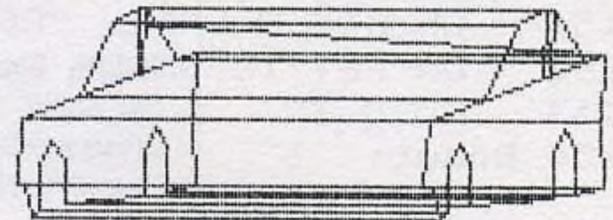
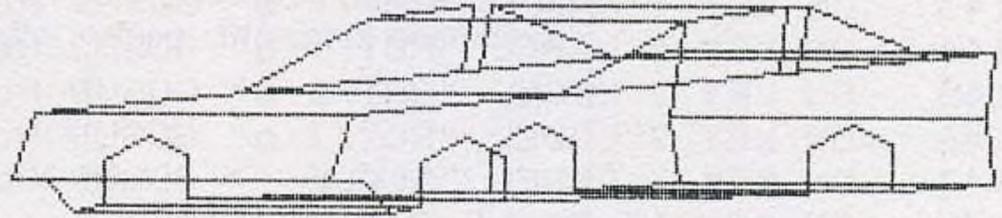
```

390 Plot:
400  MOVE 0,0
410  CALL Plot_image(Object(*),Array_size)
411  RETURN
420  END
430 Plot_image:SUB Plot_image(Points(*),Array_size)
440  ! This Plot_image routine is a special purpose routine
450  ! for use in the advanced graphics class.
460  .   OPTION BASE 1
470  .   DIM Temp(4,2)
471  .   GCLEAR
480  .   FOR J=1 TO Array_size STEP 4
490  .       IF Points(J,4)=0 THEN Points(J,4)=.0000000001
500  .       Temp(1,1)=Points(J,1)/Points(J,4)
510  .       Temp(1,2)=Points(J,2)/Points(J,4)
520  .       IF Points(J+1,4)=0 THEN Points(J+1,4)=.0000000001
530  .       Temp(2,1)=Points(J+1,1)/Points(J+1,4)
540  .       Temp(2,2)=Points(J+1,2)/Points(J+1,4)
550  .       IF Points(J+2,4)=0 THEN Points(J+2,4)=.0000000001
560  .       Temp(3,1)=Points(J+2,1)/Points(J+2,4)
570  .       Temp(3,2)=Points(J+2,2)/Points(J+2,4)
580  .       IF Points(J+3,4)=0 THEN Points(J+3,4)=.0000000001
590  .       Temp(4,1)=Points(J+3,1)/Points(J+3,4)
600  .       Temp(4,2)=Points(J+3,2)/Points(J+3,4)
612  .       FOR Plot=1 TO 4
613  .           DRAW Temp(Plot,1),Temp(Plot,2)
614  .       NEXT Plot
620  .   NEXT J
630  .   SUBEXIT
640  .   SUBEND

```

Nota : La matrice d'origine est réalisée par un autre programme. Les lignes 60 et 70 permettent de lire le fichier disque dans lequel est contenue cette matrice appelée « Object ». Le programme fonctionne à l'aide de clés préprogrammées (lignes 141 à 152). Chaque opération réalisable est divisée en deux parties : d'une part l'algorithme correspondant, d'autre part un module affectant les variables désirées.

GRAPHIQUE 3D ROTATION AXE DES X



Module 1

```

155 ! X AXIS ROTATION
156 Rotxg: Angle=-5
157       GOTO 170
160 Rotxd: Angle=5
170       CALL X_rotate(Object(*),Angle,Points(*))
180       MAT Object= Points
181       GOTO Plot

```

Sous-programme

```

650 X_rotate:SUB X_rotate(Object(*),Angle,Points(*))
660   OPTION BASE 1
670
680           ! * X_ROTATE * PROGRAM FOR X-AXIS ROTATION
690           ! Original object is Object(*)
700           ! Angle of rotation is Angle
710           ! Transformed object is Points(*)
710   DEG
720   DIM Mult_x(4,4)
730   MAT Mult_x= IDN ! 1's on the diagonal
740   !Mult_x(2,2)=COS(Angle) ! Assign values
741   Mult_x(3,3)=COS(Angle)
750   Mult_x(3,2)=SIN(Angle)
760   Mult_x(2,3)=-SIN(Angle)
770   MAT Points= Object*Mult_x ! Transform
780   SUBEXIT
790   SUBEND

```



MATRICE DE TRANSFORMATION : ROTATION AXÉ DES X

1	\emptyset	\emptyset	\emptyset
\emptyset	COS (angle)	SIN (angle)	\emptyset
\emptyset	- SIN (angle)	COS (angle)	\emptyset
\emptyset	\emptyset	\emptyset	1

L'ALGORITHME

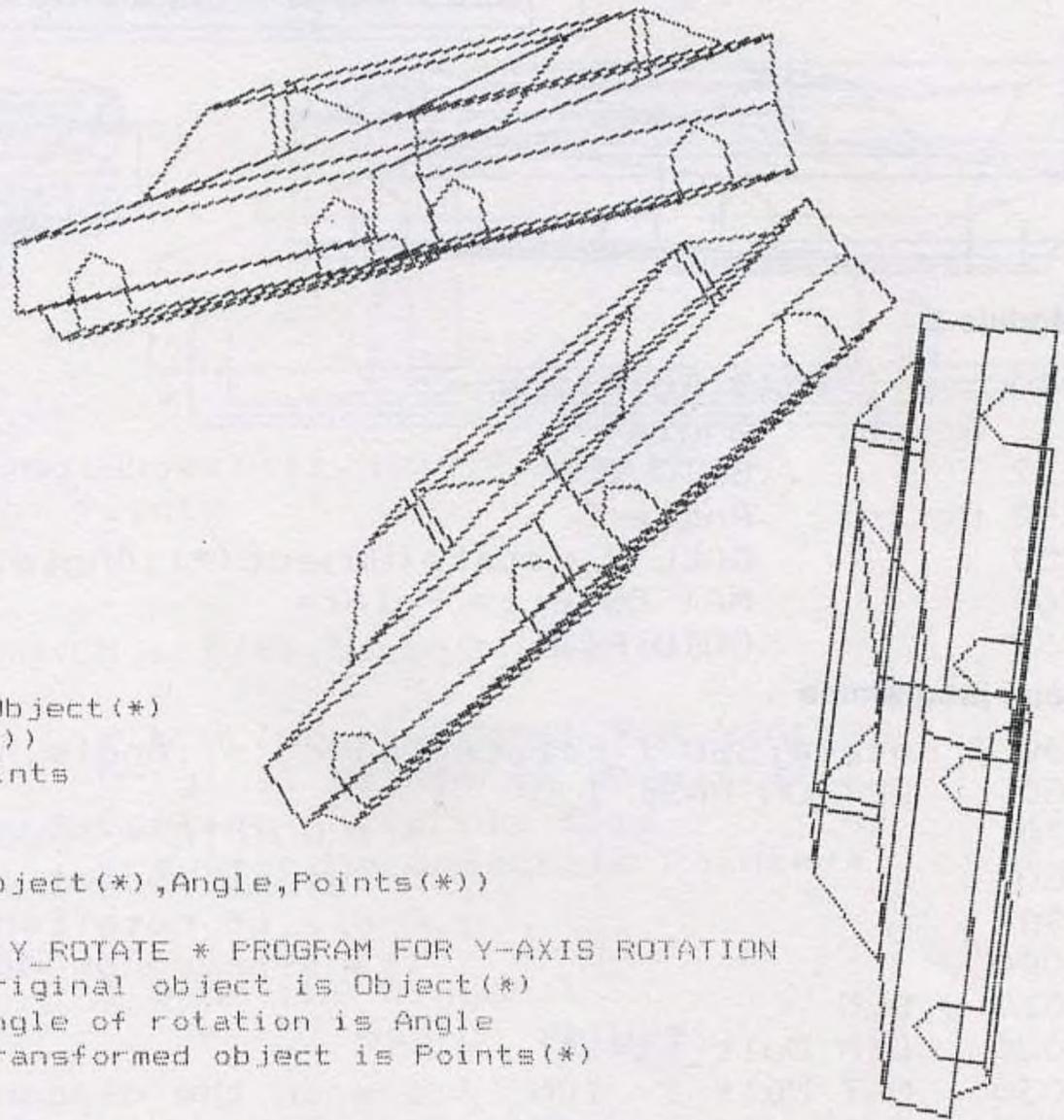
Le sous-programme de rotation dans l'axe des X se compose d'un module permettant une rotation positive ou négative dans l'axe des X. Le calcul matriciel de rotation s'effectue dans le sous-programme «X-rotate». Il part d'une matrice origine «Object» pour la transformer en une seconde matrice «Points».

Le module : en ligne 156, il y a rotation négative dans l'axe des X ;
en ligne 160, il y a rotation positive dans l'axe des X.

Nota : Le branchement aux lignes 156 ou 160 se fait grâce aux clés de fonctions préprogrammées.

Le sous-programme : On retrouve de la ligne 740 à la ligne 760 notre matrice de transformation ci-dessus. En 770, il y a multiplication de la matrice «Object» par la matrice «Mult-X» qui donne naissance à la matrice «Points» comprenant le dessin final ayant subi la transformation. En 780, retour à la ligne 180 puis appel au sous-programme dessin : «Phot-image».

GRAPHIQUE 3D ROTATION AXE DES Y



Module 2

```

190 ! Y AXIS ROTATION
191 Rotyg: Angle=-5
192      GOTO 210
200 Rotyd: Angle=5
210      CALL Y_rotate(Object(*),
                    ,Angle,Points(*))
220      MAT Object= Points
221      GOTO Plot

```

Sous-programme

```

800 Y_rotate:SUB Y_rotate(Object(*),Angle,Points(*))
810   OPTION BASE 1
820           ! * Y_ROTATE * PROGRAM FOR Y-AXIS ROTATION
830           ! Original object is Object(*)
840           ! Angle of rotation is Angle
850           ! Transformed object is Points(*)
860   DEG
870   DIM Mult_y(4,4)
880   MAT Mult_y= IDN ! 1's on the diagonal
890   Mult_y(1,1)=COS(Angle) ! Assignments
891   Mult_y(3,3)=COS(Angle)
900   Mult_y(3,1)=SIN(Angle)
910   Mult_y(1,3)=-SIN(Angle)
920   MAT Points= Object*Mult_y ! Transform
930   SUBEXIT
940   SUBEND

```

MATRICE DE TRANSFORMATION : ROTATION AXE DES Y

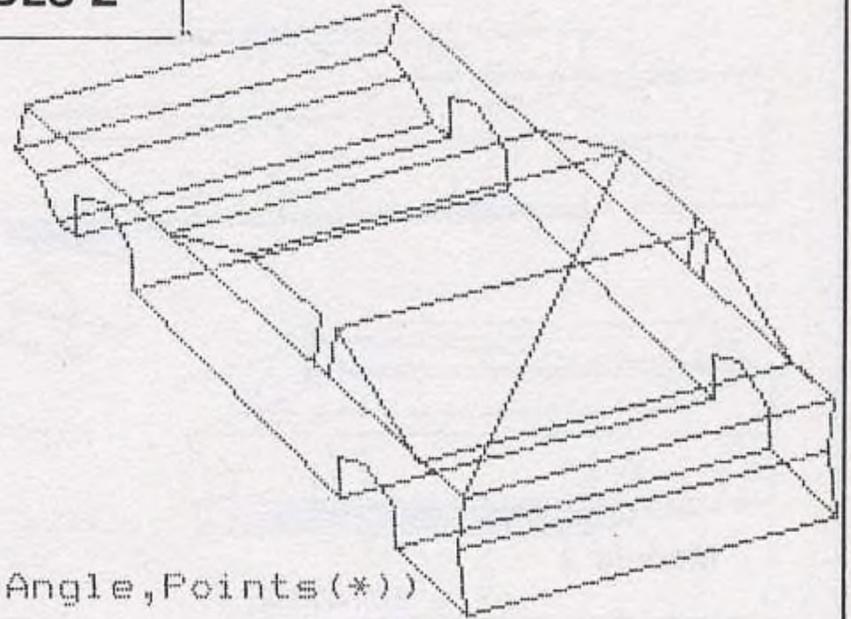
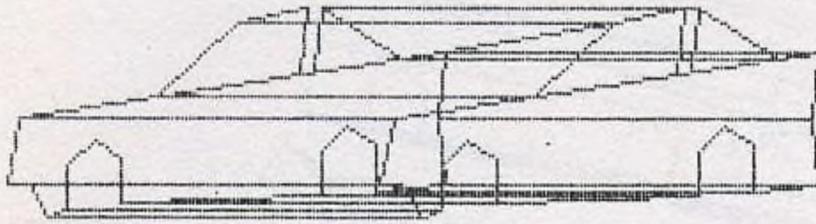
COS (angle)	\emptyset	- SIN (angle)	\emptyset
\emptyset	1	\emptyset	\emptyset
SIN (angle)	\emptyset	COS (angle)	\emptyset
\emptyset	\emptyset	\emptyset	1

L'ALGORITHME

Le sous-programme de rotation dans l'axe des Y est réalisé de la même manière que le sous-programme rotation dans l'axe des X.

Nota : Ici la matrice de transformation dans l'axe des X est remplacée par celle de transformation dans l'axe des Y (voir matrice ci-dessus).

GRAPHIQUE 3D ROTATION AXE DES Z



Module 3

```

230   ! Z AXIS ROTATION
231 Rotzg:   Angle=-5
232         GOTO 250
240 Rotzd:   Angle=5
250         CALL Z_rotate(Object(*),Angle,Points(*))
260         MAT Object= Points
261         GOTO Plot

```

Sous-programme

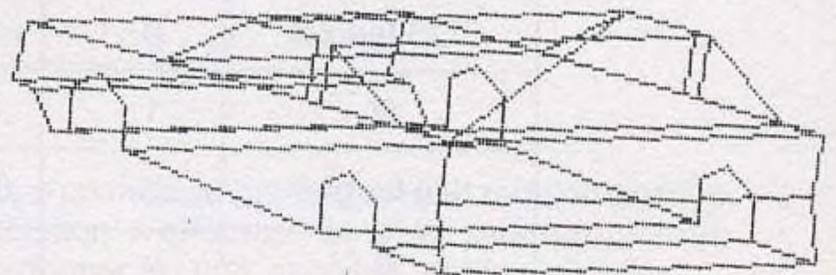
```

950 Z_rotate:SUB Z_rotate(Object(*),Angle,Points(*))
960   OPTION BASE 1
970           ! * Z_ROTATE * PROGRAM FOR Z-AXIS ROTATION
980           ! Original object is Object(*)
990           ! Angle of rotation is Angle
1000          ! Transformed object is Points(*)
1010  DEG
1020  DIM Mult_z(4,4)
1030  MAT Mult_z= IDN ! 1's on the diagonal
1040  Mult_z(1,1)=COS(Angle) ! Assignments
1041  Mult_z(2,2)=COS(Angle)
1050  Mult_z(2,1)=SIN(Angle)
1060  Mult_z(1,2)=-SIN(Angle)
1070  MAT Points= Object*Mult_z ! Transform
1080  SUBEXIT
1090  SUBEND

```

MATRICE DE TRANSFORMATION : ROTATION AXE DES Z

COS (angle)	- SIN (angle)	\emptyset	\emptyset
- SIN (angle)	COS (angle)	\emptyset	\emptyset
\emptyset	\emptyset	1	\emptyset
\emptyset	\emptyset	\emptyset	1



L'ALGORITHME

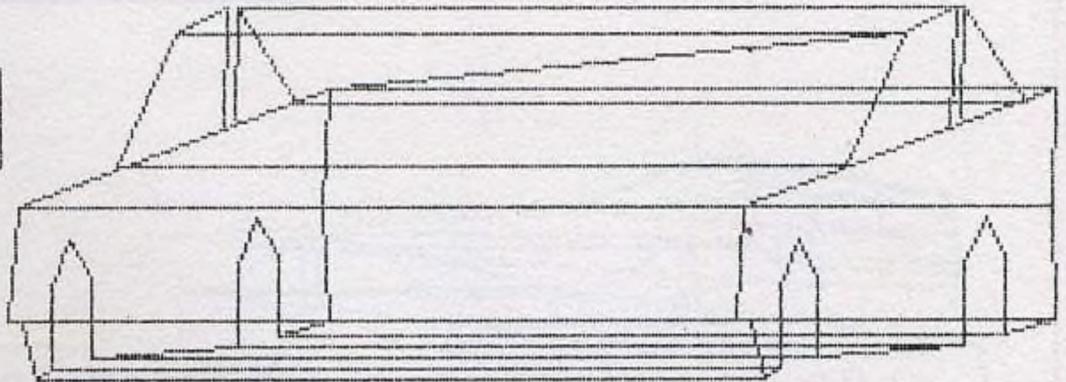
Le sous-programme de rotation dans l'axe des Z est réalisé de la même manière que le sous-programme rotation dans l'axe des X.

Nota : La matrice de transformation dans l'axe des X est remplacée par la matrice présentée ci-dessus qui autorise une rotation dans l'axe des Z.

GRAPHIQUE 3D CHANGEMENT D'ECHELLE



Module 4



```

270 ! SCALING
271 Scale_: Size=.75
272      GOTO 290
280 Scale: Size=1.33333
290      CALL Scaling(Object(*),Size,Points(*))
300      MAT Object= Points
301      GOTO Plot
  
```

Sous-programme

```

1100 Scaling:SUB Scaling(Object(*),Size,Points(*))
1110   OPTION BASE 1
1120           ! * SCALING * PROGRAM FOR SCALING
1130           ! Original object is Object(*)
1140           ! Scaling Factor is Size
1150           ! Transformed object is Points(*)
1160   DEG
1170   DIM Mult(4,4)
1180   MAT Mult= IDN      ! 1's on the diagonal
1190   Mult(1,1)=Size    ! Assign values
1191   Mult(2,2)=Size
1193   Mult(3,3)=Size
1200   MAT Points= Object*Mult      ! Transform
1220 SUBEND
  
```

MATRICE DE TRANSFORMATION : ECHELLE TROIS DIMENSIONS

X Scalaire	\emptyset	\emptyset	\emptyset
\emptyset	Y Scalaire	\emptyset	\emptyset
\emptyset	\emptyset	Z Scalaire	\emptyset
\emptyset	\emptyset	\emptyset	1

L'ALGORITHME

Le sous-programme de changement d'échelle est divisé en fait en deux parties distinctes :

1. En ligne 271, il y a réduction de l'image (Size = 0,75)

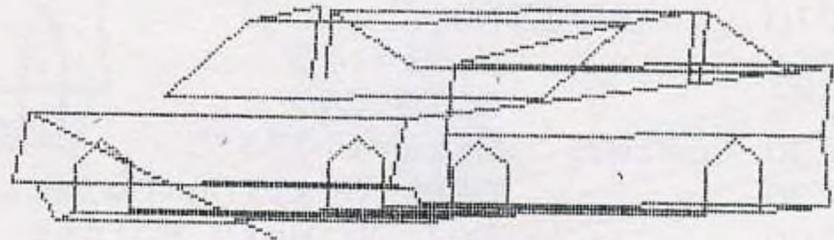
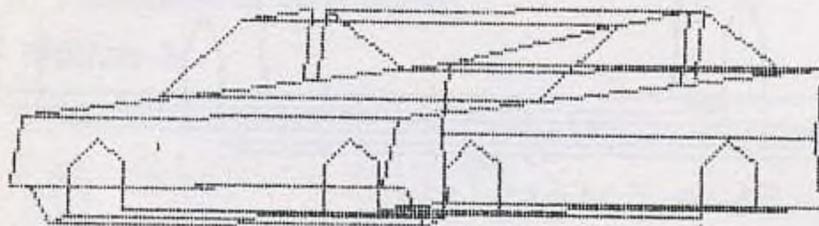
En ligne 280, il y a agrandissement de l'image (Size = 1,3333).

Nota : le branchement à la ligne 271 ou 280 se fait à l'aide de clés de fonction préprogrammées.

2. Le changement d'échelle se fait grâce à l'opération sur matrice réalisée par la routine comprise entre les lignes 1100 et 1220. A partir de la matrice de point «Object», on fabrique une opération matricielle (voir matrice ci-dessus) qui donne naissance à une seconde matrice «Points».

Une fois l'opération matricielle réalisée, il suffit de se rendre au sous-programme «Plot-image» qui tracera le dessin final à partir de la matrice «Points».

GRAPHIQUE 3D TRANSLATION



Module 5

```
310   ! TRANSLATION
320 Trans:   X=5
321           Y=5
322           Z=5
330           CALL Translate(Object(*),Points(*),X,Y,Z)
340           MAT Object= Points
341           GOTO Plot
```

Sous-programme

```
1230 Translate: SUB Translate(Object(*),Points(*),X,Y,Z)
1240   OPTION BASE 1
1250   ! Matrix multiplication for linear translation
1260   ! Original object is Object(*)
1270   ! Transformed object is Points(*)
1280   ! Perspective point is X,Y,Z
1290   ! Number of rows is Array_size
1300   DIM Translate(4,4)
1310   MAT Translate= IDN           ! 1's on diagonal
1320   Translate(4,1)=X           ! Assign delta X value
1330   Translate(4,2)=Y           ! Assign delta Y value
1340   Translate(4,3)=Z           ! Assign delta Z value
1350   MAT Points= Object*Translate ! Transform
1360   SUBEXIT
1370   SUBEND
```

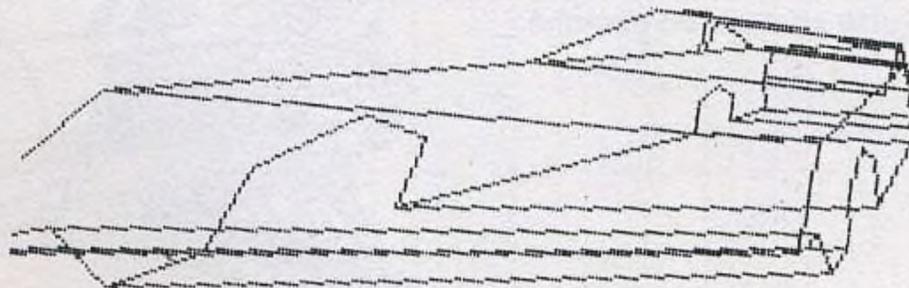
L'ALGORITHME

La translation d'un dessin en 3D est obtenue par une opération matricielle consistant à décaler des valeurs comprises dans la matrice d'origine («Object» qui contient les coordonnées d'origine du dessin), et donner naissance ainsi à une seconde matrice «Points».

Il est possible, grâce aux lignes 320 à 322 de réaliser une translation variable en fonction des valeurs données à X, Y, Z.

Le déroulement du module 5 et du sous-programme «Translate» est basé sur le même principe que les sous-programmes précédents.

GRAPHIQUE 3D PERSPECTIVE



Module 6

```
350 ! PERSPECTIVE
351 Persp_: X=-.001
352         Y=-.001
353         Z=-.001
354         GOTO 370
360 Persp: X=.001
361         Y=.001
362         Z=.001
370         CALL Perspective(Object(*),Points(*),X,Y,Z,Array_size)
380         MAT Object= Points
```

Sous-programme

```
1380 Perspective:SUB Perspective(Object(*),Points(*),X,Y,Z,Array_size)
1390     OPTION BASE 1
1391     ! Matrix multiplication for perspective transforms
1392     ! Original object is Object(*)
1393     ! Transformed object is Points(*)
1394     ! Perspective point is X,Y,Z
1395     ! Number of rows is Array_size
1400     DIM Perspective(4,4)
1410     MAT Perspective= IDN           ! 1's on diagonal
1420     Perspective(1,4)=X           ! Assign perspective X value
1430     Perspective(2,4)=Y           ! Assign perspective Y value
1440     Perspective(3,4)=Z           ! Assign perspective Z value
1450     MAT Points= Object*Perspective ! Transform
1460     FOR P=1 TO Array_size        ! Normalize X,Y,Z by homogenous coordinate
1480         Points(P,1)=Points(P,1)/Points(P,4)
1490         Points(P,2)=Points(P,2)/Points(P,4)
1500         Points(P,3)=Points(P,3)/Points(P,4)
1510     NEXT P
1520     SUBEXIT
1530 SUBEND
```

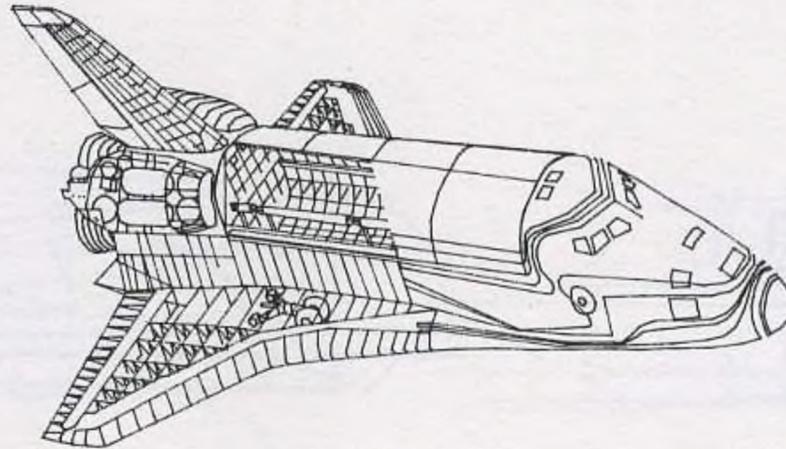
L'ALGORITHME

La réalisation d'un dessin en perspective se fait en deux étapes. Un module compris entre les lignes 350 et 380 permet de choisir entre perspective négative ou positive. Le sous-programme «Perspective» réalise une opération matricielle à partir de la matrice «Object» pour donner naissance à une seconde matrice «Points».

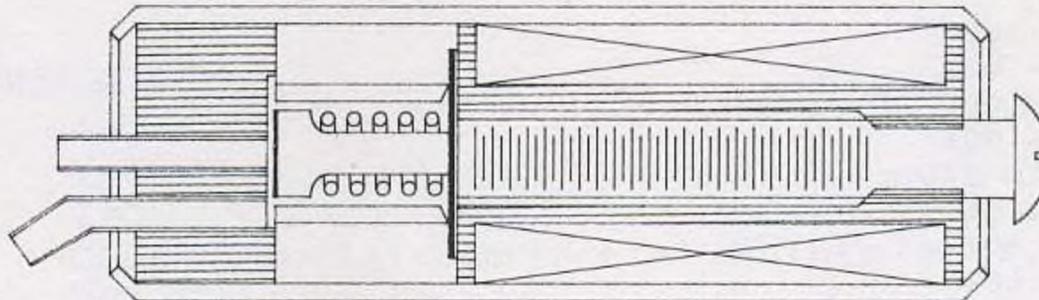
Nota : Par opposition aux différentes opérations réalisées en graphique 3D, l'opération de perspective n'est pas réversible. Il s'agit en fait d'une simulation. Il est donc impossible dans ce cas, de revenir en position initiale. En rotation, changement d'échelle et en translation il est tout à fait possible de revenir dans la position initiale de départ.

LE GRAPHISME ET LA MECANIQUE

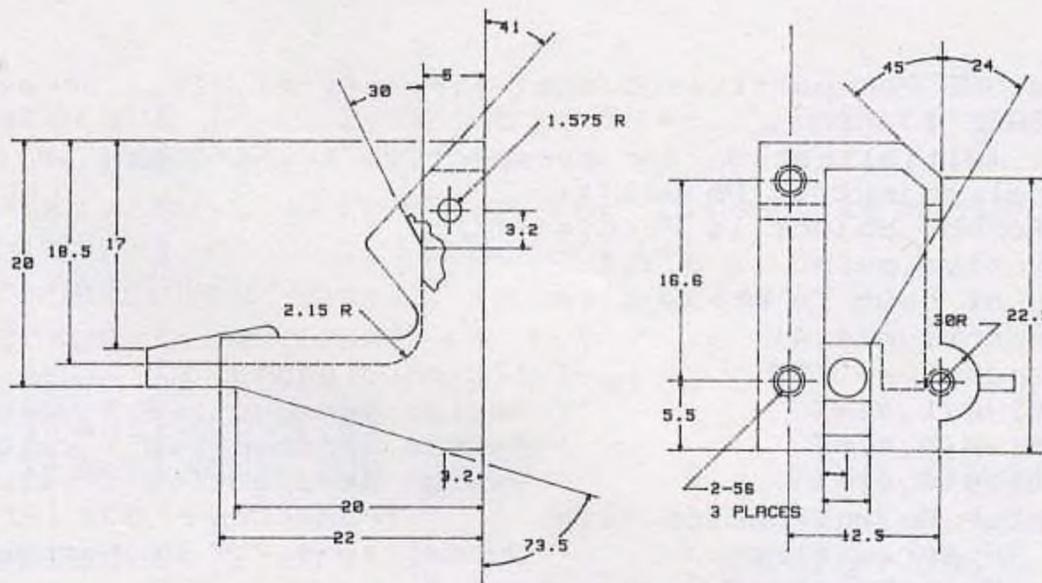
La navette spatiale en écorché.



Coupe d'une valve.



Dessin industriel.



LE GRAPHISME ET LA MECANIQUE

En mécanique, le dessin assisté par ordinateur a pris, ces dernières années, une place de plus en plus importante. Il est utilisé dans trois domaines :

1. le dessin industriel D.A.O.
2. la conception assistée par ordinateur C.A.O.
3. la conception et la fabrication assistées par ordinateur.

Dans le premier cas, il s'agit de réaliser par ordinateur les dessins industriels créés manuellement. Dans le second cas, le dessin est associé à des calculs permettant des simulations d'usage. Dans le troisième cas, le dessin réalisé en trois dimensions est lu par une machine numérique qui en fabrique automatiquement un prototype.

LE GRAPHISME ET L'ELECTRONIQUE

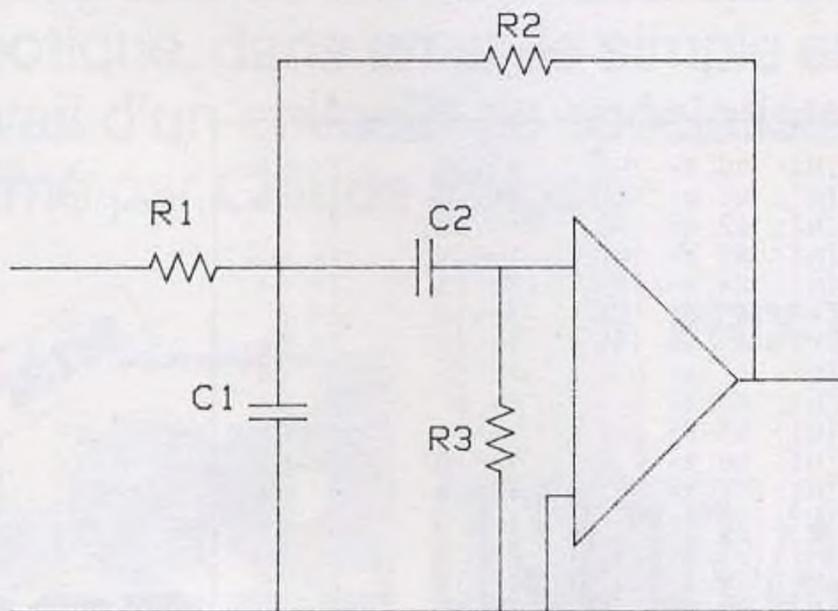


Fig. A-1

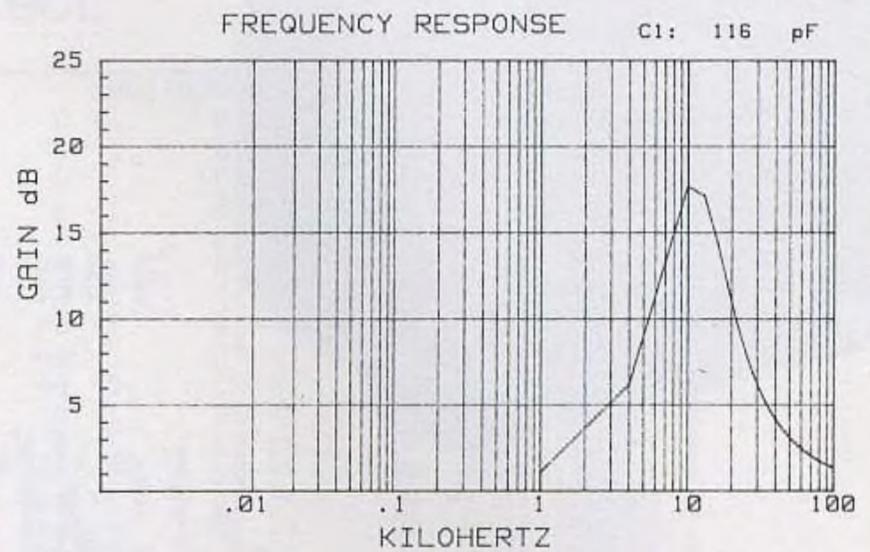


Fig. A-2

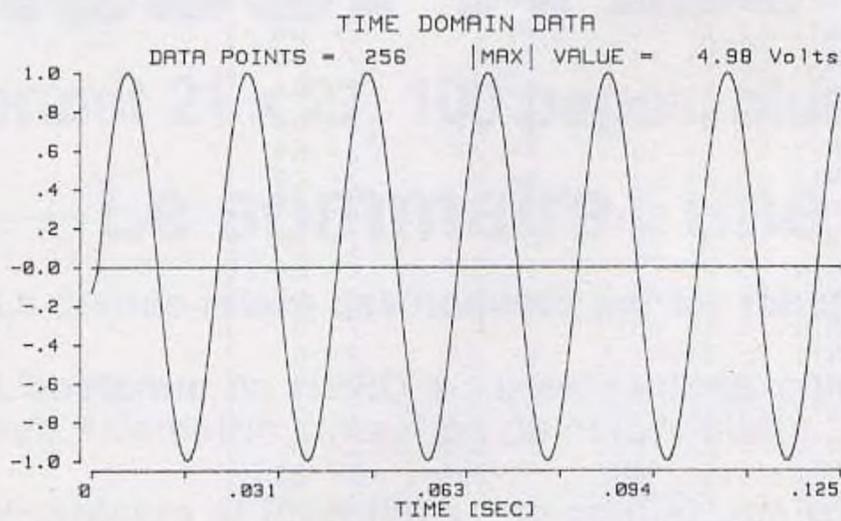


Fig. B-1 : Représentation d'une sinusoïde de 50 Hz sous 4,98 volts.

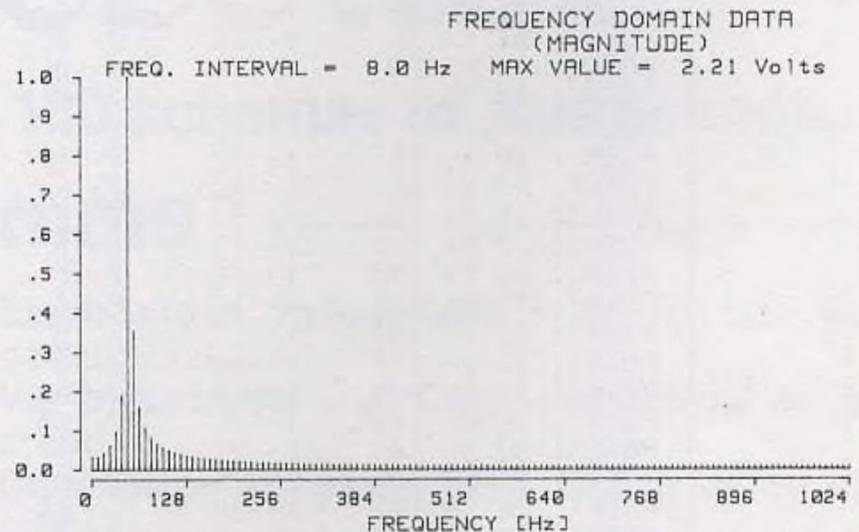


Fig. B-2 : Analyse fréquentielle de la sinusoïde.

Grâce à l'informatique, l'électronique a effectué un bon en avant spectaculaire. Aujourd'hui, l'informatique est largement utilisée dans ce domaine :

- Calcul de circuit
- Conception assistée par ordinateur-simulation
- Instrumentation associée à des convertisseurs.

Exemples :

Figures A-1 et A-2 :

Il s'agit de l'étude d'un filtre actif. Le circuit est étudié et le schéma est donné à la figure A-1. Grâce à un sous-programme de calcul et de simulation, la courbe amplitude-fréquence est présente figure A-2. Ainsi, en quelques minutes, le circuit est étudié et son fonctionnement simulé. Il est donc possible de connaître par avance les résultats d'un circuit sans le réaliser. Ceci procure un gain de temps considérable et autorise des études beaucoup plus approfondies.

Figures B-1 et B-2 :

Dans le cas de mesures en électronique, le graphisme permet une représentation optimum de données qui peuvent être captées et traitées par un ordinateur grâce à des convertisseurs analogiques digitaux. Ainsi, l'ordinateur peut se transformer en voltmètre, oscilloscope, analyseur de signaux, etc.

COORDONNEES VOITURE 3 DIMENSIONS						
POINT: 1	X= 155	Y=-10	Z= 0	POINT: 40	X= 152	Y=-20 Z= 64
POINT: 2	X= 160	Y=-20	Z= 0	POINT: 41	X= 160	Y=-20 Z= 64
POINT: 3	X= 152	Y=-20	Z= 0	POINT: 42	X= 160	Y=-20 Z= 0
POINT: 4	X= 145	Y=-25	Z= 0	POINT: 43	X= 160	Y=-20 Z= 64
POINT: 5	X= 138	Y=-24	Z= 0	POINT: 44	X= 155	Y=-10 Z= 64
POINT: 6	X= 138	Y=-24	Z= 64	POINT: 45	X= 155	Y=-10 Z= 0
POINT: 7	X= 138	Y=-17	Z= 64	POINT: 46	X= 155	Y=-10 Z= 64
POINT: 8	X= 128	Y=-13	Z= 64	POINT: 47	X= 0	Y= 0 Z= 64
POINT: 9	X= 118	Y=-16	Z= 64	POINT: 48	X= 0	Y= 0 Z= 0
POINT: 10	X= 118	Y=-23	Z= 64	POINT: 49	X= 2.5	Y=-10 Z= 0
POINT: 11	X= 118	Y=-23	Z= 0	POINT: 50	X= 2.5	Y=-10 Z= 64
POINT: 12	X= 45	Y=-22	Z= 0	POINT: 51	X= 0	Y=-20 Z= 64
POINT: 13	X= 45	Y=-22	Z= 64	POINT: 52	X= 0	Y=-20 Z= 0
POINT: 14	X= 45	Y=-15	Z= 64	POINT: 53	X= 2.5	Y=-10 Z= 0
POINT: 15	X= 35	Y=-11	Z= 64	POINT: 54	X= 2.5	Y=-10 Z= 64
POINT: 16	X= 25	Y=-14	Z= 64	POINT: 55	X= 0	Y= 0 Z= 64
POINT: 17	X= 25	Y=-21	Z= 64	POINT: 56	X= 15	Y=-1 Z= 64
POINT: 18	X= 25	Y=-21	Z= 0	POINT: 57	X= 15	Y=-1 Z= 0
POINT: 19	X= 0	Y=-20	Z= 0	POINT: 58	X= 45	Y= 7 Z= 0
POINT: 20	X= 0	Y=-20	Z= 64	POINT: 59	X= 47.5	Y=-2.5 Z= 0
POINT: 21	X= 25	Y=-21	Z= 64	POINT: 60	X= 45	Y= 7 Z= 0
POINT: 22	X= 25	Y=-21	Z= 0	POINT: 61	X= 45	Y= 7 Z= 64
POINT: 23	X= 25	Y=-14	Z= 0	POINT: 62	X= 15	Y=-1 Z= 64
POINT: 24	X= 35	Y=-11	Z= 0	POINT: 63	X= 45	Y= 7 Z= 64
POINT: 25	X= 45	Y=-15	Z= 0	POINT: 64	X= 47.5	Y=-2.5 Z= 64
POINT: 26	X= 45	Y=-22	Z= 0	POINT: 65	X= 45	Y= 7 Z= 64
POINT: 27	X= 45	Y=-22	Z= 64	POINT: 66	X= 50	Y= 7 Z= 64
POINT: 28	X= 118	Y=-23	Z= 64	POINT: 67	X= 52.5	Y=-3 Z= 64
POINT: 29	X= 118	Y=-23	Z= 0	POINT: 68	X= 50	Y= 7 Z= 64
POINT: 30	X= 118	Y=-16	Z= 0	POINT: 69	X= 75	Y= 5 Z= 64
POINT: 31	X= 128	Y=-13	Z= 0	POINT: 70	X= 75	Y= 5 Z= 0
POINT: 32	X= 138	Y=-17	Z= 0	POINT: 71	X= 50	Y= 7 Z= 0
POINT: 33	X= 138	Y=-24	Z= 0	POINT: 72	X= 52.5	Y=-3 Z= 0
POINT: 34	X= 138	Y=-24	Z= 64	POINT: 73	X= 50	Y= 7 Z= 0
POINT: 35	X= 145	Y=-25	Z= 64	POINT: 74	X= 75	Y= 5 Z= 0
POINT: 36	X= 145	Y=-25	Z= 0	POINT: 75	X= 105	Y=-6.5 Z= 0
POINT: 37	X= 145	Y=-25	Z= 64	POINT: 76	X= 105	Y=-6.5 Z= 64
POINT: 38	X= 152	Y=-20	Z= 64	POINT: 77	X= 75	Y= 5 Z= 64
POINT: 39	X= 152	Y=-20	Z= 0	POINT: 78	X= 0	Y= 0 Z= 0

RAPPEL: TOUS LES A(J,4)=1
NE PAS TENIR COMPTE DU DERNIER POINT
SCALE : -200,200,-200,200

Tableaux comprenant les coordonnées X, Y, Z, de la voiture contenues dans la matrice «Object» d'origine.

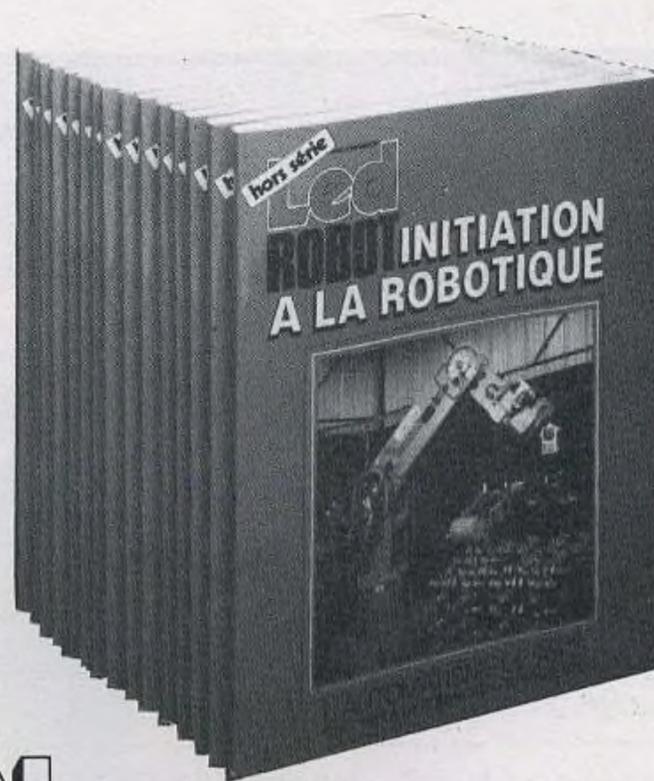
DEROULEMENT DU PROGRAMME

Exemple : rotation positive dans l'axe des X (rotation à droite)

- Ligne 141, appel de la routine rotation axe des X par la droite.
- Branchement ligne 160
 - affectation d'une rotation par angle de 5 degrés,
 - appel du sous-programme contenant l'algorithme de rotation.
- Branchement de la ligne 650 sous-programme de rotation.
- OPERATION MATRICIELLE lignes 741 à 770
 - retour au module rotation ligne 780.
- Retour ligne 180
 - affectation et changement de matrices,
 - branchement au sous-programme dessin (« plot ») en 181.
- Sous-programme de dessin : lignes 390 à 640.
- Retour au menu principal.

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

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



PRIX TTC 115 F

hors série
FRÉQUENCES D'AUJOURD'HUI
Led
ROBOT

INITIATION A LA ROBOTIQUE

Format 21 x 27, 100 pages, plus de 130 schémas et illustrations.

Le sommaire : une somme !

- **La grande relève des hommes par les robots**
- **L'anatomie de HERO 1** : bras, jambes, ouïe, vue, télémétrie, détection de mouvements.
- **Inventeurs et inventions** : ne confiez pas vos inventions avant de vous être protégé.
- **Cours de conception mécanique** : vocabulaire et notion de base - Ajustement, tolérance, excentricité, etc.
- **Cours de logique générale** : schémas et symboles.
- **Electronique industrielle** : du circuit au démultiplexeur.
- **Vie industrielle** : la CAO, assistante de la création.
- **Conception et construction** : de la tortue au robot.
- **Modules fonctionnels** : construction de la carte de départ pour commander les moteurs pas à pas à partir de votre micro.
- **Maquettes et modélisme** : le modélisme ferroviaire se renouvelle grâce à la micro-informatique.
- **Analyses et méthodes** : les rosaces d'évaluation.

BON DE COMMANDE



Je désire recevoir Led-Robot «INITIATION A LA ROBOTIQUE» (attention, cet ouvrage n'est pas vendu en kiosque) au prix de 125 F (port compris).

Nom : Prénom :

Adresse :

ATTENTION : Si je suis abonné soit à LED, soit à LED-MICRO, je bénéficierai d'une réduction de 20 % sur le prix de l'ouvrage et je ne paierai que 100 F (port compris).

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

Ci-joint un chèque bancaire chèque postal mandat .

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

PETITES ANNONCES

Vds Apple IIC + moniteur IIC + souris + logiciels. Matériel sous garantie. Tél. : 45.92.86.46 le soir.

A vendre Hector HRX 64 k (Forth+Basic) + disc. 2 2x800 k (juin 85)+moniteur couleur+imprim. Seikosha GP 100 A + 50 K7 jeux et prog. (Pyrentexte, etc.)+ diskettes prog. (Visicalc, etc.)+ 15 diskettes + nombreuses doc. Prix : 12 000 F. Tél. 56.21.86.98 ap. 20 h.

Vds ZX81 + 16 K + alim. + manuel + livres 70 progs + cass. stock-car + cass. vierge : 900 F. Bannier Patrice 3, rue des Ecoles 28330 Authon-du-Perche.

Vends doubleur de densité pour TRS ou Prof 80 avec 1791 neuf et testé : 500 F. Tél. : 40.95.08.24 (province).

Vends micro-ordinateur Tandy TRS 80, color 2 Basic étendu 64 k+K7 Tandy CCR 82+fichier color file. L'ensemble : 2 000 F. Tél. : (1) 60.16.72.98.

Vends Apple IIe, 128 K, 80 c., carte Z-80, moniteur, 2 disk, joystick, 100 disquettes avec nombreux programmes utilitaires, nombreux livres Apple, le tout très peu servi : 13 000 F. Tél. 93.05.06.34 (Alpes-Maritimes).

Vends ZX-81 + 16 K + cassettes jeux échecs + bibliothèque : 700 F à débattre. Sarotte, tél. (1) 39.51.00.00 (poste 41.14) ou soir (1) 39.69.51.91.

Cherche possesseur de MSX pour échanger des programmes, j'en possède + de 180. vends aussi un MSX (PMC 28 S 32 K) + ext. 64 K + joystick + une centaine de programmes + manuels. Faire offre à Stéphane Le Coq 16, rue Fromont 37000 Tours. Tél. (16) 47.37.50.99 après 18 h 45.

Vends ZX-81 + 16 K + clavier ABS + livre + Mg k7 : 900 F. TV jeu N60 avec écran N.B. incorporé + 2 manettes + 1 cassette jeu, 1 cassette programmation avec notice et schémas : 950 F. Tél. (1) 43.57.42.33 le soir.

Achète pour TRS-80 modèle 1, prgs Visicalc, Scripsit, Accel 3/4, New-Dos 80 version 2. Tél. (1) 43.70.06.61 après 18 heures.

Vends livres sur TI et revues. List n°s 1, 2, 3, 4, 5, 6, 8, 9. Votre ordinateur : n°s 2 à 14. SVM : n° 1 à 24. Hebdo : n°s 12, 13, 15 à 17, 57 à 59, 61, 64, 66 à 69, 77 à 78, 80, 84, 94 à 97. M. Laval Daniel rue de la Ressence 06220 Vallauris. Tél. (16) 93.64.26.12 après 20 h ou (16) 93.62.26.66 h.b.

Echangerai amicalement tous programmes pour Amstrad CPC 6128 (disquette). Tél. 35.71.92.97 Rouen.

Vends ordinateur Sharp PC 1500 84 + CE 150 + CE 155 + accessoires + documentation, l'ensemble : 3 500 F. Tél. heures de bureau : (16) 47.91.23.66.

BON DE COMMANDE

Pour compléter votre collection de Led-Micro

A retourner aux EDITIONS FREQUENCES 1, boulevard Ney - 75018 Paris

Je désire le n° (cocher le ou les n°s désirés)
 . 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26

AU PRIX DE 20 F par numéro (port compris).

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

NOM Prénom

Adresse :

Ville : Code postal :

Bulletin d'Abonnement

Je désire m'abonner à Led Micro (10 numéros). France : 160 F - Etranger : 240 F, à partir du n°

Nom Prénom

N° Rue

Ville Code Postal

Envoyez ce bon accompagné du règlement à l'ordre des Editions Fréquences à :
 EDITIONS FREQUENCES 1, boulevard Ney, 75018 PARIS

MODE DE PAIEMENT : CCP - Chèque bancaire - Mandat

TOUT SUR LES PÉRIPHÉRIQUES

NOUVEAU

dans la
COLLECTION
«ETUDES»
aux
éditions
fréquences



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

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

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

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

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

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

En vente chez votre libraire et aux Editions Fréquences.

BON DE COMMANDE

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

Nom

Adresse

A adresser aux EDITIONS FRÉQUENCES 1 boulevard Ney, 75018 Paris

Règlement ci-joint :

Par chèque bancaire

par chèque postal

par mandat

Le Victor PC ne coûte que 24.900 F n'en déplaie à [REDACTED].

Le Victor PC 15 ne coûte que 24.900 F*.

Certains d'entre vous penseront peut-être – et nous en connaissons qui aimeraient bien que ce soit vrai – qu'à 24.900 F*, il ne peut s'agir que d'un PC "bradé". Une telle réaction est d'ailleurs compréhensible quand on songe aux prix pratiqués sur le marché, en matière de PC. Prenons par exemple [REDACTED]. Son PC coûte 50% plus cher que le Victor PC 15.

Et pourtant, les performances du Victor PC 15 sont équivalentes, voire supérieures, à celles de l'[REDACTED] PC. La preuve, la voici :

Alors que la plupart des micro-ordinateurs propose une capacité de stockage de 10 Mo, le Victor PC 15, lui, offre une capacité de 15 Mo! De plus, l'utilisateur du Victor PC 15 bénéficie, grâce à un moniteur de 14 pouces, de 30% de surface écran supplémentaires (la quasi-totalité du matériel concurrent étant équipée d'un moniteur 12 pouces).

Et ce n'est pas tout! Le Victor VU – l'interface utilisateur – permet un gain de temps appréciable en guidant dans son travail l'utilisateur, par de simples messages organisés comme des menus. Finie, désormais, la consultation fastidieuse et peu pratique du manuel du système d'exploitation!

Et l'on pourrait parler des 5 emplacements d'extensions disponibles pour accroître les possibilités du PC...

Non décidément, [REDACTED] devra se faire une raison et s'accommoder de la présence sur le marché du Victor PC 15! Un PC compatible avec les standards du marché, aussi performant que celui que fabrique [REDACTED] et à un prix bien plus séduisant que celui affiché par [REDACTED].

Car au risque de le répéter et de déplaire à [REDACTED], ces 50% sont difficilement justifiables. D'ailleurs les vendeurs d'[REDACTED] doivent déjà en savoir quelque chose...

Lesquels vendeurs d'[REDACTED] ne vont sans doute guère apprécier que nous vous donnions nos coordonnées - et que vous puissiez nous contacter à Victor Technologies - Tour Horizon, 52, quai de Dion-Bouton, 92800 Puteaux (tél. : 778.14.50) ; ou encore à Lyon : (7) 234.12.45 ; Montpellier : (67) 64.71.72 ; Nantes : (40) 89.24.28. Mais l'on ne peut contenter tout le monde et [REDACTED]!



* Configuration complète avec clavier et écran monochrome. Prix H.T. au 1/9/85. (Possibilité de location financière : 700 F par mois sur 48 mois - CEGEDATA.).

VICTOR

Comme [REDACTED] moins cher qu'[REDACTED]