

Mini-système expert

pour Apple II

De quelle couleur est le stylo de François ?

Quel est le prénom du maire de la ville phocéenne ?

A quoi est égal $f(g(z))$, sachant que $x = g(z)$ et que $y = f(x)$?

Si tout logicien est incompris et que tout homme sensé est logicien, que peut-on en conclure ?

C'est à toutes ces questions, et à beaucoup d'autres, que le présent programme est capable de répondre.

Dans le numéro 44 de *Micro-Systèmes* (juillet-août 1984), nous avons présenté un programme d'« Intelligence Artificielle » en Basic qui était une première version simplifiée du logiciel présenté ici.

Le but de ce programme était de permettre la constitution et l'interrogation en langage naturel d'une base de connaissances.

Dans sa nouvelle version, le présent programme fonctionne de la même manière, mais représente cependant une amélioration par rapport à sa version précédente, en ce sens qu'il est capable de trouver les relations verticales susceptibles d'exister entre différentes assertions présentes dans la base.

Avant d'apporter davantage de précisions, donnons les grandes lignes du mode d'emploi et du fonctionnement général du programme.

Mode d'emploi : les entrées du programme

Le programme accepte trois types d'entrées :

- Des « connaissances », exprimées en langage naturel sous la forme d'assertions simples, qui ne doivent contenir **aucune ponctuation**.

- Des interrogations sur la base, également exprimées en langage naturel et **obligatoirement** terminées par un point d'interrogation ;

- Des commandes de service, exprimées sous une forme symbolique, et qui permettent de gérer la base. Ces commandes commencent par le caractère « / » (slash, ou barre de fraction) et sont au nombre de trois :

- la commande « **/L** » donne la liste de toutes les assertions présentes en mémoire, chacune étant précédée de son rang dans la base ;

- la commande « **/Dx** » permet d'effacer la xième assertion ; la commande « **/D** » sans indication numérique autorise l'effacement de la toute dernière assertion entrée (commande en cas d'erreur de saisie...) ;

- la commande « **/EFF** » fait disparaître toutes les assertions et nettoie donc entièrement la base.

Fonctionnement général

La base elle-même est une table à deux dimensions, constituée de « tiroirs » empilés les uns sur les autres et dont chacun contient une assertion. La première dimension de la table correspond au rang de l'assertion dans la base. Pour le présent programme, ce nombre total de tiroirs a été fixé à 100 (ligne 100, variable NM), mais il peut bien sûr être modifié en fonction de la taille mémoire dont dispose votre propre ordinateur.

Les **assertions** attendues par le programme sont des phrases simples, constituées de mots séparés par des espaces. Chacun de ces mots est considéré par le programme comme **signifiant** ou **non signifiant**. Seuls les mots signifiants sont conservés en mémoire, en plus de l'assertion complète, afin de procéder une recherche ultérieure lors d'une interrogation.

Prenons un exemple : l'assertion

LIMOGES EST LA CAPITALE DE LA PORCELAINE

**INTELLIGENCE ARTIFICIELLE :
Gestion relationnelle
d'une base de connaissances**
de Philippe LARVET

Après l'avoir éduqué, recherchez en utilisant le langage naturel les liens entre les connaissances de votre ordinateur.

Langage : Basic

Ordinateur : Apple II

est formée de sept mots, mais seuls les mots LIMOGES, CAPITALE et PORCELAINE sont conservés par le programme, car ils représentent l'essentiel du contenu de l'assertion. Les autres mots, considérés comme non signifiants, sont éliminés.

Les mots non signifiants reconnus par le programme sont précisés dans une liste de DATA (lignes 400-430) qui peut être facilement modifiée.

Le programme, ignorant ces mots, est donc capable d'accepter des **assertions brutes**, c'est-à-dire formées uniquement de mots signifiants. Par exemple : PIERRE FRERE JEAN. Il en est de même des interrogations, pour lesquelles on pourra se contenter de poser une question formée d'un groupe de mots signifiants suivis d'un point d'interrogation.

Cette élimination des mots non signifiants est indispensable, à la fois pour permettre au programme de « raisonner » uniquement sur la matière brute de l'assertion et pour des raisons évidentes de simplicité et de vitesse de traitement. Toutefois, il y a lieu d'en tenir compte dans le cas de certaines assertions très courtes qui pourront réservier des surprises.

Si vous entrez, en effet, une phrase du genre :

LE TAO EST LE UN

ou

LES DES SONT JETES

le programme ne retiendra qu'un seul mot signifiant (TAO ou JETES) et rejettéra l'assertion avec le message : **VOTRE PHRASE EST TROP COURTE !

En effet, comme il est précisé

plus loin, toute assertion est attendue par le programme comme une relation horizontale appliquée à un sujet, ce qui oblige à la présence d'un minimum de deux mots signifiants (dans la deuxième phrase, « DES » est assimilé à l'article indéfini et non pas au cube des jeux de hasard).

Le nombre total de mots signifiants susceptibles d'être conservés en mémoire pour chaque assertion est fixé à 4 (ligne 100, variable NP) mais peut être ramené à 3, qui est sa valeur minimum (comme expliqué plus loin).

Cette valeur définit le nombre maximum de « places » d'un tiroir donné de la base, et correspond au second indice de la table.

Le nombre de mots signifiants contenus dans une assertion doit être supérieur à un et inférieur ou égal au nombre de places autorisées à chaque étage de la base. Dans le cas contraire, un message d'erreur sera émis par le programme.

Les **interrogations** ont la même structure que les assertions : elles sont constituées de mots séparés par des espaces. Seule, la présence d'un point d'interrogation en fin de phrase autorise le programme à distinguer une interrogation d'une assertion.

Dans un premier temps, l'interrogation est traitée comme l'assertion : les mots non signifiants sont éliminés. Chacun des mots signifiants est ensuite recherché dans la base, étage par étage, et la réponse à l'interrogation est déterminée en conséquence.

Syntaxe générale des assertions

Le programme, n'ayant pas d'*intelligence* réelle, n'effectue aucune analyse **sémantique** de l'assertion entrée et ne possède donc aucune connaissance de la signification de celle-ci. Nous pouvons ainsi le « piéger » assez facilement.

Toutefois, il est important de noter que, pour obtenir une signification satisfaisante des réponses attendues, toute assertion entrée doit être conforme à un schéma **syntaxique** type, virtuellement présent dans la base et qui est de la forme :

SUJET-RELATION-OBJET.

Selon ce schéma, et pour toute assertion, le premier mot signifiant extrait est considéré comme le **sujet** de l'assertion, le second comme un **prédictat** du sujet, ou bien comme une **relation** (ou une **action**), et le troisième comme **l'objet**, au sens grammatical, de la relation ou de l'action.

Le quatrième mot extrait, s'il est présent, doit se rapporter à l'objet, pour lequel il joue le rôle d'adverbe ou de qualificatif. On peut toutefois, bien entendu, jouer avec la présence de la relation en seconde ou en troisième place, ou entrer des assertions du type « 1515 MARIGNAN ». Tout dépend ensuite des interrogations que l'on fait.

Il ne s'agit donc pas là d'une contrainte technique, mais plutôt d'une précaution à suivre afin de tirer le meilleur parti de l'utilisation du programme.

Recherche des relations horizontales

Si l'interrogation ne porte que sur un seul mot en entrée, le programme listera toutes les assertions contenant ce mot, sauf, bien entendu, si ce mot est considéré par le programme comme non signifiant.

Dans le cas de la recherche d'une **relation horizontale**, si l'interrogation porte sur plusieurs mots signifiants, il est important de noter également qu'en plus de l'**existence** dans la base de chacun des mots signifiants, l'**ordre** dans lequel ceux-ci ont été agencés au sein de l'interrogation est fondamental.

En effet, et malgré le cas fréquent en français de l'inversion du verbe et du sujet dans une

phrase interrogative, le programme ne sera capable de répondre correctement que s'il trouve une correspondance horizontale entre les mots de l'interrogation et les mots présents dans la base, et seulement si les deux ensembles de mots sont dans le même ordre.

Cette contrainte, d'ordre technique cette fois, a été imposée du fait de l'existence d'un grand nombre de verbes dont l'action n'est pas commutative.

Par exemple, l'assertion : **SABINE AIME JACQUES** ne signifie pas nécessairement que Jacques aime Sabine !

La réponse à la question : **QUI AIME SABINE ?** doit donc être **JE L'IGNORE** et non **JACQUES**.

De même, à la question : **QUI JACQUES AIME-T-IL ?** la réponse ne doit pas être **SABINE**.

On peut trouver de nombreux autres exemples qui abondent dans ce sens.

Les relations verticales

En ce qui concerne la recherche de relations verticales entre différentes assertions, le programme est capable de répondre à des questions indirectes et de mettre en évidence des liens qui peuvent être de trois types différents : l'indirection, la composition de fonctions, les syllogismes.

L'indirection

Ce que nous appelons ici **indirection** est la possibilité pour le programme de trouver le mot signifiant commun à deux assertions **distinctes**, et donc de répondre à une question portant indirectement sur ce mot commun.

Prenons l'exemple des deux assertions suivantes :

PLATON EST UN PHILOSOPHE

PLATON EST GREC

Ces deux phrases sous-entendent que nous avons entré dans la base la connaissance selon laquelle il existe un philosophe grec nommé Platon. Ceci nous paraît évident et nous sommes capables de le **déduire** malgré l'absence de relation **directe** entre les deux prédictats philosophe et grec. Il en est de même du programme, qui peut répondre correctement à l'une quelconque des questions suivantes :

QUEL PHILOSOPHE EST GREC ?

ou

QUEL GREC EST PHILOSOPHE ?

ou bien

Y A-T-IL UN PHILOSOPHE GREC ?

ou encore

QUI EST GREC ET PHILOSOPHE ?

ou, plus simplement,

PHILOSOPHE GREC ?

Prenons un autre exemple, à partir des assertions :

LE CANARI EST UN OISEAU JAUNE

JAUNE EST UNE COULEUR

Le programme – en plus, bien sûr, de la réponse aux questions directes **QUEL OISEAU EST JAUNE ?** ou **QUEL EST L'OISEAU JAUNE ?** – est capable de répondre à l'une de ces questions : **DE QUELLE COULEUR EST LE CANARI ?**

ou

QUELLE EST LA COULEUR DU CANARI ?

ou, plus simplement,

COULEUR CANARI ?

L'intérêt de l'indirection apparaît surtout lorsqu'il est nécessaire de chercher le lien existant entre deux assertions, ou plusieurs groupes de deux assertions, situées à des rangs différents dans la base et parmi un grand nombre d'autres assertions.

Reprenons le premier exemple ci-dessus : si, après cinquante nouvelles assertions, vous entrez :

ARISTOTE ETAIT GREC
et

ARISTOTE ETAIT UN PHILOSOPHE

à la question :

GREC PHILOSOPHE ?

vous ne serez pas mécontent de voir apparaître, en plus du nom d'**ARISTOTE**, celui de **PLATON**, que vous aviez peut-être oublié ou dont vous ne connaissiez pas nécessairement l'existence dans la base (cela dépend en effet de l'utilisation que l'on fait du programme : jeux, enseignement, etc.).

Un autre intérêt de l'indirection réside dans la précision supplémentaire que le programme peut apporter, dans certains cas, à ses réponses.

Dans le cas des deux assertions suivantes, par exemple : **MARIE EST UNE JOLIE FILLE**

JOLIE EST LE CONTRAIRE DE LAIDE

à la question :

MARIE EST-ELLE LAIDE ?

ou

EST-CE QUE MARIE EST LAIDE ?

la version précédente du programme (voir *Micro-Systèmes*, n° 44), ne trouvant aucune relation horizontale entre le sujet **MARIE** et le prédicat **LAIDE**, répondait simplement : **NON**.

Le présent programme apporte quelque chose de plus puisqu'il donne comme réponse le terme commun aux deux assertions : **JOLIE**. Ce qui signifie implicitement : **MARIE N'EST PAS LAIDE, ELLE EST JOLIE**. (Pour que le programme sache faire une réponse telle que celle-ci, il serait nécessaire de l'étoffer avec un solide analyseur sémantique, ce qui dépasse, de loin, ses capacités actuelles.)

A titre d'anecdote, citons une autre application de l'indirection (nous sommes sûrs que vous en trouverez bien d'autres!). Sans autoriser le calcul numérique, ce qui n'est pas son but, le programme accepte cependant des assertions plus formelles et pseudo-mathématiques, du genre :

$$X_1 = X_2$$

$$X_2 = X_3$$

ce qui permet de demander : **PAR QUOI** (c'est-à-dire par quel intermédiaire) $X_1 = X_3$? et d'obtenir la réponse : **X2**.

La composition de fonctions

Si l'indirection permet de trouver le lien existant entre deux assertions, la composition de fonctions est plus intéressante, car elle autorise la relation entre elles d'un nombre plus grand d'assertions, comprises entre 2 et 10 (pour modifier cette valeur limite, il suffit d'agrandir les tables C\$, W\$ et E).

Ces assertions peuvent être entrées dans un ordre quelconque, mais doivent remplir deux conditions pour avoir des réponses satisfaisantes :

- elles doivent toutes être construites selon le schéma type précisé plus haut : sujet-relation-objet, dans lequel la relation est considérée comme une **fonction** et l'objet comme l'**argument** de cette fonction ;
- l'objet/argument d'une assertion doit être repris comme

sujet d'une autre assertion.

Par exemple, à partir des assertions :

MARSEILLE EST LA VILLE
PHOCEENNE
DEFERRE EST LE MAIRE
DE MARSEILLE
PHOCEENNE SIGNIFIE
ORIGINAIRE DE PHOCEE
LA PHOCEE EST UNE
PROVINCE GRECQUE
GASTON EST LE PRENOM
DE DEFERRE

le programme est capable de répondre à la question :

QUEL EST LE PRENOM
DU MAIRE DE MAR-
SEILLE ?

ou

QUEL EST LE PRENOM
DU MAIRE DE LA VILLE
ORIGINAIRE D'UNE PRO-
VINCE GRECQUE ?

Le moyen de résolution employé est simple : le programme réalise l'**effacement** progressif de la question, de droite à gauche, et simplifie celle-ci jusqu'à parvenir à la réponse finale.

Par exemple, la réponse à la question :

QUEL EST LE PRENOM
DU MAIRE DE LA VILLE
PHOCEENNE ?

est déterminée de la façon suivante : VILLE PHOCEENNE est d'abord trouvé dans la base et remplacé par la valeur de son sujet : MARSEILLE. Le programme efface donc une partie de la question initiale et la remplace par la nouvelle question à résoudre, plus simple que la précédente et plus proche de la réponse :

QUEL EST LE PRENOM
DU MAIRE DE MAR-
SEILLE ?

MAIRE DE MARSEILLE est ensuite trouvé dans la base et remplacé par sa valeur : DEFERRE. Nouvel effacement et nouvelle question :

QUEL EST LE PRENOM DE
DEFERRE ?

Ces deux derniers mots sont enfin trouvés et remplacés par leur valeur : GASTON. La question ayant été entièrement effacée, la dernière valeur trouvée correspond à la réponse.

L'ordre dans lequel les assertions sont présentes dans la base est indifférent. Notons également que l'**objet** de chaque assertion peut être accompagné d'un qualificatif, qui prendra alors la quatrième « place » dans le tiroir réservé à l'assertion, mais que l'algorithme de recherche de la réponse ne tiendra pas compte de

cette quatrième place ; l'algorithme ne traite en effet que les places 2 et 3 de chaque assertion.

Par exemple, les assertions : PAUL POSSEDE UN PERROQUET BAVARD MULTICOLORE SIGNIFIE DE PLUSIEURS COULEURS UN PERROQUET EST UN ANIMAL MULTICOLORE autorisent le programme à répondre à la question : QUI POSSEDE UN ANIMAL DE PLUSIEURS COULEURS ?

sans que le qualificatif BAVARD ait été utilisé. A titre anecdotique également, le traitement des compositions de fonctions permet au programme de répondre à des interrogations plus formelles, telles que : F(G(H(K(X1))) = ? ou

A QUOI EST = F DE G DE H DE K DE X1 ?

sachant que :

Y1 = K(X1)

X2 = H(Y1)

X4 = F(X3)

et que :

X3 = G(X2)

Le programme vous donnera la réponse finale : X4 (essayez et vous verrez !). Mais attention, ici, à ne pas utiliser comme « variables » les lettres A, C, D, L, N, S, T, Y, qui sont considérées comme mots non signifiants et sont donc ignorées !

Le traitement des syllogismes

Tout le monde connaît le célèbre argument :

« Tout homme est mortel
Or Socrate est un homme
Donc Socrate est mortel. »

Il s'agit d'un « syllogisme catégorique » de la première figure. Il existe en effet plusieurs **figures** du syllogisme, dont l'origine remonte à Platon et Aristote et que F. Chenique, dans son ouvrage « Eléments de logique classique » (*), définit comme : « Une argumentation dans laquelle, de deux propositions simples disposées de façon déterminée, une troisième proposition découle nécessairement. »

Le syllogisme est donc un discours dans lequel interviennent trois propositions, construites de manière rigoureuse. Les deux premières sont appelées **prémisses** (ou antécédent) du syllogisme, et la troisième,

Tableau des variables utilisées

Alphanumériques		Numériques	
A\$	Entrée	NP	Nombre de places dans RC
B\$ (100,5)	Table de la base	NM	Nombre maxi de tiroirs
C\$ (10)	Mots signifiants extraits de l'assertion	NT L I,J,K	Nombre de tiroirs occupés LEN (A\$) Indices de boucles
E\$ (60)	Table des mots non signifiants	M A	Mémo d'indice (optimisation) Mémo de M
F\$	Mot extrait de A\$, quel qu'il soit	T,V J1	Indicateurs Mémo de J de B\$ (I,J)
G\$	Lettre extraite de A\$ pour comparaison	ND	Nombre de mots signifiants extraits de A\$
H\$	Concaténation de tous les C\$ (i)	NE	Nombre de mots non signifiants dans la table E\$
X\$	Extrait de l'assertion (petit terme) pour syllogisme	JM	Place du moyen terme dans la prémissse majeure
W\$ (10)	Sauvegarde de C\$ (i)	KM	Place du moyen terme dans la prémissse mineure
Z\$	Concaténation de deux ou trois mots signifiants et extrait de l'assertion (grand terme) pour syllogisme	NI P(2) S (100,3)	Indicateur « interrogation » Mémo ponctuelle de J Contient, pour chaque tiroir, le nombre de mots signifiants de l'assertion D (2) E (10)
			Sauvegarde de J lors de l'extraction de C\$ (i) Premier étage de la base où chaque C\$ (i) est présent

Tableau 1. – Liste des conclusions pouvant être obtenues en fonction des combinaisons des propositions initiales.

qui en « découle nécessairement », est appelée **conclusion** (ou conséquent). Les prémisses – majeure et mineure –, sont elles aussi, composées de trois termes : le grand terme (T), le petit terme (t) et le moyen terme (M).

Le grand terme sert de **prédictat** à la conclusion, dans laquelle le petit terme sert de **sujet**. Le moyen terme, quant à lui, met en rapport dans les prémisses le grand terme et le petit terme, et n'apparaît donc pas dans la conclusion.

On comprendra aisément que, selon le rapport qui relie les termes extrêmes au moyen terme et la position de ces termes au sein des prémisses, le syllogisme puisse prendre plusieurs formes, appelées **figures**.

Nous n'entrerons pas dans le détail de chaque figure, qui se

subdivise à son tour en **modes** différents, mais nous préciserons leurs structures générales, qui sont au nombre de quatre et pour lesquelles on peut dresser le **tableau 1**.

Précisons tout de suite que seules les trois premières figures sont reconnues par l'ensemble des logiciens, la quatrième, sur laquelle nous reviendrons, n'étant pas admise par tous.

Comme on le voit, la construction de la conclusion à partir des prémisses peut se faire de manière très rigoureuse, celle-ci étant toujours formée du couple :

(*) François Chenique : « Elément de logique classique, tome 2 : L'art de raisonner », éditions Dunod, série Logique et Informatique, pp. 204 et suivantes.

petit terme - grand terme conformément à un schéma syntaxique type :

sujet - prédicat.

Cette construction consiste donc à éliminer le moyen terme entre les prémisses et à concaténer le petit terme et le grand terme.

Le programme traite le syllogisme de la manière suivante :

- Son but est de construire une conclusion à partir des **deux dernières** assertions entrées qui seront considérées comme les prémisses d'un syllogisme.

- Le traitement proprement dit du syllogisme est activé par la question : DONC ? (voir les exemples ci-contre).

- Une fois cette question entrée, le programme recherche les deux dernières assertions présentes en mémoire. Nous insistons sur ce point : pour traiter un syllogisme, entrez d'abord les assertions-prémises,

ses, puis, immédiatement après, la question DONC ?

Notons que la conjonction OR n'est pas obligatoire dans la seconde prémissse.

- Dans ces deux assertions-prémises, le programme recherche le moyen terme. S'il n'en trouve pas, le message :

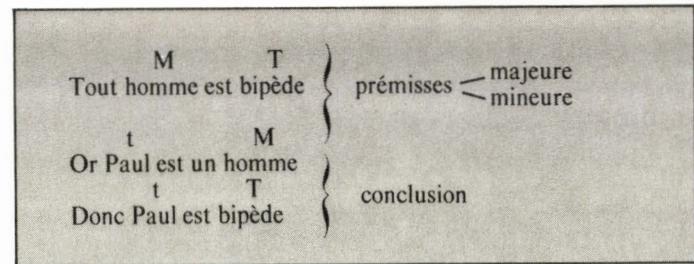
**** JE NE PEUX RIEN CONCLURE !**

est affiché.

S'il y a un moyen terme, la conclusion est construite directement par extraction puis concaténation du petit terme et du grand terme, précédés de la conjonction DONC. Seul un traitement spécifique est effectué pour les syllogismes de la troisième figure (rajout de QUELQUE ou de QUELQUES) qui représentent un cas particulier.

Donnons quelques exemples de syllogismes selon les différentes figures (ces exemples sont presque tous tirés de l'ouvrage déjà cité).

• Première figure



« Tout logicien est incompris
Or tout homme sensé est logicien
Donc tout homme sensé est incompris. »

Nous pouvons y ajouter celui-ci, bien connu :
« Tout ce qui est rare est cher
Or un cheval bon marché est rare
Donc un cheval bon marché est cher »

Précisons à ce sujet, comme le fait remarquer F. Chenique, (op. cit.) qu' « il convient de distinguer soigneusement un syllogisme VALABLE d'un syllogisme VRAI. Un syllogisme

valable est celui dont la conclusion découle nécessairement des précisions, quelle que soit d'ailleurs cette conclusion. Un syllogisme vrai est celui dont la conclusion est vraie. »

Le syllogisme ci-dessus est donc correct, ou valable, mais il est faux, car sa conclusion est fausse.

Vous vous apercevez ainsi qu'il est très facile de faire dire ce que l'on veut au programme qui, s'il construit des conclusions correctes logiquement, n'en fera pas pour autant des conclusions vraies !

SLIST

```

10 REM =====
20 REM BASE DE CONNAISSANCES
30 REM RELATIONNELLE
40 REM INTERROGEABLE EN
50 REM LANGAGE NATUREL.
51 REM
52 REM RELATIONS TRAITÉES :
53 REM HORIZONTALES ET
54 REM VERTICALES :
55 REM - INDIRECTION
56 REM - COMPOSITION DE
57 REM FONCTIONS
58 REM - SYLLOGISMES
59 REM
60 REM AUTEUR :
62 REM PHILIPPE LARVET
64 REM
70 REM COPYRIGHT 1984
90 REM =====
100 NP = 4:NM = 100
110 DIM B$(100,5)
120 DIM C$(10)
130 DIM E$(60)
132 DIM W$(10)
134 DIM D(2)
136 DIM P(2)
138 DIM E(10)
140 DIM S(100,3)
150 NT = 0
270 REM -----
280 REM MOTS NON-SIGNIFIANTS
290 REM -----
300 NE = 0
310 READ A$
320 IF A$ = "FF" THEN 1000
330 NE = NE + 1:E$(NE) = A$
340 GOTO 310
400 DATA EST,LE,LA,DE,UN,UNE
410 DATA L,DU,D,LES,DES,ET
412 DATA QU,QUE,QUI,SONT
413 DATA IL,ELLE,A,T,ETE
414 DATA EN,OU,COMMENT,AU
416 DATA N,NE,S,SE,ETAIT
420 DATA QUOI,C,CÈ,QUEL,QUELLE
422 DATA QUELS,QUELLES,PAR
424 DATA LEQUEL,LAQUELLE
426 DATA CA,SIGNIFIE,TOUT,OR
428 DATA TOUS,TOUTE,TOUTES
430 DATA =,(,),Y,FF
970 REM =====
980 REM E N T R E E
990 REM =====
1000 PRINT :NI = 0
1010 INPUT "> ";A$
1020 L = LEN (A$)
1470 REM -----
1480 REM EXAMEN DE L'ENTREE
1490 REM -----
1500 IF LEFT$ (A$,1) = "/" THEN
9950
1520 IF RIGHT$ (A$,1) < > "?" THEN 2000
1530 A$ = LEFT$ (A$,L - 1)
1540 NI = 1
1970 REM -----
1980 REM EXTRACTION DES MOTS
1990 REM -----
2000 FOR K = 1 TO NP
2010 C$(K) = "":E(K) = 0
2020 NEXT K
2030 L = LEN (A$):H$ = ""
2040 N = 1:ND = 0:J = 1
2100 I = J
2120 J = J + 1
2122 G$ = MID$ (A$,J,1)
2130 IF G$ < > " " AND G$ < >
" " AND G$ < > "-" AND G$ <
> "(" AND G$ < > ")" AND J
< = L THEN 2120
2132 F$ = MID$ (A$,I,J - I)
2133 IF (I = 1) AND (LEFT$ (F$,
2) = "QU") THEN N = 0: GOTO
2160
2135 FOR K = 1 TO NE
2136 IF E$(K) < > F$ THEN NEXT
K
2138 IF K < = NE THEN 2160
2140 ND = ND + 1
2142 IF (NI = 0) AND (ND > NP)
THEN 3600
2150 H$ = H$ + F$:C$(ND) = F$
2152 IF ND > 2 THEN 2160
2154 D(ND) = J
2160 IF J > L THEN 3000
2170 J = J + 1
2180 IF MID$ (A$,J,1) = " " AND
J < = L THEN 2170
2200 IF J < = L THEN 2100
2970 REM -----
2980 REM FIN EXTRACTION
2990 REM -----
3000 IF C$(1) = "" THEN 3700
3004 IF (NI > 0) AND (C$(1) = "D"

```

Listing du programme de constitution d'interrogation de la base.

• Deuxième figure

T	M
Tout homme est rationnel	
t	M
Or Aristote est rationnel	
t	T
Donc Aristote est un homme	
« Le Louvre est beau »	
Or j'aime tout ce qui est beau	
Donc j'aime le Louvre »	
« Tout être humain est bipède »	
Or Paul est bipède	
Donc Paul est un être humain. »	

Notons que l'on peut traiter ces assertions indifféremment en tant qu'indirections avec des questions du type QUEL PHILOSOPHE EST GREC ? ou QUEL ANIMAL EST RAISONNABLE ? ou en tant que syllogisme, avec la question DONC ?

Autre exemple de troisième figure :

« Les savants sont souvent distraits

Or tous les savants sont bavards

Donc quelques bavards sont souvent distraits. »

• Troisième figure

M	T
Tout homme est raisonnable	
M	t
Or l'homme est un animal	
t	T
Donc quelque animal est raisonnable	
« Platon est grec »	
Or Platon est philosophe	
Donc quelque philosophe est grec. »	

• Quatrième figure

T	M
Tout cheval est un équidé	
M	t
Or tout équidé est herbivore	
t	T
Donc quelque herbivore est un cheval	

A propos de cette figure, l'auteur déjà cité précise que « la combinaison constituée par

	Première figure	Deuxième figure	Troisième figure	Quatrième figure
Majeure	M - T	T - M	M - T	T - M
Mineure	t - M	t - M	M - t	M - t
Conclusion	t - T	t - T	t - T	t - T

Tableau extrait du livre de F. Chenique déjà cité.

la quatrième figure du syllogisme (...) a été suggérée, peut-être pour des raisons de symétrie, par Aristote (...). Elle a été acceptée par les logiciens scolastiques à partir du XV^e siècle, alors que les logiciens contemporains sont presque unanimes à la rejeter ou, au moins, à ne la considérer que comme une inversion de la première figure».

En effet, note par ailleurs F. Chenique : « D'un point de vue strictement grammatical, cette combinaison est possible ; d'un point de vue logique, la chose est moins sûre, car les trois premières figures épousent les combinaisons logiques des termes entre eux. » Ceci fait qu'il n'est pas évident de trouver des

exemples intéressants et corrects pour cette figure.

Adaptation du programme sur d'autres matériels

Le programme a été écrit dans un Basic standard avec un « vocabulaire » d'instructions volontairement limité, ce qui doit faciliter l'adaptation à n'importe quel micro-ordinateur.

Pour les matériels dont le Basic ne disposerait pas de la clause DATA, il suffit d'initialiser poste à poste la table E\$ et de renseigner la variable NE en conséquence (lignes 300-430). ■

```

ONC") THEN 9200
3010 IF NI > 0 THEN 6000
3020 IF ND > 1 THEN 4000
3470 REM -----
3480 REM E R R E U R
3490 REM -----
3500 PRINT "** VOTRE PHRASE EST
TROP COURTE !
3510 GOTO 1000
3600 PRINT "** VOTRE PHRASE EST
TROP LONGUE !
3610 GOTO 1000
3700 PRINT "** VOTRE QUESTION ES
T INCOMPLETE !
3710 GOTO 1000
3900 REM -----
3910 REM ASSERTION
3920 REM -----
3980 REM CONTROLE EXISTENCE
3982 REM DE L'ASSERTION
3984 REM DANS LA BASE
3990 REM -----
4000 IF NT = 0 THEN 5000
4100 FOR I = 1 TO NT
4110 Z$ = ""
4120 FOR J = 1 TO NP
4130 Z$ = Z$ + B$(I,J)
4140 NEXT J
4150 IF Z$ < > H$ THEN NEXT I
4160 IF I > NT THEN 5000
4170 REM -----
4180 REM ASSERTION DEJA CONNUE
4190 REM -----
4200 PRINT "** ASSERTION DEJA CO
NNUE !
4210 GOTO 1000
4970 REM -----
4980 REM ENRICHISSEMENT BASE
4990 REM -----
5000 FOR I = 1 TO NM
5010 IF B$(I,1) < > "" THEN NEXT I
5020 IF I < = NM THEN 5050
5030 PRINT "** STOP ! LA BASE ES
T PLEINE !
5040 STOP
5050 FOR J = 1 TO NP
5052 B$(I,J) = C$(J)
5054 NEXT J
5056 B$(I,NP + 1) = A$
5058 S(I,1) = ND:S(I,2) = D(1):S(
I,3) = D(2)
5060 PRINT "** COMPRIS
5070 IF I > NT THEN NT = I
5080 GOTO 1000
5970 REM -----
5980 REM INTERROGATION
5990 REM -----
5992 REM CONTROLE EXISTENCE
5994 REM DE CHACUN DES MOTS
5996 REM -----
6000 K = 0:M = NT
6010 T = 0:K = K + 1
6100 FOR I = 1 TO NT
6110 FOR J = 1 TO S(I,1)
6120 IF B$(I,J) < > C$(K) THEN
NEXT J
6130 IF J > S(I,1) THEN 6180
6132 T = 1
6134 IF E(K) = 0 THEN E(K) = I
6140 IF ND > 1 THEN 6170
6142 IF (J = 2) AND (L > ( LEN (
C$(K)) + 1)) THEN 6162
6150 PRINT B$(I,NP + 1)
6160 GOTO 6180
6162 PRINT LEFT$ (B$(I,NP + 1),
S(I,2))
6164 GOTO 6180
6170 A = I:I = NT
6180 NEXT I
6200 IF T = 0 THEN 9000
6210 IF ND = 1 THEN 1000
6220 IF A < M THEN M = A
6230 IF K < ND THEN 6010
6240 IF (ND > NP) THEN 8100
6970 REM -----
6980 REM RELATION HORIZONTALE
6990 REM -----
7000 V = 0
7010 FOR I = M TO NT
7020 T = 0:K = 0:J = 0
7030 K = K + 1
7040 IF K > ND THEN 7300
7050 J = J + 1
7060 IF J > S(I,1) THEN 7200
7070 IF B$(I,J) < > C$(K) THEN
7050
7080 T = T + 1
7090 IF T > 2 THEN 7030
7094 P(T) = J
7100 GOTO 7030
7200 IF T = 0 THEN 7400
7300 IF (T < > ND) OR (T < > S
(I,1)) THEN 7330
7310 PRINT "** OUI":V = 1:I = NT

```

Listing (suite).

```

7320 GOTO 7400
7330 IF T < > ND THEN 7400
7340 V = 1
7344 IF (P(1) = 1) AND (P(2) = 2)
    ) THEN 7372
7350 IF P(1) = 2 THEN 7380
7352 IF P(1) = 3 THEN 7390
7360 PRINT B$(I,NP + 1)
7370 GOTO 7400
7372 LB = LEN (B$(I,NP + 1))
7374 PRINT RIGHT$ (B$(I,NP + 1),
    ,LB - S(I,3))
7376 GOTO 7400
7380 PRINT LEFT$ (B$(I,NP + 1),
    S(I,2))
7382 GOTO 7400
7390 PRINT LEFT$ (B$(I,NP + 1),
    S(I,3))
7400 NEXT I
7410 IF V = 1 THEN 1000
7470 REM -----
7980 REM RELATION VERTICALE
7990 REM -----
8000 REM COMPOSITION
8080 REM DE FONCTIONS
8090 REM -----
8100 FOR K = 1 TO ND
8110 W$(K) = C$(K)
8120 NEXT K
8130 R = ND
8140 Z$ = W$(R - 1) + W$(R)
8150 FOR I = M TO NT
8160 IF (B$(I,2) + B$(I,3)) < >
    Z$ THEN NEXT I
8170 IF I > NT THEN 8300
8180 R = R - 1:W$(R) = B$(I,1)
8190 IF R > 1 THEN 8140
8200 PRINT W$(1)
8210 GOTO 1000
8270 REM -----
8280 REM INDIRECTION
8290 REM -----
8300 I2 = E(2):T = 0
8310 E(2) = I2
8320 FOR J1 = 1 TO S(E(1),1)
8330 Z$ = B$(E(1),J1)
8340 FOR J2 = 1 TO S(E(2),1)
8350 IF (E(1) < > E(2)) AND (Z$ =
    B$(E(2),J2)) AND (Z$ < >
    C$(1)) AND (Z$ < > C$(2))
    THEN T = 1: PRINT B$(E(1),
    ,J1):J1 = NP:J2 = NP
8360 NEXT J2
8380 NEXT J1
8400 K = 2: GOSUB 9100
8410 IF V = 1 THEN 8320
8420 K = 1: GOSUB 9100
8430 IF V = 1 THEN 8310
8440 IF T = 1 THEN 1000
8870 REM -----
8880 REM ECHEC FINAL
8890 REM -----
8900 IF N = 0 THEN 8950
8910 PRINT "** NON
8920 GOTO 1000
8950 PRINT "** JE L'IGNORE
8960 GOTO 1000
8970 REM -----
8980 REM MOT INCONNU
8990 REM -----
9000 PRINT "** JE NE CONNAIS PAS
    ";"":C$(K);"""
9020 GOTO 1000
9070 REM /////////////////
9080 REM MODULE DE RECHERCHE
9084 REM D'UN PROCHAIN E(K)
9090 REM /////////////////
9100 V = 0
9104 FOR I = E(K) + 1 TO NT
9110 FOR J = 1 TO S(E(K),1)
9120 IF (B$(I,J) = C$(K)) THEN V
    = 1:E(K) = I:J = NP:I = NT
9130 NEXT J
9140 NEXT I
9150 RETURN
9190 REM =====
9192 REM SYLLOGISME
9196 REM =====
9197 REM RECHERCHE MOYEN TERME
9198 REM -----
9200 S2 = NT:S1 = NT - 1:T = 0
9210 L1 = LEN (B$(S1,NP + 1))
9214 L2 = LEN (B$(S2,NP + 1))
9220 FOR K = 1 TO 3
9230 FOR J = 1 TO 3
9234 IF B$(S1,J) = "" THEN 9270
9240 IF B$(S2,K) < > B$(S1,J)
    THEN 9270
9250 T = 1:JM = J:KM = K
9260 J = NP:K = NP
9270 NEXT J
9280 NEXT K
9290 IF T = 0 THEN 9800
9292 REM -----
9294 REM RESOLUTION
9296 REM -----
9300 ON KM GOTO 9310,9330,9350
9310 X$ = RIGHT$ (B$(S2,NP + 1),
    L2 - S(S2,2))
9320 GOTO 9360
9330 X$ = LEFT$ (B$(S2,NP + 1),S
    (S2,2))
9340 GOTO 9360
9350 X$ = LEFT$ (B$(S2,NP + 1),S
    (S2,3))
9360 IF LEFT$ (X$,3) = "OR "
    THEN X$ = RIGHT$ (X$, LEN
    (X$) - 3)
9368 ON JM GOTO 9370,9390,9410
9370 Z$ = RIGHT$ (B$(S1,NP + 1),
    L1 - S(S1,2))
9380 GOTO 9420
9390 Z$ = LEFT$ (B$(S1,NP + 1),S
    (S1,2))
9400 GOTO 9420
9410 Z$ = LEFT$ (B$(S1,NP + 1),S
    (S1,3))
9420 IF KM + JM < > 2 THEN 9500
9430 IF LEFT$ (X$,4) = "EST "
    THEN X$ = RIGHT$ (X$, LEN
    (X$) - 4):X$ = "QUELQUE " +
    X$ + " "; GOTO 9500
9440 IF LEFT$ (X$,5) = "SONT "
    THEN X$ = RIGHT$ (X$, LEN
    (X$) - 5):X$ = "QUELQUES "
    + X$ + " "
9500 PRINT "DONC ";X$:Z$
9510 PRINT
9520 GOTO 1000
9790 REM -----
9792 REM ABSENCE DE CONCLUSION
9796 REM -----
9800 PRINT "** JE NE PEUX RIEN C
    ONCLURE !
9810 GOTO 1000
9900 REM =====
9910 REM COMMANDES DE SERVICE
9920 REM =====
9950 IF A$ = "/L" THEN 10000
9952 IF LEFT$ (A$,2) = "/D"
    THEN 11000
9954 IF A$ = "/EFF" THEN 12000
9956 PRINT "** COMMANDE INCONNUE
    !
9960 GOTO 1000
9970 REM -----
9980 REM LISTAGE DE LA BASE
9990 REM -----
10000 FOR I = 1 TO NT
10010 PRINT I;" ";B$(I,NP + 1)
10020 NEXT I
10030 GOTO 1000
10970 REM -----
10980 REM SUPPRESSION
10990 REM -----
11000 IF L > 2 THEN 11008
11004 I = NT: GOTO 11010
11008 I = VAL (RIGHT$ (A$,L - 2
    ))
11010 FOR J = 1 TO NP + 1
11020 B$(I,J) = ""
11030 NEXT J
11040 PRINT "** ASSERTION ";I;" "
    SUPPRIMEE
11050 GOTO 1000
11990 REM -----
11992 REM EFFACEMENT DE
11994 REM TOUTE LA BASE
11996 REM -----
12000 PRINT "** VOULEZ-VOUS EFFA
    CER
12010 INPUT " TOUTE LA BASE ?
    (O/N) ";A$
12020 IF A$ < > "O" THEN 1000
12030 FOR I = 1 TO NT
12040 FOR J = 1 TO NP + 1
12050 B$(I,J) = ""
12060 NEXT
12070 NEXT
12080 PRINT "** BASE ENTIEREMENT
    EFFACEE
12090 NT = 0: GOTO 1000

```

Listing (suite et fin).

EXEMPLES D'INDIRECTION

-> TOUT STYLO EST BLEU
**** COMPRIS**

-> FRANCOIS POSSEDE UN STYLO
**** COMPRIS**

-> BLEU EST UNE COULEUR
**** COMPRIS**

-> ROUGE EST UNE COULEUR
**** COMPRIS**

-> TOUT STYLO EST EN PLASTIQUE
**** COMPRIS**

-> LE PLASTIQUE EST UNE MATIERE
**** COMPRIS**

-> FRANCOIS POSSEDE-T-IL UN STYLO ROUGE?
**** NON**

-> QUELLE EST LA COULEUR DU STYLO DE FRANCOIS?
BLEU

-> RAOUL A ACHETE UN STYLO EGALLEMENT
**** COMPRIS**

-> DE QUELLE COULEUR EST LE STYLO DE RAOUL ?
BLEU

-> TOUT STYLO EST EN QUELLE MATIERE?
PLASTIQUE

-> ANNIE EST UNE JOLIE FILLE
**** COMPRIS**

-> ANNIE EST SAGE
**** COMPRIS**

-> MINNA EST UNE FILLE ELLE AUSSI
**** COMPRIS**

-> MINNA EST FOFOLE
**** COMPRIS**

-> QUELLE FILLE EST SAGE ?
ANNIE

-> ET LAQUELLE EST FOFOLE ?
MINNA

-> JOLIE EST LE CONTRAIRE DE LAIDE
**** COMPRIS**

-> EST-CE QU'ANNIE EST LAIDE?
JOLIE

-> CA SIGNIFIE-T-IL QU'ANNIE EST JOLIE
FILLE ?
**** OUI**

EXEMPLES DE COMPOSITION DE FONCTIONS

-> JEAN REGARDE MARIE
**** COMPRIS**

-> MARIE REGARDE HUGUES
**** COMPRIS**

-> QUI MARIE REGARDE-T-ELLE ?
HUGUES

-> MARIE REGARDE-T-ELLE JEAN?
**** NON**

-> HUGUES EST LE FRERE D'HENRI
**** COMPRIS**

-> HENRI EST LE FILS D'OCTAVE
**** COMPRIS**

-> OCTAVE EST L'ONCLE D'ANATOLE
**** COMPRIS**

-> QUI EST LE FILS DE L'ONCLE D'ANATOLE?
HENRI

-> ET QUI REGARDE LE FRERE DU FILS DE L'ONCLE D'ANATOLE ?
MARIE

-> L'ENTREPRISE A UN SIEGE ET UNE ANNEXE
**** COMPRIS**

-> M.BERTRAND EST RESPONSABLE DE L'ANNEXE
**** COMPRIS**

-> L'ANNEXE A 15 SERVICES DIFFERENTS
**** COMPRIS**

-> M.JACQUES EST RESPONSABLE DE LA SAISIE
**** COMPRIS**

-> M.RENE EST LE CHEF DU SERVICE COMPTABILITE
**** COMPRIS**

-> M.MARTIN EST UN AMI DE M.JACQUES
**** COMPRIS**

-> M.DUBOIS EST CHEF DU SERVICE PHOTO
**** COMPRIS**

-> LA SAISIE EST UN SERVICE DECENTRALISE
**** COMPRIS**

-> RENAUD EST LE FILS DE M.BERTRAND
**** COMPRIS**

-> DAMIEN EST LE FILS DE M.RENE
**** COMPRIS**

-> CHEF DE SERVICE ?
M.RENE
M.DUBOIS

-> RESPONSABLE ?
M.BERTRAND EST RESPONSABLE DE L'ANNEXE
M.JACQUES EST RESPONSABLE DE LA SAISIE

-> M.MARTIN EST L'AMI DE QUI ?
DE M.JACQUES

-> QUI EST LE FILS DU RESPONSABLE DES 15 SERVICES ?
RENAUD

-> QUI EST FILS D'UN CHEF DE SERVICE?
DAMIEN

-> QUI EST L'AMI DU RESPONSABLE D'UN SERVICE DECENTRALISE?
M.MARTIN

EXEMPLES DE SYLLOGISMES

-> TOUT HOMME EST MORTEL
**** COMPRIS**

-> OR SOCRATE EST UN HOMME
**** COMPRIS**

-> DONC?
DONC SOCRATE EST MORTEL

-> TOUT HOMME EST BIPEDE
**** COMPRIS**

-> OR PAUL EST UN HOMME
**** COMPRIS**

-> DONC?
DONC PAUL EST BIPEDE

-> TOUT LOGICIEN EST INCOMPRIS
**** COMPRIS**

-> OR TOUT HOMME SENSE EST LOGICIEL
**** COMPRIS**

-> DONC ?
DONC TOUT HOMME SENSE EST INCOMPRIS

-> LE LOUVRE EST BEAU
**** COMPRIS**

-> OR J'AIME TOUT CE QUI EST BEAU
**** COMPRIS**

-> DONC?
DONC J'AIME LE LOUVRE

-> TOUT CE QUI EST RARE EST CHER
**** COMPRIS**

-> UN CHEVAL.BON.MARCHE EST RARE
**** COMPRIS**

-> DONC ?
DONC UN CHEVAL.BON.MARCHE EST CHER

-> PLATON EST GREC
**** COMPRIS**

-> OR PLATON EST PHILOSOPHE
**** COMPRIS**

-> DONC?
DONC QUELQUE PHILOSOPHE EST GREC

-> QUEL GREC EST PHILOSOPHE ?
PLATON

-> LES SAVANTS SONT SOUVENT DISTRITS
**** COMPRIS**

-> OR TOUS LES SAVANTS SONT BAVARDS
**** COMPRIS**

-> DONC ?
DONC QUELQUES BAVARDS SONT SOUVENT DISTRITS

-> TOUS LES HOMMES SONT MORTELS
**** COMPRIS**

-> OR DES HOMMES SONT JUSTES
**** COMPRIS**

-> DONC ?
DONC QUELQUES JUSTES SONT MORTELS

IL EST POSSIBLE EGALEMENT D'UTILISER
LE PROGRAMME D'UNE MANIERE PLUS
FORMELLE :

-> X1 = X2
**** COMPRIS**

-> X2 = X3
**** COMPRIS**

-> PAR OUOI X1 = X3 ?
X2

-> Y1 = F(X)
**** COMPRIS**

-> X = G(T1)
**** COMPRIS**

-> V = K(W)
**** COMPRIS**

-> T1 = H(V)
**** COMPRIS**

-> F(G(H(K(W)))) = ?
Y1

-> A OUOI EST = F DE G DE H DE K DE W?
Y1

-> CHAT = ANIMAL
** COMPRIS

-> CHAT = MANGEUR(SOURIS)
** COMPRIS

-> TIGRE = ANIMAL
** COMPRIS

-> TIGRE = MANGEUR(HOMME)
** COMPRIS

-> ANIMAL MANGEUR?
CHAT
TIGRE

-> MANGEUR(HOMME) = ?
TIGRE

-> Y A-T-IL UN MANGEUR(SOURIS ET HOMME)?
** NON

-> QUI EST MANGEUR(HOMME ET SOURIS)?
** JE L'IGNORE

-> ARTABAN = CHEVAL (HENRI.IV)
** COMPRIS

-> BLANC = COULEUR(ARTABAN)
** COMPRIS

-> HENRI.IV = ROI(NAVARRE)
** COMPRIS

-> COULEUR(CHEVAL(ROI(NAVARRE)))?
BLANC

ON PEUT EGALLEMENT POSER CETTE QUESTI
ON DE LA FACON SUIVANTE :

-> QUELLE EST LA COULEUR DU CHEVAL DU
ROI DE NAVARRE ?
BLANC

VOICI POUR TERMINER LA LISTE DE
TOUTES LES ASSERTIONS PRESENTES EN
MEMOIRE :

-> /L
1 EXEMPLES D'INDIRECTION
2 TOUT STYLO EST BLEU
3 FRANCOIS POSSEDE UN STYLO
4 BLEU EST UNE COULEUR
5 ROUGE EST UNE COULEUR
6 TOUT STYLO EST EN PLASTIQUE
7 LE PLASTIQUE EST UNE MATIERE
8 RAQUIL A ACHETE UN STYLO EGALEMENT
9 ANNIE EST UNE JOLIE FILLE
10 ANNIE EST SAGE
11 MINNA EST UNE FILLE ELLE AUSSI
12 MINNA EST FOFOLE
13 JOLIE EST LE CONTRAIRE DE LAIDE
14 EXEMPLES DE COMPOSITION DE FONCTIONS
15 JEAN REGARDE MARIE
16 MARIE REGARDE HUGUES
17 HUGUES EST LE FRERE D'HENRI
18 HENRI EST LE FILS D'OCTAVE
19 OCTAVE EST L'ONCLE D'ANATOLE
20 L'ENTREPRISE A UN SIEGE ET UNE ANNEXE
21 M.BERTRAND EST RESPONSABLE DE L'ANNEX
E
22 L'ANNEXE A 15 SERVICES DIFFERENTS
23 M.JACQUES EST RESPONSABLE DE LA SAISI
E

24 M.RENE EST LE CHEF DU SERVICE COMPTAB
ILITE

25 M.MARTIN EST UN AMI DE M.JACQUES
26 M.DUBOIS EST CHEF DU SERVICE PHOTO
27 LA SAISIE EST UN SERVICE DECENTRALISE

28 RENAUD EST LE FILS DE M.BERTRAND
29 DAMIEN EST LE FILS DE M.RENE
30 EXEMPLES DE SYLLOGISMES
31 TOUT HOMME EST MORTEL
32 OR SOCRATE EST UN HOMME
33 TOUT HOMME EST BIPEDE
34 OR PAUL EST UN HOMME
35 TOUT LOGICIEN EST INCOMPRIS
36 OR TOUT HOMME.SENSE EST LOGICIEN
37 LE LOUVRE EST BEAU
38 OR J'AIME TOUT CE QUI EST BEAU
39 TOUT CE QUI EST RARE EST CHER
40 UN CHEVAL,BON,MARCHE EST RARE
41 PLATON EST GREC
42 OR PLATON EST PHILOSOPHE
43 LES SAVANTS SONT SOUVENT DISTRAITS
44 OR TOUS LES SAVANTS SONT BAVARDS
45 TOUS LES HOMMES SONT MORTELS
46 OR DES HOMMES SONT JUSTES
47 X1 = X2
48 X2 = X3
49 Y1 = F(X)
50 X = G(T1)
51 V = K(W)
52 T1 = H(V)
53 CHAT = ANIMAL
54 CHAT = MANGEUR(SOURIS)
55 TIGRE = ANIMAL
56 TIGRE = MANGEUR(HOMME)
57 ARTABAN = CHEVAL (HENRI.IV)
58 BLANC = COULEUR(ARTABAN)
59 HENRI.IV = ROI(NAVARRE)

Exemples d'indirection.

SPECTRUM GESTION



SEMAPHORE LOGICIELS CH-1283 LA PLAINE (SUISSE)

PROPOSE

la base de données multi-applications
 le traitement de textes accentué,
 la comptabilité, la facturation, la fusion
 texte - adresses, les claviers, les interfaces
 les manuels français et le service pour
DU 'PRO QUI DEGAGE'

Je m'intéresse à une documentation SémaPhore: Gestion / Jeux / Utilitaires
 que je désire recevoir sans engagement à l'adresse suivante:

Nom

Rue

Ville

Code/Pays

Ce bon est à retourner à SémaPhore Logiciels CH-1283 La Plaine-Suisse- MS-12.84

SERVICE-LECTEURS N° 117

Arbre généalogique pour Jupiter Ace

Nombre de problèmes faisant intervenir des listes chaînées ou des structures d'arbres sont élégamment résolus avec le langage Pascal qui autorise l'utilisation des variables pointeurs.

Malheureusement il n'est disponible que sur des configurations matériel riches en mémoire. Et donc difficilement accessible à la bourse de nombreux utilisateurs.

Heureusement Forth est là pour apporter des solutions tout aussi simples, pour des prix de configuration matériel nettement plus intéressants.

Services recherches

Avant de songer à la construction d'un arbre généalogique, qui n'est qu'une forme d'assemblage de données structurées, essayons de définir plus précisément ce à quoi il doit servir.

Cette question posée, on s'aperçoit que le principal service escompté est de permettre la recherche de l'existence ou non d'un lien de parenté entre deux individus quelconques, ce qui revient à rechercher s'ils ont un ancêtre commun.

Vient en second lieu l'énumération pour un individu de tous ses ascendants ou descendants présents sur la nième génération précédent ou suivant la sienne.

Les autres services plus anecdotiques ne sont que des dérivés des deux premiers.

Revenons au premier service. La recherche d'un ancêtre commun devient très aisée si l'on réalise que tout individu est la racine de deux arbres différents, à savoir l'arbre de ses ascendants et l'arbre de ses descendants.

Ces deux arbres ont une structure distincte :

- Celle des ascendantes est un arbre binaire, le passage d'une génération à la précédente se fait par l'intermédiaire des relations orientées **père** et **mère**.

- Celle des descendants est un arbre quelconque où le passage d'une génération à la suivante se fait par la relation orientée **enfant**.

S'ils sont de structures diffé-

rentes, ils doivent pouvoir être construits à partir d'un type unique de données puisqu'un même individu peut paraître en fonction des racines choisies dans l'un ou l'autre arbre à la fois.

On devine aisément que ce genre de données doit avoir une double fonction : contenir d'une part les informations requises sur les individus (date de naissance, date de décès, lieu de naissance, etc.) et pointer d'autre part sur plusieurs autres membres de la généalogie (père, fils, conjoint, frère).

Avant de définir plus précisément ces données, poursuivons la résolution de notre problème.

La recherche de l'ancêtre commun à deux individus quelconques va être celle de l'élément, en fait du couple, présent à la fois dans l'arbre des ascendants du premier individu et dans l'arbre des ascendants du second individu.

Plus précisément, comme les ascendants de ce couple vont être également présents dans les deux arbres précédents, il s'agit du couple commun situé le plus bas dans ces deux arbres.

A partir du moment où nous aurons bâti un algorithme de parcours pour arbre des ascendants, il suffira de « marquer » par un indicateur spécifique tous les éléments de l'un des arbres, puis de parcourir le second en interrogeant chacun de ses éléments pour savoir s'il porte ou non la marque précédente. Puisque l'algorithme de parcours ira en remontant sous-arbre après sous-arbre, le premier élément rencontré portant

LANGAGE : Recherche d'arborescence de M. GEORGES

La construction d'un arbre généalogique est un exemple représentatif des problèmes faisant intervenir des listes chaînées ou des structures d'arbres.

Langage : Forth
Ordinateur : Jupiter Ace

cette marque sera l'un des deux membres du couple recherché.

Le second service est encore plus simple à résoudre. Si dans chacun des algorithmes de parcours des deux arbres, on prend la peine d'incrémenter un compteur à chaque fois que l'on s'éloigne d'une génération de la racine et de le décrémenter à chaque fois que l'on s'en rapproche, il est facile alors de sélectionner les éléments de la génération voulue.

Une remarque s'impose : la mise en évidence de la double structure qui permet de résoudre nos problèmes. A se contenter du seul arbre des descendants auquel on pense spontanément en cas de problème généalogique, on aboutirait à de grandes lourdeurs, voire à des impossibilités.

Construction des données structurées nécessaires

Forth, langage tissé, va montrer maintenant toute sa puissance et simplicité.

En effet, grâce à ces mots de définitions (BUILDS ou DEFINER... DOES>), il est possible de construire sur mesure des données structurées, spécifiques de chacun des membres de la généalogie remplissant comme nous l'avons déjà établi une double fonction.

En effet, nous savons qu'en Forth les données sont constituées par une suite d'octets subdivisée en deux zones :

- une zone en-tête contenant

principalement le nom de la donnée et construite automatiquement dès que l'utilisateur a appellé le mot de définition suivi du nom choisi pour la donnée ;

- une zone paramètre construite suivant la volonté de l'utilisateur par la routine définie DEFINER ET DOES>, et dont l'adresse de début est renvoyée en phase exécution sur la pile après appel de la donnée par son nom.

Dans le cas présent, deux types d'octets sont nécessaires pour la zone paramètre des données que l'on cherche à établir. Les uns doivent contenir les informations spécifiques à chacun des membres de la généalogie. Les autres, l'adresse du début de la zone paramètre des membres de la généalogie que l'on désire pointer.

Pour laisser à l'utilisateur une certaine liberté dans le choix des données spécifiques, les octets correspondants seront placés après ceux des pointeurs. Dans la suite de notre exposé, nous les négligerons totalement car il ne s'agit que d'informations relevant des techniques

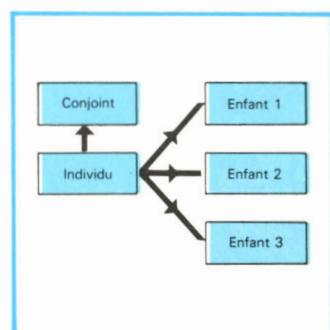


Fig. 1. – Schéma correspondant au premier choix de recherche.

classiques du fichier, avec mobilisation éventuelle d'une mémoire de masse lorsque le volume de celles-ci devient trop important.

Pour en revenir aux pointeurs, plusieurs choix possibles vont déterminer des algorithmes d'exploitation différents.

Le premier choix qui vient naturellement à l'esprit est celui correspondant au schéma de la **figure 1**.

Cette disposition se caractérise par le fait que chaque élément contient un pointeur pour son conjoint (époux ou épouse) et autant de pointeurs qu'il a d'enfants.

S'il est techniquement possible de commencer à mettre en œuvre ce choix, il en résulte rapidement les inconvénients suivants :

- compte tenu de la philosophie Forth, chaque membre de la généalogie ne peut être défini que si l'on connaît le nombre de ses enfants, de façon à pouvoir réserver le nombre d'octets nécessaires aux pointeurs ;
- il est indispensable d'incorporer ce nombre à la donnée structurée d'ensemble, de façon à en permettre la lecture, avec toutes les complications que cela représente dans la gestion de la pile de données ;
- s'il est facile de passer d'un père à ses fils, l'inverse est beaucoup plus délicat dans la mesure où, au préalable, il faut avoir descendu l'arbre à partir de sa racine en posant sur la pile de données « les petits cailloux blancs » permettant de rebrousser chemin.

Dans ces conditions il est illusoire de penser pouvoir remonter facilement un arbre des descendants dans sa totalité.

En définitive, ce premier choix s'avère donc une impasse, sauf si on le complète par des pointeurs de retour allant dans le sens fils-père. Mais même après cette adaptation, les deux premiers inconvénients subsistent.

Cependant, une autre solution dans le choix des pointeurs permet de remédier à ces inconvénients tout en accélérant les temps de parcours et en économisant la taille mémoire (**fig. 2**).

Trois différences apparaissent par rapport à la première solution :

- Chaque donnée ne pointe que sur l'un de ses enfants ; par commodité, nous l'appellerons l'ainé ;

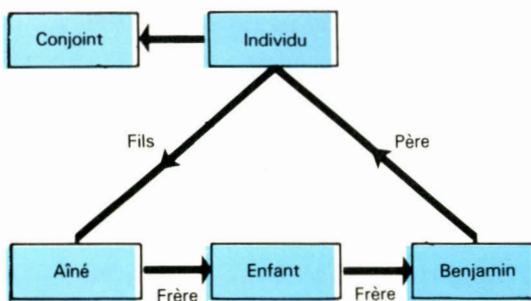


Fig. 2. – Autre solution dans le choix des pointeurs.

- chaque donnée, à l'exception d'une seule, pointe vers l'un de ses frères ; cette exception est le benjamin, qui, au lieu de pointer sur son frère ainé, pointe sur son père ;
- chaque zone paramètre a la même longueur.

Cette solution n'est viable qu'à condition de pouvoir distinguer le benjamin de ses frères. Pour cela, il suffit d'un octet indicateur accompagnant les pointeurs dans la zone paramètre.

Cet octet indicateur peut sembler dans un premier temps anti-économique au point de vue mémoire ; il serait en effet possible soit de rendre négative l'adresse contenue dans le pointeur frère du benjamin, soit d'utiliser un bit non employé dans des informations numériques telles que le mois de la date de naissance... En fait, ces deux dernières façons de procéder seraient plutôt acrobatiques, dans la mesure où, par la suite, nous aurons besoin d'autres indicateurs, notamment celui déjà mentionné qui doit porter la marque servant à l'identification d'un ancêtre commun.

Mieux vaut donc employer un octet entier offrant alors la possibilité de disposer de 8 indicateurs situés au niveau de chacun de ses bits (**fig. 3**).

Finalement, la zone paramètre de notre donnée est représentée à la **figure 4**.

Pour faciliter la compréhension de la suite, n'oublions pas que l'appel d'un individu par son nom donne directement l'adresse de cet octet indicateur, adresse qui, bien entendu,

n'est autre que le PFA de l'individu.

Le choix de l'ordre fils-frère-conjoint pour les pointeurs peut paraître moins judicieux que celui de frère-conjoint-fils, dans la mesure où le pointeur fils peut être supprimé pour un individu sur deux (pour aller à un enfant ou en revenir, on passe par le conjoint).

Pourquoi ce choix ?

- Dans nombre d'autres problèmes (réseaux sémantiques, décomposition d'un ensemble en sous-ensembles disjoints) on retrouve la même structure d'arbre, mais dépouillée de toute incidence liée à la différenciation entre sexes.
- La structure de donnée correspondante est identique à la précédente, mais sans le pointeur conjoint, qu'il vaut mieux alors placer après les autres de façon à pouvoir conserver d'un problème à l'autre le maximum d'algorithmes.

Construction d'une généalogie

Avec la structure de donnée choisie, le mot de définition d'un individu masculin se réduit à :

DEFINER MM Ø C, Ø, Ø, Ø, DOES> ;

et celui d'un individu féminin à :

DEFINER MF 16 C, Ø, Ø, Ø, Ø, DOES> ;

et la création d'une donnée à :

MM (ou MF) NOM

où NOM est le libellé du nom de l'individu accompagné d'autant de prénoms qu'il est nécessaire pour éviter toute homonymie avec un autre individu ;

chacun de ces prénoms doit être séparé des autres par un point ou tout autre séparateur autorisé pour que l'ensemble ne constitue qu'un seul mot au sens Forth du terme.

La création des liens de filiation peut se faire individu par individu ; dans ce cas, il est nécessaire de distinguer trois procédures distinctes selon que le fils est l'ainé, le benjamin ou ni l'un ni l'autre. Pour simplifier et éviter toute erreur, il est préférable d'utiliser des mots définissant en une seule fois tous les pointeurs d'une cellule familiale entière. Ce sont les mots CEL et FU.

Au préalable, le mot IND doit avoir été défini ; IND (PFA n - PFA) introduit dans l'octet indicateur la quantité n lorsqu'elle n'y est pas.

Dans ces deux mots, la quantité $100 = 4 + 32 + 64$ correspond au fait que l'individu choisi a un père (4), est le benjamin (32) et a un « frère », en fait son père (64).

De même, la quantité $68 = 64 + 4$ correspond au fait que l'individu a un frère (64) et un père (4).

CEL (PFApère PFAenfant PFAenfant2... PFAenfantn...) abrégé de cellule, est à utiliser lorsqu'il y a plusieurs enfants. n est le nombre d'enfants du père.

FU (PFApère PFAfils unique...), abrégé de famille avec fils unique, est à utiliser lorsqu'il n'y a qu'un enfant.

On remarque qu'avec ces deux mots, seuls les éléments masculins ont le « droit » de décaler une relation de filiation. Rien n'empêcherait que ce soit les éléments féminins. L'important est qu'un sexe seulement soit chargé de cette mission de déclaration des liens de filiation. Le choix en faveur du sexe masculin répond à un souci de conformité avec les règles d'état civil et à la volonté de conserver la même terminologie que dans les arbres ne faisant pas intervenir la notion de sexe.

La création des liens de mariage se fait par le mot EP, abrégé d'époux, précédé du nom des deux individus à marier.

EP (PFA PFA...) vérifie par l'intermédiaire du mot SXOP ? abrégé de sexe opposé, que les deux individus sont de sexes opposés, puis introduit dans leur octet indicateur la quantité 8.

SEX? (PFA - PFA flag), présent dans le mot précédent,

bit 8 - valeur	1 : existence d'un fils ou d'une fille 0 : pas de fils
bit 7 - valeur	1 : existence d'un frère ou d'une sœur 0 : pas de frère
bit 6 - valeur	1 : qualité de benjamin ou de benjamine 0 : n'est pas le benjamin
bit 5 - valeur	1 : personne de sexe féminin 0 : personne de sexe masculin
bit 4 - valeur	1 : existence d'un conjoint 0 : pas de conjoint
bit 3 - valeur	1 : existence d'un père 0 : pas de père
bit 2 et 1	: disponibles

Fig. 3. – Liste des 8 indicateurs situés au niveau de chacun de ses bits.

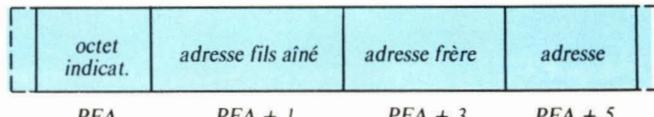


Fig. 4. – Définition de la zone paramètre des données.

Adresse	ZONE MEMOIRE		
	Nom de la zone mémoire	nb d'octets	Contenu de la zone
NFA	Zone nom	1 par caractère	1 code ASCII par
	Zone longueur du mot	2	nb d'octets de la zone paramètre + 7
LFA	Zone liaison	2	adresse de la zone long. du nom du mot précédent
	Zone longueur du nom	1	
CFA	Zone code	2	
	Zone paramètre		

Fig. 5. – Structure de l'en-tête des mots.

cherche le sexe de l'individu sans détruire son PFA.

IND? (PFA n .. PFA flag), présent dans le mot précédent, vérifie, toujours sans détruire le PFA, que l'octet indicateur contient la quantité n.

Ce mot sert dans la série de tests similaires :

FILS? vérifie si l'individu a un fils

FRERE? vérifie si l'individu a un frère

BENJAM? vérifie si l'individu est le benjamin

MARIE? vérifie si l'individu est marié

PERE? vérifie si l'individu a un père

ENFANT? vérifie si l'individu a un enfant

Chacun de ces mots préserve le PFA testé qui reste alors sur le sommet de la pile de données.

Mots d'exploitation de la généalogie

La généalogie construite, il s'agit maintenant de l'exploiter.

Nous savons que la solution à nos deux grands problèmes passe par la mise au point d'un algorithme de parcours pour chacun de nos deux arbres, algorithmes que nous allons conscrire progressivement.

Edition de l'en-tête

Tout d'abord, construisons le mot permettant d'éditer le nom de l'individu dont nous ne connaîtrions que le PFA. Il suffit pour cela d'éditer son en-tête grâce au mot TYPE précédé de l'adresse de début de l'en-tête et du nombre de caractères contenus dans cette dernière.

Dans le Jupiter Ace, la structure de l'en-tête des mots est un peu différente de celle des autres systèmes Forth (fig. 5).

Le mot AF (PFA – PFA), abrégé de l'affichage de l'en-tête dont nous avons besoin, s'en déduit immédiatement. Seule particularité pour laquelle aucun remède n'a été trouvé : la dernière lettre du mot édité apparaît en vidéo inverse.

Parcours simples

Le passage d'un individu dont le PFA est sur la pile vers son frère, son conjoint, son père, se fait très simplement à l'aide des mots suivants qui, tous, préserment les PFA de façon à permettre de nouveaux parcours : FRERE (PFA – PFA') CONJOINT (PFA – PFA') PERE (PFA – PFA')

Le passage à un enfant concrétisé par le mot ENFANT (PFA – PFA') utilise le mot FILS (PFA – PFA') qui est réservé aux seuls individus masculins.

Parcours de l'arbre des descendants

Celui-ci se fait au moyen du mot PARBDESC qui doit être précédé par le PFA de la racine de l'arbre. Au lieu d'édition l'en-tête de tous les mots rencontrés, il n'édite que celui des mots se situant à une génération fixée à l'avance (au moyen de la variable NGEN, abrégé de nombre de générations).

Pour ce faire, il est nécessaire de disposer d'un compteur, la variable COMPT, qui est incrémentée ou décrémentée à l'aide des deux mots INC et DEC qui sont sans effet sur la pile.

Le mot INC est employé chaque fois que l'on descend une génération, par l'intermédiaire du mot ENFANT. Le mot DEC chaque fois que l'on passe au « frère » d'un benjamin.

TESTPD (PFA – PFA flag), abrégé de test de poursuite de descente, vérifie si l'individu rencontré se trouve à la génération voulue. Si oui, il procède à l'affichage de l'en-tête puis dépose un 0 sur la pile et sort du mot. Le dépôt du 0 sert à indiquer que la descente ne doit pas se poursuivre et qu'il faut passer à un frère qui sera le père si l'on se trouve à un benjamin. Si non, il vérifie que l'individu a un enfant.

On remarque que l'algorithme emploie au moyen des mots O UNTIL une boucle infinie dont on ne sort que par le mot EXIT qui est déclenché lorsque l'on revient à la racine, c'est-à-dire lorsque la variable NGEN est à nouveau nulle.

ENUMDESC, abrégé de enumére les descendants de la génération n, doit trouver sur la pile le PFA de l'individu racine choisi, surmonté du numéro de la génération que l'on souhaite afficher.

Parcours de l'arbre des ascendants

Le principe de l'algorithme précédent ne peut être repris car, à chaque mouvement de retour vers la racine, il faut pour tout père ou toute mère retrouver l'enfant par lequel on était passé pour aller jusqu'à eux. La seule possibilité est donc de laisser sur la pile l'adresse de chaque élément traversé dans le sens de l'aller pour pouvoir après dépile le retrouver au retour.

Par ailleurs, ce mécanisme d'empilage-dépilage, s'il permet de garder le mécanisme précédent d'incrémentation, n'autorise pas à garder celui de décrémentation d'un compteur extérieur. La solution est alors d'accompagner chacun des éléments empilés par le numéro de sa génération (on mettra ce numéro sous chaque adresse empilée) de façon à créer la pile suivante :

Racine 1 mère 2 mère 3 mère... n mère n père.

Le mot de parcours de l'arbre des ascendants PARBASC doit être précédé par le PFA de la racine choisie. Il appelle un certain nombre de commentaires.

Grâce au mot EXECUTE précédé du rappel d'une variable-test, le mot de parcours se présente sous une forme très générale de façon à pouvoir utiliser des tests d'aiguillage différents. Différences qui sont nécessaires selon que l'arbre est parcouru pour énumérer des ancêtres, procéder au marquage d'une généalogie, rechercher cette marque puis l'effacer.

La première variable-test est TESTPM, abrégé de test de poursuite de montée. Lorsque le test est positif, on remonte par l'intermédiaire du mot PERE d'une génération en incrémentant le numéro correspondant grâce à la séquence SWAP 1 + SWAP. Si l'individu obtenu a un conjoint, on empile le PFA de ce dernier (DUP CONJOINT) ainsi que son numéro de génération (3 PICK), ROT rétablit l'ordre nécessaire dans la pile. Lorsque le test est négatif, c'est-à-dire quand on arrête le mouvement d'éloignement de la racine, il faut procéder à un double dépilage.

La seconde variable-test, TESTFP, abrégé de test de fin de parcours, sert à sortir de la boucle BEGIN... UNTIL.

Enumération des ascendants

L'énumération des ascendants de la nième génération d'un individu se fait au moyen du mot ENUMASC qui utilise le mot PARBASC dans lequel la variable test de poursuite de montée a pour valeur le CFA du mot TPM1 et la variable test de fin de parcours, celle du CFA du mot TFP1.

TPM1 (PFA - PFA flag) fonctionne de la même façon que le mot TESTPD.

TFP1 (PFA - PFA flag) vérifie si après un double dépliage, on se trouve en présence du PFA de la racine.

On remarquera la façon dont les variables-test se voient affecter leur valeur : utilisation des deux mots [et] pour que, lors de la compilation, on passe en mode exécution le temps que le mot FIND aille chercher le PFA voulu et le dépose sur la pile ; le rôle du mot LITERAL est alors de prendre ce PFA et de le compiler dans la zone paramètre du mot en cours de chargement dans le dictionnaire.

De la même façon que ENUMDESC, le mot ENUMASC doit être précédé, lors de son emploi, du nom de l'individu choisi et du numéro désiré.

Après exécution du mot PARBASC, le PFA de la racine reste sur la pile. D'où le DROP en fin de mot. Nous verrons par la suite pourquoi ce DROP n'a pas été intégré au mot PARBASC comme il aurait semblé logique de le faire.

Dernière observation importante : faire très attention à toute modification du dictionnaire avant les mots employant LITERAL, car les adresses compilées par ce dernier risquent de ne plus être les bonnes.

Recherche d'un ancêtre commun

Nous avons vu qu'il fallait marquer les ascendants de l'un des deux individus, puis remonter l'arbre de l'autre en recherchant le premier ascendant portant la marque « précédent ».

Pour marquer chacun des ascendants rencontrés, on utilise le bit 1 de l'octet indicateur resté jusqu'à présent disponible. L'état de ce bit est modifié chaque fois que l'on utilise le mot MARQ (PFA - PFA), ce qui permet avec le même mot de procéder au marquage d'un individu, puis à l'effacement de la

marque. Le mécanisme utilisé réside dans la séquence 1 XOR du mot.

Le processus de marquage de la totalité d'un arbre est concrétisé par le mot MARQARB, abrégé de marquage de l'arbre des ascendants. Ce mot utilise PARBASC dans lequel le test de poursuite de montée concrétisé par TPM2 se réduit au mot PERE ? précédé du mot MARQ, tandis que le test de fin de parcours reste le même, c'est-à-dire TFP1. Le DROP final efface la racine restant sur la pile.

La recherche de la marque dans le second arbre se fait au moyen de RECHMARQ, abrégé de recherche de la marque, lui aussi utilisant le mot PARBASC dans lequel le test de poursuite de montée se réduit à PERE ? Quant au test de fin de parcours, TFP2, il consiste à vérifier si l'octet indicateur contient la quantité 1, séquence 1 IND ?, et dans l'affirmative à sortir du mot TFP2 après avoir édité le nom de l'individu trouvé et déposé un 1 sur la pile pour sortir ensuite de la boucle BEGIN UNTIL du mot PARBASC.

En interrompant de la sorte le parcours, la pile reste encombrée de tous les éléments qui y avaient été placés en vue des mouvements de retour vers la racine. D'où le mot EFF qui procède à leur effacement dans le mot final LIENPAR ? (PFA PFA'...) extrêmement concis, dont on lit facilement les trois étapes, marquage, recherche de la marque et effacement des marques. Le mot OVER sert à conserver un point de départ pour le dernier parcours.

Conclusion

Arrivé au terme de notre problème, il est évident que le programme proposé, opérationnel sur Jupiter Ace et facilement adaptable sur d'autres systèmes moyennant la refonte du mot AF, peut être encore perfectionné surtout au niveau des messages de recherche infructueuse.

Néanmoins, nous pensons avoir apporté notre contribution à la démonstration des possibilités de Forth capable, à notre sens, de tailler des croupières à Pascal puisqu'il permet de visualiser l'organisation en mémoire des données structurées complexes. ■

```

DEFINER MM 0 C, 0, 0, 0, DOES>;
DEFINER MF 16 C, 0, 0, 0, DOES>;
: IND? OVER @ AND 0> ;
: FILS? 128 IND? ;
: FRERE? 64 IND? ;
: BENJAMIN 32 IND? ;
: SEXE? 16 IND? ;
: MARIE? 8 IND? ;
: PERE? 4 IND? ;
: ENFANT? 128 IND? SWAP 152 IND? ROT OR ;
: SXOP? SEXE? ROT SEXE? ROT XOR 0= IF . "erreur"
ABORT THEN ;

: FILS 1+ @ ; ;
: FRERE 3+ @ ; ;
: CONJOINT 5+ @ ; ;
: ENFANT SEXE? IF CONJOINT THEN FILS ;
: PERE BEGIN BENJAM? 0 = WHILE FRERE
REPEAT FRERE ;

: >IND OVER C@ OR OVER C! ;
: EP SXOP? OVER OVER SWAP 8 >IND 5+ ! 8 >IND 5+ ! ;
: FU OVER OVER SWAP 128 >IND 1+ ! 100 >IND 3+ ! ;
: CEL DUP >R 1+ PICK OVER 100 >IND 3+ ! R> 1-0 DO
OVER 68 >IND 3+ LOOP SWAP 128 >IND 1+ ! ; ;

0 VARIABLE COMPT
0 VARIABLE NGEN
0 VARIABLE RACINE

: INC COMPT DUP @ 1+ SWAP ! ;
: DEC COMPT DUP @ 1- SWAP ! ;
: AF DUP 3- DUP C@ SWAP 4- OVER - SWAP TYPE
SPACE ;

: TESTPD NGEN @ COMPT @ = IF AF 0 EXIT THEN
ENFANT ? ;
: PARBDESC ENFANT? 0= IF DROP . "pas d'enfant" THEN
BEGIN TESTPD
IF ENFANT INC
ELSE BEGIN BENJAM?
WHILE FRERE DEC COMPT @ 0= IF DROP EXIT
THEN
REPEAT FRERE
THEN 0
UNTIL ;
: ENUMDESC NGEN ! 0 COMPT ! PARBDESC ; ;

0 VARIABLE TESTPM
0 VARIABLE TESTFP
: PARBASC DUP RACINE ! 0 OVER
BEGIN TESTPM @ EXECUTE
IF PERE SWAP 1+ SWAP MARIE? IF DUP CONJOINT
3 PICK ROT THEN
ELSE DROP DROP
THEN TESTFP @ EXECUTE
UNTIL ;
: TPM1 OVER NGEN @ = IF AF 0 ELSE PERE? THEN ;
: TFP1 DUP RACINE @ = ;
: ENUMASC NGEN ! [FIND TPM1] LITERAL TESTPM !
[FIND TFP1] LITERAL TESTFP ! PARBASC DROP ;
: ENUMTASC 0 DO DUP I ENUMASC LOOP ;
: MARQ DUP C@ 1 XOR OVER C! ;
: TPM2 MARQ PERE? ;
: MARQARB [FIND TFP1] LITERAL TESTFP ! [FIND
TPM2] LITERAL TESTPM ! PARBASC DROP ;
: TFP2 1 IND? IF AF 1 EXIT THEN TFP1 ;
: RECHMARQ [FIND TFP2] LITERAL TESTFP ! [FIND
PERE?] LITERAL TESTPM ! PARBASC ;
: EFF BEGIN RACINE @ = UNTIL ;
: LIENPAR? OVER MARQARB RECHMARQ EFF
MARQARB ; ;

```

Listing du programme.

Une tortue Logo en Basic

Si le logiciel proposé ici est développé sur un micro-ordinateur Vic 20 équipé de la cartouche super-expander et d'une 16 Ko, il peut également fonctionner sur n'importe quel matériel avec quelques modifications qui seront indiquées. Rien ne vous empêche de changer le nom des mots clefs, ou d'en rajouter d'autres, etc.

Le programme

Tel qu'il est présenté (**tableau 1**), le programme se compose de 7 ordres différents qui ne peuvent être employés que manuellement.

Au nombre de 16, les instructions dites « initiales » (**tableau 2**), contrairement aux ordres, ne sont exploitables que dans le mode programme, afin de définir d'autres mots qui sont utilisables d'une part dans un programme et d'autre part manuellement.

Le nombre de mots définis est fonction de la capacité mémoire du micro-ordinateur. Ils peuvent être manipulés autant de fois qu'on le désire et être ainsi imbriqués jusqu'à cent fois « les uns dans les autres ».

De plus, si la mémoire de votre ordinateur l'autorise, rien ne vous empêche d'augmenter ce chiffre ! La seule précaution à prendre lors des imbrications est de ne pas constituer des boucles interminables qui se finiraient tout de même par une erreur de dimension.

Gestion des erreurs

Le Vic 20 ne disposant pas d'instruction ON ERROR, celle-ci est simulée à la ligne 200.

L'astuce consiste à inscrire dans le tampon clavier : GOTO 4 (RETURN), ce qui donne le même effet que ON ERROR GOTO 4. Ainsi, sur d'autres micro, la ligne 200 devient ON ERROR GOTO 4. De cette manière, une erreur de dépassement d'écran par exemple ne nous fait pas quitter Logo.

Les mots inconnus, ou encore mal employés, déclenchent un message tel que : « Je ne connais pas xxxx » ; si le mot désigne un ordre ou s'il se trouve dans un autre mot, l'ordinateur affiche : « Comment faire xxx » ; si en plus le mot

inconnu ne se trouve pas dans celui ayant servi d'ordre, notre micro répond : « comment faire xxxx dans yyyy » (**tableau 3**).

Au cœur du programme : la pile

Afin de simuler la récurrence, une pile est constituée (variable PL(n)) ainsi qu'un pointeur de pile (P %).

Chaque instruction composant un mot est codée par le numéro du mot dont elle fait partie et par son classement dans celui-ci (variable MCS(n,p)). A noter que la taille des mots est limitée à 4 lettres pour des raisons de mémoire.

Examinons maintenant le fonctionnement de la pile. Lors de l'exécution d'un mot, le code de la prochaine instruction à exécuter est stocké dans PL(P %) de la façon suivante : les milliers désignent le numéro du mot dont elle fait partie, les centaines, dizaines et unités nous donnant son classement (par exemple PL(n)=3012 désigne dans le mot numéro 3 la 12^e instruction).

De la même façon, l'exécution de « REPE n » inscrit dans PL(P %) n fois le code de l'instruction suivant « REPE n ». Dans ce cas, le pointeur de pile (P %) est incrémenté de « n ».

L'éditeur pleine page

Si vous disposez en Basic d'un éditeur pleine page comme c'est le cas sur le Vic, vous pouvez, pour corriger un mot, le lister tout d'abord grâce à la commande « EDIT xxxx », puis remonter le curseur sur EDIT et écrire à la place POUR, et enfin sauter de lignes en lignes avec la touche RETURN.

Si le mot à corriger n'est pas trop long, ou si vous ne disposez pas d'un tel éditeur, vous pouvez le réécrire avec l'ins-

LANGAGE : Logo

de Stéphane SABBAGUE

A l'heure où la programmation en Logo fait de grands pas, ce programme permet de simuler la partie graphique de Logo sur votre ordinateur si celui-ci possède la haute résolution.

Langages : Basic + Assembleur 6502

Ordinateur : Commodore Vic 20

MOTS	: donne la liste des mots employés
EDIT xxxx	: liste le mot xxxx sur l'écran
LIST xxxx	: liste le mot xxxx sur l'imprimante
POUR xxxx	: définition du mot xxxx
FIN	: fin de la définition
ENRE xxxx	: enregistre sur cassette le mot xxxx
CHAR xxxx	: charge le mot xxxx qui est sur la cassette

Tableau 1. – Les sept ordres définissant le programme.

AVAN n	: avance de n, si n < 0 alors on recule
DROI n	: tourne à droite d'un angle de n degrés.
GAUC n	: tourne à gauche d'un angle de n degrés
REPE n	: répète n fois les instructions qui suivent
INCR n	: incrémenté l'avancement de n
AUGM n	: augmente l'angle de n degrés
AFFI xxxxxx	: affiche à l'écran xxxxxx
COUL n p q r	: couleur écran, bords et caractères
EFFA	: efface l'écran
PEIN	: peint
CENT	: positionne la tortue au centre
MAIS	: tortue à la maison (en haut à gauche)
CACH	: cache la trace de la tortue
MONT	: remonte la trace
CTOR	: cache la tortue à la fin du dessin
MTOR	: montre la tortue à la fin du dessin

Tableau 2. – Les seize instructions dites « initiales ».

EDIT TYPE	EDIT POUX
CENT	VIVE
POUX	FIN
FIN	
TYPE	
COMMENT FAIRE VIVE DANS POUX ?	
(Il se produit une erreur si on a oublié de définir VIVE.)	

Tableau 3. – Exemple de message d'erreur.

GRAPHIC2	: mise en mode haute résolution (1,500)
GRAPHIC0	: mise en mode texte (1, 20, 594)
Les POKE L.30	: supprime le «?» après INPUT
Les POKE L.90	: met le curseur en mode définition
Les POKE L.200	: ERROR GOTO 4
PRINT «xxxx»	: joue les notes xxxx (220, 420, 594)
DRAW2, x, ytoz,v	: trace une ligne de (x, y) à (z, v)
COLOR a,b,c,d	: change les couleurs
SNCLR	: efface l'écran graphique (1400)
PAINT2,x,y	: peint à partir du point (x,y)
CHARx,y,a\$: affiche a\$ ligne x, colonne y (1992)
WAIT 653,4	: attend la frappe de « ctrl »

Remarque : Pour revenir au texte lorsque le dessin est fini, il faut appuyer sur la touche « CTRL ».

Adaptation du programme à d'autres micro-ordinateurs.

Liste des variables

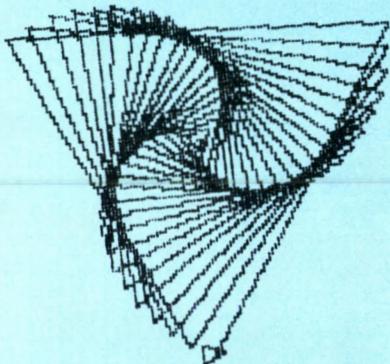
AG	: angle
AM	: ancien angle
D	: n° d'instruction
M\$ (n)	: mots utilisateurs
N\$ (n)	: mots d'origine
MC\$ (n,p)	: instructions
PB	: ancien avancement
PL(n)	: pile
P%	: pointeur de Pile
(XT, YT)	: anciennes coordonnées
(XN, YN)	: nouvelles coordonnées

Exemples de créations graphiques

Pour programmer TRIA il faut écrire :

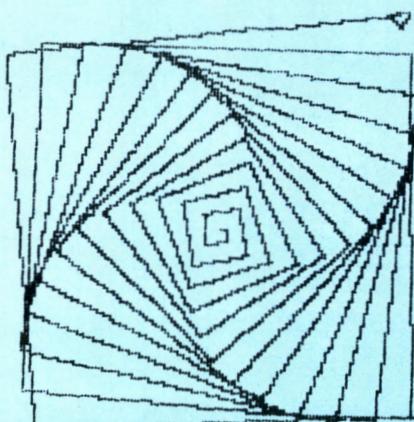
```
POUR TRIA
CENT
REPE 60
INCR 13
DROI 118
FIN
```

L'ordinateur affiche READY. Puis, il suffit d'écrire TRIA suivi de (RETURN)



Pour réaliser CARE, le processus est identique, seules les valeurs numériques changent.

```
POUR CARE
CENT
REPE 50
INCR 16
DROI 88
FIN
```



```

10 REM*****
20 REM* COPIE ECRAN *
30 REM* HAUTE RESOLU. *
40 REM*****
50 REM STEPHANE SABBAGUE 1983
52 :
60 GRAPHIC2:GRAPHICO
70 S=1024
80 REM*CODAGE EN HEXA#
90 REPEAT: IF R$="*THENPRINT":LA ROUTINE S'APPELLE AVEC SYS 1024":END
100 R1$=LEFT$(R$,1)
110 N1=ASC(R1$)-48:IF N1>9THEN N1=N1-7
120 R1$=RIGHT$(R$,1)
130 N=ASC(R1$)-48:IF N>9THEN N=N-7
140 N1=N1*16+N:POKES,N1:S=S+1:GOTO 90
150 DATA09,01,R2,04,R0,00,20,BR,FF,R9,00,20,BI,FF,20
160 DATA00,FF,R2,01,20,C9,FF,R9,08,20,I2,FF,R9,00,85
170 DATA01,85,62,R9,10,85,02,85,63,R9,14,85,64,R9,08
180 DATA05,65,R9,06,65,B1,62,4A,CA,D0,FC,26,FB,88
190 DATA10,F3,A5,FB,09,80,20,I2,FF
195 DATA0A,ER,ER,ER,ER,ER
200 DATA0C,65,D0,DF,ER,ER,C6,64,F0,10,18,R5,62,69
210 DATA00,FF,R5,62,R5,63,69,00,85,63,18,90,C5,R9,00,20
220 DATA02,FF,18,R5,01,89,07,85,01,R5,02,69,00,85,02
230 DATA09,10,00,08,R5,01,C9,R1,D0,02,F0,0B,R5,02,85
240 DATA03,65,01,85,62,18,90,98,R9,0F,20,I2,FF,20,CC
250 DATAFF,R9,01,20,C3,FF,60,ER,*
```

300 :
310 : EN REMPLACANT LA LIGNE 195 PAR :DATA 20,I2,FF,ER,ER,ER
315 : ON OBTIENT LA COPIE EN DOUBLE LARGEUR
320 : 325 : EN REMPLACANT LA LIGNE 195 PAR :DATA 20,I2,FF,20,I2,FF
330 : 335 : ON OBTIENT LA COPIE EN TRIPLE LARGEUR
340 :
READY.

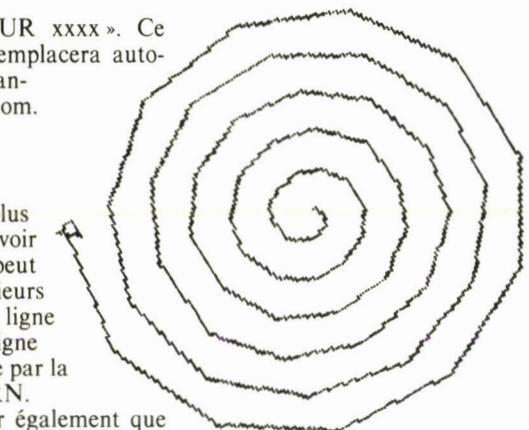
Routine écrite en assembleur 6502, pour effectuer des copies d'écran.

instruction : « POUR xxxx ». Ce nouveau mot remplacera automatiquement l'ancien du même nom.

La syntaxe

La chose la plus importante à savoir est que l'on ne peut pas mettre plusieurs instructions par ligne et que chaque ligne doit être validée par la touche RETURN.

Il faut savoir également que les espaces entre les mots sont obligatoires sauf s'il s'agit d'une instruction et de son paramètre (par exemple, AVAN30 est aussi correct que AVAN 30) l'espace étant mis automatiquement lors de l'édition par le programme.



Pour rajouter des commandes, il faut les mettre en 3400 (voir les lignes 3100 à 3150 par exemple).

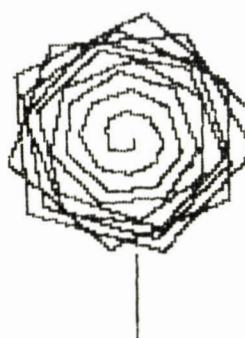
Et si, en plus...

Si vous disposez de l'imprimante graphique du Vic (type GP-100VC ou CBM) et que vous désirez stocker vos plus beaux dessins sur papier, nous vous donnons la routine écrite en assembleur 6502 effectuant la copie de l'écran haute résolution du Vic. Il vous suffit pour cela de faire « tourner » cette routine, puis de charger le programme Logo et d'inclure à la ligne 2100 la commande SYS 1024. Ainsi, l'écran sera automatiquement stocké sur l'imprimante dès la fin du dessin.

S'il vous était arrivé de « rebomber » dans Basic, il faut faire un « GOTO 20 » pour revenir à Logo sans avoir perdu la mémoire ! ■

Pour étendre Logo

Pour intégrer d'autre mots clefs, il faut redimensionner n\$ (ligne 2), puis agir sur les lignes 15, 16, 600 et programmer le mot désiré (voir lignes 700 à 760 par exemple).



LOGO

```

0 REM-----INITIALISATION-----
1 GRAPHIC2:GRAPHIC0
2 DIMPL(100),MC$(30,30),M$(80),N$(16):GOT010
3 REM-----ERREUR-----
4 PRINT"■■■■■":PRINT" ";:GOT020
5 :
6 REM*****DEBUT+MOTS CLEFS*****
7 :
10 PRINT"*** VIC-LOGO V2 ***"
12 PRINT"***3200 WORDS FREE"
15 N=16:FORT=1TO16:READN$(T):NEXT
16 DATAAVAN,DROI,GAUC,REPE,CACH,MONT,INCR,EFFA,COUL,PEIN,MAIS,CENT,CTOR,MTOR,AUG
M,AFFI
17 CM=1:TR=1
19 REM-----ENTREE+INITIA TAMPON CLAVIER-----
20 GRAPHIC0
23 AG=0:PB=0
24 PX=0:AM=0
25 XT=0:YT=0
30 PRINT:PRINT"READY.":D=0:POKE631,20:POKE632,20:POKE633,17:POKE198,3
35 INPUTA$:
36 :
37 REM*****TEST*****
38 :
40 L$=LEFT$(A$,4)
50 IFL$<>"POUR"THEN200
52 REM-----SP-ECRITURE-----
55 PRINT
60 V$=MID$(A$,6,4)
70 FORT=1TOM:IFV$=M$(T)THENK=T:FORY=1TO12:MC$(T,Y)=""NEXTY:GOT090
80 NEXTT:M=M+1:K=M:M$(M)=V$
90 POKE631,157:POKE632,157:POKE633,32:POKE634,32:POKE198,4:INPUTA$:
100 IFA$="FIN"THEN20
102 IFMID$(A$,5,1)<>" "ANDLEN(A$)>4THENA$=LEFT$(A$,4)+" "+MID$(A$,5,9)
110 D=D+1:MC$(K,D)=A$:GOT090
190 REM-----SP-EDITION-----
200 POKE631,71:POKE632,207:POKE633,52:POKE634,13:POKE198,4
205 IFL$<>"EDIT"THEN300
210 PRINT:ED$=MID$(A$,6,4):FORT=1TOM
220 IFED$<>M$(T)THENNEXT:PRINT"■38":PRINTED$" M'EST INCONNU!":GOT020
230 K=0
240 K=K+1
250 IFMC$(T,K)<>""THENPRINT" "MC$(T,K):GOT0240
255 PRINT" FIN"
260 GOT020
290 REM-----LISTE DES MOTS-----
300 IFL$<>"MOTS"THEN400
310 FORT=1TON:PRINTN$(T)"/":NEXT
320 FORT=1TOM:PRINTM$(T)"/":NEXT:GOT020
400 FORT=1TOM:IFL$=M$(T)THEN500
410 NEXT
415 GOT03000:REM----SAUT POUR DEFINIR D'AUTRES ORDRES-----
418 REM-----INSTRUCCION INCONUE-----
420 PRINT"■38":PRINT"JE NE CONNAIS PAS : ";A$:GOT020

```

Listing du programme.

```

470 :
480 REM*****EXECUTE LES PROGRAMMES*****
490 :
500 GRAPHIC2
510 P% = P%+1:D=0:REM----INCREmente LE POINTEUR DE PILE---
520 D=D+1
530 PL(P%)=D+1+T*1000:REM----RANGE ADRESSE PROCHAINE INSTRUTION DANS LA PILE---
535 REM----TEST SI LE MOT EST CONNU---

540 IF MC$(T,D)=="THEN2000"
550 FOR S=1 TO M%
560 IF MC$(T,D)=M$(S) THEN T=S:GOT0510
570 NEXT
580 FOR S=1 TO N%
590 IF LEFT$(MC$(T,D),4)=N$(S) THEN NEXT:GOT0594
592 GOT0600
594 GRAPHIC0:PRINT "#33A"
595 PRINT "COMMENT FAIRE "MC$(T,D): IF P%-1>0 THEN PRINT "DANS "M$(INT(PL(P%)/1000))
598 GOT020
599 REM---REVOIT VERS EXECUTION---

600 ON S600 SUB 700,800,900,1000,1100,1200,1300,1400,1500,1600,1700,1800,1900,1950,1
980,1990
610 GOT0520
620 REM----EXECUTE AVAN-
630 J=LEN(MC$(T,D)):B=VAL(MID$(MC$(T,D),6,J))
640 FB=B
650 XN=B*COS(2*pi*AG/360)+XT
660 YN=B*SIN(2*pi*AG/360)+YT
670 IF CM=1 THEN DRAW2(XT,YT,XN,YN)
680 XT=XN:YT=YN
690 RETURN
700 REM----EXECUTE DROI-
710 J=LEN(MC$(T,D)):B=VAL(MID$(MC$(T,D),6,J))
720 AG=AG+B:AM=B:RETURN
730 REM----EXECUTE GAUCH-
740 J=LEN(MC$(T,D)):B=VAL(MID$(MC$(T,D),6,J))
750 AG=AG-B:AM=-B:RETURN
760 REM----EXECUTE REPE-
770 J=LEN(MC$(T,D)):B=VAL(MID$(MC$(T,D),6,J))
780 FOR Z=P% TO M%+B-1
790 PL(Z)=PL(P%):NEXT:P%=P%+B-1
800 RETURN
810 REM----EXECUTE CACH-
820 CM=0:RETURN
830 REM----EXECUTE MONT-
840 CM=1:RETURN
850 REM----EXECUTE INCR-
860 J=LEN(MC$(T,D)):B=VAL(MID$(MC$(T,D),6,J))
870 B=B+PB:GOT0710
880 REM----EXECUTE EFFA-
890 SCNCLR:RETURN
900 REM----EXECUTE COUL-
910 J=LEN(MC$(T,D)):A=VAL(MID$(MC$(T,D),6,1))

```

Listing (suite).

```

1510 J=LEN(MC$(T,D)):B=VAL(MID$(MC$(T,D),8,1))
1520 J=LEN(MC$(T,D)):C=VAL(MID$(MC$(T,D),10,1))
1530 J=LEN(MC$(T,D)):E=VAL(MID$(MC$(T,D),12,1))
1540 COLORA,B,C,E:RETURN
1550 REM-----EXECUTE PEIN-----
1600 PAINT2,XT,YT:RETURN
1650 REM-----EXECUTE MAIS-----
1700 XT=0:YT=0:RETURN
1750 REM-----EXECUTE CENT-----
1800 XT=512:YT=512:RETURN
1850 REM-----EXECUTE CTOR-----
1900 TR=0:RETURN
1940 REM-----EXECUTE MTOR-----
1950 TR=1:RETURN
1970 REM-----EXECUTE AUGM-----
1980 J=LEN(MC$(T,D)):B=VAL(MID$(MC$(T,D),6,J))
1982 RM=AM+B:AG=AG+RM
1984 RETURN
1985 REM-----EXECUTE AFFI-----
1990 J=LEN(MC$(T,D)):B$=MID$(MC$(T,D),6,J-5)
1992 CHARYT/51.2,XT/51.2,B$:RETURN
1995 REM-----TEST LA FILE EN FIN D'INSTRUCTION-----
2000 IF P%-1>0 THEN P%=P%-1:T=INT(PL(P%)/1000):D=INT(((PL(P%)/1000)-T)*1000+.5):GOT
0530
2005 REM-----CALCUL POUR DESSINER LA TORTUE-----
2010 XB=25*COS((AG+90)*2*pi/360):YB=25*SIN((AG+90)*2*pi/360):X1=XB+XT:Y1=YB+YT
2020 X2=XT-XB:Y2=YT-YB:X3=XT+30*COS(AG*2*pi/360):Y3=YT+30*SIN(AG*2*pi/360)
2025 IF TR=0 THEN 2100:REM-----DOIT-ON LA DESSINER?-----
2030 DRAW2,X1,Y1TOX2,Y2TOX3,Y3TOX1,Y1
2100 WAIT653,4:GOTO20:REM-----ATTEND LA FRAPPE DE 'CTRL'-----
3000 IF L$<>"ENRE"THEN3100
3005 REM-----EXECUTE ENRE-----
3010 J$=MID$(A$,6,4):FORT=1TOM:IF J$<>M$(T)THENNEXT:A$=J$:GOTO420
3020 OPEN1,1,1,J$
3030 D=0
3040 D=D+1:IF MC$(T,D)<>""THENPRINT#1,MC$(T,D):GOTO3040
3045 PRINT#1,"@"
3050 CLOSE1:GOTO20
3100 IFL$<>"CHAR"THEN3200
3105 REM-----EXECUTE CHAR-----
3110 J$=MID$(A$,6,4):M=M+1:M$(M)=J$
3120 OPEN1,1,0,J$
3130 D=0
3140 D=D+1:INPUT#1,O$:IF O$<>"@*"THENMC$(M,D)=O$:GOTO3140
3150 CLOSE1:GOTO20
3200 IFL$<>"LIST"THEN3400
3202 REM-----EXECUTE LIST-----
3205 OPEN4,4
3210 ED$=MID$(A$,6,4):FORT=1TOM
3220 IF ED$<>M$(T)THENNEXT:PRINT"■S3A":PRINTED$" M'EST INCONU!":CLOSE4:GOTO20
3230 K=0:PRINT#4,CHR$(14)" "ED$/CHR$(15):PRINT#4
3240 K=K+1
3250 IF MC$(T,K)<>""THENPRINT#4,MC$(T,K):GOTO3240
3260 PRINT#4,"FIN":PRINT#4:CLOSE4:GOTO20
3400 GOT0420

```

EN TOURAINE 37

PLUS DE 20 MODÈLES DISPONIBLES

ZX 81*/SPECTRUM*
578 F ZX 81

ORIC 48 K*
ATMOS

DRAGON 32*
32 K - 64 K
NOMBREUX LOGICIELS

MULTITECH MPF 2
COMPATIBLE GRANDE MARQUE
avec MODIFICATION

COMMODORE

THOMSON

LYNX*
48 K - 96 K - 128 K - 192 K
48 K: 2980 F

ADVANCE 86

PAP 16 BIT 128 K

VENTE DIRECTE DÉPÔT
J-60



LOGICIELS

EPISTOLE* TRANSFORME votre J-50, J-60 ou autre grande marque en une puissante machine à écrire, stock, fichier, etc.

SAARI POUR COMPTABILITÉ : paie, gestion, stock, facturation, etc.

NOMBREUX LOGICIELS
pour particuliers et professionnels

LE COIN COMPATIBLE
PRODUITS POUR J-50, J-60,
MULTITECH modifié ou autre grande
marque (Apple marque déposée)

LECTEURS DE DISQUETTES
KATO **2080 F**
TANDON* **2590 F**
SIEMENS **2690 F**
HITACHI 3"500 K **3150 F**

CARTE PROFESSIONNELLE
nombreux modèles*

Z 80
80 col
16 K
DRIVE
128 K, 192 K, etc.

NOUS CONSULTER

VENTILATEUR
POIGNÉE DE JEUX

ORDINATEURS PROS
GARANTI 1 AN

J-50 48 K*	4950 F
J-50 64 K*	5850 F
J-60 64 K*	6800 F

avec Z 80-6502, clavier détachable,
fonction intégrée

DES MILLIERS
DE PROGRAMMES DISPONIBLES
POUR J-50, J-60

IMPRIMANTES MX 80 - BX 80* - GP 100* - GP 700* couleur, etc.
PRIX PROMO INCROYABLE SUR* BMC BX 80 PROFESSIONNEL

MONITEUR*
42 cm, couleur **2780 F**

TAXAN-PRO
36 cm, haute définition couleur

VERT ZENITH*	980 F
VERT PHILIPS*	1180 F
AMBRE PHILIPS	1580 F

DÉPÔT 1000 m²

JCC ELECTRONIC

Z.I. - Boulevard de l'Avenir
37400 NAZELLES-AMBOISE
T. (47) 57.44.22 lignes groupées

Vente directe dépôt
S.A.V. ASSURÉ

*Stock important

2000 ARTICLES EN STOCKS

DISPONIBILITÉ SUIVANT STOCK.
PRIX INDICATIFS SELON FLUCTUATION MONETAIRE

MAGASIN

JCC ELECTRONIC

53, rue de la Fuye

37000 TOURS

Tél. (47) 46.24.97 - 46.24.98
Ouvert 10 h à 13 h - 14 h à 19 h

POSSIBILITÉ DE CRÉDIT TOTAL

RÈGLEMENT 2 MOIS APRÈS

POSSIBILITÉS LOCATION-VENTE

CREDIT CETELEM

OUVERTURE
MARDI AU SAMEDI 9 H A 12 H ET 14 H A 19 H

CATALOGUE CONTRE 5 F
DISPONIBLE AVRIL

Graphisme assisté par ordinateur sur Canon X 07

Le Canon X 07 possède 64 caractères graphiques redéfinissables au gré de l'utilisateur par la fonction FONT\$. Mais le calcul des paramètres de cette fonction devient vite fastidieux. Si vous souhaitez composer un dessin utilisant plusieurs dizaines de « FONT\$ » la tâche devient herculéenne !

Le logiciel proposé ici vous permet de dessiner en toute liberté sur l'écran, de calculer puis de mémoriser automatiquement les caractères graphiques créés, et surtout d'archiver sur cassette le dessin réalisé. En prime, une recopie d'écran sur X-710 (rapide !) vous laisse un souvenir impérissable.

Initialisation

Le programme est lancé par RUN. Si vous êtes plusieurs semaines sans utiliser le logiciel, il est possible de revoir rapidement le rôle des touches utiles.

Après les explications (éventuelles) une séquence de musique aléatoire réjouit vos oreilles pendant quelques secondes.

Une pression sur une touche quelconque permet de passer aux choses sérieuses.

L'entrée des paramètres et le choix des options

Tout d'abord, il est nécessaire d'entrer les dimensions du cadre dans lequel s'inscrira le dessin. Ces dimensions sont exprimées en nombre de caractères alphanumériques ($L \leq 20$ et $H \leq 4$, L et H représentant respectivement le nombre de colonnes et le nombre de lignes).

En outre, l'utilisateur ne disposant que de 64 caractères graphiques pour mémoriser un dessin, le programme refuse tout produit $L \times H > 64$.

Cette limitation est mise à profit pour créer un espace de « dialogue » entre l'utilisateur et le programme. Cet espace est situé soit dans les trois colonnes de droite si $L \leq 16$, soit sur la dernière ligne dans le cas contraire (en effet, la surface-

écran maxi utilisable est soit 16×4 , soit 20×3).

- Le programme laisse ensuite la possibilité de définir le numéro du premier caractère graphique à utiliser pour la mémorisation (dans la limite, bien sûr, des 64 possibles). Cette option est particulièrement intéressante si vous voulez sauvegarder plusieurs petits dessins à la fois (par exemple 4 dessins de 4×4), utilisables ensuite pour une animation graphique.

- Le choix de l'option « sauvegarde sur cassette » vous permet de donner un titre au dessin. Ce titre sera mémorisé en début d'enregistrement.

- Il est possible de sortir sur imprimante les paramètres décimaux des « FONT\$ » calculés (fig. 1). A ce moment, la sortie n'est possible que sur imprimante puisque l'écran est « protégé » pendant la phase de mémorisation. Par contre, à l'issue de cette phase, la sortie des paramètres des FONT\$ (en hexadécimal) est possible sur écran et/ou sur imprimante.

Le dessin proprement dit

La zone graphique apparaît, délimitée sur l'écran par un cadre. Le curseur est situé en (0,0) et clignote. Pour dessiner, l'utilisateur dispose alors des possibilités suivantes :

- Les quatre touches de déplacement du curseur (\blacktriangleleft , \blacktriangleright , \blacktriangledown , \blacktriangleup) permettent de dessiner des droites dans les quatre directions, plus les diagonales à 45° par appui simultané sur deux de ces touches.
- Pour effacer, il suffit d'appuyer sur la touche F6 située au milieu du pavé des touches de déplacement. Un bip sonore est émis et une étoile (*) apparaît

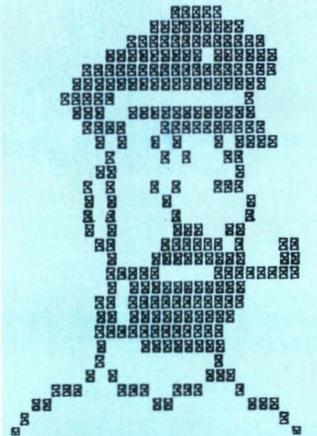
**UTILITAIRE :
« Course aux FONT\$ »
d'Alain NOGUES**

**Le tracé de formes sur un micro-ordinateur peut être fastidieux.
Avec cet utilitaire, cela devient un régal.**

**Langage : Basic
Ordinateur : Canon X 07**

```
1 , *****  
2 , * *****  
3 , * COURSE AUX FONT$ *  
4 , * sur Canon X-07 *  
5 , * *****  
6 , * AUTEUR: Alain NOGUES *  
7 , * *****  
8 , *****  
9 ,  
20 CLEAR 600:DEFSTRA-D:DEFINTE-Z  
21 CONSOLE0,4,,1,  
25 CLS:LOCATE6,0:PRINT"Bonjour ?"  
30 LOCATE4,1:PRINT"Voulez-vous "  
31 LOCATE1,2:INPUT"des explications":D  
32 IFLEFT$(D,1)="O":THENGOSUB200  
33 '-----PRESENTATION-----  
--  
35 J=VAL(RIGHT$(TIME$,2))/4  
36 FORI=0TOJ:K=15*RND(1):NEXT:J=1  
40 CLS:FORI=1TO18:LOCATEI,0:PRINT"*";  
41 N=K+J*10*RND(1)  
42 LOCATEI,3:PRINT"*";BEEPN,3:J=-J  
43 NEXT  
45 LOCATE0,1:PRINT"* COURSE AUX FONT$ *"  
;  
46 LOCATE0,2:PRINT"* sur Canon X-07 *"  
;  
47 GOSUB20  
48 '-----ENTREE DES PARAMETRES-----  
--  
50 CLS:PRINT"Fixez les dimensions du cadre du dessin :";  
51 INPUT"LARGEUR L":L  
52 INPUT"HAUTEUR H":H  
55 IFL>0ANDH<21GOT060  
56 GOSUB72:PRINT"L doit être compris entre 1 et 20 inclus";  
57 GOT067  
60 IFH>0ANDH<5GOT065  
61 GOSUB72:PRINT"H doit être compris entre 1 et 4 inclus";  
62 GOT067  
65 IFL*H<65GOT075  
66 GOSUB72:PRINT"Le produit de L par H ne peut excéder 64";  
67 GOSUB20:GOT050  
70 IFLNKEY$=="THENJ0ELSERETURN  
72 CLS:LOCATE3,0:PRINT"ATTENTION ?":RETURRN  
73 '---CHOIX DE LA ZONE DE "DIALOGUE"---  
--  
75 IFL<17THENQ=0ELSEQ=1  
76 IFQ=1ANDH=3THENQ=2  
77 XL=L*6-1:YL=H*8-1:N=L*H
```

Listing du programme Basic.



```
*MDDOCK
FONTC 128 )=[00,00,00,00,00,00,00,00,00,00]
FONTC 129 )=[00,00,00,00,00,00,00,00,00,00,00,00,00]
FONTC 130 )=[00,00,00,00,00,00,00,00,00,00,00,00,00,00,00]
FONTC 131 )=[00,00,E0,F8,FC,FC,FC,FC]
FONTC 132 )=[00,00,00,00,00,00,00,00,00,00,00,00,00,00,00]
FONTC 133 )=[00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00]
FONTC 134 )=[F8,70,3C,14,08,18,28,28]
FONTC 135 )=[00,FC,IC,28,14,08,28,18]
FONTC 136 )=[00,FC,FC,2C,18,08,18,04]
FONTC 137 )=[00,00,50,C0,00,00,00,00]
FONTC 138 )=[00,00,00,00,00,00,00,00,00]
FONTC 139 )=[28,28,18,08,00,00,18,18]
FONTC 140 )=[00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00]
FONTC 141 )=[00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00]
FONTC 142 )=[00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00]
FONTC 143 )=[00,00,00,00,00,00,04,18,28,00]
FONTC 144 )=[FC,10,10,28,C4,00,00,00,00,00]
FONTC 145 )=[FC,2C,00,00,CC,30,00,00,00,00]
FONTC 146 )=[E0,E0,20,70,88,04,00,00,00,00]
FONTC 147 )=[00,00,00,00,00,00,00,00,00,00]
```

Fig. 1. – Listing des FONT\$ calculées.

dans le coin inférieur droit, signalant le mode « effacement ». Le déplacement du spot efface alors les points rencontrés par ce dernier.

Pour revenir en mode « dessin », il suffit d'une nouvelle pression sur **F6**. Un nouveau bip est émis et l'étoile disparaît. ● Il existe deux vitesses de déplacement du spot. Au départ, c'est le mode « rapide » qui est sélectionné. Pour passer en mode « lent », pressez la **barre d'espacement**. Un bip est émis et les coordonnées du spot s'affichent en clignotant dans la zone réservée au « dialogue ».

Une nouvelle pression sur la barre autorise le retour en mode rapide. Les coordonnées ne sont plus affichées.

Pour une prise en compte plus rapide de toutes les commandes en mode lent, pressez la touche désirée lorsque les coordonnées « s'effacent » au cours du clignotement.

● Pour positionner le spot au point de coordonnées ($x < L * 6$, $y < H * 8$), pressez la touche **P**. $X = ?$ est alors affiché. Il faut entrer x et appuyer sur **RETURN**. Si $x \geq L * 6$, la valeur entrée n'est pas acceptée.

```
78 CONSOLE,,,0
79 '--NUMERO PREMIER CARACTERE GRAPHIQUE
--
80 CLS:INPUT"Voulez-vous fixer le numero du premier caractere graphique";D
81 IFLEFT$(D,1)<>"0"THENR=128:GOTO95
85 CLS:INPUT"Numero";R
86 IFR>127ANDR<160ORR>223ANDR<256GOTO90
87 GOSUB72:PRINT"Cette valeur R doit etre >127 et <160 ou >223 et <256";
88 GOSUB70:GOT085
90 S=-(192-R)*(R<160)-(256-R)*(R>223)
91 IFN<=SGOTO95
92 GOSUB72:PRINT" Numero trop eleve pour les dimensions du cadre";
93 GOSUB70:GOT085
95 N=L*X
96 '----OPTION SAUVEGARDE SUR CASSETTE---
100 CLS:INPUT"Voulez-vous archiver le dessin sur cassette";D
101 IFLEFT$(D,1)="0"THENE=1ELSEE=0:GOTO10
102 CLS:PRINT"Pensez a positionner la bande et a regler tonalite et volume a";
103 LOCATE0,3:PRINT"S/10 ; O.K.?";
104 GOSUB70
105 CLS:INPUT" Voulez-vous donner un titre a l'enregistrement";D
106 IFLEFT$(D,1)="N"THENCT="SANS TITRE":GOTO110
107 CLS:LINEINPUT" Quel est ce titre ?";
CT
108 '----OPTION SORTIE PARAMETRES---
110 CLS:PRINT" Voulez-vous editer les FONT$ des caracteres graphiques en";
111 LOCATE0,3:INPUT"notation decimale";D
112 IP=0:IFLEFT$(D,1)<>"0"THEN120
113 CLS:PRINT"L'imprimante est en service ? ; couleur selectionnee ? ";
114 LOCATE3,3:PRINT"Alors O.K. ?";
115 IP=1:GOSUB70
116 '*****DEBUT DU MODULE GRAPHIQUE*****
120 CLS:LINE(L*6,0)-(L*6,H*8):LINE-(0,H*8)
125 X=0:Y=0:P=1:F=1:Z=1
126 GOT0140
127 '-----VITESSE LENTE-----
130 A=STR$(X):A=RIGHT$(A,LEN(A)-1):GOSUB270
131 A=STR$(Y):A=RIGHT$(A,LEN(A)-1):GOSUB272
135 BEEP0,7:BEEP0,1
136 A="" :GOSUB270:GOSUB272
137 '-----VITESSE RAPIDE-----
140 T=STICK(0)+1:ONTGOSUB150,170,172,175
141 PRESET(X,Y):IFP=1THENPSET(X,Y)
142 IFZ=1THEN140ELSE130
145 '-----SCRUTATION DU CLAVIER-----
150 PSET(X,Y):BEEP0,2:BEEP0,1
151 PRESET(X,Y)
152 IFSTRIG(1)GOTO200
153 IFSTRIG(0)GOTO210
155 IFTKEY("D")GOTO215
156 IFTKEY("P")GOTO220
157 IFTKEY("C")GOTO240
158 IFTKEY("L")GOTO250
159 IFTKEY("M")GOTO300
160 IFTKEY("E")GOTO120
161 IFTKEY("A")GOTO350
162 RETURN
165 '--TRAITEMENT DES TOUCHES DU CURSEUR
--
170 Y=Y-1:IFY<0THENY=0
171 RETURN
172 Y=Y-1:X=X+1:IFY<0THENY=0
173 IFX>XLTHENX=XL
174 RETURN
175 X=X+1:IFX>XLTHENX=XL
176 RETURN
177 X=X+1:Y=Y+1:IFX>XLTHENX=XL
178 IFY>YLTHENY=YL
179 RETURN
180 Y=Y+1:IFY>YLTHENY=YL
181 RETURN
182 Y=Y+1:X=X-1:IFY>YLTHENY=YL
183 IFX<0THENX=0
184 RETURN
185 X=X-1:IFX<0THENX=0
186 RETURN
187 X=X-1:Y=Y-1:IFX<0THENX=0
188 IFY<0THENY=0
189 RETURN
195 '-----EFFACEMENT(F6)-----
200 P=1-P:BEEP10,4:BEEP0,1
201 IFP=1THENA=" " ELSEA="*"
202 GOSUB274
203 IFP=0THENX1=X:Y1=Y:RETURN
204 IFP=1THENX2=X:Y2=Y:RETURN
206 '-----CHANGEMENT DE VITESSE-----
210 Z=1-Z:BEEPS,4:BEEP0,1
211 IFZ=1THENA=" " :GOSUB270:GOSUB272
212 RETURN
214 '-----TRACE D'UNE DROITE-----
215 LINE(X1,Y1)-(X2,Y2):RETURN
217 '-----POSITIONNEMENT DU SPOT-----
220 PSET(X,Y):A="X=?":GOSUB270
221 A="" :GOSUB272:GOSUB280:X=VAL(C)
222 A="" :GOSUB272:IFX>XLGOTO221
225 A="Y=?":GOSUB270
226 A="" :GOSUB272:GOSUB280:Y=VAL(C)
227 A="" :GOSUB272:IFY>YLGOTO226
229 A="" :GOSUB270:GOSUB272
230 IFF=1THENX1=X:Y1=Y:F=0:RETURN
231 IFF=0THENX2=X:Y2=Y:F=1:RETURN
235 '-----TRACE D'UN CERCLE-----
240 A="R=?":GOSUB270
241 A="" :GOSUB272:GOSUB280:W=VAL(C)
242 IFW<0ORW>127THENA=" " :GOSUB272:GOT0241
243 CIRCLE(X,Y),W
244 A="" :GOSUB270:GOSUB272
245 RETURN
247 '-----AFFICHAGE D'UN CHR$-----
250 A="X=?":GOSUB270
251 A="" :GOSUB272:GOSUB280:U=VAL(C)
252 A="" :GOSUB272:IFU<0ORU>LGOTO251
255 A="Y=?":GOSUB270
256 A="" :GOSUB272:GOSUB280:J=VAL(C)
257 A="" :GOSUB272:IFU<0ORU>HGOTO256
259 A="C=?":GOSUB270
260 A="" :GOSUB272:GOSUB280:W=VAL(C)
261 A="" :GOSUB272:IFW<320RW>255GOTO260
0
262 A="" :GOSUB270
263 LOCATEU,U:PRINTCHR$(VAL(C));
```

Listing du programme (suite).

```

264 RETURN
267 ,-----DIALOGUE-----
270 IFQ=0THENLOCATE17,0ELSELOCATES,3
271 GOT0275
272 IFQ=0THENLOCATE17,1ELSELOCATE10,3
273 GOT0275
274 LOCATE17,3
275 PRINTA;
276 IFQ=2ANDLEFT$(A,1)==" ANDZ=1THENLINE
[0,24)-(6*L,24)
277 RETURN
278 '-----EQUIVALENT DE INPUT-----
280 J=STRIG(0):C=""
281 D=INKEY$:IFD==" THEN281
282 IFD=CHR$(13)THENRETURN
283 PRINTD;:C=C+D:GOT0281
285 ,
286 '*****FIN DU MODULE GRAPHIQUE*****
***  

287 ,
300 PSET(X,Y):GOSUB400
301 CLS:INPUT "Voulez-vous revoir le dessin ";D
302 IFLEFT$(D,1)="0"THENGOSUB500
310 CLS:PRINT "Voulez-vous editer les FONT$ des carac-teres graphiques en";
311 LOCATE0,3:INPUT"notation HEXA";D
312 IFLEFT$(D,1)<>"0"THEN330
320 CLS:INPUT" SORTIE EN CONTINU";D
322 IFLEFT$(D,1)="0"THENW=1ELSEW=0
325 GOSUB360
330 CLS:INPUT" Voulez-vous reproduire le dessin sur papier";D
331 IFLEFT$(D,1)="0"THENGOSUB540
335 IFE=0GOTO340
336 CLS:INPUT"Voulez-vous verifier l'enregistrement";D
337 IFLEFT$(D,1)="0"THENCONSOLE@,1:GOSUB
600
340 CLS:INPUT" Voulez-vous initialiser les caracteresgraphiques";D
341 IFLEFT$(D,1)="0"THENCONSOLE@,1
350 CLS:INPUT"Voulez-vous faire un autre dessin";D
352 CONSOLE,,,1,1:IFLEFT$(D,1)="0"GOT050
355 CLS:END
356 '---SORTIE DES FONT$ EN HEXADECIMAL-----
360 LPRINTCT
365 S=R
370 FORI=0TOH-1:FORJ=0TOL-1
371 CLS:PRINT"FONT$(";S;")";FONT$(S)
372 LPRINT"FONT$(";S;")";FONT$(S)
373 IFW=0THENGOSUB70
375 S=S+1:IFS>159ANDS<224THENS=224
376 NEXTJ,I
377 RETURN
380 '--MODULE DE SAUVEGARDE SUR CASSETTE
400 IFE=0GOTO420
402 A="CAS":GOSUB270
403 A="OK?":GOSUB222
404 BEEP0,3:BEEP0,1
405 A="":GOSUB272
406 BEEP0,3:BEEP0,1
407 IFINKEY$==" THEN403
408 J=STRIG(0):GOSUB270
409 '----BLANC SONORE DE SEPARATION-----
410 MOTORON:BEEP0,60:BEEP0,1
411 INIT#1,"CASO:"
```

Listing du programme (suite).

```

415 PRINT#1,CT,L,H,N,R
420 A="NUM":GOSUB270
421 S=R:F=0:K=0:D=CHR$(34)
425 IFIP=1THENLPRINTCT:LPRINT
427 '-----DEBUT BALAYAGE ECRAN-----
430 FORY=0TOH-1:FORX=0TOL-1
431 K=K+1
432 '--AFFICHAGE DU NUMERO DU CARACTERE-
433 A=RIGHT$(STR$(S),3):GOSUB272
434 GOSUB460:IFE=1THENGOSUB440
435 NEXTX,Y:RETURN
437 '-----FIN BALAYAGE-----
440 IF(LEN(D)+LEN(C)>253)GOT0445
441 D=D+C
442 IFK<NTHENRETURNSEF=1
445 INIT#1,"CASO":
446 D=D+CHR$(34)
447 PRINT#1,F,D
450 D=CHR$(34)+C
451 IFK=NANDF=0THEN442ELSERETURN
455 '-----CALCUL DU FONT$(S)-----
460 C="":FORI=Y*8TOY*8+2
461 U=0:U=0
462 FORJ=X*6TOX*6+5
463 IFPOINT(J,I)THENU=U+2^(7-U)
464 U=U+1:NEXTJ
465 A=STR$(U)
466 C=C+RIGHT$(A,LEN(A)-1)+""
467 NEXTI
468 BEEP10,2:BEEP0,1
469 '-----MEMORISATION-----
470 A=LEFT$(C,LEN(C)-1)
471 FONT$(S)=A
473 '--IMPRESSION DES PARAMETRES DECIMAUX---
475 IFIP=1THENLPRINT"FONT$(";S;")";CHR$(34);A;CHR$(34)
476 S=S+1:IFS>159ANDS<224THENS=224
477 RETURN
490 '----AFFICHAGE D'UN DESSIN-----
500 CLS:S=R:FORY=0TOH-1
501 FORX=0TOL-1
502 LOCATEX,Y:PRINTCHR$(S);
503 S=S+1:IFS>159ANDS<224THENS=224
505 NEXTX,Y
506 GOSUB70:RETURN
520 '----DESSIN SUR IMPRIMANTE X-710-----
540 CLS:PRINT" L'imprimante est en service ? ; couleur selectionnee ? ";
541 LOCATE3,3:PRINT"Alors O.K. ?";
542 GOSUB70
545 '-----TRACE DU CADRE-----
550 XD=64*X:H=240-XD/2:XI=-XD
551 YD=36*L-1:YI=-YD
552 LPRINTCHR$(17):LPRINTCHR$(18)
553 LPRINT" M"+STR$(X)+" "+STR$(YI)
554 LPRINT" I"
560 LPRINT" J"+STR$(XD)+" , 0, 0, "+STR$(YD)+
", "+STR$(XI)+" , 0, 0, "+STR$(YI)
561 LPRINT" S"
562 YD=YD-1
565 GOSUB500
567 '----RECOPIE POINT PAR POINT-----
570 FORI=0TOYL
571 IF(IMOD2)=1GOT0585
```

Y = ? est alors affiché et, à nouveau, il faut entrer *y*, suivi de RETURN.

De la même façon une valeur, *y* $\geq H * 8$, n'est pas acceptée.

- Pour tracer une droite entre deux points A (*x*₁, *y*₁) et B (*x*₂, *y*₂), deux possibilités sont offertes :

– Tout d'abord, le tracé à « vue » où l'on ne connaît pas les coordonnées de A et B. Il faut amener le spot sur A, presser F6 (les coordonnées (*x*₁, *y*₁) sont alors mémorisées). On déplace ensuite le curseur en mode « effacement » jusqu'à B et on appuie à nouveau sur F6, puis sur la touche D.

– Ensuite, le tracé précis, pour lequel on utilise « l'option » P pour introduire successivement (*x*₁, *y*₁), puis (*x*₂, *y*₂). L'appui de D provoque le tracé de la droite désirée.

- Pour tracer un cercle de centre (*x*, *y*) et de rayon R, il est nécessaire de positionner d'abord le curseur au point (*x*, *y*), puis presser la touche C. « R = ? » apparaît.

Il faut alors introduire le rayon suivi de RETURN.

- Pour « incruster » un caractère ASCII (code ≥ 32), il faut presser la touche L.

« X = ? » est affiché.

On introduit alors la colonne X ($0 \leq X \leq L - 1$) suivie de RETURN.

« Y = ? » est ensuite affiché.

Il faut alors introduire la ligne Y ($0 \leq Y \leq H - 1$) suivie de RETURN.

Enfin, « C = ? » est affiché.

Le code ASCII du caractère doit être fourni, suivi d'un RETURN.

- Pour effacer tout l'écran, il faut presser la touche E.

- Pour changer de dessin ou arrêter, il suffit de presser la touche A.

- Une fois le dessin achevé, une pression de la touche M lance la mémorisation.

Mémorisation et archivage

Dès que M a été pressée, si l'option sauvegarde a été demandée initialement, le programme affiche un « CAS », suivi d'un « OK ? ».

Ceci pour rappeler aux étourdis de mettre le magnétophone en position « enregistrement ».

Une pression sur une touche quelconque lance l'exécution. Les « FONTS » sont calculées

```

522 XD=8*(I+1)
525 LPRINT"Q3":LPRINT"M"+STR$(XD)+",0"
580 FORJ=0TOXL
581 IFPOINT(J,I)THENLPRINT"P"ELSEPRINT
" "
582 NEXTJ:GOT0595
585 XD=8*I
586 LPRINT"Q1":LPRINT"M"+STR$(XD)+","+ST
R$(YD)
590 FORJ=XTO0STEP-1
591 IFPOINT(J,I)THENLPRINT"P"ELSEPRINT
" "
592 NEXTJ
595 NEXTI:RETURN
596 '----RELECTURE D'UN ENREGISTREMENT---
597
600 CLS:PRINT" Bande positionnée ? PLAY
? Volume et tonalité = 7/10 ?";
601 LOCATE1,3:PRINT" Alors allez-y ?";
602 GOSUB70
610 INIT#1,"CASI:"
612 INPUT#1,CT,L,H,N,R
613 CLS:PRINTCT,L;H;N;R
617 '----LECTURE TANT QUE F=1-----
620 S=R
621 INPUT#1,F,C
622 A="" :I=1
625 FORJ=1TOLEN(C)-1
626 D=MID$(C,J,1):IFD<>CHR$(44)THEN=A+D
:GOT0629
627 Q(I)=VAL(A):A=""
628 I=I+1:IFI>8THENGOSUB640
629 NEXTJ:Q(8)=VAL(A):GOSUB640
630 IFF=0GOT0621
631 'F=1 signale la fin d'enregistrement
632
635 GOSUB500:RETURN
637 '-----MEMORISATION-----
638
640 FONT$(S)="Q(1),Q(2),Q(3),Q(4),Q(5),Q
(6),Q(7),Q(8)"
642 CLS:PRINT"FONT$(";S;")=";
643 FORP=1TO7:PRINTQ(P);";":NEXT
644 PRINTQ(8);
650 S=S+1:IFS>159ANDS<224THENS=224
651 A="" :I=1:RETURN
652
653 '-----EXPLICATIONS-----
654
700 FORI=1TO18
702 READA:CLS:PRINTA;
704 GOSUB70:NEXT:RETURN
710 DATA" Avec un écran fixe, PRESSEZ sur
une touche QUELCONQUE pour continuer"
715 DATA" Pour dessiner, utiliser les 4 touches de déplacement du curseur"
720 DATA" Pour tracer une diagonale, appuyer sur les 2 touches appropriées"
725 DATA" Pour passer en MODE LENT, pressez la barre d'espacement"
730 DATA" Dans ce cas, les coordonnées X,Y du point s'affichent et cliquent"
735 DATA" Pour revenir au MODERAPIDE, pressez à nouveau la barre"
740 DATA" Pour EFFACER, pressez d'abord la touche défonction F6 située au centre"
745 DATA" Dans ce cas une * est affichée en bas, à droite de l'écran"
750 DATA" Pour revenir en MODE ECRITURE, pressez à nouveau la touche F6"
755 DATA" Pour positionner un point, pressez P, et introduisez ses coordonnées"
760 DATA" Pour tracer une droite entre 2 points A et B, pressez D...
765 DATA"...après avoir introduit les co

```

ordonnées de A et B par l'option P"

770 DATA" Pour tracer un CERCLE, pressez C et introduisez le rayon"

775 DATA" Pour inscrire un caractère, pressez L et introduisez .. "

780 DATA... la position (X,Y) et le code ASCII du caractère"

785 DATA" Pour effacer TOUT l'écran, pressez E"

790 DATA" Pour interrompre, pressez A"

795 DATA" Une fois le dessin achevé, pressez M pour lancer la mémo-risation"

Listing du programme (suite et fin).

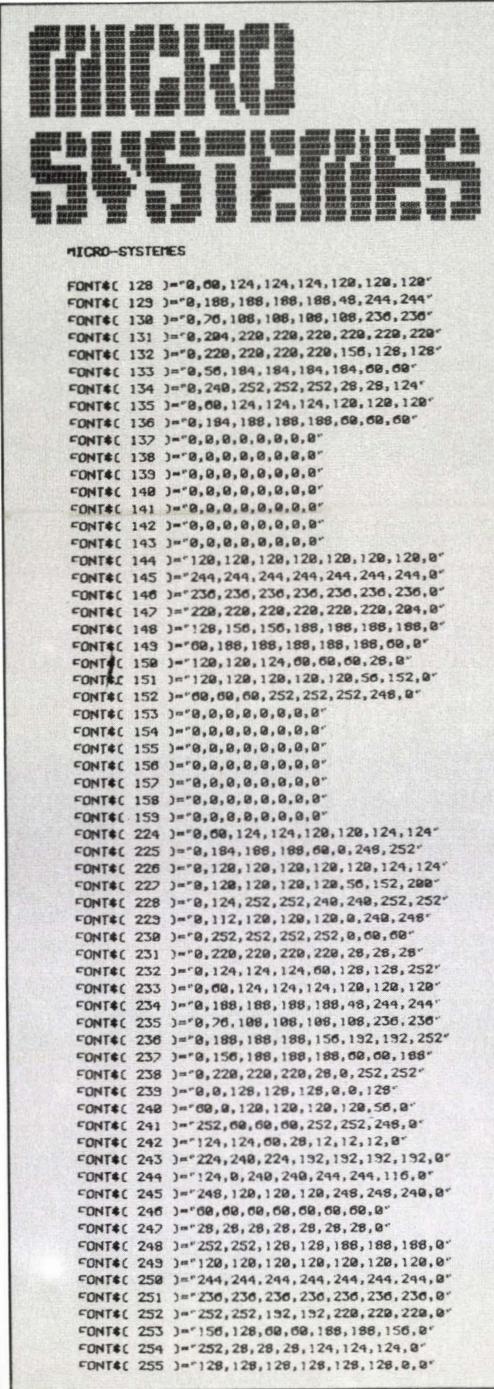


Fig. 2. - Le Logo de Micro-Systèmes tel qu'il est imprimé. On constate que les points ne sont pas carrés comme sur l'écran du X 07.

une à une et éditées éventuellement sur l'imprimante. La sauvegarde s'effectue par « paquets » sur la cassette.

Chaque « FONT\$ » mémorisée entraîne l'affichage de son numéro sur l'écran. Un petit bip est émis.

Les questions auxquelles il a fallu répondre

- Comment créer un INPUT sans « dévastation » de l'écran ?

Le dialogue au cours des « options » P, C, L nécessite l'équivalent de la fonction INPUT classique. Or, il est impératif d'éviter tout déroulement dévastateur de l'écran. Le sous-programme des lignes 280 à 283 remplit cet office.

- Comment effectuer une recopie point par point « rapide » ?

L'idéal serait de disposer d'un pavé « carré » programmé dans les caractères de l'imprimante X-710. Malheureusement, ce n'est pas le cas.

A défaut, on utilise le caractère CHR\$(127) qui imprime un X (fig. 2) sur le papier. Son symbole sur l'écran est le fameux point d'interrogation à l'envers (;) qui n'est accessible par aucune touche. Pour l'introduire dans les lignes 581 et 591, il faut procéder de la façon suivante :

Faire CLS, puis
?CHR\$(127) et RETURN

Le symbole ; est produit et il suffit de le déplacer sur l'écran pour l'introduire dans le texte des lignes 581 et 591 à l'endroit adéquat.

L'inconvénient (mineur) de l'utilisation de ce caractère est une légère déformation dans le sens de la hauteur de l'image imprimante par rapport au dessin présenté sur l'écran. Par contre, la recopie qui se fait dans les deux sens est relativement rapide.

- Comment compacter les enregistrements sur cassette ?

L'examen du module « sauvegarde » permet de voir que plusieurs « FONT\$ » successives sont « accrochées » les unes aux autres pour obtenir une variable D\$ d'une longueur aussi proche que possible de la longueur maximum autorisée pour une chaîne de caractères, soit 255 caractères.

Grâce à cette astuce, on limite le nombre d'enregistrements et on diminue considérablement le temps de sauvegarde (où, à l'inverse, de lecture d'un enregistrement). ■

Tableau des principales variables utilisées

Variables	Valeurs possibles	Rôle
A\$	quelconque	Affichage dans zone de dialogue
CT\$	quelconque	Titre de l'enregistrement
E	0 1	Pas d'enregistrement sur cassette Enregistrement sur cassette
F	0 1	Vaut zéro pour tout enregistrement d'une chaîne de FONT\$, sauf la dernière Signale la fin de l'enregistrement
H	1 ≤ H ≤ 4	Hauteur du cadre (nombre de lignes)
IP	0 1	Pas de sortie des FONT\$ sur imprimante Sortie sur imprimante
L	1 ≤ L ≤ 20	Largeur du cadre (nombre de colonnes)
N	1 à 64	Surface du dessin (N = H * L)
P	0 1	Mode effacement Mode dessin
Q	02 1 2	Localisation de la zone de dialogue : Utilisation des colonnes 17, 18, 19 Utilisation de la ligne 3 Restauraton du trait du cadre lorsque Q = 1 et H = 3
R	128 à 159 224 à 255	Numéro du premier caractère graphique qui sera utilisé en mémorisation
S	idem R	Numéro du caractère courant en phase de mémorisation
T	0 à 8	Etat des touches de déplacement du curseur
X, Y	0 ≤ X ≤ L 0 ≤ Y ≤ YL	Coordonnées du spot
XL	6 à 120	Limite de x
YL	8 à 32	Limite de y
Z	1	Vitesse rapide Vitesse lente

Tableau des variables.

Particularités du programme

Le listing (fig. 3) est commenté et permet de repérer facilement la structure du programme :

Lignes 20 à 115 : module d'initialisation.

Lignes 120 à 283 : module graphique. C'est le « cœur » du programme.

Lignes 300 à 355 : traitements du dessin.

Lignes 400 à 477 : module de sauvegarde sur cassette et de mémorisation des FONT\$.

Lignes 500 à 506 : module d'affichage.

Lignes 540 à 595 : module de recopie point par point sur l'imprimante X-710.

Lignes 600 à 651 : module de relecture d'un enregistrement. Ce module est à recopier dans tout programme qui utilise un stockage de dessins sur cassette.

Lignes 700 à 795 : explications des fonctions offertes à l'utilisateur.

SUPER VEGAS MONTE

EXCEPTIONNEL

UNITE CENTRALE:

- microprocesseur 6809
- 64 Ko de mémoire R.A.M.
- 1 lecteur de disquettes double face (320Ko) possibilité d'extension à 4 lecteurs
- interfaces parallèles pour imprimante (type "Centronics") et manettes
- 2 E/S série RS 232C
- horloge temps réel avec batterie de sauvegarde
- sorties vidéo monochrome et couleur (RGB)

CLAVIER:

- clavier ergonomique 101 touches (détachable)
- pavés machine à écrire (Azerty), numérique et de fonctions
- adaptation parfaite au traitement de textes

LOGICIEL DE BASE:

Flex et S. Basic

OPTIONS:

- carte graphique THR 512 x 512 } 8 couleurs
- carte graphique HR 256 x 256 } 2 plans
- boîtier SS30 pour connexions des cartes :
 - interface S.A.S.I. (SCSI)
 - interface IEEE 488
 - digitalisation d'images
 - programmeur d'EPROMS
 - convertisseurs AD et DA
 - synthèse vocale
 - ANTIPOE
 - etc..



LANGAGES ET LOGICIELS :

- langage C, PL 9, Pascal, Forth, assembleurs
- nombreux outils de développement et dessin assisté.

Monté en coffret noir métal, connecteurs très haute fiabilité (type militaire). Prix:

14 950 F TTC

Offre valable dans la limite des stocks disponibles

Vegas

Circuit imprimé + 2 Eproms
+ 1 disque Flex + dossier de montage _ 1 200 F

Clavier

Azerty 101 touches matricé X,Y _____ 590 F
Codeur en kit interface // _____ 350 F
Coffret _____ 290 F



15, Quai Jules Guesde 94400 Vitry - Tél. (1) 681.88.37