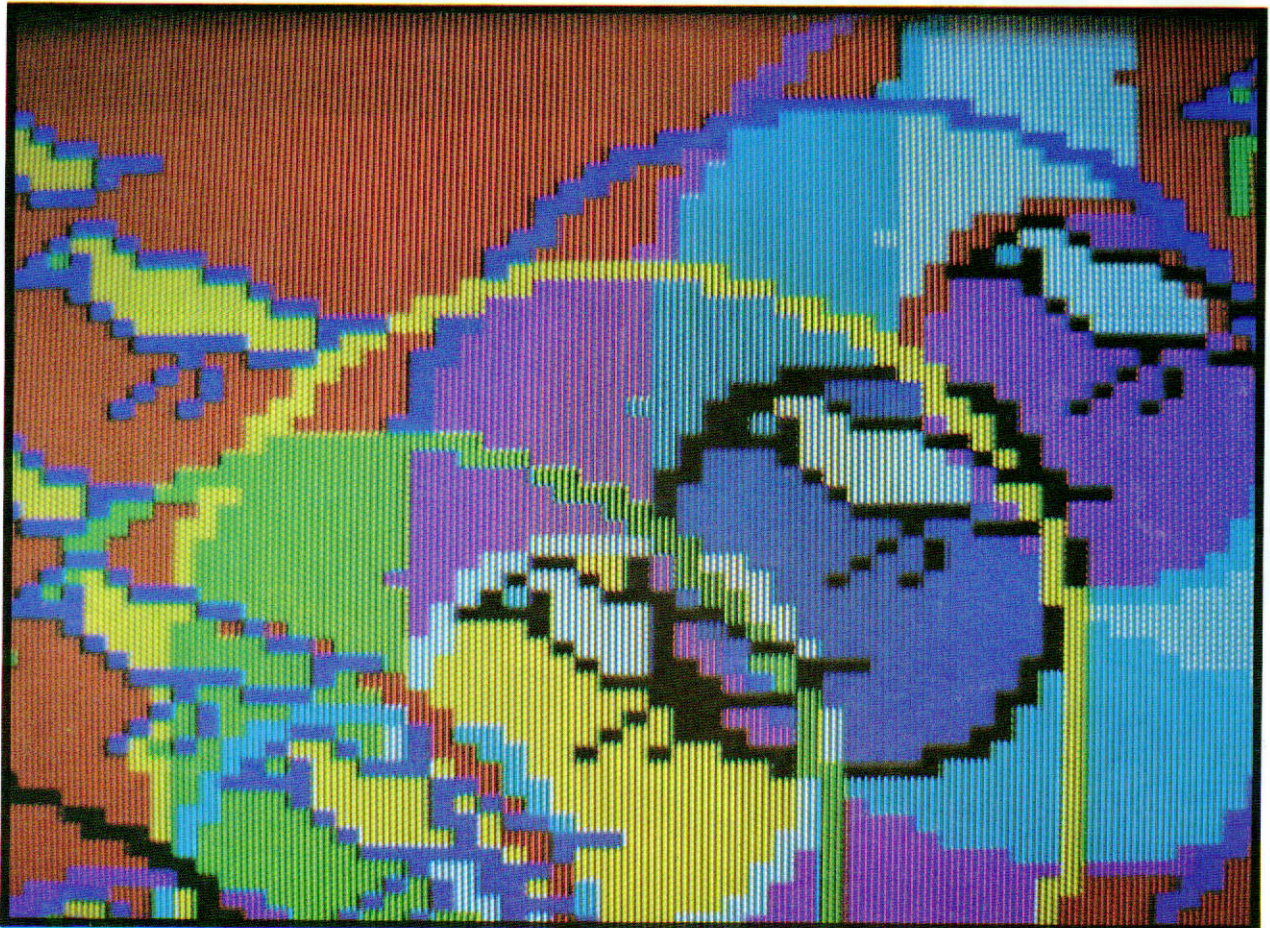


goupil **revue**



Les utilitaires du FLEX ■ Les ruses de GOUPIL ■ Gestion des fichiers séquentiels ■ Mission Delta
■ Sleuth ■ Le SBASIC de plus en plus efficace ■ Programmez vos fiches de paye ■ Encore des
courbes mathématiques ■ LSD FOX ■



l'ordinateur pour tous

EN REGION PARISIENNE GOUPIL C'EST MICROFRANCE

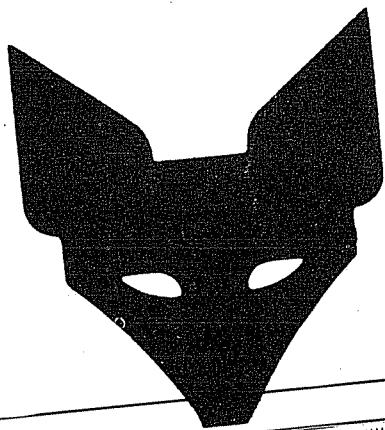
Microfrance est un point de rencontre pour 70 distributeurs. Microfrance assure maintenance et service après-vente. Microfrance a choisi des partenaires de grande compétence pour développer des applications performantes.

microfrance

Distributeur Officiel de Goupil
pour Paris et Ile de France

**OUVERTURE PROCHAINE
D'UN SECOND
CENTRE A PARIS
54 Bd. Davout 75020 Paris**





SOMMAIRE



Notre couverture : Premier contact avec graphigoupil, par Marie Langlois.
Cliché A.D.A.O. (Association pour le développement de l'Art par l'Ordinateur).

LOGICIELS DE BASE

Le SBASIC de plus en plus efficace	6
Les ruses de Goupil	8
Les utilitaires du FLEX	14
A propos de la récursivité	17
SLEUTH	18
LSD - FOX	24

LOGICIELS D'APPLICATION

Le jeu des étoiles	26
Programmez vos fiches de paye	30
Encore des courbes mathématiques	42
Musigoupil	44
Génécar	46

GOUPIL Revue - CB Editions 24 rue Marc Séguin
75018 PARIS - Tél. 201.04.50

Trimestriel

goupil revue

N° 6
LE NUMERO 20 F

ABONNEMENT
France : 70 F

ADMINISTRATION

Directeur de la publication : Jean-Louis KARL

REDACTION

Rédacteur en chef : Julien SPIESS

Rédacteur en chef adjoint : Sabine GRANDADAM

Secrétaire de rédaction : Marie-Ange NOGUER

ABONNEZ-VOUS A



goupil revue

Chaque trimestre "les ruses
de Goupil" dans votre boîte à lettres

ga go go **goupil** revue



4 parutions 70 F
— le numéro F



EDITORIAL

*Avec son numéro de rentrée,
Goupil-Revue voit s'élargir son équipe rédactionnelle : c'est ainsi
que les "Ruses de Goupil" ont été confiées à Georges Gallerand,
un lecteur de la première heure, qui nous propose d'améliorer
l'affichage des messages d'erreurs ainsi que l'entrée de la date
dans le FLEX.*

*Nous souhaitons aussi la bienvenue dans notre équipe à
David Mery, jeune responsable d'association, qui nous détaille ici
les commandes du SLEUTH, ainsi qu'à Marc Abramson qui nous
propose, depuis sa Provence natale, un programme générant des
caractères et un gestionnaire de fiches de paye.*

*Nous retrouvons avec plaisir Yves Martin et ses utilitaires du
FLEX. Jack-Michel Cornil nous présente les avantages de la
nouvelle version du S BASIC et Didier Cugy nous parle de son
très mystérieux "L.S.D. FOX". Notre ami Jean Saquet,
mathématicien de haut vol, nous invite à tracer de nouvelles
courbes, tandis que nous republions, mais dans le bon ordre et
avec toutes nos excuses, son précédent article sur la récursivité.*

*Avec Gérard Betouche, notre Goupil redevient musicien
et s'il nous reste un peu de temps pour jouer,
Bernard Beville nous convie à son "Jeu des Etoiles"...
Un sommaire qui, nous l'espérons, satisfera chacun.*

*Notre couverture donne déjà un avant-goût de ce que sera
l'article sur le graphisme dans notre numéro 7, qui comptera,
outre nos traditionnelles rubriques, un maximum d'applications
pratiques, ainsi que vous nous le demandez dans vos lettres ou
dans vos appels téléphoniques. Faites-nous toujours part de vos
réalisations, logicielles ou matérielles, de vos réflexions et
n'hésitez pas à nous envoyer vos listings. Notre prochain numéro
devant paraître en janvier 84, nous vous souhaitons d'ores et
déjà de passer de bonnes fêtes de fin d'année, en vous
rappelant, une fois encore, que cette revue est la vôtre.*

Julien Spiess

LE SBASIC DE PLUS EN PLUS EFFICACE

Le SBASIC a fait peau neuve et la dernière cuvée ou si vous préférez la dernière release, tout en restant compatible avec la précédente, constitue un progrès non négligeable dont ne pourront que se réjouir tous ceux qui développent des programmes en SBASIC. Voici en quelques lignes les principales améliorations.



Le nouvel éditeur de ligne

Le SBASIC dispose maintenant d'un véritable éditeur de ligne VIDEOTEX, ce qui est quand même bien plus agréable que celui de la première version où les caractères accentués apparaissaient, lors de l'édition, sous forme de la triple séquence permettant de les coder et qui obli-

geait à une gymnastique incroyable pour chaque remplacement d'un caractère normal par un caractère accentué et réciproquement.

Mais malgré son importance on peut qualifier la précédente amélioration de mineure quand on regarde les nouvelles fonctions qui sont proposées.

La touche [Ctrl-R]

- Lors de l'édition d'une ligne,

elle permet de se déplacer directement en début ou en fin de la ligne en cours. Si le curseur est en fin de ligne, il retourne au début, dans les autres cas il se positionne derrière le dernier caractère de la ligne.

- En dehors du mode édition, lorsque SBASIC affiche PRET sur l'écran, cette même touche permet de retrouver le dernier contenu du buffer d'édition.



C'est certainement cette dernière utilisation qui remplira le plus de joie les habitués du SBASIC. En effet, qui n'a jamais pesté lorsque, l'analyseur syntaxique rejetant dédaigneusement le fruit d'une frappe peut-être laborieuse sous prétexte d'un mauvais balancement des parenthèses, il se voyait contraint de réentrer toute la ligne qu'il venait de concocter patiemment, en surveillant d'un œil méfiant ces maudites parenthèses. Il suffit maintenant de taper [Ctrl-R] pour récupérer la ligne et corriger l'erreur précédemment fatale.

Attention : Il faut taper directement [Ctrl-R], toute autre touche vidant le buffer. Evitez donc les retours chariots rageurs après l'entrée d'une ligne ou l'apparition d'un message d'erreur.

Cette nouvelle fonction rend aussi plus aisée l'utilisation sous SBASIC des commandes du FLEX, et l'écriture d'instructions exécutées en mode direct. [Ctrl-R] permet aussi de les récupérer et éventuellement de les modifier pour une nouvelle exécution.

La touche [TAB]

Cette touche, dont l'utilisation déplace le curseur de huit caractères vers la droite, permet de balayer beaucoup plus rapidement la ligne, pour se positionner sur les caractères à modifier.

La touche [Ctrl-X]

Cette touche permet de sortir du mode édition en perdant la fin du buffer, ce qui est quand même bien plus rapide que l'utilisation répétée de la touche [EFF]

En contrepartie des améliorations précédemment décrites, il faut signaler la perte de l'utilisation en mode direct des touches correspondant à certains caractères de contrôle tels que CHR\$(12) pour effacer l'écran ou [Ctrl-Q] pour récupérer le curseur, mais on peut toujours taper PRINT CHR\$(12), et l'utilisation de [Ctrl-R] permet de retrouver le curseur.

Il faut aussi regretter que cet éditeur soit une particularité SBASIC et ne fasse partie du FLEX, ce qui est assez vexant lorsque l'on utilise directement le système d'exploitation.

En résumé, même si ce n'est ni VOLTAIRE, ni AUTOGRAPHE, cet éditeur de ligne est quand même très confortable en comparaison de ce que l'on peut trouver dans les autres BASICs.

INPUT et INPUT LINE

Lorsqu'un programme fait appel à l'une de ces fonctions l'utilisateur du programme disposera de toutes les fonctions du nouvel éditeur de ligne précédemment décrit pour entrer ses données, ce qui lui facilite la vie tout en évitant au programmeur la gestion fastidieuse et "octivore" d'un tel éditeur.

Gestion des fichiers

Il est maintenant possible d'ouvrir un fichier, de le lire, d'y écrire en mode direct. La sortie en erreur, l'arrêt sur un STOP ne ferment plus systématiquement tous les fichiers. Seuls END et l'introduction de nouvelles lignes de programmes forcent la fermeture de tous les fichiers.

C'est une modification qui permet aussi de gagner beaucoup de temps lors de la mise au point des programmes SBASIC. Il devient en effet possible de tester en mode direct toutes les routines mettant en jeu des lectures et des écritures fichier

Enfin signalons la nouvelle interprétation par SBASIC de l'instruction CLOSE. Elle ferme tous les fichiers référencés qui sont encore ouverts. Pour ceux qui sont déjà fermés ou n'ont pu être ouverts aucune erreur ne se produit. Cela facilite énormément l'écriture des programmes de gestion des erreurs d'entrée/sortie sur disque

CONT

L'instruction CONT permet de reprendre l'exécution d'un programme sur l'instruction suivant immédiatement le STOP qui a provoqué son arrêt. Cela permet donc de positionner dans le programme des points d'arrêts où l'on peut tester certaines variables avant de continuer, même si l'on travaille sur des programmes mettant en œuvre des fichiers, qui ne sont pas fermés par l'instruction STOP.

Nouvelles instructions

STRING\$ permet de construire une chaîne répétant un même motif. La syntaxe est

A\$ = STR\$(exp1), [exp2])
où [exp1] est une expression chaîne de caractères et [exp2] une expression entière. Le contenu de A\$ sera obtenu en concaténant [exp2] fois le contenu de [exp1].

LTRIM\$ et RTRIM\$ remplacent respectivement RTRM\$ et LTRM\$. Ces derniers mots clés sont toujours compris par l'interpréteur mais ils sont automatiquement remplacés par leurs nouveaux homologues.

BREAKON et BREAKOFF permettent respectivement de valider et d'inhiber l'utilisation de la touche [Ctrl-C]. Cela évite le recours à l'instruction POKE qui peut toujours jouer de mauvais tours.

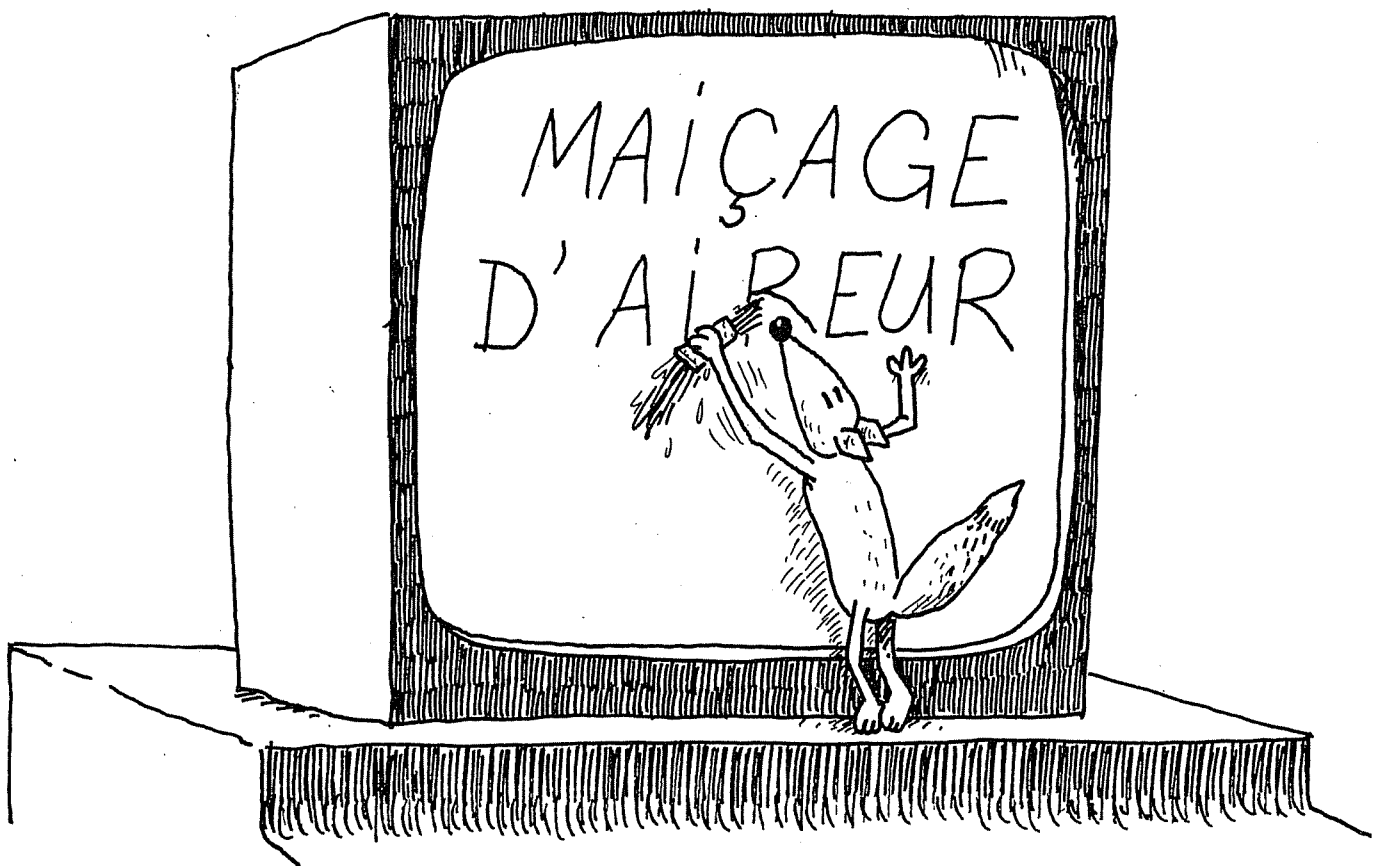
Dernière heure !

A l'heure où j'écris cet article (29.08.83) je peux révéler une nouvelle qui ravira tous les auteurs de logiciels écrits sous SBASIC : On commence à voir circuler, en test, dans les milieux autorisés une version du SBASIC fonctionnant sous CP/M 86. Je pense pouvoir affirmer sans risque qu'elle sera opérationnelle avant le SICOB 83. Souhaitons lui une carrière aussi brillante que celle de la version FLEX.

Jack-Michel Cornil

LES RUSES DE GOUPIL

Messages d'erreur



Le programme en SBASIC sert à écrire un nouveau fichier ERRORSYS.

Le SBASIC possède une fonction qui lui permet d'afficher textuellement les messages d'erreurs à la place du laconique "Erreur # ... à la ligne..." qui nous oblige à chercher dans les manuels pour avoir l'équivalent d'un numéro.

Malheureusement, le SBASIC va

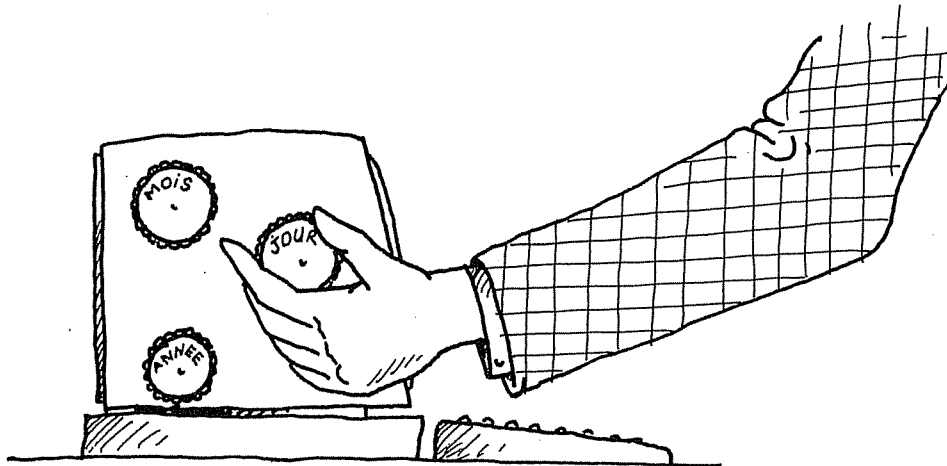
chercher ces messages dans le fichier ERRORSYS. qui ne contient que les erreurs du FLEX, celles qui surviennent le moins souvent en BASIC. Ce programme se charge donc de réécrire le fichier avec toutes les erreurs, du n° 1 au n° 128, telles qu'on les trouve dans le manuel du SBASIC.

De plus, si vous disposez de la commande FLEX ERREUR.CMD,

on peut appeler le fichier ainsi, sous SBASIC : "+ ERREUR ␣ <Numéro>" ce qui donnera l'équivalent "compréhensible" du numéro d'erreur. N'oubliez surtout pas l'espace (␣) pour assurer la conformité syntaxique avec le FLEX. Nous espérons que ces deux "Ruses" vous seront utiles.



Entrée de la date dans le FLEX



A l'initialisation, le Goupil 3 ne demande pas la date mais affiche simplement : "Date courante...". Il faut alors la rentrer manuellement par l'intermédiaire de l'utilitaire DATE.CMD du FLEX. C'est fastidieux et on ne peut la faire entrer s'il y a un fichier STARTUP.TXT qui branche directement sous SBASIC par exemple. Ce programme en Assembleur demande donc la date, la teste (mois de 29,30 ou 31 jours, pas plus de 12 mois dans l'année, etc.) et la range en mémoire à l'endroit où FLEX 9 et SBASIC iront normalement la chercher. Si la date entrée ne correspond pas, le programme affiche : "Date fantaisiste" et redemande la date. La date est demandée sous la forme JJ, MM, AA. Les séparateurs admis sont la virgule, le point, le slash (/), le signe moins et

l'espace (on peut les modifier aux lignes 47 à 55).

Le programme tourne sur 6809, mais les codes sont écrits en langage 6800 à l'exception de l'instruction MUL propre au 6809 (ligne 102) qui multiplie les registres A et B entre eux et range le résultat dans le registre 16 bits D, composé des registres A et B ensemble (A : poids forts, B : poids faibles). De toute façon, le résultat ne doit pas dépasser 8 bits (99 pour être précis) et est donc contenu en totalité par B. Cette instruction peut donc être remplacée aisément par un petit sous-programme pour tourner sur 6800.

Ce programme utilise le tampon de ligne de Flex 9, c'est-à-dire que si on l'entre dans un fichier STARTUP, les commandes éventuelles situées à droite de son nom ne

seront pas prises en compte par le FLEX. Par exemple, il ne faut pas mettre dans STARTUP. "INPUT DAT : SBASIC" mais "EXEC STARTUP2" et créer un fichier STARTUP2.TXT contenant :

```
"INPUT DAT -
SBASIC" pour ne pas se retrouver sous FLEX après l'entrée de la date.
```

Attention : ce programme doit être assemblé avec l'extension CMD pour tourner à l'appel de son nom sans autre manipulation. Il doit donc être appelé comme n'importe quelle commande du FLEX.

Ce programme ne se substitue pas à l'utilitaire DATE-CMD du FLEX dont le but est différent. Ne lui donnez donc pas le même nom. INPUTDAT, comme dans l'exemple, correspond d'ailleurs bien à sa fonction.

Georges Gallerand



1
2
3
4
5
6
9
10
11
12



```
*** PROGRAMME D'ENTREE DE LA DATE DANS LE FLEX ***
*****
```

```
CD03 FLEX EQU $CD03
CD1B INBUFF EQU $CD1B
CD1E PSTRING EQU $CD1E
```

```

13          CD27  NXTCH  EQU    $CD27
14          CD24  PCRLF  EQU    $CD24
15          CC0F  JOUR2  EQU    $CC0F
16          CC0E  MOIS2  EQU    $CC0E
17          CC10  ANNEE2  EQU    $CC10
18
19          0008  JOUR    EQU    $08
20          0009  MOIS    EQU    $09
21          000A  ANNEE   EQU    $0A
22          0003  SEPRE   EQU    $03
23          0004  AUTRE   EQU    $04
24          0005  TRANSF  EQU    $05
25          0007  TYPE    EQU    $07
26
27          *****
**
28
29          *** Debut du programme ***
30
31          *****
**
32
33          C100          ORG    $C100
34
35          C100 5F          DEBUT  CLR    CLR
36          C101 0F          07          CLR    TYPE
37          C103 0F          06          CLR    TYPE-1
38          C105 0F          03          CLR    SEPRE
39          C107 0F          04          CLR    AUTRE
40          C109 8E          C1BC        LDX    $MESS
41          C10C BD          CD1E        JSR    PSTRING  Affichage de "Date (JJ/MM/AA
) ? ".
42
43          C10F BD          CD1B        JSR    INBUFF  Demande de la date.
44
45
46          C112 BD          CD27        TESTA  JSR    NXTCH  Test du caractere courant
47          C115 81          20          CMPA   $#20    Separateur admis (SPC ; , ;
; / ; -) ?
48          C117 27          13          BEQ    TRAITA
49          C119 81          2C          CMPA   $',
50          C11B 27          0F          BEQ    TRAITA
51          C11D 81          2F          CMPA   $'/
52          C11F 27          0B          BEQ    TRAITA
53          C121 81          2E          CMPA   $'.
54          C123 27          07          BEQ    TRAITA
55          C125 81          2D          CMPA   $'-
56          C127 27          03          BEQ    TRAITA
57          >C129 7E          C147        JMP    TESTB
58
59          C12C 34          02          TRAITA PSHA
60          >C12E BD          C15A        JSR    SAVE
61
62
63
64
65
66          C131 96          03          SUITE3  LDAA  SEPRE
67          C133 81          00          CMPA   $0
68          C135 26          07          BNE   SUITE  Separateur deja defini ?
69          C137 35          02          PULA  Non -> Definition
70          C139 97          03          STAA  SEPRE
71          >C13B 7E          C112        JMP    TESTA
72          C13E 35          02          SUITE  PULA
73          C140 91          03          CMPA  SEPRE  Separateur correct ?
74          C142 27          CE          BEQ   TESTA  Oui -> Caractere suivant.
75          >C144 7E          C1AD        JMP   ERREUR
76
77          *****
**
78
79          C147 81          0D          TESTB  CMPA  $#0D

```





Entrée de la date dans le FLEX



80	C149 27	27	BEQ	CRLF	Test: Retour Chariot ?
81	C14B 81	2F	CMPA	F	Test si les codes ASCII sont
82	C14D 23	5E	BLS	ERREUR	compris entre \$30 et \$39
83	C14F 81	3A	CMPA	A	
84	C151 24	5A	BHS	ERREUR	
85	C153 80	30	SUBA		
86	C155 97	05	STAA	TRANSF	
87	>C157 7E	C164	JMP	SUITE2	



88
89
90
91
92



93	C15A 9E	06	SAVE	LDX	TYPE-1
94					
95	C15C E7	08		STAB	B, X
96	C15E 0C	07		INC	TYPE
97	C160 5F			CLRB	
98	C161 0F	04		CLR	AUTRE
99	C163 39			RTS	



100	C164 0C	04	SUITE2	INC	AUTRE
101	C166 86	0A		LDAA	�A
102	C168 3D			MUL	
103	C169 81	00		CMPA	�
104	C16B 26	40		BNE	ERREUR
105	C16D DB	05		ADDB	TRANSF
106	>C16F 7E	C112		JMP	TESTA



107
108
109



110	>C172 BD	C15A	CRLF	JSR	SAVE
111	C175 96	07		LDAA	TYPE
112	C177 81	03		CMPA	
113	C179 26	32		BNE	ERREUR
114	C17B 96	09		LDAA	MOIS



entre 1 et 12.

115	C17D 81	00		CMPA	�
116	C17F 23	2C		BLS	ERREUR
117	C181 81	0D		CMPA	
118	C183 24	28		BHS	ERREUR
119	C185 96	08		LDAA	JOUR



120	C187 8E	C1E7		LDX	&#DATAS-1
121	C18A D6	09		LDAB	MOIS
122	C18C 3A			ABX	
123	C18D A1	84		CMPA	0, X
124	C18F 22	1C		BHI	ERREUR



repond au mois.

125
126
127
128



129	C191 81	00		CMPA	�
130	C193 27	18		BEQ	ERREUR
131	C195 96	0A		LDAA	ANNEE



0 et 99

132	C197 81	64		CMPA	d
133	C199 24	12		BHS	ERREUR



*** Transfert de la date correcte dans le FLEX ***



140					
141	C19B 96	08		LDAA	JOUR
142	C19D B7	CC0F		STAA	JOUR2
143	C1A0 96	09		LDAA	MOIS
144	C1A2 B7	CC0E		STAA	MOIS2
145	C1A5 96	0A		LDAA	ANNEE
146	C1A7 B7	CC10		STAA	ANNEE2



```

147
148 C1AA 7E CD03 JMP FLEX
149
150 *****
*
151
152 C1AD BD CD24 ERREUR JSR PCRLF
153 C1B0 8E C1D2 LDX $MESS2
154 C1B3 BD CD1E JSR PSTRING
155 C1B6 BD CD24 JSR PCRLF
156 C1B9 7E C100 JMP DEBUT
157
158 *****
**
159
160 C1BC 20 20 44 61 MESS FCC ' Date (JJ/MM/AA) ? '
C1C0 74 65 20 28
C1C4 4A 4A 2F 4D
C1C8 4D 2F 41 41
C1CC 29 20 3F 20
161 C1D0 00 04 FCB 0,4
162 C1D2 44 61 74 65 MESS2 FCC 'Date fantaisiste !!!'
C1D6 20 66 61 6E
C1DA 74 61 69 73
C1DE 69 73 74 65
C1E2 20 21 21 21
163 C1E6 00 04 FCB 0,4
164 C1E8 1F 1D 1F 1E DATAS FCB 31,29,31,30,31,30,31,31,30,31,30,31
C1EC 1F 1E 1F 1F
C1F0 1E 1F 1E 1F

165
166 *****
**
167
168
169 END DEBUT

0 erreur(s) dBetectBee(s)
TABLE DES SYMBOLES :
ANNEE 000A ANNEE2 CC10 AUTRE 0004 CRLF C172 DATAS C1E8
DEBUT C100 ERREUR C1AD FLEX CD03 INBUFF CD1B JOUR 0008
JOUR2 CCOF MESS C1BC MESS2 C1D2 MOIS 0009 MOIS2 CCOE
NXTCH CD27 PCRLF CD24 PSTRIN CD1E SAVE C15A SEPARE 0003
SUITE C13E SUITE2 C164 SUITE3 C131 TESTA C112 TESTB C147
FRANSE 0005 TYPE 0007

```



SORESE

SOCIÉTÉ D'ORGANISATION, D'ÉTUDES ET DE SERVICES

23, boulevard des Capucines. 75002 Paris

DEPARTEMENT INFORMATIQUE REPARTIE

- Développement de programmes spécifiques
- Adaptation de progiciels
- Formation utilisateurs

CENTRE DE DEVELOPPEMENT

8, rue Meriel - 93100 Montreuil
Tél. 859.36.00 — M. Stevens - Mme Mejean - M. Mangeol

LES UTILITAIRES DU FLEX

Après inventaire, nous avons répertorié une centaine d'utilitaires courants (99 pour être précis) fonctionnant sur le GOUPIL 3 ainsi que huit modules (dont l'extension est .SYS) nécessaires pour le fonctionnement correct de certaines commandes.

Dans la liste ci-dessous, les utilitaires précédés d'un ■ sont en réalité des modules systèmes, ceux qui sont précédés d'un • sont décrits dans la suite de cet article alors que le o indique qu'ils seront expliqués dans GOUPIL REVUE n° 7. Les autres utilitaires ont déjà été décrits dans les numéros précédents.

• **BIP**

Emet un son puis revient au FLEX.

• **CLS**

- Remet tous les attributs de l'écran à leur valeur par défaut
- règle la taille du curseur
- positionne le curseur en clignotement rapide
- met le SCROLLING implicitement ON
- sélectionne le format de l'écran en 24 x 80 avec la carte contrôleur 25 x 80
- met le curseur en haut à gauche de l'écran
- efface la ligne
- envoie un retour chariot suivi d'un passage à la ligne suivante
- passe au jeu alphanumérique standard
- allume le curseur
- efface l'écran.

• **COMPILE** b <Nom Fich> b <Ign deb> ,

<Ign Fin> , <nveau Ign deb> b + options

Utilisé uniquement sous SBASIC, il code le programme basic présent en mémoire de <Ign deb> (par défaut 1) à la <Ign Fin> (par défaut 32767) et met le résultat dans un fichier qui portera le nom Nom Fich et qui aura l'extension .BAC.

On peut éventuellement déplacer le programme pour que le nouveau numéro de ligne début ait la valeur <nveau Ign deb>.

Les options : C supprime les REM et les espaces. Le programme reste listable
S supprime la table des symboles. Le programme devient illisible
P rend le programme non listable. Il peut seulement être exécuté.

Le programme, même codé avec aucune option peut être chargé avec l'instruction BLOAD, deux à trois fois plus rapidement qu'avec l'instruction LOAD.

• **CSORT** b <Fich dep> b <Fich Fin> b

+ <Specif. dep> b + <Specif. Fin>

Permet de trier un fichier en indiquant les paramètres de tri sur la même ligne que la commande.

Le fichier de départ <Fich dep> est trié et rangé sur disque sous le nom <Fich Fin>.

Spécifications d'entrée <Specif. dep> :

- W = n Numéro du lecteur de travail
- L = n Longueur d'enregistrement fixe,
ou
E = 'X ou \$ NN caractère de fin d'enregistrement
ou
C = n Nombre de champs
F = 'X ou \$ NN caractère séparateur de champs
- T ou B
T : fichier de type texte
B : fichier de type binaire



- U si ce paramètre est présent, les minuscules sont considérées comme équivalentes aux majuscules.
- K conservation des enregistrements dont la clé est vide
- indications sur la clé de tri - Exemple : 1-5

Spécifications de sortie Specif. Fin :

- O = I ou K, ou T
I : l'enregistrement de sortie est le même que l'enregistrement de départ
K : l'enregistrement de sortie est la clé de tri
T : l'enregistrement de sortie provient d'un fichier pointeur
- E = X ou \$ NN ou N (nul) caractère de fin d'enregistrement
- T ou B fichier de type Texte ou de type Binaire
- M suppression des messages de travail
- Indications sur la forme de l'enregistrement de sortie exemple : 10-20, 1-9 ou 1-10 b b 'Numéro', 11-20

• **EXTRACT** b <nveau fich> b <fich cmd>

Permet de créer un nouveau fichier texte à partir d'autres fichiers.

Le <fich cmd> doit contenir la liste de tous les fichiers avec éventuellement la liste des lignes.

Exemple : si le fichier de commande contient :

) TOTO b 10-50 b 60-

) TATA

TEXTE

) TUTU b -50, 15

Il recopiera dans le nouveau fichier les lignes 10 à 50 et toutes les lignes à partir de 60 du fichier TOTO puis le fichier TATA, puis insérera une ligne où il y aura marqué TEXTE puis les cinquantes premières lignes du fichier TUTU et la ligne 15 du fichier TUTU.

• **MEMEND**

Sert à consulter la dernière position mémoire du FLEX

MEMEND b ? affichera la valeur courante

MEMEND test mémoire pour la déterminer et la mettre

MEMEND b <adr Hexa> modifie la valeur



• **MEMTU**

Cette commande est similaire à la commande MEMTEST, elle permet de tester la zone mémoire occupée par le FLEX 9 (de 48 à 60 k). Elle détruit FLEX9 qui devra être rechargé.

• **MERGE** **b** <Fich 1> **b** <Fich 2> **b** <Fich 3>... etc.
Permet de fusionner plusieurs fichiers. Lors de l'exécution de cette commande, et avant toute fusion, des questions sont posées à l'utilisateur. Ce sont exactement les mêmes que celles posées par la commande SORT.

• **OKI**

Permet de fixer les paramètres internes d'une imprimante de type OKI 82 ou 83.

Répondre aux questions

MODES :

Compressé

Normal

Semi étendu

Étendu

ESPACEMENT :

6 lignes - pouces

8 lignes - pouces

ESPACEMENT-CARACT. :

<A> ligne courte

 ligne longue

taper pour sortir de la commande.

• **PMERGE** **b** <Fich Param> **b** <Fich 1> **b** <Fich 2>... etc.

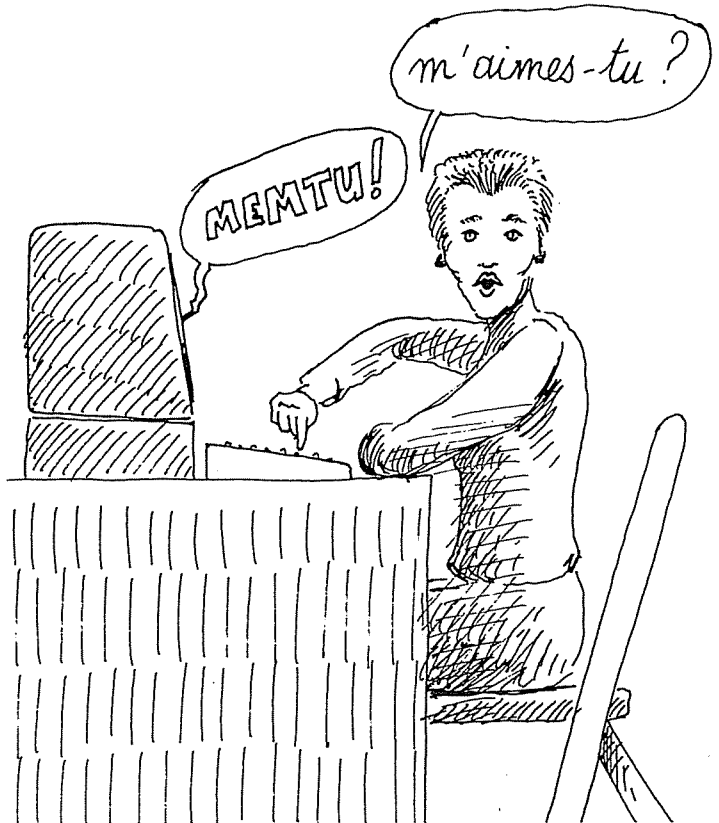
ou **PMERGE** **b** <Fich Param> **b** <Fich Sortie> **b** <Fich 1> **b** <Fich 2>... etc.

Permet de faire la fusion des <Fich 1>, <Fich 2>... etc. à partir des paramètres contenues dans le fichier paramètres <Fich Param>.

Il est possible de spécifier le fichier de sortie au cas où il serait différent de celui indiqué dans le fichier paramètres.

• **PRINT** **b** <Nom Fich> **b** + <nb>

Permet de lancer l'impression d'un fichier en "spooling" c'est-à-dire pendant que le GOUPIL exécute un travail diffé-



rent. <nb> est le nombre de répétitions.

• **QCHECK**

Sert à examiner et à modifier le contenu de la queue d'impression c'est-à-dire la liste des fichiers en attente d'impression.

Au lancement de cette commande, la liste de tous les fichiers est affichée suivie du message "Commande ?" (l'impression s'arrête).

Liste de tous les utilitaires et de tous les modules systèmes (dont l'extension est .SYS)

- | | | | | |
|---------------|--------------|---------------|---------------|--------------|
| ASN.CMD | APPEND.CMD | ASMB.CMD | BACKUP.CMD | BUILD.CMD |
| •BIP.CMD | COPY.CMD | CAT.CMD | •CONFDISK.CMD | •COMPILE.CMD |
| •CSORT.CMD | CHECK.CMD | CMPMEM.CMD | CONTIN.CMD | CONCAT.CMD |
| •CLS.CMD | CONVERGE.CMD | COPYR.CMD | •COPYNEW.CMD | DELETE.CMD |
| DATE.CMD | DIR.CMD | DUMP.CMD | DUP.CMD | DYNAMIC.CMD |
| ▪ERRORS.SYS | EXEC.CMD | EDIT.CMD | •EDITG2.CMD | •EDITG3.CMD |
| •ERREUR.CMD | ECHO.CMD | •EXTRACT.CMD | EXAMINE.CMD | FILES.CMD |
| FIND.CMD | FREE.CMD | FILTY.P.CMD | FILETEST.CMD | FLAW.CMD |
| ▪G3FLEX02.SYS | HECHO.CMD | •HELP.CMD | I.CMD | INSTALL.CMD |
| INTEG.CMD | JUMP.CMD | LINK.CMD | LIST.CMD | MEMTEST.CMD |
| •MEMTU.CMD | •MERGE.CMD | MAP.CMD | •MEMEND.CMD | MEMOVE.CMD |
| MEMDUMP.CMD | MEMFILL.CMD | NEWDISK.CMD | O.CMD | •OLOAD.CMD |
| •OKI.CMD | ▪PSERIE.SYS | •PRINT.SYS | P.CMD | PROT.CMD |
| •PRINT.CMD | •PSORT.CMD | •PMERGE.CMD | PDEL.CMD | •PATTERN.CMD |
| •QCHECK.CMD | QUICK.CMD | RENAME.CMD | •REFSUB.CMD | •RUN.CMD |
| REPLACE.CMD | RPT.CMD | REMSPC.CMD | RECOVER.CMD | RANDOM.CMD |
| RAWCOPY.CMD | REBUILD.CMD | ▪SBERRORS.SYS | ▪SSPRINT.SYS | SAVE.CMD |
| ▪SRTMRG.SYS | •SORT.CMD | •SBASIC.CMD | SAVE.LOW | SPLIT.CMD |
| SLEEP.CMD | ▪STARTUP.TXT | TEST.CMD | TTYSET.CMD | •TRIVAR.CMD |
| •TASVAR.CMD | TYPOS.CMD | UNDELETE.CMD | VERSION.CMD | VERIFY.CMD |
| •VIDEO23.CMD | VALIDATE.CMD | WORDS.CMD | WALK0.CMD | WALK1.CMD |
| XOUT.CMD | | •ZAP.CMD | | |



Les utilitaires du FLEX

Commandes possibles :

- Q imprime à nouveau la liste
relance l'impression
- R, =N, X répète X fois le fichier qui est à la position N
- D, =N supprime de la liste le fichier qui est à la position N
- T interrompt le travail d'impression en cours et lance le travail suivant sauf si c'est le dernier fichier de la liste qui est en cours d'impression.
- N, =M le fichier qui est à la position =M sera le prochain fichier à imprimer une fois le travail en cours terminé donc modifie l'ordre de la liste.
- S arrête l'impression dès que le fichier en cours est terminé d'être imprimé
- G permet de relancer l'impression interrompue par S
- K supprime le fichier en cours d'impression et supprime le nom de tous les fichiers en attente dans la queue d'impression.

• REFSUB

Utilisée uniquement sous SBASIC, cette commande permet d'afficher la liste de tous les sous-programmes définis par l'instruction SUB, établit cette liste à partir du programme présent en mémoire.

- **RUN** $\text{b} \langle \text{adr chg} \rangle, \langle \text{instruction} \rangle (1)$
ou
RUN / $\langle \text{adr chg} \rangle, \langle \text{instruction} \rangle (2)$

Permet de charger à partir d'une adresse différente de celle où l'utilitaire s'exécute habituellement. Cet utilitaire ne doit pas dépendre d'une position mémoire définie. Dans le cas de la syntaxe (1) l'utilitaire est exécuté dès le chargement effectué sinon il ne l'est pas (2).

• SBASIC $\text{b} \langle \text{Nom Fichier} \rangle$

Cette commande passe le contrôle à l'interpréteur SBASIC. Lorsque le nom d'un fichier est précisé, l'interpréteur est chargé ainsi que le programme indiqué puis ce programme est exécuté directement.

Par défaut l'extension du programme est .BAC. Il faut donc préciser .BAS s'il n'est pas codé.

• SORT $\text{b} \langle \text{Nom Fichier} \rangle$

Cet utilitaire permet de trier le fichier spécifié. Avant le tri des questions sont posées à l'utilisateur pour lui permettre de définir les caractéristiques exactes des fichiers. La description des questions est beaucoup trop longue pour figurer dans cette rubrique. Il est préférable de se reporter à

la documentation fournie avec la disquette pour y répondre correctement.

Cette commande peut constituer un "fichier paramètre" qui contiendra toutes les spécifications des fichiers et qui pourra être utilisé avec la commande PSORT.

• TASVAR

Utilisée uniquement sous SBASIC, cette commande permet de "faire le ménage" dans la table des variables du programme présent en mémoire.

Elle supprime tous les noms des variables qui ne sont plus utilisées.

• TRIVAR

Utilisée uniquement sous SBASIC, cette commande permet d'afficher la table des références appelée aussi tables des références croisées du programme présent en mémoire.

Cette table est la liste de toutes les variables utilisées dans le programme suivie chacune des numéros de ligne où la variable est utilisée.

VIDEO 23

Cette commande est destinée à résoudre les conflits entre les codes utilisés pour la gestion de l'écran sur GOUPIL 2 et GOUPIL 3. Elle permet d'utiliser sur GOUPIL 3 des programmes écrits en langage BASIC prévus pour être exécutés sur l'écran 24 x 80 du GOUPIL 2.

Cet utilitaire modifie des paramètres en mémoire puis revient directement au FLEX. Pour revenir au mode Ecran 25 x 80 du GOUPIL 3, il faut recharger le FLEX.

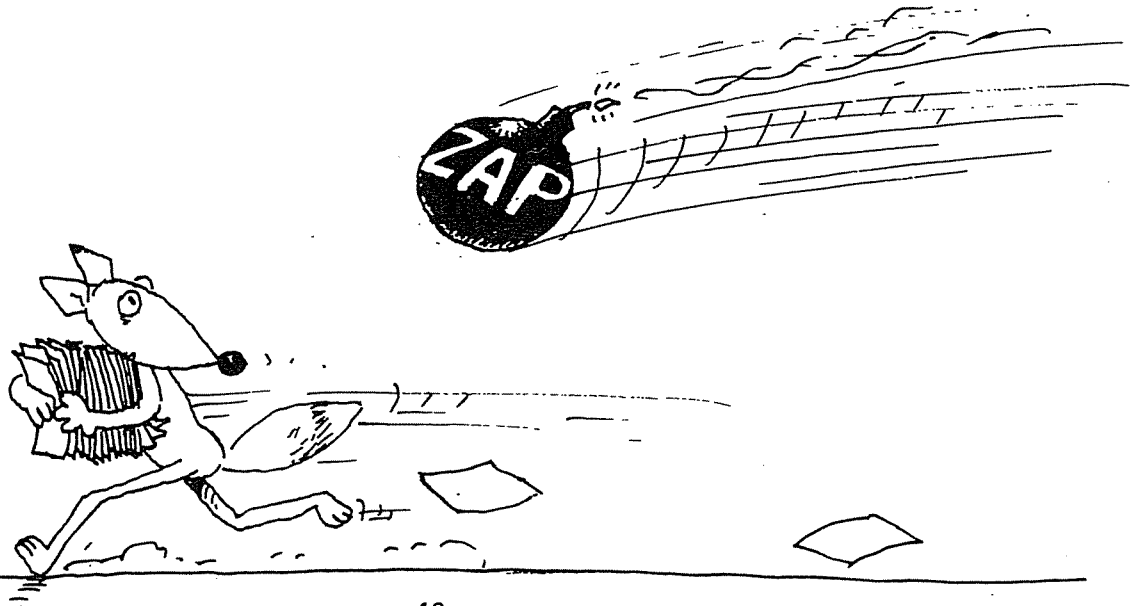
Cette commande se charge à l'adresse \$E0000 et occupe un demi K octet.

• ZAP

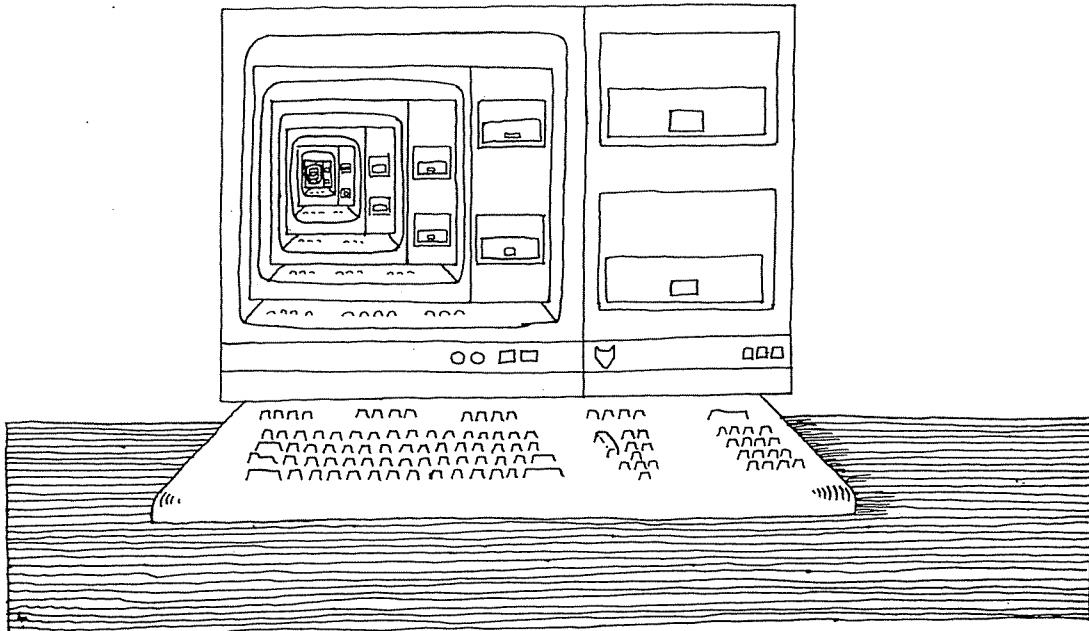
Attention : cette commande détruit les fichiers sans intervention. Avec la même syntaxe que PDEL ou CAT, cet utilitaire "delete" les fichiers indiqués en le signalant mais sans intervention de l'utilisateur.

Toutes les commandes qui existent sur le GOUPIL 2 se retrouvent sur le GOUPIL 3 avec souvent des améliorations sensibles de leur utilisation. Seuls les utilitaires dont l'extension est .LOW n'ont pas été transférés car il suffit de reloger ces utilitaires à l'aide de la commande RUN.

Yves Martin



A PROPOS DE LA RÉCURSIVITÉ



Dans le numéro précédent de Goupil-*Revue*, vous avez pu étudier un programme récursif en BASIC présenté dans deux versions : XBASIC et SBASIC. Ces programmes permettaient de tracer sur l'écran graphique du Goupil (2 ou 3), la courbe dite courbe de Hilbert.

Ce type de programme est toutefois assez difficile à comprendre, compte tenu de la récursivité qui n'est pas familière aux programmes BASIC.

Afin de mieux comprendre ce mécanisme, nous prendrons cette fois-ci un exemple plus simple : il s'agit d'entrer une phrase au clavier et de la réécrire à l'envers.

En BASIC standard (et non récursif), cela peut s'écrire :

```
10 INPUT "une phrase" ; A$
20 FOR I = LEN(A$) TO 1 STEP - 1
30 PRINT MID$(A$, I, 1) ;
40 NEXT I
50 PRINT
60 END
```

On aurait pu également lire caractère par caractère.

Les stocker dans un tableau et relire ce tableau à l'envers. Cette méthode peut s'avérer indispensable si on ne dispose pas de l'instruction MID\$ (cas de certains BASIC restreints) et surtout, c'est cette méthode que nous allons employer pour "récursiver" le problème, en SBASIC bien entendu :

```
10 PRINT "Ecrivez une phrase"
20 CALL LITCAR
30 PRINT : END
```

```
100 SUB LITCAR
110 LOCAL CAR$
120 CAR$ = INCH$(0) : PRINT CAR$ ;
130 IF CAR$( < > )CHR$(13) THEN CALL
LITCAR ELSE PRINT
140 PRINT CAR$ ;
150 RETURN
```

Ce programme mérite quelques explications.

Le programme principal (lignes 10 à 30) se contente d'afficher un message et d'appeler le sous-programme LITCAR.

Celui-ci comporte une variable locale CAR\$, dans laquelle on met le caractère lu par INCH\$(0). Si ce caractère n'est pas le retour chariot, on appelle de nouveau LITCAR avec par conséquent une nouvelle variable CAR\$, puisque le niveau de sous-programme a changé.

Ainsi tant que l'on tape des caractères autres que retour chariot, ceux-ci sont stockés dans les CAR\$ successifs (1 par niveau de sous-programme).

Exemple : si on tape Goupil, nous avons

- au niveau 1 : CAR\$ = "G"
- au niveau 2 : CAR\$ = "O"
- au niveau 3 : CAR\$ = "U"
- au niveau 4 : CAR\$ = "P"
- au niveau 5 : CAR\$ = "I"
- au niveau 6 : CAR\$ = "L"
- au niveau 7 : CAR\$ = retour chariot

Pour écrire le mot à l'envers, il suffit alors de remonter les appels de sous-programme en écrivant à chaque fois le CAR\$ du niveau dans lequel on se trouve. Ceci est réalisé par les lignes 140 et 150 : on écrit le dernier CAR\$ (retour

chariot) et on remonte d'un niveau de sous-programme grâce à RETURN. On retrouve alors le CAR\$ du niveau 6 (ici L) et on l'écrit et ainsi de suite jusqu'à G. Le RETURN permet alors de revenir au programme principal (ligne 30), où l'on passe à la ligne par PRINT. Ce programme peut bien sûr être écrit en PASCAL :

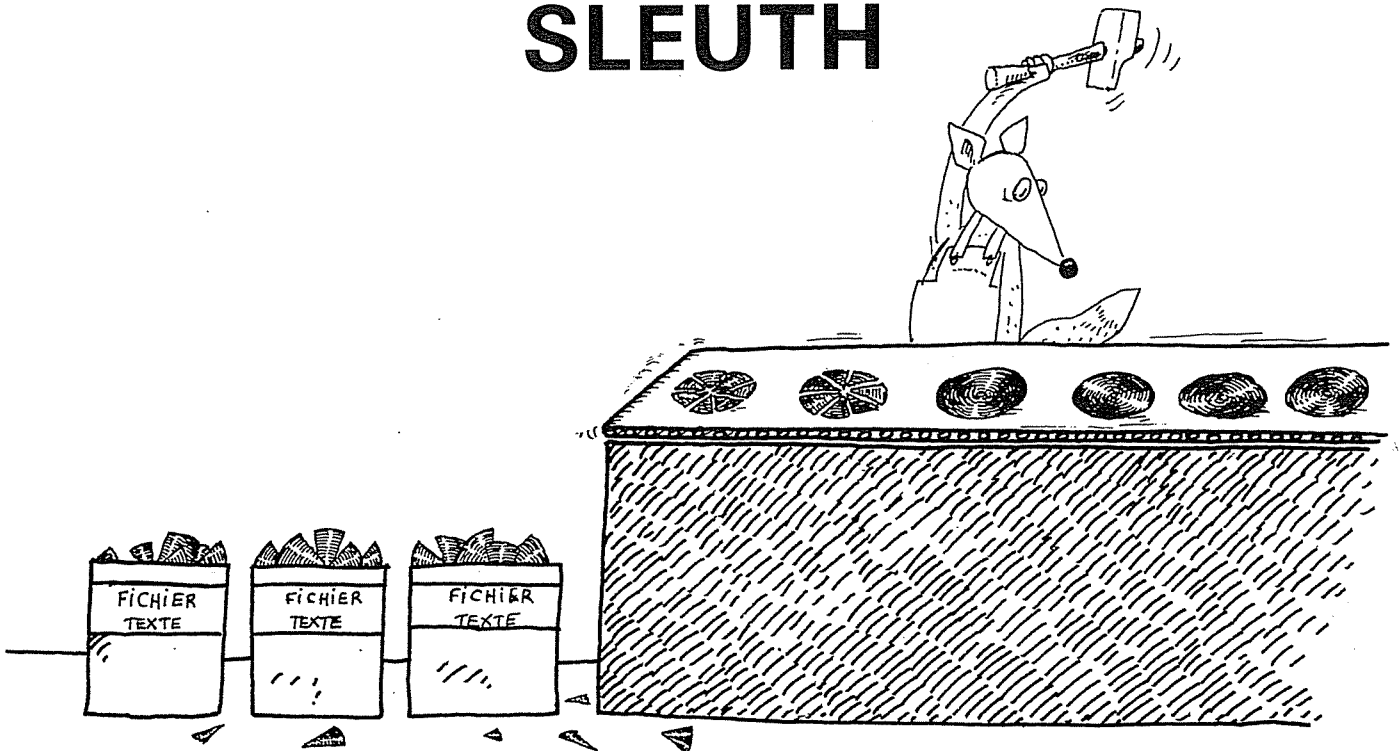
```
PROGRAM INVERSE ;
PROCEDURE LITCAR ;
VAR CAR : CHAR ;
BEGIN
  READ (CAR) ;
  IF NOT EOLN THEN LITCAR ;
  WRITE (CAR) ;
  END ;
BEGIN
  WRITELN ('Ecrivez une phrase') ;
  LITCAR ;
  WRITELN ;
  END.
```

On remarque la ressemblance entre les versions PASCAL et SBASIC, les seules différences notables étant l'ordre de la procédure et du programme principal et quelques PRINT (ou plutôt WRITE) en moins, compte tenu du fait que READ (CAR) a un écho sur l'écran, contrairement à INCH\$(0).

Bien entendu, le premier programme en BASIC standard était plus simple, les deux autres ayant seulement pour but de faire comprendre sur un exemple simple le mécanisme de la récursivité, laquelle n'est pas indispensable dans le cas présent.

Jean Saquet

SLEUTH



SLEUTH est un utilitaire FLEX qui permet de désassembler un programme binaire (code objet) se trouvant sur disque ou en mémoire centrale. Le code ainsi obtenu peut-être sauvegardé sur un disque en un fichier texte et être réassemblé avec la commande ASMB. On peut donc modifier le source avec la commande EDIT. L'emploi de cet utilitaire requiert une parfaite connaissance de l'assembleur. Le signe => précède un texte qui est affiché par SLEUTH après une commande.

Commandes de SUPER SLEUTH

- Commande <??> Affiche le menu des commandes.

Exemple:

??

Operational commands:

O / v -disassemble/view object code

F-exit to FLEX

Q / M -query/modify object code

T-fill address range with hex value

U-fill address range with hex value

W-write new object code file

Y-find hex string in object code

Address range commands:

A=F0B,C=F0C,H=F0B,I=code,J=code+ASCII,K=ignored

Mode change commands:

B-flip cross-assembler switch

E-flip separate-label switch

P-flip position independence switch

Z-change CPU mode

Miscellaneous commands

G-specify auxiliary input/output file

L-list control information

N-set new disassembly range



D-set offset load value
 R-restart program
 S-specify input file name
 X-set transfer address

OPERATIONAL COMMANDS

Commande <V> Visualise la zone de code objet spécifiée et son désassemblage sans label

Exemple:

?V

Enter T(terminal), N(no output) T

Start=C100 End=C215 Xfer=C102

separate-label switch off

CPU mode 6809

position-independence switch off

cross-assembler switch off

Input file name 0.ASN.COM

C100		ORG	\$C100
C100 00		FCB	\$00
C101 008D		FDB	\$008D
C103 6827		FDB	\$6827
C105 3000CD27		LEAX	?\$8E30,PCR3
C107 84DF		ANDA	£#DF
C108 8157		CMPA	£\$57
C10D 2700		BER	\$C11C

Commande <D> Désassemble la zone de code objet spécifiée en code source avec des labels

=> Output file name

On doit préciser le nom du fichier texte dans lequel va être sauvegardé le source ou <RETURN> pour ne pas créer de fichier sur disque

=> NAME

Nom du programme (directive NAM)

=> OPTIONS

Options d'assemblage (directive OPT)

Exemple:

?D

Output file name ASN.TXT

Enter T(terminal), N(no output) T

NAME ASN G3FLEX

OPTIONS PAG

Start=C100 End=C215 Xfer=C102

separate-label switch off

CPU mode 6809

position-independence switch off

cross-assembler switch off

Input file name 0.ASN.COM

Phase 1

Phase 2

		NAM	ASN G3FLEX
		OPT	PAG
008D	Z008D EQU		\$008D
6827	Z6827 EQU		\$6827
8E30	Z8E30 EQU		\$8E30
C100		ORG	\$C100
C100 00	ZC100 FCB		\$00



sleuth

Commande <F> Retour au FLEX

Exemple:

?FF

+++

Commande <Q> Dump 256 par 256 octets la zone spécifiée en précisant après chaque octet comment il sera désassemblé:

!: FDB £: FCC (: FCB): code *: code+ASCII +: ignoré

<S> Positionne le curseur sur l'octet 00

Se placer sur l'octet que l'on veut modifier à l'aide des flèches.

Si l'on est dans la partie hexadécimale, il faut taper des chiffres hexadécimaux. Si l'on est dans la partie ASCII, il faut taper des caractères ASCII.

<Q> Redonne la main

<P> Dump les 256 octets précédents

<N> Dump les 256 octets suivants

Exemple:

```

.0 .1 .2 .3 .4 .5 .6 .7 .8 .9 .A .B .C .D .E .F 0123456789ABCDEF
C100 00(00!8D!6B!27!30)8D)CD)27)84)0F)81)57)27)0D)81) ...k'0=M'-.W'.. Start
C110 53)26)17)8D)65)25)13)87)CC)00)20)07)8D)5C)25)0A) S&..e%.7L. .. %. =C100
C120 87)CC)0C)8D)4A)26)0F)7E)CD)03)8E)C8)40)C6)1A)E7) 7L..J&_-M..H&F.g End
C130 01)8D)CD)3F)20)F1)5F)F7)C1)00)8E)C1)8A)8D)CD)1E) .=M? q_wA..A:=M. =C215
C140 8E)C1)00)86)CC)08)87)C1)01)2A)05)8D)C1)A9)20)03) .A.6L.7A.*.=A) . A=!
C150 8D)CD)39)8E)C1)06)8D)CD)1E)86)CC)0C)87)C1)01)2A) =M9.AV=M.6L.7A.* C=£
C160 05)8D*C1*A9*20*C1*8E*C1*00*5F*8D*CD*39*20*B8*B6* .=A) A.A._=M9 86 H=(
C170 etc. I=)
C180 J=*
C190 K=+
C1A0 Scrn
C1B0 Quit
C1C0 Prev
C1D0 Next
C1E0 addr
C1F0 ?

```

Commande <M> Pour modifier des octets

=> Start=

Adresse à partir de laquelle il y a une modification

Pour changer la valeur de l'octet, tapez la nouvelle valeur, pour avancer se servir des flèches, pour sortir tapez <RETURN>

Exemple:

?M Start=0100

0100 00 25

0101 00

Commande <T> Permet de remplir une zone avec un chiffre hexadécimal

=> Start=

Adresse de début de zone

=> End=

Adresse de fin de zone

=> Value=

Valeur hexadécimale

Exemple:

?T Start=0100 End=0109 Value=FA

Commande <U> Permet d'entrer une commande FLEX

=> +++

Commande FLEX

Exemple:

?U+++CAT 0



```
Catalogue du lecteur 0
Disque: GOUFIL 20
Nom      Type Taille   Date   Prt
G3FLEX02.SYS  26   3-Mar-83
ERRORS .SYS    9   30-Avr-83
.PRINT  .SYS    1   3-Mar-83
etc.
```

Commande <W> Permet de sauver le code objet de la zone spécifiée par la commande <N> ou <S>

=> Output file-name
On doit préciser le nom du fichier binaire dans lequel va être sauvegardé le code objet

Exemple:
?W
Output file name ASN.BIN
Start=C100 End=C1F7
Start=C1F8 End=C215
Xfer=C102

Commande <Y> Recherche dans une zone une chaîne d'octets

=> Start=
Adresse de début de zone
=> End=
Adresse de fin de zone
=> Value=
Chaîne d'octets

Exemple:
?Y Start=C100 End=C11F Value=235F

ADDRESS RANGE COMMANDS:

Commandes spécifiant comment seront désassemblées les zones indiquées

Commande <A> La zone spécifiée sera considérée comme des FDB

Exemple:
?A Start=C101 End=C104

Commande <D> La zone spécifiée sera considérée comme des FCD

Exemple:
?D Start=C10A End=C215

Commande <H> La zone spécifiée sera considérée comme des FCH

Exemple:
?H Start=C100 End=C100

Commande <I> La zone spécifiée sera considérée comme du code valide

Exemple:
?I Start=C105 End=C160

Commande <J> Comme I, mais affiche les codes ASCII dans la zone commentaire

Exemple:
?J Start=C100 End=C215

Commande <K> La zone spécifiée sera ignorée dans le désassemblage

Exemple:
?K Start=C215 End=C219



Pour toutes les commandes ci-dessus
=> Start=
Adresse de début de la zone
=> End=
Adresse de fin de la zone

MODE CHANGE COMMANDS:

Commandes spécifiant le mode de désassemblage
Sauf <Z> ces commandes agissent en bascule (pour voir l'état actuel:
commande <L>

**Commande ** Cross-assembleur:
Si en position OFF, désassemble le code source du microprocesseur
spécifié par la commande <Z>
Si en position ON et si le microprocesseur n'est pas le 6502,
désassemble en mnemoniques 6809
Exemple:
?BB

Commande <E> Si en position ON, affiche les labels sur une ligne fictive sur
laquelle ne figure pas de code objet
(Ex: C15C 39 RTS
C15D ZC15D ERU *
C15D 8DF5 8SR ZC154)
Si en position OFF, affiche les labels à gauche du source
(Ex: C15C 39 RTS
C15D 8DF5 ZC15D 8SR ZC154)
Exemple:
?EE

Commande <P> Si en position ON et si le microprocesseur est le 6809, désassemble
tous les adressages mémoire absolus, interne au programme, en un
mode d'adressage équivalent indépendant de la position
(Ex: LEAX #C140,PCR ; JMP #C105,PCR)
Si en position OFF, désassemble normalement
(Ex: LDX #C140 ; JMP #C105)
Exemple:
?PP

Commande <Z> Spécifie sur quel microprocesseur a été écrit le code objet
=> 0=6800/2/8,1=6801/3,2=6502,5=6805,9=6809
Numero (0,1,2,5 ou 9) correspondant au microprocesseur
Exemple:
?Z
0=6800/2/8,1=6801/3,2=6502,5=6805,9=6809 9

MISCELLANEOUS COMMANDS:

Commande <G> Charge ou sauve le fichier texte spécifiant tous les paramètres
de SLEUTH
=> Auxiliary Input file name
Nom du fichier de paramètres que l'on veut charger, ou <RETURN> si
l'on veut en sauve un
=> Auxiliary Output file name
Nom du fichier que l'on veut sauve
Exemple:
?G
Auxiliary Input file name
Auxiliary Output file name ASNA.TXT

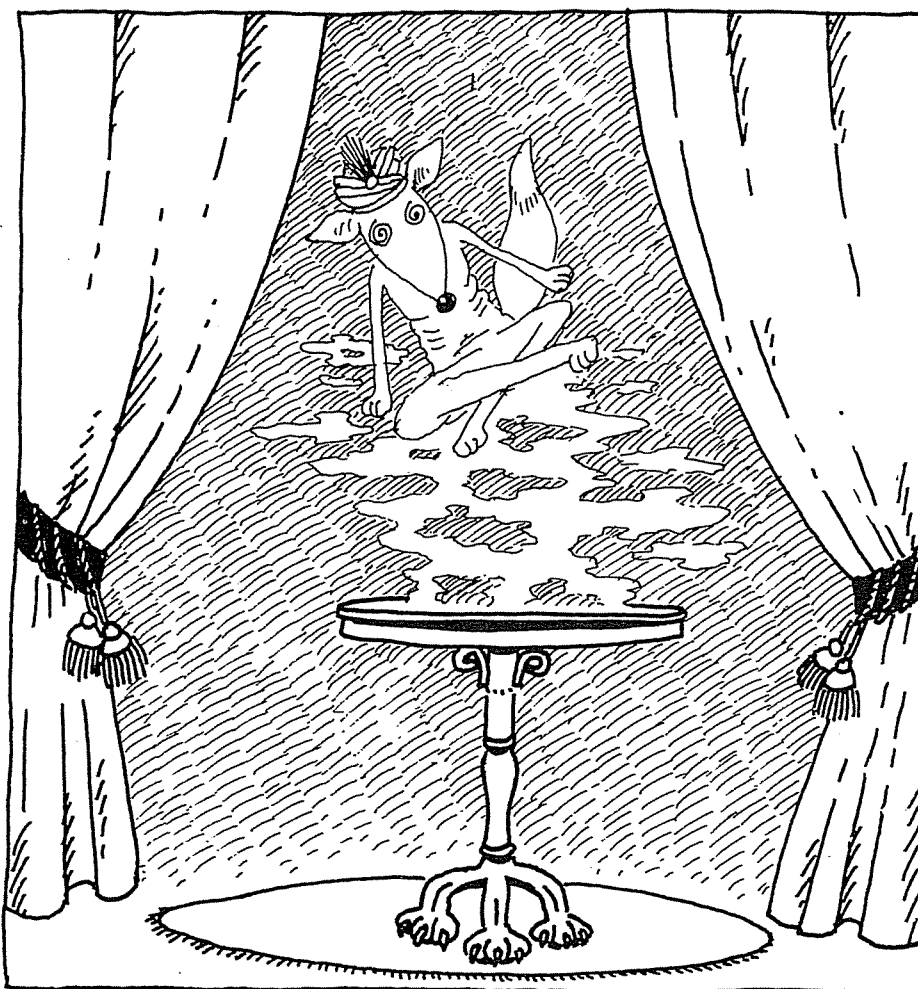


- Commande <L>** Liste l'état des paramètres
Exemple:
?L
J C100 C215
C C18A C213
H C100 C100
A C101 C104
I C105 C160
K C215 C219
C C215 C215
Start=C100 End=C215 Xfer=C102
separate-label switch off
CPU mode 6809
position-independence switch off
cross-assembler switch off
Input file name 0.ASN.COM
- Commande <N>** Spécifie une nouvelle zone de désassemblage
=> Start=
Adresse de début de la zone de désassemblage
=> End=
Adresse de fin de la zone de désassemblage
=> Xfer=
Adresse lancement du programme
Exemple:
?N Start=C100 End=C219 Xfer=C105
- Commande <O>** Spécifie une valeur à ajouter à l'adresse de chargement du programme
=> offset load value=
Valeur entre 0000 et FFFF
Exemple:
?O offset load value 0010
- Commande <R>** Réinitialisation de SLEUTH
Exemple:
?RR
CSC Super Sleuth V1.3
by Bud Pass, Conveyers, GA. C1980
Enter ? for help
- Commande <S>** Spécifie le nom du fichier binaire (.CMD, .BIN) que l'on veut désassembler
=> Input file name
Nom du fichier
Exemple:
?S
Input file name 0.ASN.COM
Start=C100 End=C215 Xfer=C102
separate-label switch off
CPU mode 6809
position-independence switch off
cross-assembler switch off
- Commande <X>** Spécifie l'adresse de lancement du programme
=> Xfer=
Adresse
Exemple:
?X Xfer=C102

(Nous ne vous apprendrons rien en vous disant que SLEUTH, en anglais, signifie limier)

David Mery

LOGICIELS DE BASE



Après le Sar Rabindranah Duval, le gourou Marahadji, le comte Jules Sneipi, la prêtresse Chantal Ferfeg, le maître John Risopedeau et le mage Sten Berebasinard, FLEX entre en transes et s'ouvre à la communication spatio temporelle avec l'au-delà.

Videotex, bases de données, télémaintenance, télédistribution n'ont plus de secrets pour lui grâce au logiciel le plus speed de l'année j'ai cité :

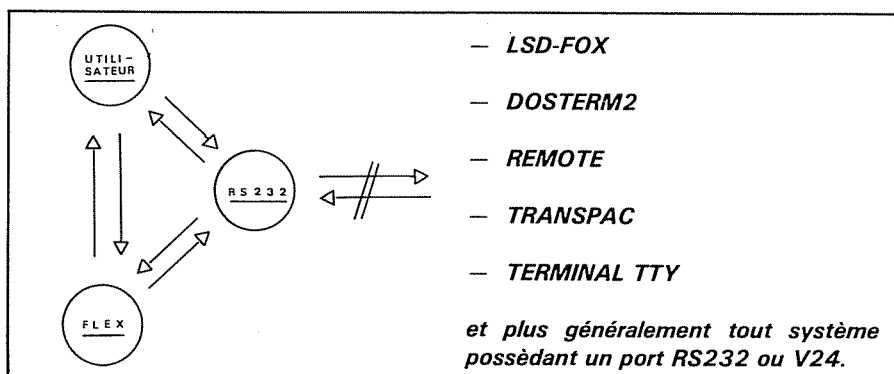
LSD - FOX

LSD-FOX ou le trépid communi-quant, il est le seul, l'unique pouvant prétendre à ce qualificatif après le gai ri donc car il permet en effet une communication totale avec l'au-delà.

Un simple schéma permet de s'en convaincre.

Les constituants

Le noyau LSD-FOX.COMD prend la main sur FLEX et contrôle les entrées-sorties.





LINELINK.SYS

contient les sous-programmes d'entrée sortie caractère et assure la portabilité de LSD-FOX d'une machine à une autre.

EXIT.CMD

permet de retourner sous contrôle du moniteur de FLEX

Des utilitaires de paramétrage de la liaison

SET.CMD et PROFIL.CMD

permettent de contrôler la vitesse de transmission, la longueur des caractères transmis, la parité, le nombre de bits d'arrêt, l'écho local et ligne, les protocoles d'émission et réception, le mode remote, etc.

Des utilitaires de transfert

TX.CMD

analogue à la commande P.CMD

SEND.CMD

émission fichier

RECEIVE.CMD

réception fichier

MSG.CMD

gestion de séquence de messages

Une commande spécifique du GOUPIL 3 moniteur 1.2

VIDEOTEX.CMD

adapte LSD-FOX.CMD aux protocoles télétextes.

Un utilitaire supplémentaire à FLEX

EX.CMD

commande EXEC relocatable avec passage possible de paramètres dans la ligne de commandes.

Ce qu'apporte LSD-FOX par rapport aux autres logiciels de communication :

— Portabilité d'une machine FLEX à une autre liée au fichier LINELINK.SYS

LSD-FOX fonctionne actuellement sur :

GOUPIL 2 (GPFLEX)
GOUPIL 3 moniteurs 1.1 et 1.2
THEMIS
CANNON CX1
Système cartes europe EFCIS
Système PEEP
TAVERNIER 6800
TAVERNIER 6809
MS1
et bientôt sur VEGAS 6809

— Assure des transferts de fichiers entre ces machines de façon fiable et facile.

— Conserve l'usage de toutes les commandes FLEX lors de son fonctionnement.

— Permet une télémaintenance réelle et efficace sur toutes ces machines.

— Procure un confort d'utilisation sans pareil lors de la consultation de bases de données.

— Permet l'automatisation de procédures de connexion.

— S'adapte facilement à des procédures de transmission particulière (polling buroughs, etc.)



En conclusion

Pour un prix très raisonnable quant à ses possibilités, LSD-FOX est le logiciel de communication que l'on se doit de posséder dans sa bibliothèque de logiciels FLEX au même titre que l'éditeur, l'Assembleur ou le basic.

Didier Cugy

* LSD-FOX est distribué par le réseau S.M.T. pour de plus amples renseignements :

S.M.T.

Service logiciel
4 impasse Garnier
75015 PARIS



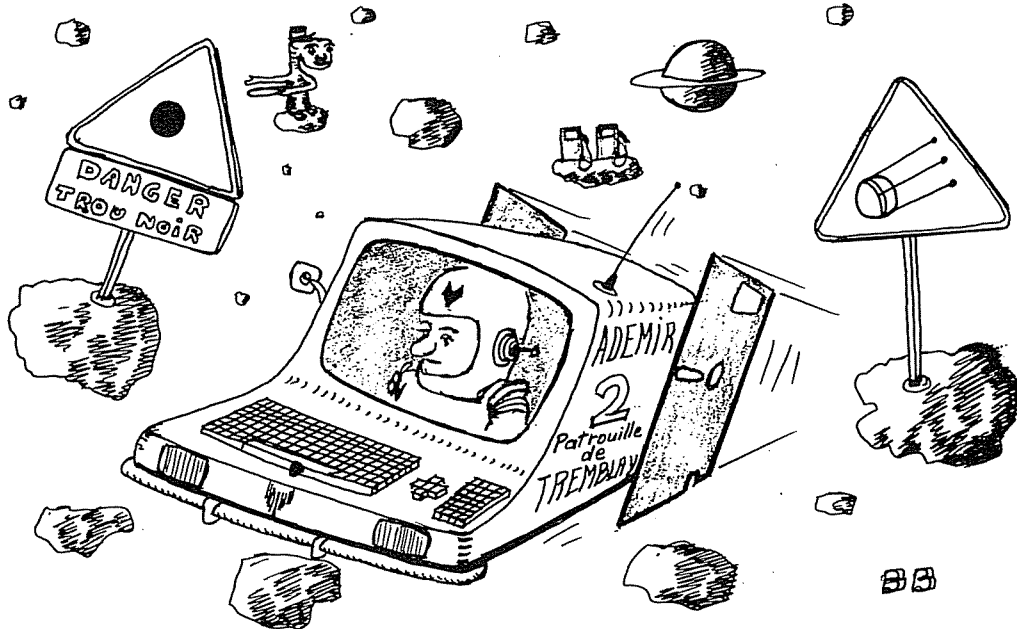
SEMINAIRES DE LA FEDERATION NATIONALE MICROTEL

AVEC L'AIDE DE L'ADI

BA 1	BASIC NIVEAU 1 Initiation au Micro-Ordinateur et à la programmation en langage Basic	14/11 au 17/11/83 26/12 20 29/12/83	1700 F
BA 2	BASIC NIVEAU 2 Perfectionnement à la pratique des tableaux des fichiers séquentiels et des fichiers à accès directs en langage Basic	28/11 au 1/12/83	1900 F
TX 1	Initiation à la pratique de programmes de traitement de texte	21 novembre	600 F
GF 1	Initiation à la pratique de programmes gestionnaires de fiches	22 novembre	600 F
TA 1	Initiation à la pratique de programmes du type tableur	23 novembre	600 F
TB 1	Initiation à la télématique et à la bureautique. Norme Vidéotex, Pratique du Télétel, Application pratique	12/12 au 15/12/83	2100 F

Pour tout renseignement : Jean-Marie LEGER
Département Formation. Fédération Nationale Microtel
9, rue Huysmans 75006 PARIS. Tél. (1) 544 70 23

LE JEU DES ETOILES



En fait, le jeu du serpent annoncé dans l'éditorial du numéro précédent s'est transformé en jeu des Etoiles offrant beaucoup plus de possibilités. Ce programme, bien qu'autonome, ne constitue pas vraiment un programme de jeu mais plutôt une base de départ de nombreux jeux possibles sur ce genre de thème.

Comme le programme de la « brique volante », celui-ci utilise aussi les possibilités d'affichage de la carte 24 x 80 et fonctionne sous le X BASIC du Goupil 2. Le principe : un « vaisseau spatial » se déplace dans un univers limité, il part d'un point et doit arriver à un autre. Quoi de plus simple ?

A chaque déplacement, un certain nombre d'obstacles, des étoiles supplémentaires, surgissent de façon aléatoire dans cet univers en création et risquent de couper la route.

Au début apparaît un cadre qui se remplit d'étoiles, c'est le petit morceau de la galaxie dans laquelle vous allez vous promener, dans une petite soucoupe bien ronde représentée par un '0'.

Le départ du voyage se situe en haut et à gauche de l'écran, l'arrivée en bas à droite. Pour déplacer votre vaisseau spatial, vous disposez de commandes qui sont les chiffres de la claviers numérique. (voir figure 1).

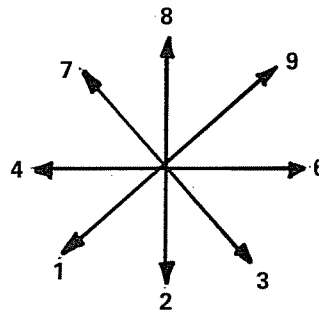


Figure 1.

Si vous heurtez une étoile, l'accident est sans danger, mais vous est signalé par un bip sonore.

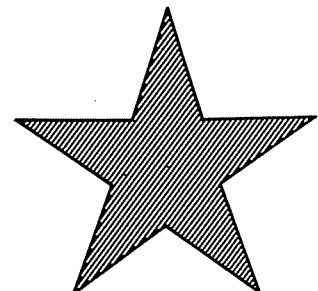
A chaque fois que vous jouez, d'autres étoiles apparaissent. Vous avez 80 coups à jouer pour parvenir à la sortie de l'espace.

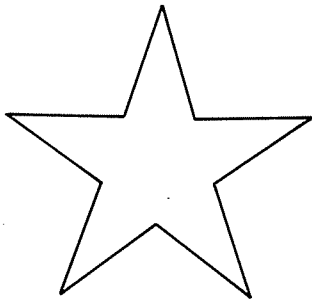
Comme le programme ne prévoit pas le cas où le chemin qui mène à l'arrivée est complètement bouché, et afin de ne pas jouer les 80 coups nécessaires pour terminer la partie (ligne 2220) il est possible d'arrêter le programme à tout moment en appuyant sur la touche "F" (ligne 2110).

A partir de là, on peut imaginer tout un tas de sous-programmes venant rendre cette situation un peu plus intéressante.

Quelques idées

- On peut déjà améliorer l'affichage en remplaçant les étoiles (affichées "x") par ceci "x", le vaisseau par " " (ou autre) de façon à rendre l'affichage un peu plus "carré". Il faut alors multiplier par deux les coordonnées en Y et diviser par deux le nombre d'éléments du tableau sur chaque ligne.
- Les coordonnées des points de départ et d'arrivée peuvent devenir aléatoires.
- On peut créer la nécessité de se ravitailler en carburant sur des bases en cours de route (rajouter alors l'affichage d'une "jauge") avec des risques de panne sèche...
- Il peut y avoir des zones à éviter, des "trous noirs" dont on ne peut plus sortir.
- On peut imaginer qu'en cas de collision avec une étoile, la pénalité soit justement d'en voir apparaître 36 d'un coup !





— Si le chemin vers l'arrivée est bouché, on peut imaginer de pouvoir forcer le passage, au prix d'une consommation accrue de carburant (dommage si c'est justement pour pouvoir se rendre à une base de ravitaillement).

— On peut imaginer (attention ça devient savant !) que cette promenade, qui se déroule sur un seul plan, se situe dans un espace à trois dimensions (avec la possibilité de visualiser les différents plans pour pouvoir calculer sa trajectoire).

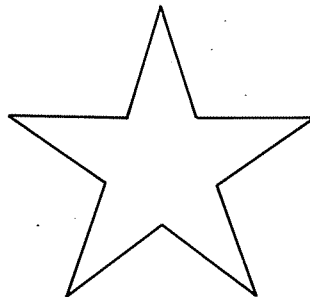
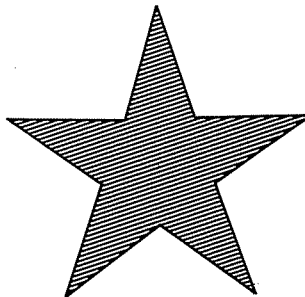
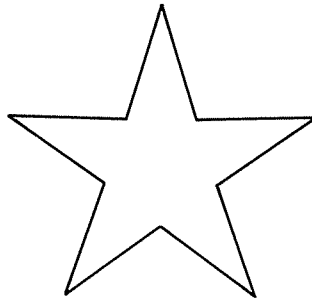
— On peut ignorer où se trouve la base d'arrivée (ou bien un autre vaisseau à rejoindre, qui se déplacerait en plus !) mais on pourrait connaître sa distance (et Pythagore ouvre un œil et commence à rêver d'hypothénuses à calculer en trois dimensions).

— Au lieu d'avoir une entrée de déplacement bloquante par INCH\$ (0) on peut faire se déplacer le vaisseau de façon continue à l'aide de PEEK appropriés (voir dans les "ruses du goupil" des numéros précédents) ou passer en SBA-SIC et utiliser l'instruction INCH\$ (-1), mais gare aux collisions ! Il faut alors bien entendu pouvoir commander la vitesse et les directions dans l'espace (aïe).

— Bien entendu de gentils E.T. peuvent intervenir dans cette aventure...

Voilà plein d'idées qui, je suis sûr, vous serviront à le modifier à l'aide de vos propres idées et qui feront peut être que ce programme deviendra un autre jeu comme le jeu du serpent est devenu le jeu des étoiles...

Bernard Beville
Microtel Ademir Tremblay



Variables utilisées

A\$(X, Y) = "" affichage d'une étoile en coordonnées X, Y

A\$(X1, Y1) = "0" affichage du vaisseau en X1, Y1

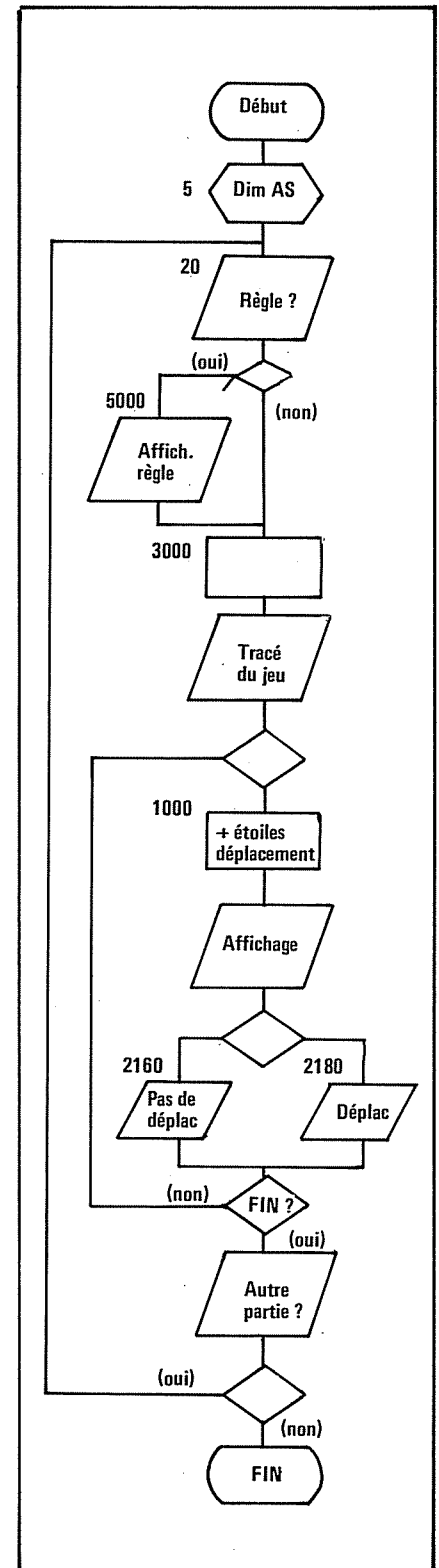
A\$(X2, Y2) = "" effacement de la position précédente

Variables de la ligne N° 50

K = nombre de coups dans une partie

H = nombre d'étoiles qui naissent à chaque coup joué

G = nombre d'étoiles posées avant le départ.



```

5 DIM A$(20,60)
10 PRINT "*****"
*****
11 PRINT "***"
***
12 PRINT "***"
***
13 PRINT "*****"
*****
14 PRINT
15 PRINT TAB(20)"Auteur : B.Beville ADEMIR-TREMBLAY "
16 PRINT TAB(20)"Version pour GOUPIL2 carte 24 x 80"
  
```

Le jeu des étoiles

```
17 PRINT TAB(20)"    Sous G2FLEX en XBASIC"
18 PRINT
20 PRINT"Voulez-vous les r)gles du jeu    (O/N)";:R#=INCH$(0)
30 IF R#="O" OR R#="o" THEN GOSUB 5000
45 PRINT CHR$(27);CHR$(69)
50 K=80:H=5:G=100
90 GOSUB 3000
1000 REM*****INITIALISATIONS*****
1010 FOR I=1 TO H
1015 PRINT CHR$(27);CHR$(28);
1020 X=INT(20*RND(0))+1
1030 Y=INT(60*RND(0))+1
1040 IF X)=20 THEN X=20
1050 IF Y)=60 THEN Y=60
1060 IF X)<=2 THEN X=2
1070 IF Y)<=2 THEN Y=2
1080 IF A$(X,Y)="O" THEN 1110
1084 A$(X,Y)="*"
1085 IF A$(2,2)="*" THEN A$(2,2)=" "
1086 IF A$(20,60)="*" THEN A$(20,60)=" "
1100 PRINT CHR$(27);"=";CHR$(X);CHR$(Y);A$(X,Y)
1110 NEXT I
2000 REM*****DEPLACEMENT*****
2010 Y2=Y1:X2=X1
2020 R#=INCH$(0)
2030 IF R#="3" THEN X1=X1+1:Y1=Y1+1
2040 IF R#="1" THEN X1=X1+1:Y1=Y1-1
2050 IF R#="9" THEN X1=X1-1:Y1=Y1+1
2060 IF R#="7" THEN X1=X1-1:Y1=Y1-1
2070 IF R#="2" THEN X1=X1+1
2080 IF R#="8" THEN X1=X1-1
2090 IF R#="6" THEN Y1=Y1+1
2100 IF R#="4" THEN Y1=Y1-1
2110 IF R#="F" THEN END
2120 IF X1)<=2 THEN X1=2
2130 IF Y1)<=2 THEN Y1=2
2140 IF X1)=20 THEN X1=20
2150 IF Y1)=60 THEN Y1=60
2160 IF A$(X1,Y1)="*" THEN 2170 ELSE 2180
2170 PRINT CHR$(7):X1=X2:Y1=Y2
2175 GOTO 2210
2180 A$(X1,Y1)="O":A$(X2,Y2)=" "
2190 PRINT CHR$(27);"=";CHR$(X2);CHR$(Y2);A$(X2,Y2)
2200 PRINT CHR$(27);"=";CHR$(X1);CHR$(Y1);A$(X1,Y1)
2210 IF X1)=20 AND Y1)=60 THEN 2500
2220 IF K)<=0 THEN 2500
2230 K=K-1
2235 PRINT CHR$(27);"I";
2240 PRINT CHR$(27);"=";CHR$(22);CHR$(20);"Reste @ jouer ";K;" coups"
2245 PRINT CHR$(27);"J";
2250 GOTO 1000
2500 REM***** FIN DE PARTIE *****
2510 PRINT CHR$(27);"C"
2520 PRINT CHR$(12)
2530 INPUT " Voulez vous jouer une autre partie ";R#
2540 IF R#="O" OR R#="o" THEN 10
2550 END
3000 REM*****TRACE DU JEU*****
3010 PRINT CHR$(12)
3015 PRINTCHR$(27);CHR$(28);
3020 PRINT CHR$(142);
3030 FOR I=1 TO 60:PRINTCHR$(27);CHR$(150);:NEXT I
3040 PRINT CHR$(27);CHR$(141)
3050 FOR I=1 TO 19
3060 PRINT CHR$(27);CHR$(149);
3070 FOR L=1 TO 60:PRINT" ";:NEXT L
3080 PRINT CHR$(27);CHR$(149)
3090 NEXT I
3100 PRINT CHR$(27);CHR$(143);
3110 FOR I=1 TO 60:PRINTCHR$(27);CHR$(150);:NEXT I
3120 PRINT CHR$(27);CHR$(144)
4000 REM*****POSE DES ETOILES*****
4010 FOR I=1 TO G
4015 PRINT CHR$(27);CHR$(28);
4020 X=INT(20*RND(0))+1
4030 Y=INT(60*RND(0))+1
```

```

4040 IF X)=20 THEN X=20
4050 IF Y)=60 THEN Y=60
4060 IF X(<=2 THEN X=2
4070 IF Y(<=2 THEN Y=2
4080 IF A$(X,Y)="0" THEN 4110
4085 A$(X,Y)="*"
4090 PRINT CHR$(27);"=";CHR$(X);CHR$(Y);A$(X,Y)
4100 NEXT I
4105 PRINT CHR$(28);
4110 A$(2,2)="D":X1=2:Y1=2
4120 PRINT CHR$(27);CHR$(2);CHR$(2);A$(2,2)
4121 A$(20,60)="A"
4122 PRINT CHR$(27);"=";CHR$(20);CHR$(60);A$(20,60)
4130 RETURN
5000 REM*****REGLE DU JEU*****
5010 PRINT CHR$(12)
5020 PRINT"          LA PROMENADE DANS LES ETOILES"
5040 PRINT"          Au debut apparait un cadre qui se remplit d'etoiles,c'est le
petit"
5050 PRINT"morceau de galaxie dans laquelle vous allez vous promener , dans un
e "
5060 PRINT"petite soucoupe bien ronde representee par un '0' "
5070 PRINT
5080 PRINT"          Le depart du voyage se situe en haut et a gauche de l'ecran , l
'arrivee"
5090 PRINT"en bas a droite."
5110 PRINT"          Pour deplacer votre vaisseau spatial , vous disposez de command
es "
5120 PRINT"qui sont les chiffres du clavier numerique "
5130 PRINT" Les directions sont donnees par leur position          7 8 9"
5140 PRINT"          + ! /"
5150 PRINT"          4 - - 6"
5160 PRINT"          / ! +
5170 PRINT"          1 2 3"
5180 PRINT
5190 PRINT"          Si vous heurtez une etoile , l'accident est sans danger , mais
vous"
5200 PRINT"est signale par un bip sonore ."
5220 PRINT"          A chaque fois que vous jouez , d'autres etoiles apparaissent et
"
5230 PRINT"risquent de vous boucher le chemin qui vous amene a la sortie ! "
5240 PRINT"          Vous avez 80 coups a jouer pour parvenir a la sortie de l'espac
e "
5250 PRINT"          BON VOYAGE !"
5255 PRINT"(pour demarrer le jeu appuyez sur une touche)"
5260 R$=INCH$(0)
5270 RETURN

```



noyema
INFORMATIQUE

vous propose
sur
GOUPIL 2 et 3

- Votre gestion dentaire NOGEDENT
- Votre dessin assisté par ordinateur DAMOG avec table à digitaliser (2794 X 2794 points)
- Toutes vos cartes d'interfaces sur études ou en standart standard E/S analogiques et digitales .

NOGÉMA Centre d'Affaires LES NATIONS Tél: (8) 356.89.57 54500 VANDOEUVRE

AV2i (M. Vidal) 16, rue de l'Abbé Grégoire

92130 ISSY-LES-MOULINEAUX • Tél. (1) 644.86.47

AVICALC

- Un gestionnaire de tableaux électroniques
 - Un outil pour responsables et décisionnaires
 - Un véritable tableau de bord
- Facile d'emploi et adapté aux utilisateurs français

Logiciels : ASTRID I - ASTRID II - ASTRID III - AVICALC

INFORMATIQUE

Spécialistes en France

POUR APPARAÎTRE
ICI

239.37.47



PROGRAMMEZ VOS FICHES DE PAYE

Ce programme permet de gérer la paye du personnel d'une entreprise. Il est capable d'établir une feuille de paye complète à partir du nom, de l'adresse, du prix horaire d'un employé, de créer un fichier contenant toutes les informations sur le personnel, de réutiliser ce fichier pour les mois suivants, de le modifier et de faire des recherches par critères. Il peut gérer plusieurs fichiers de noms différents.

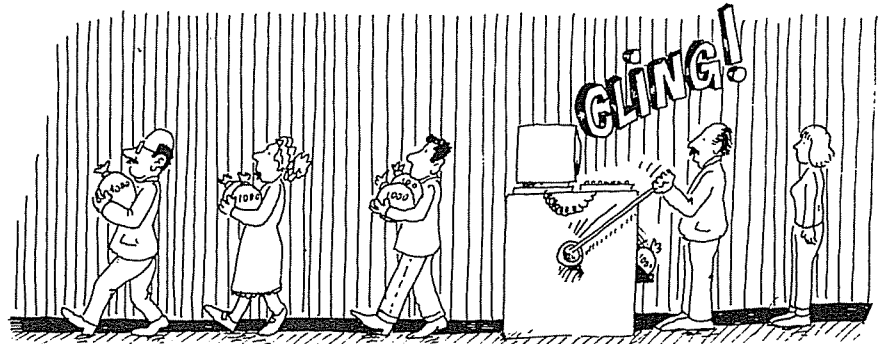


Table des variables utilisées

A.....	: longueur du nom du fichier
Bf.....	: stockage des DATA
C.....	: prix de l'heure
Ca.....	: pourcentage de la retenue chômage
Cc.....	: pourcentage de la retenue retraite
Ch.....	: n° du choix lors du tri ou de la modification
Ci.....	: cumul imposable
Cp.....	: nombre de fiche dans un fichier
D.....	: salaire brut
EE.....	: nombre de DATA à lire
F.....	: acompte
I.....	: compteur de boucle
Ns.....	: longueur de la donnée à trier ou à modifier
Nsa.....	: longueur totale des données situées avant NS
Nu.....	: numéro de la fiche choisie
Rc.....	: retenue chômage
Rr.....	: retenue retraite
Rs.....	: retenue sécurité sociale
S.....	: numéro de la fiche choisie
Sn.....	: salaire net
Ss.....	: pourcentage de la retenue sécurité sociale
Tp.....	: flag de retour du programme d'affichage écran
Tr.....	: total des retenues
Wx.....	: compteur de boucle
A\$.....	: nom
B\$.....	: prénom
C\$.....	: adresse
Cc\$.....	: prix de l'heure
Ce\$.....	: acompte
Ch\$.....	: chaîne contenant le n° du choix lors du tri ou de la modification
Cp\$.....	: chaîne contenant le nombre de fiche du fichier
D\$.....	: ville
Df\$.....	: message de présentation au pluriel ou au singulier suivant longueur de V\$
E\$.....	: chaîne contenant le code postal
F\$.....	: chaîne contenant l'acompte
Fi\$.....	: nom du fichier
Fi1\$.....	: ancien nom du fichier



Fi2\$..... : nouveau nom du fichier
 Fic\$..... : nom du fichier plus extension
 Nn\$..... : ancienne donnée
 Nu\$..... : chaîne contenant le numéro de la fiche choisie
 R\$..... : donnée à chercher lors du tri
 Re\$..... : variable de stockage des données
 S\$..... : stockage des données précédant la donnée à lire
 St\$..... : identique à S\$
 U\$..... : donnée à lire

 V\$..... : chaîne contenant les numéros des fiches trouvées lors du
 Z\$..... : contient une suite de = pour souligner les titres

Tables des références

TABLES DES REFERENCES

A.....	1030	1040	1060	1060	1720	1730
	1730	1770	2560	2570	2720	2720
BF.....	3190	3410	3410	5200	5200	
C.....	550	550	550	560	600	1080
	1090	1240	2160	2160	2160	2170
	2210	2730	2740	2900	4180	4240
	4660	4660	4660	4690	4710	
CA.....	100	580	1120	2180	2770	4700
CC.....	110	580	1110	2180	2760	4700
CH.....	3310	3320	3320	3330	3370	3410
	3450	3450	4470	4480	4480	4490
	4510	5150	5200	5350	5350	
CI.....	1150	1320	2800	2980		
CP.....	340	350	350	350	410	870
	880	890	940	940	950	950
	960	2630	2640	2650	3550	3560
	4090	4130	4130	4140	4140	4150
	4870	4900	4900	4910	4910	4920
	4980	5000	5040			
D.....	560	580	580	580	590	1090
	1100	1110	1120	1140	1290	2170
	2180	2180	2180	2200	2740	2750
	2760	2770	2790	2950	4680	4690
	4700	4700	4700			
EE.....	3410	3410	5200	5200		
F.....	590	590	590	600	1080	1140
	1300	2200	2200	2200	2210	2730
	2790	2960	4180	4250	4680	4680
	4680	4710				
I.....	410	640	650	1040	1040	1060
	1060	2010	2050	2250	2270	2270
	2280	2570	2570	2720	2720	3010
	3010	4520	4720	5000	5010	5020
	5030					
NS.....	3420	3460	3540	3860	5210	5220
NSA.....	3190	3410	3410	3540	3860	5200
	5200	5200	5220			
NU.....	930	940	950	960	960	960
	960	1010	1050	1060	2690	2700
	2710	2720	3580	3800		
RC.....	1120	1130	1270	2770	2780	2930
RR.....	1110	1130	1260	2760	2780	2920
RS.....	1100	1130	1250	2750	2780	2910
S.....	4120	4130	4140	4150	4150	4160
	4520	4900	4910	4920	4920	4920
	4930	5000	5230	5370		
SN.....	1140	1150	1310	2790	2800	2970
SS.....	90	580	1100	2180	2750	4700
TP.....	730	1330	3780			
TR.....	1130	1140	1280	2780	2790	2940
WX.....	3560	3570	3580	3590		
A\$.....	320	460	900	1190	1710	1720
	1940	2060	2660	2840	3820	4040
	4190	4660	4570	4990	5510	
B\$.....	320	470	900	1200	1940	2080
	2660	2850	3820	4040	4200	4560
	4580	4990				



Programmez vos fiches de paye



C\$.....	320	480	900	1210	1940	2090
	2660	2860	3820	4040	4210	4560
	4590	4990				
CC\$.....	320	600	900	1080	1940	2210
	2660	2730	3820	4040	4180	4560
	4710	4990				
CE\$.....	330	340	360	540	550	570
	590	2150	2160	2190	2200	4650
	4660	4670	4680			
CH\$.....	3300	3310	3440	3630	3720	3880
	3890	3900	3910	3920	3930	3940
	4460	4470	5290	5440	5450	5460
	5470	5480	5490	5500		
CP\$.....	310	360	820	870	1930	2010
	2280	2600	2630	3520	3550	4030
	4090	4550	4850	4870	5040	
D\$.....	320	490	900	1220	1940	2100
	2660	2870	3820	4040	4220	4560
	4600	4990				
DF\$.....	3700	3700	3720			
E\$.....	320	520	900	1230	1940	2130
	2660	2880	3820	4040	4230	4560
	4630	4990				
F\$.....	320	600	900	1080	1940	2210
	2660	2730	3820	4040	4180	4560
	4710	4990				
FI\$.....	200	210	210	220	290	720
	740	740	750	800	1020	1030
	1350	1430	1850	1860	1860	1870
	1910	2440	2450	2450	2460	2550
	2560	2580	3030	3470	3480	3480
	3500	3630	3720	3990	4000	4000
	4010	4810	4820	4820	4830	4930
	5570	5580	5580	5590	5630	
FI1\$.....	5700	5710	5710	5720		
FI2\$.....	5730	5740	5740	5750		
FIC\$.....	290	300	800	810	1910	1920
	2580	2590	3500	3510	3810	3850
	4010	4020	4530	4830	4840	4980
	5160	5630	5640			
FIC1\$.....	5720	5760				
FIC2\$.....	5750	5760				
NN\$.....	5300	5340	5440	5450	5460	5470
	5480	5490	5500			
NU\$.....	920	930	2670	2690		
R\$.....	3450	3630	3720			
RE\$.....	220	230	230	240	240	460
	470	480	490	510	510	520
	750	760	760	770	770	1350
	1360	1360	1370	1370	1870	1880
	1880	1890	1890	2060	2070	2070
	2080	2090	2100	2120	2120	2130
	2380	2460	2470	2470	2480	2480
	3030	3040	3040	3050	3050	3440
	3450	3450	3450	3460	3460	3460
	3580	4570	4580	4590	4600	4620
	4620	4630	4940	4950	4950	4960
	4960	5340	5350	5360	5400	5410
	5410	5420	5420	5590	5600	5600
	5610	5610				
S\$.....	3540	3860	4110	4120	4890	4920
	5220	5240	5360			
ST\$.....	5240	5360				
U\$.....	3540	3580	3860	5220	5290	5360
V\$.....	3490	3630	3640	3700	3720	3800
	3800					
Z\$.....	1040	1040	1040	1040	1060	1060
	1060	1070	1070	2570	2570	2570
	3000	3010	3010	3010		



Un conseil : ce programme étant très long, ne pas attendre d'avoir tout tapé pour le sauvegarder. Une coupure de courant est si vite arrivée.

Marc Abramson



+++++ FICHE DE PAYE +++++

Fichier exemple

=====

fiche n° 1

=====

NOM : Dupond
 PRENOM : Jacques
 ADRESSE : 8 , rue des crabes fleuris
 VILLE : Triffouilli les olives
 CODE POSTAL: 86321

Prix de l'heure : 150.00
 Retenue S. S. : 2611.05
 Retenue Retraite : 1318.20
 Retenue chomage : 334.62
 Total des retenues : 4263.87
 Salaire brut : 25350.00
 Accompte : 25.00
 Salaire net : 21061.13
 Cumul imposable : 252733.56

+++++



```
*****
*
* GESTION DE FICHIERS
* PAR Marc Abramson
* COPYRIGHT 1983 GOUPIL REVUE ET L'AUTEUR
*
*****
```

```
10 REM
20 REM
30 REM
40 REM
50 REM
60 REM
70 REM
80 REM
90 SS=10.3 : REM +++retenue securite sociale en % ++++
100 CA=1.32 : REM +++retenue chomage en % ++++
110 CC=5.20 : REM +++retenue retraite en % ++++
120 PRINT CHR$(12)
130 ON ERROR GOTO 1420
140 EXEC, "ttyset ps=n"
150 GOTO 1550
160 REM
170 REM ***** creer un nouveau fichier *****
180 REM
190 PRINT CHR$(12)
200 INPUT "nom du fichier"; FI$
210 FI$=LEFT$(FI$, 7)
220 CURSOR 3, 2 : PRINT "est ce bien "; FI$; " (O/N)" : RE$=INCH$(0)
230 IF RE$="O" OR RE$="o" THEN GOTO 260
240 IF RE$="n" OR RE$="N" THEN RETURN
250 CURSOR 3, 2 : PRINT CHR$(24) : GOTO 220
260 PRINT CHR$(12)
270 PRINT "***** FICHE DE PAYE *****"
280 PRINT
290 FIC$=FI$+".mar"
300 OPEN NEW FIC$ AS 1
310 FIELD #1 , 5 AS CP$
```



Programmez vos fiches de paye

```
320 FIELD #1 ,30 AS A$,30 AS B$,50 AS C$,30 AS D$,5 AS E$,7 AS CC$, 7 AS F$
330 CURSOR 4,1:INPUT "COMBIEN DE PERSONNEL ";CE$
340 CP=VAL(CE$)
350 IF CP(<=0 OR INT(CP) <> CP THEN CURSOR 4,1 :PRINT CHR$(24) :GOTO 330
360 LSET CP$=CE$
370 REM
380 REM ***** ecriture du nombre de donnees sur le disque *****
390 REM
400 PUT #1,RECORD 1
410 FOR I=1 TO CP
420 REM
430 REM ***** entree des donnees *****
440 REM
450 PRINT CHR$(12)
460 CURSOR 4,20 : PRINT "Nom:"; GOSUB 2380: LSET A$=RE$
470 CURSOR 5,20: PRINT "Prenom:"; GOSUB 2380: LSET B$=RE$
480 CURSOR 6,20: PRINT "Adresse "; GOSUB 2380: LSET C$=RE$
490 CURSOR 7,20: PRINT "Ville "; GOSUB 2380: LSET D$=RE$
500 CURSOR 8,20: PRINT "Code postal:"; GOSUB 2380
510 IF VAL(RE$)=0 OR VAL(RE$) >99999 THEN CURSOR 8,20: PRINT CHR$(24): GOTO500
520 LSET E$=RE$
530 CURSOR 11,10: PRINT "Nombre d'heures:169"
540 CURSOR 12,10:INPUT "Prix de l'heure ";CE$
550 C=VAL(CE$) :IF C <=0 OR C >1000 THEN CURSOR 12,10: PRINT CHR$(24): GOTO 540
560 D=C*169
570 CURSOR 13,10: INPUT "Accompte:";CE$
580 D=D-D*(SS+CA+CC)/100
590 F=VAL(CE$): IF F<=0 OR F>D THEN CURSOR 13,10: PRINT CHR$(24): GOTO570
600 LSET CC$=STR$(C): LSET F$=STR$(F)
610 REM
620 REM ***** ecriture sur fichier *****
630 REM
640 PUT #1 ,RECORD I+1
650 NEXT I
660 CLOSE 1
670 RETURN
680 REM
690 REM ***** consulter les fiches d'un ancien fichier *****
700 REM
710 PRINT CHR$(12)
720 INPUT "Nom du fichier";FI$
730 TP=0
740 FI$=LEFT$(FI$,7)
750 CURSOR 3,2: PRINT "Est ce bien "; FI$; " (O/N)": RE$=INCH$(0)
760 IF RE$="O" OR RE$="o" THEN GOTO 790
770 IF RE$="n" OR RE$="N" THEN RETURN
780 CURSOR 3,2: PRINT CHR$(24): GOTO 750
790 PRINT CHR$(12)
800 FIC$=FI$+".mar"
810 OPEN OLD FIC$ AS 1
820 FIELD #1 ,5 AS CP$
830 REM
840 REM ***** lecture du nombre de fiches dans le fichier *****
850 REM
860 GET #1 ,RECORD 1
870 CP=VAL(CP$)
880 PRINT "Ce fichier contient "; CP ;"donnee";
890 IF CP>1 THEN PRINT "s" ELSE PRINT
900 FIELD #1,30 AS A$,30 AS B$,50 AS C$,30 AS D$,5 AS E$,7 AS CC$, 7 AS F$
910 CURSOR 20,3:PRINT "Quei numero de personnel ";
920 CURSOR 20,26: INPUT NU$
930 NU =VAL(NU$)
940 IF NU>CP THEN CURSOR 22,4 : PRINT "Ce fichier ne contient que";CP;"donnee";
950 IF NU>CP THEN IF CP > 1 THEN PRINT "s" ELSE PRINT
960 IF NU(<=0 OR NU<>INT(NU) OR NU>CP THEN CURSOR 20,26:PRINT CHR$(24): GOTO 920
970 PRINT CHR$(12)
980 REM
990 REM ***** lecture de la fiche numero NU *****
1000 REM
1010 GET #1,RECORD NU+1
```



```
1020 CURSOR 1,5: PRINT "Fichier ";FI$
1030 A=LEN(FI$)+8
1040 Z$="": FOR I=1 TO A: Z#=Z#+ "=" : NEXT I: CURSOR 2,5: PRINT Z$
1050 CURSOR 4,5: PRINT "Fiche n0";NU
1060 Z$="": A=LEN(STR$(NU)): FOR I= 1 TO A+8: Z#=Z#+ "=" : NEXT I
1070 CURSOR 5,5: PRINT Z$ :Z$=""
1080 C=VAL(CC$) :F=VAL(F$)
1090 D=C*169
1100 RS=D*SS/100
1110 RR=D*CC /100
1120 RC=D*CA/100
1130 TR=RS+RR+RC
1140 SN=D-TR-F
1150 CI=SN*12
1160 REM
1170 REM ***** affichage des donnees sur l'ecran *****
1180 REM
1190 CURSOR 6,23 :PRINT "Nom :";A$
1200 CURSOR 7,23 :PRINT "Prenom :";B$
1210 CURSOR 9,23 :PRINT "Adresse :";C$
1220 CURSOR 10,23 :PRINT "Ville :";D$
1230 CURSOR 11,23 :PRINT "Code postal :";E$
1240 CURSOR 14,15 : PRINT "    Prix de l'heure      :";: PRINT USING'#####.##',C
$250 CURSOR 15,15 : PRINT "    Retenue S.S.          :";: PRINT USING'#####.##',R
1260 CURSOR 16,15: PRINT "    Retenue Retraite     :";: PRINT USING'#####.##',RR
1270 CURSOR 17,15: PRINT "    Retenue chomage      :";: PRINT USING'#####.##',RC
1280 CURSOR 18,15: PRINT "    Total des retenues   :";: PRINT USING'#####.##',TR
1290 CURSOR 19,15: PRINT "    Salaire brut         :";: PRINT USING'#####.##',D
1300 CURSOR 20,15: PRINT "    Accompte             :";: PRINT USING'#####.##',F
1310 CURSOR 21,15: PRINT "    Salaire net          :";: PRINT USING'#####.##',SN
1320 CURSOR 22,15 :PRINT "    Cumul imposable     :";: PRINT USING'#####.##',CI
1330 IF TP=1 THEN CLOSE 1: RETURN
1340 CURSOR 23,13
1350 PRINT"Un autre numero dans le fichier ";FI$;" (O/N)":RE$=INCH$(0)
1360 IF RE$="0" OR RE$="o" THEN PRINT CHR$(12): GOTO 910
1370 IF RE$="n" OR RE$="N" THEN CLOSE 1: RETURN
1380 CURSOR 23,13: PRINT CHR$(24): GOTO 1350
1390 REM
1400 REM ***** traitement des erreurs *****
1410 REM
1420 IF ERR=4 THEN CLOSE 1
1430 IF ERR=4 THEN PRINT "Le fichier ";FI$;" n'existe Cpas sur cette disquette"
1440 IF ERR=4 THEN RESUME 1780
1450 IF ERR=34 THEN END
1460 PRINT "Erreur de manipulation du proqramme" : RESUME 1780
1470 REM
1480 REM ***** catalogue des fichiers *****
1490 REM
1500 PRINT CHR$(12) :EXEC," CAT 1 .MAR"
1510 RETURN
1520 REM
1530 REM ***** menu *****
1540 REM
1550 PRINT CHR$(12)
1560 PRINT"***** M E N U *****"
1570 PRINT
1580 CURSOR 4,2: PRINT "voulez vous:"
1590 CURSOR 6,15: PRINT "1 : consulter le catalogue des fichiers"
1600 CURSOR 7,15: PRINT "2 : consulter les donnees d'un ancien fichier"
1610 CURSOR 8 ,15:PRINT "3 : creer un nouveau fichier"
1620 CURSOR 9,15: PRINT "4 : completer un ancien fichier"
1630 CURSOR 10,15: PRINT "5 : sortir une fiche sur imprimante"
1640 CURSOR 11,15: PRINT "6 : recherche d'une fiche par critere connu"
1650 CURSOR 12,15: PRINT "7 : modifier une fiche "
1660 CURSOR 13,15: PRINT "8 : detruire une fiche "
1670 CURSOR 14,15: PRINT "9 : detruire un fichier"
1680 CURSOR 15,14: PRINT "10 : modifier le nom d'un fichier"
1690 CURSOR 16,14:PRINT "11 : quitter le programme"
1700 CURSOR 23,2: PRINT "votre choix";
1710 CURSOR 23,15: INPUT A$
```



Programmez vos fiches de paye

```
1720 A=VAL(A$)
1730 IF A<=0 OR A>11 THEN GOTO 1710
1740 REM
1750 REM ***** aiguillage vers un sous programme *****
1760 REM
1770 ON A GOSUB 1500, 710, 190, 1810, 2430, 3160, 3980, 4750, 5560, 5690, 3100
1780 PRINT "appuyer sur une touche pour revenir au menu"
1790 GOSUB 5510
1800 GOTO 1550
1810 PRINT CHR$(12)
1820 REM
1830 REM ***** completer un fichier *****
1840 REM
1850 INPUT "nom du fichier";FI$
1860 FI$=LEFT$(FI$,7)
1870 CURSOR 3,1: PRINT "est ce bien "; FI$; " (O/N)": RE$=INCH$(0)
1880 IF RE$="O" OR RE$="o" THEN GOTO 1910
1890 IF RE$="n" OR RE$="N" THEN RETURN
1900 CURSOR 3,1: PRINT CHR$(24): GOTO 1870
1910 FIC$=FI$+".mar"
1920 OPEN OLD FIC$ AS 1
1930 FIELD #1,5 AS CP$
1940 FIELD #1,30 AS A$,30 AS B$,50 AS C$,30 AS D$,5 AS E$,7 AS CC$, 7 AS F$
1950 REM
1960 REM ***** lecture du nombre de donnees dans le fichier *****
1970 REM
1980 GET #1,RECORD 1
1990 PRINT CHR$(12)
2000 PRINT"tapez ZZZZ comme nom pour revenir au menu"
2010 I=VAL(CP$)+1
2020 REM
2030 REM ***** entree des donnees *****
2040 REM
2050 CURSOR 3,23 :PRINT "fiche n0";I
2060 CURSOR 4,30: PRINT "Nom:";: GOSUB 2380: LSET A$=RE$
2070 IF RE$="ZZZZ" OR RE$="zzzz" THEN GOTO 2280
2080 CURSOR 5,30: PRINT "PRENOM:";: GOSUB 2380: LSET B$=RE$
2090 CURSOR 6,30: PRINT "Adresse ";: GOSUB 2380: LSET C$=RE$
2100 CURSOR 7,30: PRINT "Ville ";: GOSUB 2380: LSET D$=RE$
2110 CURSOR 8,30: PRINT "Code postal:";: GOSUB 2380
2120 IF VAL(RE$)=0 OR VAL(RE$)99999 THEN CURSOR 8,30: PRINT CHR$(24): GOTO 2110
2130 LSET E$=RE$
2140 CURSOR 11,10: PRINT "Nombre d'heures:169"
2150 CURSOR 12,10: INPUT "Prix de l'heure ";:CE$
2160 C=VAL(CE$): IF C<=0 OR C>1000 THEN CURSOR 12,10: PRINT CHR$(24): GOTO 2140
2170 D=C*169
2180 D=D-D*(SS+CA+CC)/100
2190 CURSOR 13,10: INPUT "Accompte:";CE$
2200 F=VAL(CE$): IF F<=0 OR F>D THEN CURSOR 13,10: PRINT CHR$(24): GOTO 2190
2210 LSET CC$=STR$(C): LSET F$=STR$(F)
2220 REM
2230 REM ***** ecriture des donnees sur le disque *****
2240 REM
2250 PUT #1,RECORD I+1
2260 PRINT CHR$(12)
2270 I=I+1 :GOTO 2050
2280 LSET CP$=STR$(I-1)
2290 REM
2300 REM ***** ecriture du nombre de donnees sur le disque *****
2310 REM
2320 PUT #1,RECORD 1
2330 CLOSE 1
2340 RETURN
2350 REM
2360 REM ***** routine d'entree des donnees *****
2370 REM
2380 INPUT LINE RE$
2390 RETURN
2400 REM
2410 REM ***** sortie sur imprimante *****
```



```
2420 REM
2430 PRINT CHR$(12)
2440 INPUT "Nom du fichier";FI$
2450 FI$=LEFT$(FI$,7)
2460 CURSOR 3,1: PRINT "Est ce bien "; FI$ ;" (O/N)": RE$=INCH$(0)
2470 IF RE$="O" OR RE$="o" THEN GOTO 2510
2480 IF RE$="n" OR RE$="N" THEN RETURN
2490 CURSOR 3,1: PRINT CHR$(24): GOTO 2460
2500 PRINT CHR$(12)
2510 LPRINT: LPRINT "+++++++ FICHE DE PAYE ++++++"
2520 REM
2530 REM ***** ecriture du nom du fichier *****
2540 REM
2550 LPRINT :LPRINT :LPRINT "Fichier ";FI$
2560 A=LEN(FI$)+8
2570 FOR I=1 TO A: Z#=Z#+ "=" : NEXT I: LPRINT Z#
2580 FIC#=FI$+".mar"
2590 OPEN OLD FIC$ AS 1
2600 FIELD #1,5 AS CP#
2610 GET #1,RECORD 1
2620 PRINT CHR$(12)
2630 CP=VAL(CP#)
2640 PRINT "Ce fichier contient ";CP;"donnee";
2650 IF CP>1 THEN PRINT "s" ELSE PRINT
2660 FIELD #1,30 AS A$,30 AS B$,50 AS C$,30 AS D$,5 AS E$,7 AS CC$, 7 AS F#
2670 CURSOR 5,1: INPUT "Quei numero de personnel";NU$
2680 PRINT CHR$(12);
2690 NU=VAL(NU$)
2700 GET #1,RECORD NU+1
2710 LPRINT :LPRINT "fiche n0";NU
2720 A=LEN(STR$(NU)): FOR I= 1 TO A+8: LPRINT "="; : NEXT I: LPRINT
2730 C=VAL(CC#): F=VAL(F#)
2740 D=C*169
2750 RS=D*SS/100
2760 RR=D*CC /100
2770 RC=D*CA/100
2780 TR=RS+RR+RC
2790 SN=D-TR-F
2800 CI=SN*12
2810 REM
2820 REM ***** ecriture des donnees sur imprimante *****
2830 REM
2840 CURSOR 6,23 :LPRINT "      NOM           ";A$
2850 CURSOR 7,23 :LPRINT "      PRENOM        ";B$
2860 CURSOR 9,23 :LPRINT "      ADRESSE       ";C$
2870 CURSOR 10,23 :LPRINT "      VILLE         ";D$
2880 CURSOR 11,23 :LPRINT "      CODE POSTAL  ";E$
2890 LPRINT
2900 CURSOR 14,23 :LPRINT "      Prix de l'heure      :";: LPRINT USING'#####.##',C
2910 CURSOR 15,23:LPRINT "      Retenue S.S.         :";:LPRINT USING'#####.##',RS
2920 CURSOR 16,23:LPRINT "      Retenue Retraite     :";:LPRINT USING'#####.##',RR
2930 CURSOR 17,23:LPRINT "      Retenue chomage      :";:LPRINT USING'#####.##',RC
2940 CURSOR 18,23:LPRINT "      Total des retenues   :";:LPRINT USING'#####.##',TR
2950 CURSOR 19,23:LPRINT "      Salaire brut         :";:LPRINT USING'#####.##',D
2960 CURSOR 20,23:LPRINT "      Accompte             :";:LPRINT USING'#####.##',F
2970 CURSOR 21,23:LPRINT "      Salaire net          :";:LPRINT USING'#####.##',SN
2980 CURSOR 22,23:LPRINT "      Cumul imposable      :";:LPRINT USING'#####.##',CI
2990 LPRINT
3000 Z#=""
3010 FOR I=1 TO 80: Z#=Z#+ "+": NEXT I: LPRINT Z#
3020 CLOSE 1
3030 CURSOR 23,33:PRINT"Autre numero dans le fichier ";FI$;" (O/N)":RE$=INCH$(0)
3040 IF RE$="O" OR RE$="o" THEN PRINT CHR$(12): GOTO 2510
3050 IF RE$="n" OR RE$="N" THEN RETURN
3060 CURSOR 23,33: PRINT CHR$(24):GOTO 3030
3070 REM
3080 REM ***** fin du programme *****
3090 REM
3100 PRINT CHR$(12): CURSOR 12,5
3110 PRINT "Nous esperons que ce programme vous a donne entiere satisfaction"
```



Programmez vos fiches de paye

```
3120 END
3130 REM
3140 REM ***** recherche d'une fiche par critere connu *****
3150 REM
3160 PRINT CHR$(12)
3170 PRINT "++++recherche d'une fiche par critere connu++++"
3180 RESTORE
3190 NSA=0 :BF=0
3200 CURSOR 4,3: PRINT "quel est votre critere de tri"
3210 CURSOR 6,20: PRINT "1 : le nom"
3220 CURSOR 8,20: PRINT "2 : le prenom"
3230 CURSOR 10,20: PRINT "3 : l'adresse"
3240 CURSOR 12,20: PRINT "4 : la ville "
3250 CURSOR 14,20 :PRINT "5 : le code postal"
3260 CURSOR 16,20 :PRINT "6 : le prix de l'heure"
3270 CURSOR 18,20: PRINT "7 : l'acompte"
3280 CURSOR 20,20 :PRINT "8 : RETOUR AU MENU"
3290 CURSOR 22,4: PRINT "votre choix "
3300 CURSOR 22,16: CH#=INCH$(0)
3310 CH=VAL(CH#)
3320 IF CH=(0 OR CH)8 THEN CURSOR 22,16: PRINT CHR$(24) :GOTO 3290
3330 IF CH=8 THEN RETURN
3340 REM
3350 REM ***** choix du critere de tri *****
3360 REM
3370 ON CH GOSUB 3880 ,3890 ,3900 ,3910 ,3920 ,3930 ,3940
3380 REM
3390 REM ***** calcul du field *****
3400 REM
3410 FOR EE=1 TO CH: READ BF :NSA=NSA+BF :NEXT EE
3420 READ NS
3430 PRINT CHR$(12)
3440 PRINT "Quel est ";CH#;: INPUT RE#
3450 R#=RE#: IF CH=6 OR CH=7 THEN R#="+R#"
3460 IF LEN(RE#) <NS THEN R#=R#+": GOTO 3460
3470 INPUT "Quel est le nom du fichier";FI#
3480 FI#=LEFT$(FI#,7)
3490 V#=""
3500 FIC#=FI#+".mar"
3510 OPEN OLD FIC# AS 1
3520 FIELD #1,5 AS CP#
3530 GET #1,RECORD 1
3540 FIELD #1,NSA AS S# ,NS AS U#
3550 CP=VAL(CP#)
3560 FOR WX=1 TO CP
3570 GET #1,RECORD WX+1
3580 IF R#=U# THEN NU=WX: GOTO 3780
3590 NEXT WX
3600 REM
3610 REM ***** cette donnee n'existe pas dans le fichier *****
3620 REM
3630 IF V#="" THEN PRINT CH#;" ";R#;" n'existe pas dans le fichier ";FI#
3640 IF V#(">)" THEN GOTO3700
3650 CLOSE 1
3660 RETURN
3670 REM
3680 REM ***** affichage des numeros des fiches contenant la donnee cherchee ***
3690 REM
3700 IF LEN(V#)>3 THEN DF#="les fiches." ELSE DF#="la fiche"
3710 PRINT CHR$(12) : PRINT : PRINT
3720 PRINT CH#;" ";R#;" est contenu dans ";DF#;" n° ";V#;" du fichier ";FI#
3730 PRINT: PRINT
3740 CLOSE 1 :RETURN
3750 REM
3760 REM ***** affichage d'une fiche contenant la donnee cherchee *****
3770 REM
3780 TP=1: CLOSE 1
3790 PRINT CHR$(12)
3800 V#=V#+STR$(NU)
3810 OPEN OLD FIC# AS 1
```



```
3820 FIELD #1,30 AS A$,30 AS B$,50 AS C$,30 AS D$,5 AS E$,7 AS CC$, 7 AS F$
3830 GOSUB 1010
3840 PRINT "Appuyer sur une touche pour continuer": GOSUB 5510
3850 OPEN OLD FIC$ AS 1
3860 FIELD #1,NSA AS S$ ,NS AS U$
3870 GOTO 3590
3880 CH$="le nom": RETURN
3890 CH$="le prenom": RETURN
3900 CH$="l'adresse" :RETURN
3910 CH$="la ville" :RETURN
3920 CH$="le code postal" :RETURN
3930 CH$="le prix de l'heure": RETURN
3940 CH$="l'acompte" :RETURN
3950 REM
3960 REM ***** modification d'une fiche *****
3970 REM
3980 PRINT CHR$(12) :PRINT "Modification d'une fiche"
3990 INPUT "nom du fichier";FI$
4000 FI$=LEFT$(FI$,7)
4010 FIC$=FI$+".mar"
4020 OPEN OLD FIC$ AS 1
4030 FIELD #1,5 AS CP$
4040 FIELD #1,30 AS A$,30AS B$,50AS C$,30AS D$,5AS E$,7AS CC$, 7AS F$
4050 REM
4060 REM ***** lecture du nombre de fiche dans le fichier *****
4070 REM
4080 GET #1,RECORD 1
4090 CP=VAL(CP$)
4100 CURSOR 5,5: PRINT"n° de la fiche:  "
4110 CURSOR 5,22: INPUT S$
4120 S=VAL(S$)
4130 IF S >CP THEN CURSOR 22,4 :PRINT "Ce fichier ne contient que";CP;"donnee";
4140 IF S>CP THEN IF CP >1 THEN PRINT "s" ELSE PRINT
4150 IF S(<=0 OR S)CP THEN CURSOR 5,22: PRINT CHR$(24): GOTO 4110
4160 GET #1,RECORDS+1
4170 CLOSE 1
4180 F=VAL(F$): C=VAL(CC$)
4190CURSOR 6,23 :PRINT "NOM :";A$
4200 CURSOR 7,23 :PRINT "PRENOM :";B$
4210 CURSOR 9,23 :PRINT "ADRESSE :";C$
4220 CURSOR 10,23 :PRINT "VILLE :";D$
4230 CURSOR 11,23 :PRINT "CODE POSTAL :";E$
4240 CURSOR 14,23 : PRINT "    Prix de l'heure      " :PRINTUSING'#####.##',C
4250 CURSOR 15,23 : PRINT "    Acompte          " :PRINT USING'#####.##',F
4260 PRINT "Appuyez sur une touche pour continuer  "
4270 REM
4280 REM ***** modification de toute la fiche *****
4290 REM
4300 GOSUB 5510
4310 PRINT CHR$(12)
4320 REM
4330 REM ***** choix de la modification *****
4340 REM
4350 CURSOR 4,3 :PRINT "que voulez vous modifier "
4360 CURSOR 6,20: PRINT "1 : le nom"
4370 CURSOR 8,20: PRINT "2 : le prenom"
4380 CURSOR 10,20: PRINT "3 : l'adresse"
4390 CURSOR 12,20 :PRINT "4 : la ville "
4400 CURSOR 14,20: PRINT "5 : le code postal"
4410 CURSOR 16,20: PRINT "6 : le prix de l'heure"
4420 CURSOR 18,20: PRINT "7 : l'acompte"
4430 CURSOR 20,20 :PRINT "8 : toute la fiche"
4440 CURSOR 22,20 :PRINT "9 : RETOUR AU MENU"
4450 CURSOR 23,4: PRINT "votre choix  "
4460 CURSOR 23,16 :PRINT "?": CH$=INCH$(0)
4470 CH=VAL(CH$)
4480 IF CH(<=0 OR CH)9 THEN CURSOR 23,16: PRINT CHR$(24) :GOTO 4460
4490 IF CH=9 THEN RETURN
4500 PRINT CHR$(12)
4510 IF CH=8 THEN GOTO 4520 ELSE GOTO 5150
```

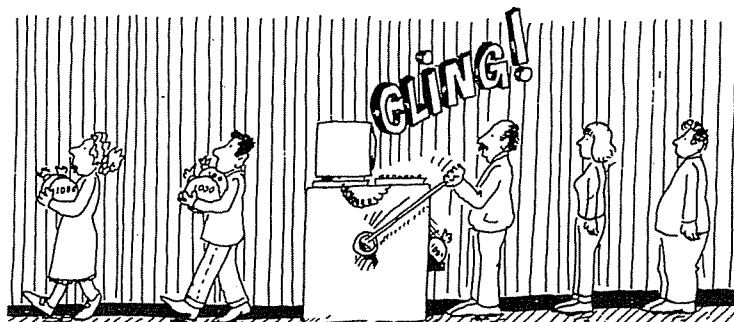


Programmez vos fiches de paye

```
4520 I=S
4530 OPEN OLD FIC$ AS 1
4540 PRINT CHR$(12)
4550 FIELD #1,5 AS CP$
4560 FIELD #1,30 AS A$,30 AS B$,50 AS C$,30 AS D$,5 AS E$,7 AS CC$, 7 AS F$
4570 CURSOR 4,20: PRINT "Nom:";: GOSUB 2380: LSET A$=RE$
4580 CURSOR 5,20: PRINT "Prenom:";: GOSUB 2380: LSET B$=RE$
4590 CURSOR 6,20: PRINT "Adresse ";: GOSUB 2380: LSET C$=RE$
4600 CURSOR 7,20: PRINT "Ville ";: GOSUB 2380: LSET D$=RE$
4610 CURSOR 8,20: PRINT "Code postal:";: GOSUB 2380
4620 IF VAL(RE$)=0 OR VAL(RE$)99999 THEN CURSOR 8,20: PRINT CHR$(24): GOTO 4610
4630 LSET E$=RE$
4640 CURSOR 11,10: PRINT "NOMBRES D'HEURES:169"
4650 CURSOR 12,10: INPUT "prix de l'heure ";:CE$
4660 C=VAL(CE$): IF C<=0 OR C>1000 THEN CURSOR 12,10: PRINT CHR$(24) :GOTO 4650
4670 CURSOR 13,10: INPUT "ACCOMPTE:";:CE$
4680 F=VAL(CE$): IF F<=0 OR F>D THEN CURSOR 13,10: PRINT CHR$(24): GOTO 4670
4690 D=C*169
4700 D=D-D*(SS+CA+CC)/100
4710 LSET CC$=STR$(C): LSET F$=STR$(F)
4720 PUT #1,RECORD I+1
4730 CLOSE 1
4740 RETURN
4750 PRINT CHR$(12)
4760 RESTORE
4770 REM
4780 REM ***** destruction d'une fiche *****
4790 REM
4800 PRINT "+++++ destruction d'une fiche +++++"
4810 INPUT "nom du fichier";FI$
4820 FI$=LEFT$(FI$,7)
4830 FIC$=FI$+".mar"
4840 OPEN OLD FIC$ AS 1
4850 FIELD #1,5 AS CP$
4860 GET #1, RECORD 1
4870 CP=VAL(CP$)
4880 CURSOR 5,5: PRINT "n° de la fiche:  "
4890 CURSOR 5,22: INPUT S$
4900 IF S >CP THEN CURSOR 22,4 :PRINT "Ce fichier ne contient que";CP;"donnee";
4910 IF S >CP THEN IF CP>1 THEN PRINT "s" ELSE PRINT
4920 S=VAL(S$): IF S<=0 OR S >CP THEN CURSOR 5,22: PRINT CHR$(24): GOTO 4880
4930 PRINT "Etes vous sur de vouloir detruire la fiche n°";S;"du fichier ";FI$;
4940 PRINT "(O/N) ": RE$=INCH$(0)
4950 IF RE$="o" OR RE$="O" THEN GOTO 4980
4960 IF RE$="n" OR RE$="N" THEN RETURN
4970 PRINT "repondez par O ou N ": GOTO 4930
4980 IF CP=1 THEN KILL FIC$:RETURN
4990 FIELD #1,30 AS A$,30 AS B$,50 AS C$,30 AS D$,5 AS E$,7 AS CC$, 7 AS F$
5000 FOR I=S TO CP-1
5010 GET #1,RECORD I+2
5020 PUT #1,RECORD I+1
5030 NEXT I
5040 LSET CP$=STR$(CP-1)
5050 PUT #1,RECORD 1
5060 CLOSE 1
5070 RETURN
5080 REM
5090 REM ***** data pour les fields *****
5100 REM
5110 DATA 0, 30, 30, 50, 30, 5, 7, 7
5120 REM
5130 REM ***** modification de la fiche (suite) *****
5140 REM
5150 ON CH GOSUB 5440, 5450, 5460, 5470, 5480, 5490, 5500
5160 OPEN OLD FIC$ AS 1
5170 REM
5180 REM ***** calcul du field *****
5190 REM
5200 RESTORE: NSA=0 :FOR EE=1 TO CH: READ BF: NSA=NSA+BF: NEXT EE
5210 READ NS
```




```
5220 FIELD #1,NSA AS S$,NS AS U$
5230 GET #1,RECORD S+1
5240 ST$=S$
5250 PRINT CHR$(12)
5260 REM
5270 REM ***** affichage de l'ancienne donnee *****
5280 REM
5290 CURSOR 10,10: PRINT CH$; " : "; U$
5300 CURSOR 11,10 :PRINT NN$;
5310 REM
5320 REM ***** entree de la nouvelle donne *****
5330 REM
5340 CURSOR 11,(10+LEN(NN$)+1): INPUT RE$
5350 IF (CH=6 OR CH=7) AND VAL(RE$)<=0 THEN GOTO 5340
5360 LSET S$=ST$: LSET U$=RE$
5370 PUT #1,RECORD S+1
5380 CLOSE 1
5390 PRINT "Avez vous termine vos modifications sur cette fiche (o/n)"
5400 RE$=INCH$(0)
5410 IF RE$="o" OR RE$="O" THEN RETURN
5420 IF RE$="n" OR RE$="N" THEN GOTO 4310
5430 GOTO 5390
5440 CH$="ancien nom" : NN$="nouveau nom": RETURN
5450 CH$="ancien prenom": NN$="nouveau prenom": RETURN
5460 CH$="ancienne adresse": NN$="nouvelle adresse": RETURN
5470 CH$="ancienne ville": NN$="nouvelle ville": RETURN
5480 CH$="ancien code postal": NN$="nouveau code postal": RETURN
5490 CH$="ancien prix de l'heure": NN$="nouveau prix de l'heure": RETURN
5500 CH$="ancien accompte": NN$="nouvel accompte": RETURN
5510 A$=INCH$(0)
5520 RETURN
5530 REM
5540 REM ***** destruction d'un fichier *****
5550 REM
5560 PRINT CHR$(12)
5570 INPUT "nom du fichier a detruire";FI$
5580 FI$=LEFT$(FI$,7)
5590 PRINT"etes vous sur de vouloir detruire ";FI$;" (O/N)": RE$=INCH$(0)
5600 IF RE$="o" OR RE$="O" THEN GOTO 5630
5610 IF RE$="n" OR RE$="N" THEN RETURN
5620 PRINT"repondez par O ou N ": GOTO 5590
5630 FIC$=FI$+".mar"
5640 KILL FIC$
5650 RETURN
5660 REM
5670 REM ***** changement du nom du fichier *****
5680 REM
5690 PRINT CHR$(12)
5700 INPUT "ancien nom du programme";FI1$
5710 FI1$=LEFT$(FI1$,7)
5720 FIC1$="1."+FI1$+".mar"
5730 INPUT"nouveau nom du programme";FI2$
5740 FI2$=LEFT$(FI2$,7)
5750 FIC2$="1."+FI2$+".mar"
5760 RENAME FIC1$ , FIC2$
5770 RETURN
```



ENCORE DES COURBES MATHÉMATIQUES

Le but de cet article est de proposer un schéma général de programme de tracé de courbes à partir d'une équation. Cet exercice classique, cauchemar des élèves, a souvent été programmé sur microordinateur. Le SBASIC du Goupil permet de résoudre de façon élégante les problèmes particuliers qui se posent dans la réalisation du programme.

Il est extrêmement facile de programmer le tracé d'une courbe sur tout ordinateur possédant un basic et un graphisme suffisamment fin (programme 1) : la boucle sur X est destinée à balayer l'écran de gauche à droite ; Les lignes 40 et 60 opèrent le changement de variable nécessaire pour visualiser correctement la courbe sinus et la ligne 50 définit la fonction.

Si on désire modifier l'échelle du tracé, il suffit de modifier les lignes 40 et 60. On peut bien entendu améliorer ce programme en demandant à l'utilisateur l'échelle désirée (programme 2). Il est ainsi possible de placer l'origine du repère en un point quelconque de l'écran et de choisir l'échelle pour chacun des deux axes.

Les problèmes sérieux commencent lorsqu'on désire changer de fonction. On peut bien entendu remplacer la ligne 50 mais cela conduit à une modification du programme lui-même à chaque fois qu'on désire étudier une nouvelle fonction.

Ce procédé est peu élégant, de plus il interdit pratiquement la superposition de deux courbes sur le même graphique et d'autres améliorations.

L'idéal est de pouvoir entrer l'équation de la fonction sous forme d'une chaîne de caractères, et de la traiter ensuite comme une fonction. Ce n'est pas facile, car il y a une différence fondamentale entre l'ordre basic $Y = \text{SIN}(X)$, codé en mémoire avec les codes spécifiques au basic utilisé, et la chaîne de caractères "Y = SIN(X)", codée en ASCII.

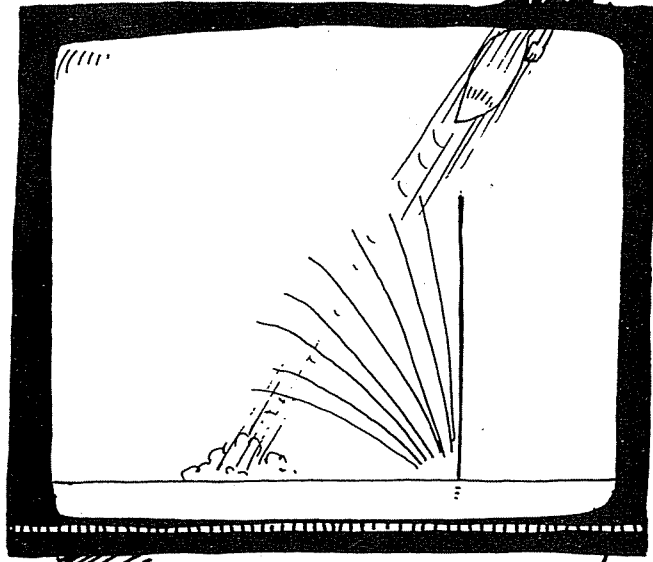
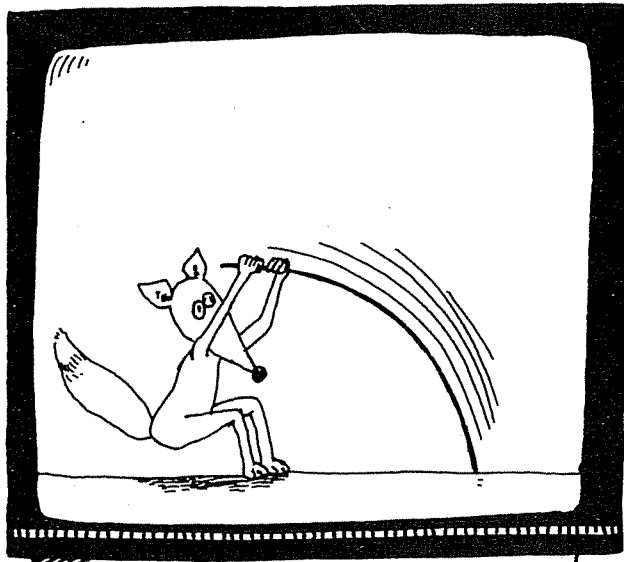
A ce problème, il y a plusieurs solutions :

— ou bien on écrit un programme permettant de transformer la chaîne de caractères en ordre basic : ceci impose de connaître parfaitement le codage interne du basic utilisé et l'organisation mémoire de l'ordinateur.

— ou bien on incorpore la chaîne de caractères dans un fichier disque qui n'est autre que le programme du tracé que l'on exécute ensuite automatiquement. Ce procédé nécessite des accès disques assez longs.

— ou bien on utilise une particularité du SBASIC : l'ordre EXECUTE. Cela donne le programme n° 3 : l'équation est entrée à la ligne 95 comme chaîne de caractères et EXECUTÉE comme un ordre basic à la ligne 50

Jean Saquet





Programme 1

```

5 COLOR 0
10 GR
20 COLOR 1
30 FOR X=0 TO 255
40 X1=(X-128)/40
50 Y1=SIN(X1)
60 Y=INT(128-Y1*120)
70 PLOT X,Y
80 NEXT X
90 END

```

Programme 2

```

2 GOSUB 100
5 COLOR 0
10 GR
20 COLOR 1
30 FOR X=0 TO 255
40 X1=(X-X0)/E1
50 Y1=SIN(X1)
60 Y=INT(255-Y0-Y1*E2)
70 PLOT X,Y
80 NEXT X
90 END
100 INPUT "coordonnees du centre (x0,y0) :";X0,Y0
110 INPUT "echelle en nb de points /unité (e1,e2) :";E1,E2
120 RETURN

```

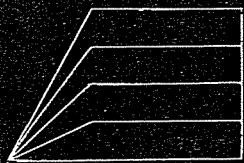
Programme 3

```

2 GOSUB 95
5 COLOR 0
10 GR
20 COLOR 1
30 FOR X=0 TO 255
40 X1=(X-X0)/E1
50 EXECUTE A#
60 Y=INT(255-Y0-Y1*E2)
70 PLOT X,Y
80 NEXT X
90 END
95 INPUT "fonction à tracer (Y1=F(X1)) :";A#
100 INPUT "coordonnees du centre (x0,y0) :";X0,Y0
110 INPUT "echelle en nb de points /unité (e1,e2) :";E1,E2
120 RETURN

```

READY



POLYPHOT

17 rue de la plaine
75020 PARIS
Tel: 373 81 28

POLYDOC logiciel de recherche documentaire

DIDEROT logiciel de gestion automatisé de bibliothèque



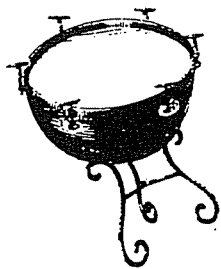
POSSIBILITE D'ADAPTATION SUR CAHIERS DES CHARGES

labo · sciences informatique

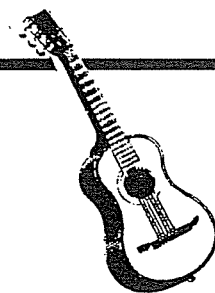
6, rue Saint Dominique 75007 Paris 705.98.89

- Applications scientifiques et techniques sur GOUPIL
- Analyse et programmation en physique, chimie, biologie
- Assistance totale matériel et logiciel.



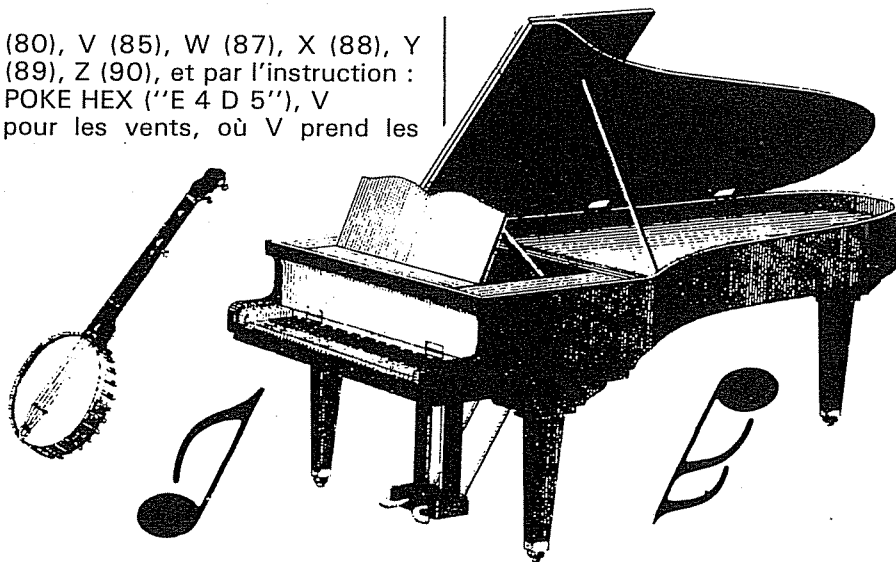


LE BASIC MUSICAL DE GOUPIL 2



Lors de la description du "clavier musical" en langage Basic pour Goupil 2, les commandes de « TIMBRES » et d'« INSTRUMENTS » n'ont pas été fournis. Dans le petit programme ci-joint — en quelques lignes d'essais — vous constaterez qu'il est nécessaire d'adresser directement la mémoire par l'instruction : POKE HEX ("E4 D4"), T pour les timbres, où T peut prendre neuf valeurs qui sont les valeurs ASCII codées décimal des lettres G (71), H, (72), J (74), P

(80), V (85), W (87), X (88), Y (89), Z (90), et par l'instruction : POKE HEX ("E 4 D 5"), V pour les vents, où V prend les



```

10 EXEC, "TTYSET PS=N"
20 REM ***** GOUPIL-REVUE --- Rubrique musicale *****
30 REM *****Programme d'essai des TIMBRES musicaux de GOUPIL 2 *****
40 REM ***** Quelques notes de "La lettre a Elise" pour essais *****
50 REM
60 DIM A(9)
70 FOR I=1 TO 9:READ A(I):NEXT I      :REM Mise en tableau des timbres
80 FOR V=0 TO 1                      :REM Instrument : 0=vent , 1=cordes
90 FOR J=1 TO 9
100 FOR Z=0 TO 1                     :REM Changement d'octave
110 T=A(J)
120 PRINT CHR$(12)
130 PRINT "TIMBRE = ";T
140 POKE HEX("E4D4"),T              :REM Adresse memoire du TIMBRE
150 IF V=0 THEN PRINT "Vent = 0"
160 IF V=1 THEN PRINT "cordes = 1"
170 POKE HEX("E4D5"),V              :REM Adresse memoire de l'instrument
180 TEMPO=190
190 PRINT "TEMPO = 190":REM Tempo fixe a 100 pour l'essai
200 FOR I=1 TO 19
210 READ A$,K,P$,A                  :REM Lecture de la " partition "
220 PRINT "Octave = ";K+Z:PRINT CHR$(11);
230 K=K+Z:NOTE=A$,K,P$,A           :REM Interpretation
240 NEXT I
250 RESTORE 330                     :REM Reinitialisation de la "partition"
260 NEXT Z:NEXT J:NEXT V
270 PRINT:PRINT:PRINT
280 PRINT "**** FIN D'ESSAI ****":STOP
290 REM ***** Valeur des 9 timbres *****
300 REM ***** (dec. ASCII) de G, H, J, P, U, W, X, Y, Z *
310 DATA 71, 72, 74, 80, 85, 87, 88, 89, 90
320 REM ***** Extrait de partition *****
330 DATA "MI", 4, "D", 0, "RE", 4, "D", 1, "MI", 4, "D", 0, "RE", 4, "D", 1
340 DATA "MI", 4, "D", 0, "SI", 3, "D", 0, "RE", 4, "D", 0, "DO", 4, "D", 0
350 DATA "LA", 3, "C", 0, "SL", 3, "D", 0, "DO", 3, "D", 0, "MI", 3, "D", 0
360 DATA "LA", 3, "D", 0, "SI", 3, "C", 0, "SL", 3, "D", 0, "MI", 3, "D", 0
370 DATA "so", 3, "d", 1, "si", 3, "d", 0, "do", 4, "c", 0
    
```



valeurs Ø pour l'"instrument à vent", 1 pour l'"instrument à cordes".

Remarque d'un utilisateur de Musigoupil

Lors de l'écriture d'une partition

musicale, pour éviter les passages qui doivent être joués deux fois consécutives, l'élément de partition à rejouer se termine par un symbole caractéristique (voir exemple).

Or, pour l'écriture avec Musigou-

pil, il n'est pas prévu de réécriture de phrase musicale automatique : il faut tout réentrer au clavier. Le problème est à l'étude. Mais si un lecteur peut proposer une solution, l'outil qu'est Musigoupil en serait grandement amélioré. Merci de bien vouloir nous écrire.

PARISETTA

POLKA

FRANCIS POPY

Gérard Bétouche

PETITE FLÛTE R^b
(ou Violon monté en Ré)

Op. 1 T^o di Polka

Exemple :
Phrase à ré-écrire

Exemple :
Phrase à ré-écrire



Centre d'affaires LES NATIONS Boul. de l'Europe
54500 VANDOEUVRE Tél. (8) 356.89.57

NOGEDENT
INFORMATIQUE

NOGEDENT
Programme de gestion dentaire. Gestion complète de cabinet, patients, soins, facturation, comptabilité, etc...

INFORTHO
Programme d'analyses téléradiographiques et d'aide au pré-diagnostic pour orthodontistes.

DAMOG
Programme de dessin assisté par ordinateur avec ou sans table à digitaliser pour dessins en couleur, vecteurs, points, modifications, déplacements, cercles, diagrammes, (création d'images utilisables sous flex ou BASIC), etc...

NOGEFICHE
Programme de gestion de fichiers, saisie (fiches de longueur variable), modification, tris, possibilités de calculs, éditions paramétrables, etc...

NOGETEXTE
Programme de traitement de texte avec des fonctions supplémentaires du type calculs, augmentations, cumuls, recherche, remplacement automatique de mot, etc...

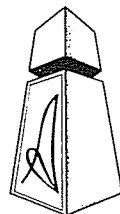
HARDCOPY
Programme de recopie d'écrans couleurs sur imprimante graphique (OKI, CENTRONICS, etc...)

ALPHAMOU
Programme permettant de régénérer de nouveaux caractères qui seront accessibles sous Flex ou BASIC.

SINSTALL
Programme qui permet de supprimer les problèmes d'accoutumation sur Goupil 3 et sur imprimante et permet de modifier 30 caractères par touche de fonction.

HORLOGE
Programme vous offrant une base de temps accessible sous saisies analogiques sous 8 bits, 12 bits - Sortie analogique sous 12 bits - Entrées et sorties logiques 5, 12, 24 volts.

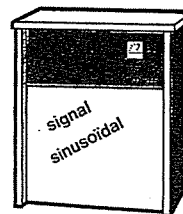
CARTES D'INTERFACES
Saisies analogiques sous 8 bits, 12 bits - Sortie analogique sous 12 bits - Entrées et sorties logiques 5, 12, 24 volts.



A.I. Informatique et Industrie

batec village cidex 8653, tresques 30330 conaux
tel: (66) 89.75.48 usine à 30160 bessèges
Gard France

**ALIMENTATIONS
AUTONOMES
DE
SECURITE**



puissance de 160 VA à 10 KVA

- plus de coupures
- plus de micro-coupures
- plus de courants parasites
- tension et fréquence stables
- une autonomie de plus de 2 heures

BON POUR UNE DOCUMENTATION GRATUITE
Complétez et retournez ce bon à l'adresse ci-dessus

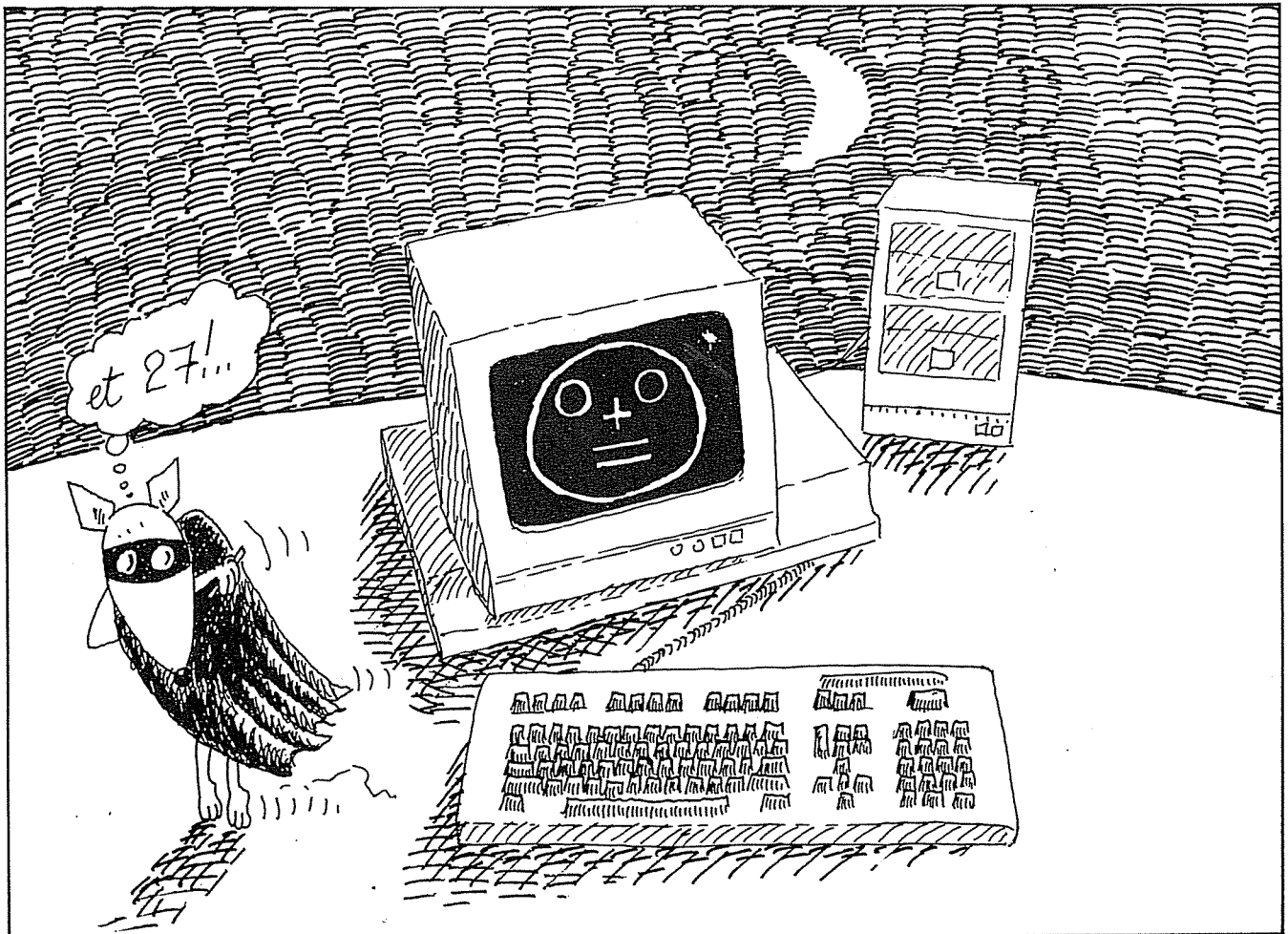
Nom _____ Société _____

Adresse _____ Tél. _____

Code postal _____ Localité _____



GENECAR : générateur de caractères



Voici un programme grâce auquel vous pourrez créer 26 nouveaux caractères graphiques dans une matrice 8*12, ce qui permet une résolution théorique de 300*640 points avec un GOUPIL 3 et un écran normal 25*80 (on est en fait limité par le faible nombre de caractères programmables).

Ces caractères remplaceront les anciens pavés graphiques de code 65 à 90. De plus, ils pourront être sauvegardés dans un fichier disque puis relus grâce à la partie du programme

comprise entre les lignes 1710 et 2150.

Explication du principe

Le caractère est dessiné dans une matrice 8*12, puis il est codé par groupes de 6 bits, chaque bit à 1 représentant un point allumé. Il est ensuite ajouté 64 à chaque groupe de 6 bits ce qui donne un octet. On

envoie alors, grâce à des "PRINT", un en-tête composé des codes 31, 35, 32. un nombre entre 64 et 127 correspondant au jeu graphique, puis les codes 72, 49, 59, 65, 80, 96, le code ASC du nouveau caractère et les 16 octets de codage.

Pour utiliser les caractères ainsi créés, il faut taper CTRL N puis la touche correspondante au caractère créé, puis CTRL O.

Marc Abramson



```
10 PRINT CHR$(12)
20 PRINT :PRINT :PRINT
30 PRINT CHR$(31)
```



```
40 PRINT "*****"
50 PRINT "*"
60 PRINT "": GENERATEUR DE CARACTERES "*"
70 PRINT "": copyright 1983 goupil revue et MARC ABRAMSON "*"
80 PRINT "*"
90 PRINT "*****"
```

```

100 EXEC , "ttyset ps=n"
110 CURSOR 20,10
120 PRINT "veuillez avoir l'obligeance de mettre votre clavier en majuscule"
130 PRINT "      puis de taper 0"
140 A$=INCH$(0)
150 IF A$ (> "0" THEN GOTO 140
160 PRINT CHR$(12)
170 CLEAR
180 SON$=CHR$(27)+ "Z": SOF$=CHR$(27)+ "Y"
190 INV$=CHR$(27)+ "<" : NO$=CHR$(27)+ "↓"
200 MOU$=CHR$(27)+ "o" : NMOU$=CHR$(15)
210 ON ERROR GOTO 2190
220 DIM NC(15)
230 GOSUB 1510
240 REM
250 REM ***** creation du caractere *****
260 REM
270 PRINT CHR$(12)
280 CURSOR 20,05
290 PRINT "A quelle touche voulez vous faire correspondre le caractere cree ?"
300 S$=INCH$(0)
310 E=ASC(S$)
320 IF E(65 OR E)90 THEN GOTO 350
330 PRINT CHR$(12)
340 CURSOR 8,35 :PRINT SON$: "GENERATION D'UN NOUVEAU CARACTERE"; SOF$
350 CURSOR 10,35 :PRINT "Deplacer votre curseur :touches fleches"
360 CURSOR 11,35 :PRINT "Allumer votre curseur :touche A"
370 CURSOR 12,35 :PRINT "Eteindre votre curseur :touche E"
380 CURSOR 13,35 :PRINT "Recommencer votre caractere:touche EFF"
390 CURSOR 15,35:PRINT"Votre caractere terminé,tapez la touche EXEC"
400 FOR I = 1 TO 8 : CURSOR 3,I*2+10 :PRINT I; :NEXT I
410 FOR J = 1 TO 12
420 CURSOR 3+J,9 :PRINT J; :CURSOR 3+J,12 :PRINT" . . . . . "
430 NEXT J
440 POX =1: POY=1 :POX2=1 :POY2=1
450 CURSOR 3+POY,POX*2+11
460 A$=INCH$(0)
470 A=ASC(A$)
480 REM
490 REM ***** gestion du curseur *****
500 REM
510 IF A=11 THEN POY2=POY-1
520 IF A=29 THEN POX2=POX+1
530 IF A=10 THEN POY2=POY+1
540 IF A=8 THEN POX2=POX-1
550 IF A=219 THEN GOTO 920
560 IF A=223 THEN GOTO 270
570 IF A=65 THEN GOSUB 660
580 IF A=69 THEN GOSUB 790
590 IF POX2(<1 OR POX2 >8 THEN POX2=POX
600 IF POY2(<1 OR POY2 >12 THEN POY2=POY
610 POX=POX2 :POY=POY2
620 GOTO 450
630 REM
640 REM ***** allumage d'un point *****
650 REM
660 POR=(POY-1)*8+POX-1
670 YP=INT(POR/6)
680 XP=POR-YP*6
690 A=NC(YP)
700 EE=5-XP
710 IF(A AND 2>EE)(>0) THEN GOTO 730
720 A=A+2>EE
730 NC(YP)=A
740 CURSOR 3+POY,POX*2+11 :PRINT INV$ ;" " ;NO$
750 RETURN
760 REM
770 REM ***** Extinction d'un point *****
780 REM
790 POR=(POY-1)*8+POX-1
800 YP = INT(POR/6)
810 XP=POR-YP*6
820 A=NC(YP)
830 EE=5-XP

```



```
840 IF A AND 2<EE =2<EE THEN GOTO 850 ELSE GOTO 860
850 A=A-2<EE
860 NC(YP)=A
870 CURSOR 3+POY,POX*2+11 :PRINT". ";
880 RETURN
890 REM
900 REM **mise en place du caractere cree a la place d'un caractere graphique **
910 REM
920 CURSOR 20,5 :FOR I=1 TO 14 :READ A$
930 A=HEX(A$)
940 PRINT CHR$(A);
950 NEXT I
960 PRINT S$;
970 FOR I=0 TO 15
980 PRINT CHR$(NC(I)+64);
990 NEXT I
1000 PRINT CHR$(12) ; "voici le caractere cree: ";MOU$; CHR$(ASC(S$)) ;NMOU$
1010 PRINT "pour l'utiliser taper "; SON$ ;"ctrl N"; SOF$;
1020 PRINT " puis la touche ";S$; " puis ";SON$; "ctrl O";SOF$
1030 REM
1040 REM ***** aiguillage vers un sous-programme *****
1050 REM
1060 PRINT "voulez-vous stocker ce caractere sur un fichier (o/n) ?"
1070 A$=INCH$(0)
1080 IF A$="O" OR A$="o" THEN GOTO 1120
1090 IF A$="n" OR A$="N" THEN GOTO 2400
1100 GOTO 1070
1110 DATA 1F ,23 ,20 ,40 ,48 ,31 ,3B ,31 ,3B ,41 ,50 ,60 ,1F ,23
1120 PRINT "fichier deja cree (o/n) ?"
1130 A$=INCH$(0)
1140 IF A$="O" OR A$="o" THEN GOSUB 1200 :GOTO 2400
1150 IF A$="n" OR A$="N" THEN GOSUB 1460 :GOTO 2400
1160 REM
1170 REM ***** completer un ancien fichier *****
1180 REM
1190 GOTO 1130
1200 INPUT "nom du fichier" ;A$
1210 IF LEN(A$) > 8 THEN A$= LEFT$(A$,8)
1220 RRE$= A$ :A$= A$+".gen"
1230 OPEN OLD A$ AS 1
1240 REM
1250 REM ***** ecriture du caractere sur la disquette *****
1260 REM
1270 FIELD #1,3 AS R$,3 AS Q$,3 AS B$,3 AS C$,3AS D$,3 AS E$,3 AS F$,3 AS G$
1280 FIELD #1,3 AS H$,3 AS I$,3 AS K$,3 AS L$, 3 AS M$, 3 AS N$,3 AS O$, 3 AS P$
1290 LSET R$="0"
1300 LSET Q$= STR$(NC(0)) :LSET B$= STR$(NC(1)) :LSET C$=STR$(NC(2))
1310 LSET D$= STR$(NC(3))
1320 LSET E$= STR$(NC(4)): LSET F$= STR$(NC(5)): LSET G$=STR$(NC(6))
1330 LSET H$= STR$(NC(7))
1340 LSET I$= STR$(NC(8)): LSET J$= STR$(NC(9)) :LSET K$= STR$(NC(10))
1350 LSET L$= STR$(NC(11))
1360 LSET M$= STR$(NC(12)) :LSET N$= STR$(NC(13)) :LSET O$= STR$(NC(14))
1370 SS=ASC(S$)-64
1380 LSET P$= STR$(NC(15))
1390 PUT#1,RECORD SS
1400 PRINT R$
1410 CLOSE 1
1420 RETURN
1430 REM
1440 REM ***** creer un nouveau fichier *****
1450 REM
1460 INPUT"nom du fichier";A$
1470 IF LEN(A$) >8 THEN A$= LEFT$(A$,8)
1480 A$= A$+".gen"
1490 OPEN NEW A$ AS 1
1500 GOTO 1270
1510 PRINT "voulez-vous charger des caracteres a partir de votre disquette ";
1520 PRINT " :tapez 1"
1530 PRINT "ou creer de nouveaux caracteres ";
1540 PRINT " :tapez 2"
1550 A$=INCH$(0)
1560 IF A$="1" THEN GOTO 1620
1570 IF A$="2" THEN RETURN
```




```

1580 GOTO 1550
1590 REM
1600 REM ***** relire des caracteres sur disquette *****
1610 REM
1620 PRINT CHR$(12); :INPUT "nom du fichier" ;A$
1630 IF LEN(A$) >8 THEN A$=LEFT$(A$,8)
1640 RRE$=A$
1650 A$=A$+".gen"
1660 OPEN OLD A$ AS 1
1670 FIELD #1,3 AS R$,3 AS Q$,3 AS B$,3 AS C$,3AS D$,3 AS E$,3 AS F$,3 AS G$
1680 FIELD #1,3 AS H$,3 AS I$,3 AS K$,3 AS L$, 3 AS M$, 3 AS N$,3 AS O$, 3 AS P$
1690 PRINT "tapez la touche correspondante au caractere que vous voulez charger"
1700 PRINT "ou 0 pour charger tout les caracteres"
1710 A$=INCH$(0)
1720 IF A$="0" THEN GOTO 1970
1730 REM
1740 REM ***** charger un caractere *****
1750 REM
1760 IF A$("<"A" OR A$) "Z" THEN GOTO 1710
1770 AA=ASC(A$)
1780 AA=AA-64
1790 GET #1,RECORD AA
1800 IF R$("<"0 " THEN GOTO 2330
1810 NC(0)=VAL(Q$)+64 : NC(1)=VAL(B$)+64 : NC(2)=VAL(C$)+64 : NC(3)=VAL(D$)+64
1820 NC(4)=VAL(E$)+64 : NC(5)=VAL(F$)+64 : NC(6)=VAL(G$)+64 : NC(7)=VAL(H$)+64
1830 NC(8)=VAL(I$)+64 : NC(9)=VAL(J$)+64 : NC(10)=VAL(K$)+64 : NC(11)=VAL(L$)+64
1840 NC(12)=VAL(M$)+64 : NC(13)=VAL(N$)+64 : NC(14)=VAL(O$)+64 : NC(15)=VAL(P$)+64
1850 PRINT CHR$(31); CHR$(35); CHR$(32); CHR$(64); CHR$(72); CHR$(49); CHR$(59);
1860 PRINT CHR$(49); CHR$(59); CHR$(65); CHR$(80); CHR$(96)
1870 PRINT CHR$(31); CHR$(35);
1880 PRINT CHR$(AA+64);
1890 FOR I=0 TO 15
1900 PRINT CHR$( NC(I));
1910 NEXT I
1920 PRINT CHR$(12) ; "voici le caractere cree: ";MOU$; CHR$(ASC(S$)) ;NMOU$
1930 CLOSE 1
1940 PRINT "pour l'utiliser taper "; SON$ ;"ctrl N"; SOF$;
1950 GOTO 2400
1960 PRINT " puis la touche ";A$; " puis ";SON$; "ctrl O";SOF$
1970 PRINT CHR$(31); CHR$(35); CHR$(32); CHR$(64); CHR$(72); CHR$(49); CHR$(59);
1980 REM
1990 REM ***** chargement de tou< les caracteres d'un fichier *****
2000 REM
2010 PRINT CHR$(49); CHR$(59); CHR$(65); CHR$(80); CHR$(96)
2020 FOR I=1 TO 26
2030 GET#1,RECORD I
2040 IF R$("<"0 " THEN GOTO 2140
2050 NC(0)=VAL(Q$)+64 : NC(1)=VAL(B$)+64 : NC(2)=VAL(C$)+64 : NC(3)=VAL(D$)+64
2060 NC(4)=VAL(E$)+64 : NC(5)=VAL(F$)+64 : NC(6)=VAL(G$)+64 : NC(7)=VAL(H$)+64
2070 NC(8)=VAL(I$)+64 : NC(9)=VAL(J$)+64 : NC(10)=VAL(K$)+64 : NC(11)=VAL(L$)+64
2080 NC(12)=VAL(M$)+64 : NC(13)=VAL(N$)+64 : NC(14)=VAL(O$)+64 : NC(15)=VAL(P$)+64
2090 PRINT CHR$(31) ;CHR$(35);
2100 PRINT CHR$(I+64);
2110 FOR T=0 TO 15
2120 PRINT CHR$(NC(T));
2130 NEXT T
2140 NEXT I
2150 GOTO 2350
2160 REM
2170 REM ***** traitement des erreurs *****
2180 REM
2190 IF ERR=4 AND ERL=3070 THEN GOTO 2240
2200 IF ERR=4 AND ERL=6190 THEN GOTO 2290
2210 IF ERR=24 AND ERL=6190 THEN 2330
2220 IF (ERR=24 OR ERR=8) AND ERL=6420 THEN 2350
2230 GOTO 2400
2240 PRINT CHR$(12); :CURSOR 20,10: PRINT "le fichier ";RRE$;" n'existe pas"
2250 FOR TP=1 TO 5000 : NEXT TP
2260 PRINT CHR$(12)
2270 CLOSE 1
2280 RESUME 1060
2290 PRINT CHR$(12); :CURSOR 20,10:PRINT"le fichier ";RRE$;" n'existe pas"
2300 FOR TP= 1 TO 5000 :NEXT TP
2310 PRINT CHR$(12)

```



```

2320 RESUME 15
2330 PRINT "le caractere correspondant a la touche ";A$;" n'existe pas encore "
2340 GOTO 2400

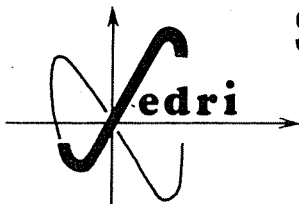
2350 PRINT CHR$(12); "voici les nouveaux caracteres"
2360 FOR K=1 TO 25
2370 PRINT " ";
2380 PRINT MOU$ ;CHR$(64+K); NMOU$
2390 NEXT K

2400 PRINT "voulez-vous creer d'autres caracteres (o/n) ?"
2410 A$=INCH$(0)
2420 IF A$="O" OR A$="o" THEN GOTO 160
2430 IF A$= "n" OR A$="N" THEN END
2440 GOTO 2410

```

LISTE DES VARIABLES

Son\$..... : souligne ON
 Sof\$..... : souligne OFF
 Inv\$..... : inversion video ON
 No\$..... : inversion video OFF
 Mou\$..... : passage en mode alphabet graphique
 Nmou\$..... : retour a l'alphabet normal
 S\$..... : lettre correspondant au nouveau caractere
 E..... : code de S\$
 Pox et Pox2..... : position du curseur en X
 Poy et Poy2..... : position du curseur en Y
 Nc(0) a nc(15).... : codage du nouveau caractere sur 16 octets
 A..... : code de la derniere touche enfoncee lors de la creation du caractere
 Por..... : position absolue du curseur dans la matrice 8*12
 Yp..... : numero de l'octet correspondant a la position du curseur
 Xp..... : numero du bit dans l'octet Yp



SOCIETE D'ETUDE DE DEVELOPPEMENT ET DE RECHERCHE INDUSTRIELLE

3, rue de la Manufacture - B.P. 13 - 02410 - SAINT GOBAIN

Tél. (23) 52 86 87

PRESENTATION DE LA SOCIETE

Son activité consiste à étudier, rechercher et fabriquer tous les produits industriels concernés par la Micro-électronique et par Micro-informatique.

La Société S.E.D.R.I. est composée de 4 secteurs distincts :

1° MICRO-INFORMATIQUE : M. FEUVRIER

Avec distribution du matériel français « Goupil ».

Réalisation et développement de programmes sur demande.

Ce département est supervisé par M. Feuvrier, qui assume la double fonction de technico-commercial et de conseiller auprès de la clientèle réelle et potentielle.

2° MICRO-ELECTRONIQUE : M. HERON

Développement, recherche et fabrication de produits propres ou pour le compte de donneurs d'ordres.

Monsieur Héron supervise l'atelier de fabrication électronique de Rouen et de Saint-Gobain. Il est chargé de la création et du développement de tout procédé électronique inexistant sur le marché.

3° VIDEO, SURVEILLANCE, GARDIENNAGE : M. GUYOT

M. Guyot étudie et adapte aux besoins de chaque client tout système d'alarme et de surveillance à l'aide de matériel vidéo de notre fabrication ou de notre distribution.

4° CONTROLE ET REGULATION - THERMIE, SECURITE : M. DEFORT

Etude et adaptation aux besoins industriels de tout système de contrôle, régulation à l'aide d'un matériel de notre fabrication ou de notre distribution.

Développement autour du GOUPIL

LOGICIEL : gestion PME - gestion LYCEE - gestion CABINET MEDICAL

MATERIEL : carte d'interface industrielle - code barre - connexion périphérique spécifique

Autographe

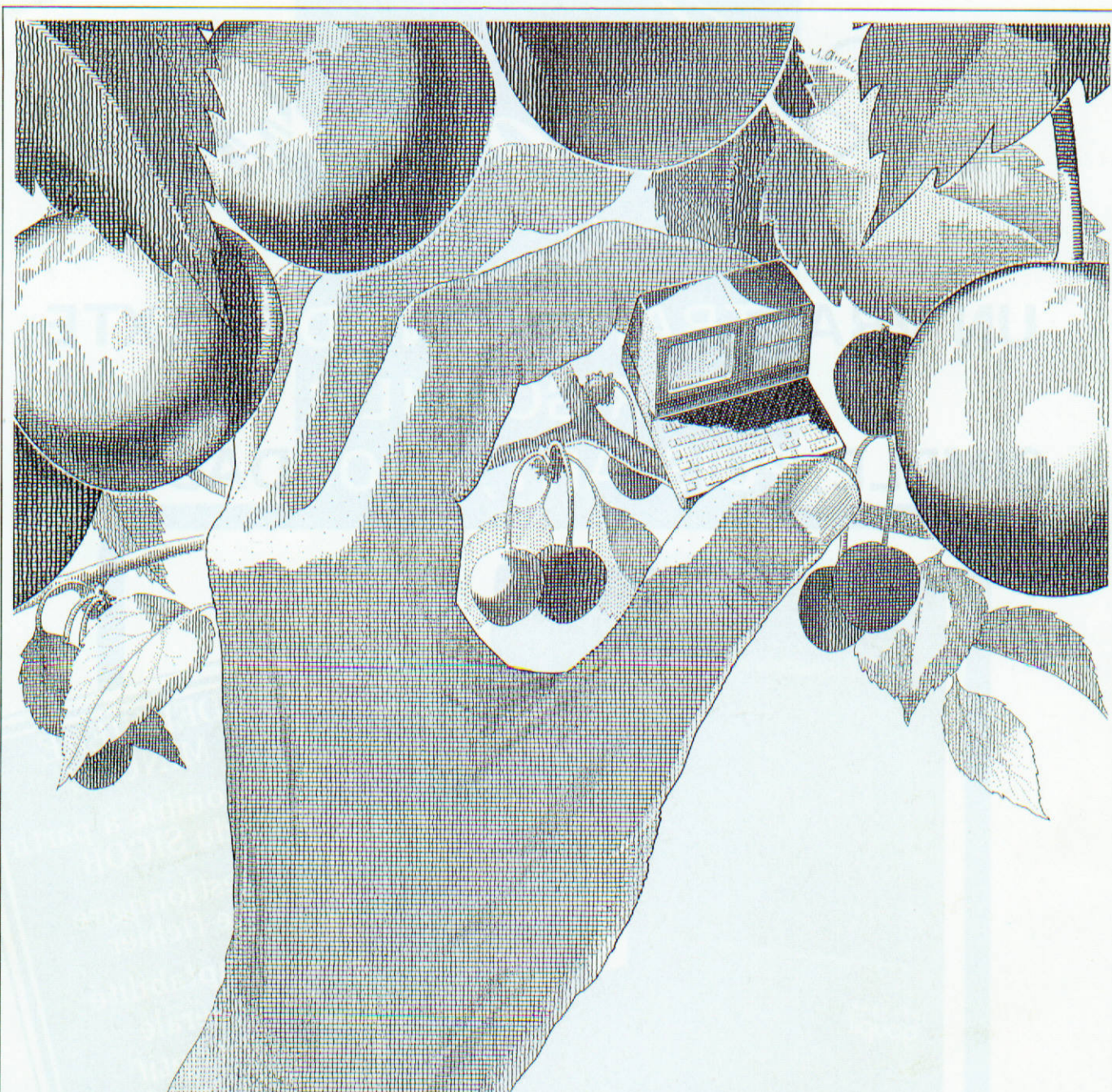
**UN VRAI TRAITEMENT DE TEXTE
POUR GOUPIL 3
SIMPLE, PUISSANT, ABORDABLE...**



**DERNIERE
MINUTE**
Disponible à partir
du SICOB
Gestionnaire
de fichier
Comptabilité
générale
(descriptif
sur demande)

**SOMMA FRANCE
3, rue Ruhmkorff
75017 PARIS**

Tél. (1) 572 17 38 — Télex : 642255



GOUPIL RÉINVENTE LE MICRO ORDINATEUR

Des milliers de personnes utilisent aujourd'hui les micro-ordinateurs Goupil. En moins de trois ans, par le professionnalisme de son réseau de distribution (le premier réseau français de micro-informatique), par l'exigence et l'enthousiasme de ses clients, SMT-Goupil a réussi à bâtir une véritable industrie.

Premier taux de croissance européen, SMT-Goupil franchit aujourd'hui un pas décisif. Avec l'ordinateur Goupil 3, elle ouvre une nouvelle époque de l'informatique en proposant une machine qui réalise la plus étonnante synthèse des développements actuels :

- une machine puissante pouvant utiliser 3 processeurs différents et donc accéder aux plus prestigieux logiciels du marché ;
- une machine évolutive capable de répondre aux besoins spécifiques de chacun ;
- une machine adaptable en permanence aux progrès des technologies ;
- une machine d'un confort d'utilisation exceptionnel et d'une esthétique raffinée.

Ordinateur résolument professionnel, mais aussi ordinateur pour tous, Goupil 3 concrétise la réussite d'un projet à la mesure de notre temps, et d'une entreprise dont l'ambition est de gagner désormais une part significative du marché mondial.

Où que vous soyez, et quels que soient votre profession, vos exigences et vos désirs, Goupil 3, le nouveau fruit de SMT, s'offre à vous ; il est à la portée de votre main.



 **goupil**
L'ORDINATEUR POUR TOUS

SMT Goupil - 22 rue St-Amard, 75015 Paris - Tél. 533.61.39.