

tremplin micro

**Créez vos
caractères
point
par point !**

**Hard-copy
graphique**

**En direct du
microprocesseur**

**Des trucs
pour mieux
maîtriser
votre Apple**

N° 6 - Bimestriel - Première année
9 Janvier - 2 Mars 1986
201 FB - 10 FS - 30 F

Le premier logiciel de **TREMLIN MICRO**
est signé **CLAUDE AUBRY** et c'est :

MINIE

AMPER-INTERPRETEUR _____
_____ BASIC SOURIS



Vous aimeriez en savoir plus... lisez plutôt un extrait de la première page de cet excellent logiciel, et n'hésitez pas à le commander, en utilisant le bulletin de la page 66.

PRÉSENTATION DE MINIE

BONJOUR ! Vous programmez en BASIC ou en assembleur et vous voulez utiliser la souris APPLE simplement, mais efficacement : créer des menus déroulants ou de type boîte, gérer des fenêtres style MACINTOSH, etc. MINIE est faite pour vous... Avec une vingtaine de commandes ajoutées au BASIC ou accessibles depuis l'assembleur, vous réaliserez ce que font les logiciels les plus professionnels et cela sans aucune difficulté.

Pour utiliser MINIE, vous transférerez tout d'abord le programme MINIE, sur la disquette destinée à votre application à l'aide du programme TRANSFERT. Vous incluez dans votre programme HELLO la ligne suivante : PRINT CHR\$(4);"BRUN MINIE."

Dès lors, MINIE sera à votre service. Vous communiquerez avec elle à l'aide d'une interface logicielle : L'AMPER-INTERPRETEUR MINIE-MUM, si vous êtes sous BASIC ou Le Machine Langage Interpreter MICKEY, si vous êtes en assembleur 6502.

Une instruction MINIE débute toujours par la lettre & (AMPERSAND).

Le & est suivi d'un code, c'est-à-dire : UNE LETTRE ou UN MOT décrivant la fonction de la commande. Par exemple W ou WINDOW pour fenêtre. Selon le cas, des paramètres suivent ou non le code, nombres et (ou) chaînes.

Seule la première lettre du code de commande est significative...

PRIX TTC : 170 F _____

tremplin micro

6

Apple et ProDOS (noms et logos) sont des marques déposées d'Apple Computer, Inc.

BIMESTRIEL

Le numéro : 30 F
Abonnement d'un an 180 F
(6 numéros)

EDITIONS JIBENA

Direction-Rédaction :
Editions JIBENA

Guy-HACHETTE

La Petite Motte — Senillé
86100 CHÂTELLERAULT.

Téléphone :
49-93-66-66

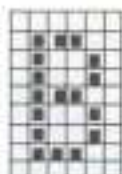
PUBLICITÉ :
Joëlle (même numéro)

Commission paritaire :
Demande en cours.

Les revues qui choisissent d'être réellement au service du Lecteur, en ne l'obligeant pas à glaner, dans plusieurs magazines, les renseignements concernant sa machine, ne bénéficient pas du numéro de Commission Paritaire, et pas davantage des tarifs postaux réduits.

TREMPLIN MICRO — Bimestriel — C'est une publication des Editions JIBENA, 4, rue de la Cour-des-Neues, 75020 PARIS — S.A. au capital de 3 600 000 F — Imprimé par CITÉ-PRESS/PARIS — Dépôt légal à la date de parution — Inscription à la Commission Paritaire des Publications et Agences de Presse : en cours — Directeur de la Publication : Guy-Clément COGNÉ — Diffusion N.M.P.P.

Meilleurs vœux pour 1986 !



ONNE année, amie Lectrice, et à vous aussi, ami Lecteur... et bonne programmation, sur votre Apple ! Il vaut mieux s'adonner à cette passionnante activité que de dire du mal de son voisin ! non ? Justement, parlons de notre plus proche... voisin : Apple.

Quelle(s) surprise(s), évidemment agréable(s) nous réservent donc les dirigeants de Cupertino ? Noël 85 a vu les prix tomber à leur niveau le plus bas et l'Apple IIc restait, sur un marché dominé par les compatibles IBM, une bonne affaire et un bon choix. Peut-on lui promettre une longue vie ? J'en doute. Contrairement à ce que l'on pensait chez Apple, les machines fermées n'ont guère d'avenir, actuellement du moins. L'utilisateur, lorsqu'il refuse le prêt-à-porter informatique, aime se bidouiller un système à ses mesures... et il a donc besoin d'un nombre raisonnable de slots d'extension. Ce sont eux qui ont permis à l'Apple II de devenir une machine universelle... dont le géant IBM s'est largement inspiré lorsqu'il a décidé de lancer son PC. Et je parie que Mac aurait doublé le chiffre de ses ventes s'il avait été équipé, comme son aîné, de quelques fentes destinées à autant de cartes... D'ailleurs, après avoir persévéré dans l'erreur, Apple y vient. Mais laissons Mac à nos confrères spécialisés !

Quand naîtra l'Apple IIx (il paraît qu'il conservera ses chiffres romains), celui que vous et moi attendons avec curiosité... et intérêt ? Je ne suis pas dans le secret des dieux, et je l'ignore. Il semble, si j'en crois des milieux généralement bien informés, que la date de son annonce soit beaucoup plus proche que ne le laissent entendre les dirigeants d'Apple (d'ailleurs muets sur la question, je le reconnais). Ce sera donc un 16 bits, présentant l'éminente qualité d'émuler l'Apple II actuel (équipé d'un 65C02), mais réellement différent de son ancêtre : clavier séparé, bloc numérique d'origine, écran couleur, unidisk, emplacement pour disque dur... Presque un Mac, mais avec la philosophie que nous apprécions dans l'Apple II, ce qui n'exclut nullement l'utilisation de la souris et la multiplication des fenêtres.

Ah ! j'oubliais : l'Apple IIx sera livré, d'origine, avec un autre Basic, comprenant des tas d'instructions nouvelles... mais qu'il faudra charger (que l'on se rassure : avec la SOFTCARD, ce sera instantané !). Adieu la ROM ! Quant aux programmes de *Tremplin Micro*, je vous promets qu'ils ne tourneront plus autour du "Print using", des "Transferts d'écrans"... et autres astuces ; nous aurons mieux à faire avec de merveilleux petits logiciels, utilisant et la souris... et toutes les fonctions qui y sont rattachées.

Vous pensez que je rêve ? Peut-être. L'avenir le dira, mais comptez en tout cas sur *Tremplin Micro*, pendant toute cette année, pour vous aider à mieux connaître et utiliser votre Apple IIe, IIc... ou IIx. Ce n'est pas demain que Guy-Hachette et ses collaborateurs pourront abrégier leurs longues et captivantes veillées !

Encore mes meilleurs vœux, et bonne programmation !

GUY-HACHETTE.

INITIATION

- Trajet minuté
(une routine créée à la demande
d'un lecteur de Tremplin Micro) 9
- Point par point 12
- Saisie de mots 51
- Multiplication en assembleur 62

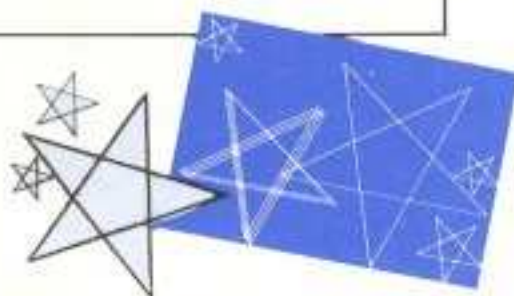
LOGICIELS

- Le Jongleur de mots, un excellent
traitement de texte 39
- Mouse Desk indispensable avec
l'Unidisk 63
- L'anglais sur Apple avec Assimil .. 63

DES TRUCS

- Inverse-normal ou comment vous
passer, dans un programme, des
instructions INVERSE et NORMAL,
remplacées par CTRL-I et CTRL-N 40
- SAUVÉ après un NEW
malencontreux 59

POINT PAR POINT



Où il est démontré qu'il est facile, sur une imprimante ImageWriter ou sur une DMP, d'éditer les pages graphiques de l'Apple 12



ET LE PROGRAMME IMP.HGR.1

Il fonctionne sur imprimante DMP, ImageWriter, Mannesmann et autres compatibles 17

GRAPHISME

- Animez vos triangles 3
- Point par point 12
- Le programme IMP.HGR.1 17
- GROSSIR ? par B. GOURC 35

CARACTÈRES PERSONNALISÉS

PAGE 24

- Un programme pour DMP, ImageWriter et ImageWriter II
- Une première application :
En créant 8 caractères spéciaux, tous les encadrés deviennent possibles.

UTILITAIRES

- VISU-PUCE 41
- Conversion : Décimal — Hexadécimal
— Binaire 53

VISU-PUCE • par Maurice Chavelli •

Pour voir ce qui se passe dans votre Apple lorsque "tourne" un programme 41

LISEZ AUSSI :

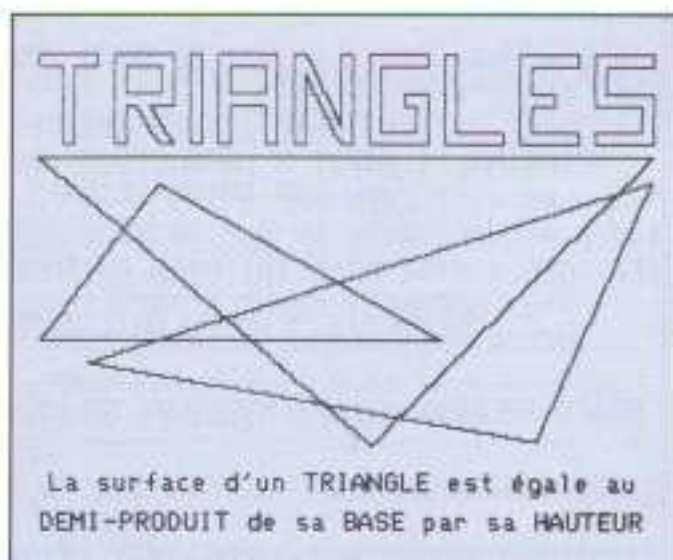
- Votre bibliothèque
informatique 8 — 34 — 64
- Nouveaux matériels
 - Une imprimante Mannesmann
compatible Apple 65
 - ImageWriter II 52
 - Unidisk 3.5 52
- La conversion ?
c'est dans la poche ! 55
- Votre fiche n°6 21 — 22
- Le courrier des lecteurs
et les nombreuses réponses
apportées par Tremplin Micro 57
- Bulletin de commande
et d'abonnement 66

TREMPIN MICRO

Le numéro 7 paraîtra le 3 mars.

ANIMEZ VOS TRIANGLES !

C E programme se veut "éducatif", mais il n'a pas la prétention d'être un modèle du genre. La courte "leçon" qu'il présente comporte bien des lacunes et bon nombre de nos lecteurs enseignants voudront sans doute le compléter. La page graphique de présentation est entièrement réalisée en utilisant la commande HLOT, mais en langage machine. Nous n'avons pas jugé utile d'en publier le programme source. Il suffit de lister le sous-programme binaire pour en comprendre facilement le fonctionnement.



LOMEM protège votre routine en langage machine ➔
Affichage présentation ➔
Rappel utile, sous le graphisme ➔

```
10 LOMEM: 5600: PRINT CHP# (4)*BLDAD TRIBIN,A4700":
   REM #125C (LONGUEUR 858)
20 TEXT : NORMAL : PRINT CHR# (21): HOME :Z = 0
30 GOTO 550
40 VTAB 21: PRINT " La surface d'un TRIANGLE est égal
   e au DEMI-PRODUIT de sa BASE par sa HAUTEUR"
```

PENDANT L’AFFICHAGE DE LA PAGE GRAPHIQUE, VOUS PRÉPAREZ LA SUITE EN MODE "TEXT"

Vers la sous-routine d'attente ➔

```
50 INVERSE : FOR I = 5 TO 15: VTAB I: HTAB 5: PRINT "
   "; HTAB 36: PRINT " "; IF I = 5 OR I = 15 THEN F
   OR J = 6 TO 35: VTAB I: HTAB J: PRINT " "; NEXT
60 NEXT
70 GOSUB 480: NORMAL : GOSUB 470: TEXT
80 VTAB 8: HTAB 9: PRINT "DONNEZ LES DIMENSIONS DU":
   VTAB 10: HTAB 9: PRINT "TRIANGLE QUE VOUS DESIREZ
   ": VTAB 12: HTAB 9: PRINT "OBSERVER SUR CET ECRAN..
   "
90 NORMAL : HCOLOR= 2: HLOT 2,2: CALL - 3082: GOSUB
   470
```

Effacement invisible de la page graphique. ➔

LA SAISIE DES DONNÉES COMMENCE ICI

```
100 CALL - 198: VTAB 21: INPUT "QUELLE EST SA BASE (2
   0 A 279) -> ";R#: IF R# = "" THEN GOSUB 470: GOTO
   420
```

```

110 B = VAL (R$): IF B < 20 OR B > 279 THEN 100
120 GOSUB 470
130 CALL - 190: VTAB 21: INVERSE : PRINT "BASE = "B;:
    NORMAL : INPUT " SA HAUTEUR (10 A 150) -> ";R$: IF
    R$ = "" THEN 90
140 H = VAL (R$): IF H < 10 OR H > 150 THEN 130
150 GOSUB 470: VTAB 22: HTAB 8: INVERSE : PRINT "BASE
    =";: NORMAL : PRINT " "B" ";: INVERSE : PRINT "HA
    UTEUR =";: NORMAL : PRINT " "H

```

← Vers effacement

Il est préférable de choisir les plus grandes dimensions pour obtenir un bon résultat.

TRACÉ DE TRIANGLES SUCCESSIFS DE MÊMES BASE ET HAUTEUR

```

160 HCOLOR= 3: H PLOT 2,H TO 2,Z TO B,H TO Z,H: GOSUB 5
    80
170 VTAB 24: HTAB 4: INVERSE : PRINT "PRESSEZ UNE TOUC
    HE SUIVANT VOTRE GRE": NORMAL : VTAB 22: GOSUB 520
180 VTAB 24: HTAB 6: PRINT " LA HAUTEUR RESTE CONSTAN
    TE ";: VTAB 20: PRINT ""
190 FOR I = 7 TO 8 STEP 7: GOSUB 500: GOSUB 510: NEXT
    : GOSUB 500:I = 8: GOSUB 510
200 GOSUB 470: HTAB 5: VTAB 22: PRINT "ICI, C'EST UN T
    RIANGLE ";: INVERSE : PRINT "RECTANGLE": NORMAL : G
    OSUB 480
210 GOSUB 470: HTAB 7: VTAB 22: PRINT "ET VOICI LE MEM
    E... ";: INVERSE : PRINT "ISOCELE": NORMAL
220 HGR : H PLOT B / 2,Z TO Z,H TO B,H TO B / 2,Z: GOSU
    B 480
230 VTAB 7: HTAB 8: PRINT "COMME VOUS LE SAVEZ, CES ":
    VTAB 9: HTAB 8: PRINT "TRIANGLES AVAIENT MEME BASE
    ": VTAB 11: HTAB 8: PRINT "MEME HAUTEUR, MEME ";: F
    LASH : PRINT "SURFACE": NORMAL :
240 GOSUB 470: TEXT : GOSUB 480: HOME

```

← Tracé, puis GOSUB vers l'affichage de la page

← Tracé du triangle isocèle

ON PASSE AUX QUESTIONS DE CONTRÔLE

```

250 T$ = " Q U A T R E   Q U E S T I O N S " : INVERSE :
    VTAB 2: HTAB 5: PRINT T$:T$ = "
    ": VTAB 1: HTAB 5: PRINT T$: VTAB 3: HT
    AB 5: PRINT T$
260 NORMAL : PRINT : HTAB 5: PRINT "Quelle est la bonn
    e définition ?"
270 VTAB 16: INVERSE : PRINT "A";: NORMAL : PRINT " De
    ux COTES EGAUX et deux ANGLES EGAUX qui leur son
    t opposés": PRINT
280 INVERSE : PRINT "B";: NORMAL : PRINT " Un angle DR
    OIT": PRINT
290 INVERSE : PRINT "C";: NORMAL : PRINT " Trois COTES
    EGAUX (angles= 60 degrés)": PRINT
300 INVERSE : PRINT "D";: NORMAL : PRINT " Trois COTES
    INEGAUX"
310 POKE 34,7: POKE 35,15
320 0 = 2:E = 2

```

* Le nombre de questions peut évidemment être modifié, et celles-ci améliorées.

Le programme est perfectible, ô combien !

Réponses en DATA



```
330 Q = Q + 1: READ Q#: READ RB#
340 VTAB 5 + Q * 2: CALL - 198: HTAB 5: INVERSE : PRI
    NT Q#;; NORMAL : PRINT " LETTRE-CLE -> ";; GET R#:
    PRINT R#: IF R# ( ) RB# THEN E = E + 1: GOTO 340
350 VTAB 5 + Q * 2: HTAB 5: PRINT Q#
360 IF Q < 4 THEN 330
370 GOSUB 490: TEXT
```

APPRECIATION... SI L'ON PEUT DIRE

Plus il y aura d'apprécia-
tions... et plus votre pro-
gramme sera long. Evident,
non ?



```
380 VTAB 15: CALL - 958: PRINT : HTAB 5: IF E = 2 THE
    N PRINT " "; FLASH : PRINT "BONNES REPNSES: TRE
    S BIEN!"; GOTO 420
390 INVERSE : IF E = 1 THEN PRINT "UNE LACUNE QUI SER
    A DONC COMBLEE"; GOTO 420
400 IF E < 4 THEN PRINT "UNE BONNE REVISION S'IMPOSE.
    ..": GOTO 420
410 PRINT "CONNAISSANCES TRES INSUFFISANTES"
```

ON CONTINUE... OU ON ARRÊTE ?

Si l'on désire en rede-
mander, c'est tout de
même prévu...



```
420 NORMAL : VTAB 22: CALL - 198:X = 0: GOSUB 460: PR
    INT "ENU DE DISQUETTE ";; GOSUB 460: PRINT "ECOMMEN
    CER ";; GOSUB 460: PRINT "ERMINE ";; GET R#: PRINT
    R#: IF R# = "R" THEN CLEAR : GOTO 20
430 IF R# = "M" THEN PRINT : PRINT CHR# (4)"RUN MENU
    "
440 IF R# ( ) "T" THEN 420
450 HOME : END
460 INVERSE :X = X + 1: PRINT MID# ("MRT",X,1);: NORM
    AL : RETURN
470 VTAB 21: CALL - 958: RETURN
```

Lettres en mode inverse



PETITS SOUS-PROGRAMMES DIVERS

Attente classique
(CTRL-C fonctionne)



```
480 VTAB 24: HTAB 6: PRINT "PRESSEZ UNE TOUCHE QUELCON
    QUE"; VTAB 20: PRINT " "
490 CALL - 198: POKE - 16368,0: WAIT - 16384,128,12
    7: POKE - 16368,0: RETURN
```

Z=0 et on efface



```
500 HCOLOR= 2: HPLLOT 2,H TO 1 - 7,2 TO B,H: HPLLOT 1 -
    7,2 TO 1 - 7,H - 1: RETURN
```

TRACÉ (HCOLOR=3)



```
510 HCOLOR= 3: HPLLOT 1,2 TO 1,H - 1: HPLLOT 2,H TO 1,2
    TO B,H
```

Emission d'un BZZZ



```
520 FOR ZZ = 1 TO 10:X = PEEK (49200): NEXT
```

```
530 X = PEEK (49152): IF X < 128 THEN 530
```

```
540 POKE 49168,0: VTAB 20: PRINT "": RETURN
```

```
550 POKE 8192,0:A# = "2001<2000.3FFEM D823G"
```

```
560 FOR I = 1 TO LEN (A#): POKE 511 + I, ASC ( MID# (
    A#,I,1)) + 128: NEXT : POKE 72,0: CALL - 144
```

Une méthode origi-
nale d'effacement de
la page graphique



Vers le sous-programme
en assembleur



```
570 HGR : TEXT : HCOLOR= 3: CALL 4700: GOSUB 580: GOTO
    40
```

Affichage HGR sans
effacement



```
580 POKE 49236,0: POKE 49235,0: POKE 49239,0: POKE 492
    32,0: RETURN
```

```
590 DATA RECTANGLE,B,ISOCELE,A,SCALENE,D,EQUILATERAL,C
```


TRIBIN (ROUTINE DE PRÉSENTATION)

```
125C:A2 09 A0 00 A9 28 20 57 F4 A9 0E A2
01 A0 28 20 3A F5 A9 96 A0 96 20 3A F5
A9 09 A0 20 20 3A F5 A2 3C A0 00 A9 32 2
0 57 F4 A9 0A A2 00 A0 6E 20 3A F5 A9 B4
A0 6E 20 3A F5 A9 3C A0 32 20 3A F5 A2
0D A0 01 A9 32 20 57 F4 A9 1E A2 00 A0 7
7 20 3A
```

```
12AD:F5 A9 DC A0 95 20 3A F5 A9 0D A2 01
A0 32 20 3A F5 A2 09 A0 00 98 20 57 F4
A9 22 A2 00 A0 00 20 3A F5 A9 22 A0 05 2
0 3A F5 A9 18 A0 05 20 3A F5 A9 18 A0 23
20 3A F5 A9 13 A0 23 20 3A F5 A9 13 A0
05 20 3A F5 A9 09 A0 05 20 3A F5 A9 09 A
0 00 20
```

```
12FE:3A F5 A2 29 A0 00 98 20 57 F4 A9 42
A2 00 A0 00 20 3A F5 A9 42 A0 14 20 3A
F5 A9 38 A0 14 20 3A F5 A9 42 A0 23 20 3
A F5 A9 3C A0 23 20 3A F5 A9 32 A0 14 20
3A F5 A9 2E A0 14 20 3A F5 A9 2E A0 23
20 3A F5 A9 29 A0 23 20 3A F5 A9 29 A0 0
0 20 3A
```

```
134F:F5 A2 2E A0 00 A9 05 20 57 F4 A9 3D
A2 00 A0 05 20 3A F5 A9 3D A0 0F 20 3A
F5 A9 2E A0 0F 20 3A F5 A9 2E A0 05 20 3
A F5 A2 49 A0 00 98 20 57 F4 A9 4E A2 00
A0 00 20 3A F5 A9 4E A0 23 20 3A F5 A9
49 A0 23 20 3A F5 A9 49 A0 00 20 3A F5 A
2 55 A0
```

```
13A0:00 98 20 57 F4 A9 6E A2 00 A0 00 20
3A F5 A9 6E A0 23 20 3A F5 A9 69 A0 23
20 3A F5 A9 69 A0 14 20 3A F5 A9 5A A0 1
4 20 3A F5 A9 5A A0 23 20 3A F5 A9 55 A0
23 20 3A F5 A9 55 A0 00 20 3A F5 A2 5A
A0 00 A9 05 20 57 F4 A9 69 A2 00 A0 05 2
0 3A F5
```

```
13F1:A9 69 A0 0F 20 3A F5 A9 5A A0 0F 20
3A F5 A9 5A A0 05 20 3A F5 A2 75 A0 00
98 20 57 F4 A9 75 A2 00 A0 23 20 3A F5 A
9 7A A0 23 20 3A F5 A9 7A A0 08 20 3A F5
A9 89 A0 23 20 3A F5 A9 8E A0 23 20 3A
F5 A9 8E A0 00 20 3A F5 A9 89 A0 00 20 3
A F5 A9
```

```
1442:89 A0 1C 20 3A F5 A9 7A A0 00 20 3A
F5 A9 75 A0 00 20 3A F5 A2 AE A0 00 98
20 57 F4 A9 95 A2 00 A0 00 20 3A F5 A9 9
5 A0 23 20 3A F5 A9 AE A0 23 20 3A F5 A9
AE A0 0F 20 3A F5 A9 A4 A0 0F 20 3A F5
A9 A4 A0 14 20 3A F5 A9 A9 A0 14 20 3A F
5 A9 A9
```

Si vous êtes néophyte, tapez un CALL—151 (suivi de RETURN) pour passer en mode moniteur. Puis, immédiatement après l'étoile, tapez chacune des séquences suivantes (RETURN après chacune d'elles).

Pour terminer :

BSAVE TRIBIN, A4700, L858
(ou **BSAVE TRIBIN, A\$125C, L858**)

Ensuite, tapez CTRL-C, puis RETURN et, en direct, la formule de contrôle que vous trouverez à la page suivante. Le résultat devra être **96808**.

DEUX ADRESSES

\$F457 **HPL0T**

Trace un point (placer dans X et Y les parties basses et hautes de la coordonnée X du point en question, et dans A la coordonnée Y).

Ainsi, à partir de \$125C,

LXD \$S09

LDY \$S00

LDA \$S28

JSR \$F457

va tracer un point en \$9-\$28 (9-40 en décimal)... ou encore à 9 du bord gauche de l'écran et à 40 du haut de ce même écran).

\$F53A **HLIN**

Trace une ligne, de la même manière. Cette fois, on place le nombre de points horizontaux en A,X (toujours en hexadécimal) et la coordonnée Y dans Y (accumulateur).

Dans notre programme, à partir de l'adresse \$1265, cela donne :

LDA \$S0E] \$10E = 270

LXD \$S01

LDY \$S28 \$28 = 40

JSR \$F53A

Une ligne va aller du point précédent au point situé à 296 du bord gauche et à 40 du haut de l'écran.

Un sous-programme en langage machine paraît terriblement long (celui-ci est élémentaire et ne fait que tracer des lignes successives), mais il n'en est rien. Sans être un chef-d'œuvre (et pas davantage un modèle), l'écran graphique ainsi obtenu occuperait les 33 secteurs d'une disquette. Ici, il ne comporte que 858 octets... 22 lignes de BASIC !

```

1493:A0 1E 20 3A F5 A9 9A A0 1E 20 3A F5
A9 9A A0 05 20 3A F5 A9 AE A0 05 20 3A
F5 A9 AE A0 00 20 3A F5 A2 B5 A0 00 98 2
0 57 F4 A9 BA A2 00 A0 00 20 3A F5 A9 BA
A0 1E 20 3A F5 A9 CE A0 1E 20 3A F5 A9
CE A0 23 20 3A F5 A9 B5 A0 23 20 3A F5 A
9 B5 A0
14E4:00 20 3A F5 A2 D5 A0 00 98 20 57 F4
A9 EE A2 00 A0 00 20 3A F5 A9 EE A0 05
20 3A F5 A9 DA A0 05 20 3A F5 A9 DA A0 0
F 20 3A F5 A9 E4 A0 0F 20 3A F5 A9 E4 A0
14 20 3A F5 A9 DA A0 14 20 3A F5 A9 DA
A0 1E 20 3A F5 A9 EE A0 1E 20 3A F5 A9 E
E A0 23
1535:20 3A F5 A9 D5 A0 23 20 3A F5 A9 D5
A0 00 20 3A F5 A2 0E A0 01 A9 00 20 57
F4 A9 F5 A2 00 A0 00 20 3A F5 A9 F5 A0 1
4 20 3A F5 A9 09 A2 01 A0 14 20 3A F5 A9
09 A2 01 A0 1E 20 3A F5 A9 F5 A2 00 A0
1E 20 3A F5 A9 F5 A0 23 20 3A F5 A9 0E A
2 01 A0
1586:23 20 3A F5 A9 0E A2 01 A0 0F 20 3A
F5 A9 FA A2 00 A0 0F 20 3A F5 A9 FA A0
05 20 3A F5 A9 0E A2 01 A0 05 20 3A F5 A
9 0E A2 01 A0 00 20 3A F5 60

```

BSAVE TRIBIN, A\$125C, L858

FORMULE DE CONTRÔLE (EN DIRECT)

```

$T=0:FOR I=4700 TO 4700+857:T=T+PEEK(I):NEX
T:PRINT T

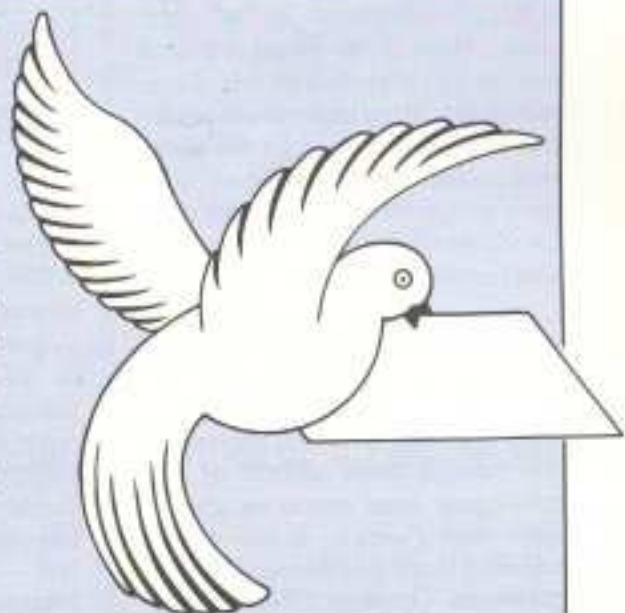
```

Le résultat devra être 96808

Vous nous adressez vos programmes, soit pour qu'ils soient publiés dans *Tremplin Micro*, soit parce que vous butez sur une difficulté apparemment insurmontable.

Bien. Nous sommes vos amis... mais de grâce, joignez toujours à votre envoi la disquette correspondante, avec les sources éventuelles. De plus, si un retour s'impose, n'oubliez jamais que *Tremplin Micro* ne roule ni sur l'or ni sur les timbres-poste !

Alors, amicalement, joignez une enveloppe timbrée et libellée à votre adresse.



Votre bibliothèque INFORMATIQUE

• par NESTOR



• BASIC PLUS... 80 routines pour Apple II*

Michel Martin connaît bien le basic Applesoft et désire communiquer sa science aux programmeurs moins qualifiés que lui : la bonne idée que voilà ! Comment structurer un programme ? Comprendre et utiliser les fonctions logiques et notamment DEF FN ? Deux questions auxquelles Michel Martin répond simplement et efficacement. Mais il va beaucoup plus loin, et l'on n'en finirait pas d'énumérer les titres des divers chapitres et sous-chapitres de cet excellent recueil de programmes, judicieusement structuré... mais oui ! On aimerait que tous les bouquins aient cette tenue. Bravo !

• MATHÉMATIQUES ET GRAPHISME SUR APPLE II**

Marc Ducamp et Alain Reverchon ont associé leurs efforts et leur talent pour nous concocter un bouquin dont j'avoue, à ma grande honte, n'avoir pas recopié les programmes. Chapeau ! Personnelle-

ment, je n'ai qu'un regret, c'est de ne pas pouvoir acheter la disquette... contenant toutes les routines de cet ouvrage. Pourquoi ne pas la commercialiser ? Il est vrai que les auteurs désirent — d'abord et surtout — initier leurs lecteurs aux méthodes de calcul numérique, ce qui passe par une analyse sérieuse des programmes... et donc par une frappe directe, au clavier. Je conseille néanmoins à la librairie Eyrolles de diffuser la disquette : elle le mérite !



• LE TOUT MICRO***

Tout. Vous saurez tout sur la micro-informatique individuelle, et même l'adresse de votre *Tremplin Micro*. L'édition 1985/1986 de cet ouvrage, discret — mais suffisant — par sa pagination, séduisant par la richesse de son illustration (beaucoup de couleur) et de sa mise en pages, indispensable par l'abondance et l'exactitude de sa documentation qui m'a paru plus complète que la précédente, déjà fort intéressante. Matériels, tableaux comparatifs, périphé-

riques, logiciels, livres, adresses... LE TOUT MICRO mérite son titre !



• Basica et GWBASIC Micro-soft**

Je sais : vous possédez un Apple et non un IBM PC, mais est-ce une raison suffisante pour vous désintéresser complètement du GWBASICA (ou BASICA) ? Sûrement pas. L'ouvrage de Bruno Vanryb et Roger Politis tient compte des dernières versions (celles installées sur IBM PC et OLIVETTI M24), mais les instructions les plus récentes sont séparées des autres, ce qui permet de ne pas pénaliser les possesseurs des premières versions. Comme on aimerait trouver, sur Apple, certaines des extensions (et pourquoi pas toutes ?) du BASICA ! Il est vrai qu'il y a le Basic de Mac... mais je préfère, quant à moi, sage Nestor, rêver d'un Apple II... 16 bits !

NESTOR.

* PSI
** Eyrolles
*** Hachette Informatique

TRAJET MINUTÉ

Programme demandé

Il s'agit de noter, pour chaque étape d'un parcours (un rallye... ou un déplacement aérien), le temps prévu.

Le programme va donc additionner les minutes et les heures, fournir, au fur et à mesure de la saisie, le temps total prévu :

- en heures et minutes
- en minutes
- en heures et centièmes de minute.

On peut corriger les données en tapant un "*", voir et revoir le tableau aide-mémoire en tapant un "C"... etc. Tout cela figure dans le programme.

Ajoutons que l'impression des étapes et de leur durée est aussi prévue.

De plus, il n'est pas interdit de modifier et de compléter cette routine, créée à la demande d'un lecteur de *Tremplin-Micro*.

```

100 TEXT : PRINT CHR$(21): HOME
105 DIM T(30),T$(30)
110 FOR I = 1 TO 21:U$ = U$ + " ": NEXT
115 FOR I = 1 TO 11:PT$ = PT$ + "...": NEXT
120 :
125 REM *****
130 REM **      PETIT MENU      **
135 REM *****
140 :
145 HTAB 9: INVERSE : PRINT " LES ETAPES DU PARCOURS "
150 NORMAL : PRINT
155 PRINT "MENU EXPRESS -----"
160 PRINT : PRINT "      1. SAISIE"
165 PRINT : PRINT "      2. IMPRESSION"
170 PRINT : PRINT "      3. TERMINE"
175 PRINT : PRINT "----- CHOIX -> ";:C = POS
(X) + 1:CV = PEEK(37) + 1
180 PRINT : GOSUB 600
185 VTAB CV: HTAB C: CALL - 950: CALL - 190: GET R$
190 ON VAL(R$) GOTO 250,400,195: GOTO 185
195 TEXT : HOME : END
200 ON R GOTO 250,400,540: GOTO 190
205 :
210 REM *****
215 REM **      SAISIE DES DONNEES      **
220 REM *****
225 :
230 X = 0: FOR I = 1 TO 21 STEP 7:X = X + 1: VTAB X: HT
AB 33: PRINT MID$(T$,I,7): NEXT : RETURN
235 T$ = U$: GOSUB 230: GOTO 185
240 TT = TT - (T(N - 1)):N = N - 2: GOTO 435
245 :
250 INVERSE : PRINT R$: NORMAL
255 NORMAL
260 N = 0
265 N = 0:TT = 0
270 N = N + 1
275 T$ = "LIBELLEDE      L'ETAPE": INVERSE : GOSUB 230:
NORMAL
280 PRINT : VTAB 14: CALL - 860: PRINT N" ";:VI = PO
S(X): PRINT PT$;; HTAB VI + 1: CALL - 190: IF E$(
N) < > "" THEN PRINT E$(N);
285 HTAB VI + 1: INPUT " ";E$
290 IF E$ = "" AND E$(N) < > "" THEN VTAB 14: HTAB V
I + 1: PRINT E$(N)

```



```

295 IF E$ = "" AND E$(N) < > "" THEN 330
300 IF E$ = "F" THEN POKE 35,22: GOTO 235
305 IF E$ = "C" THEN GOSUB 600: GOTO 280
310 IF E$ = "" THEN 280
315 IF E$ = "*" AND N > 1 THEN 240
320 IF E$ = "?" THEN 235
325 E$(N) = E$:E$ = ""
330 T$ = "RENTREZ VOTRE HORAIRE": INVERSE: GOSUB 230:
    NORMAL
335 VTAB 16: HTAB 24: PRINT "TEMPS "N" -> ";:V = POS
    (X) + 1
340 CALL - 868: CALL - 190: IF T(N) THEN PRINT T$(N
    );
345 HTAB V: INPUT "":T$
350 IF T$ = "?" THEN 235
355 IF T$ = "C" THEN GOSUB 600: GOTO 335
360 IF T$ = "" AND T(N) THEN 430
365 IF T$ = "" THEN 185
370 IF T$ = "*" AND N > 1 THEN 240
375 IF T$ = "F" THEN POKE 35,22: GOTO 235
380 P = 0:PP = 0:LT = LEN (T$): IF NOT LT THEN 280
385 FOR I = LT TO 1 STEP - 1:C$ = MID$(T$,I,1): IF
    (C$ < "0" OR C$ > "9") AND C$ < > "." THEN 280
390 IF C$ = "." THEN P = 1:PP = I
395 IF P AND I = LT - 1 THEN T$ = LEFT$(T$,I) + "0"
    + RIGHT$(T$,1): GOTO 380
400 NEXT
405 IF NOT P THEN TM% = 60 * VAL (T$): GOTO 420
410 TX% = INT ( VAL (T$)):TY% = VAL ( RIGHT$(T$,LT -
    PP)):TM% = (60 * TX%) + TY%
415 IF NOT PP THEN T$ = T$ + ".00"
420 IF TM% > 300 THEN VTAB 21: HTAB 1: PRINT TM%" MIN
    UTES - CONFIRMEZ PAR OUI SVP ";: GET R$: PRINT : VT
    AB 21: CALL - 868: IF R$ < > "0" THEN 280
425 T$(N) = T$:T(N) = TM%
430 TT = TT + T(N)
435 H$ = "":M$ = "":H$ = STR$( INT (TT / 60)): IF ( V
    AL (H$) * 60) < > TT THEN M$ = STR$(TT - ( VAL (
    H$) * 60))
440 HC$ = H$: IF M$ < > "" THEN HC$ = HC$ + "," + STR
    $( INT ( VAL (M$) * 100 / 60))
445 H$ = H$ + "h" + M$
450 GOSUB 570: GOTO 270
455 :
460 REM *****
465 REM **      IMPRESSION      **
470 REM *****
475 :
480 PRINT : VTAB 20: CALL - 868: PRINT "ALLUMEZ VOTRE
    IMPRIMANTE S.V.P.": GET R$: PRINT : VTAB 20: CALL
    - 868
485 PRINT : PRINT CHR$(4)*PR1": PRINT : PRINT CHR$(
    9)*80N": PRINT CHR$(27)*E": CHR$(27)*L020"

```

On aurait sans doute pu simplifier cette partie de la saisie en utilisant une seule variable, évitant les doubles "IF ceci... ou cela", mais on n'est pas — ici en tout cas — à quelques octets de plus ou de moins.

Il existe différentes méthodes pour additionner les heures et les minutes. Nous en avons choisi une, pour différentes raisons... tout à fait arbitraires.

A noter que, si vous saisissez 1.1, le programme transforme automatiquement en 1.01. Là encore, il s'agit d'un choix délibéré.

Si CHR\$(80N) est une commande standard, CHR\$(27)*E et CHR\$(27)*L020 sont destinées aux imprimantes DMP et IMAGEWRITER.

Voici un PRINT USING de derrière les fagots. Si vous avez mieux... n'hésitez pas !

Si, lors de la saisie de la valeur n°1, on indique l'HEURE DE DÉPART, on obtiendra ici l'HEURE D'ARRIVÉE.

Nous avons fait un effort méritoire pour afficher, par la frappe d'un simple "C", l'aide-mémoire ci-contre.

Programme classique, en basic, de transfert d'écran.

```

490 PRINT CHR$(14)"DEPLACEMENT DU" CHR$(15)PT$: PRINT : PRINT
495 FOR I = 1 TO N - 1:E$ = " ": IF I > 9 OR N - 1 < 10 THEN E$ = ""
500 PRINT E$;I;". ";E$(I);: POKE 36,50
505 IF T(I) < 600 THEN PRINT " ";
510 IF T(I) < 61 THEN PRINT " ";
515 T = VAL (T$(I))
520 T$ = STR$( (I + T * 1000) / 1000):L = LEN (T$):T$ = LEFT$(T$,L - 1): PRINT T$;" mn"
525 NEXT
530 PRINT : PRINT
535 SV = 1: GOSUB 570: PRINT : PRINT CHR$(4)"PRÉ"
540 POKE 35,24: GOTO 185
545 :
550 REM *** * : *****
555 REM ** TEMPS TOTAL **
560 REM *****
565 :
570 VTAB 23: PRINT "TEMPS TOTAL:"H$( "TT" mn) OU "; INVERSE : PRINT HC$;: NORMAL : PRINT " ";: RETURN
575 :
580 REM *****
585 REM ** AIDE-MEMOIRE **
590 REM *****
595 :
600 GOSUB 670
605 VTAB 11: CALL - 950: FOR I = 1 TO 10: PRINT "----";: NEXT : PRINT
610 INVERSE : PRINT "PENDANT LA SAISIE:": NORMAL : PRINT
615 PRINT "- (<?) VOUS RENVOIE AU MENU"
620 PRINT "- (<F) INDIQUE LA FIN DE LA SAISIE"
625 PRINT "- (<*) PERMET DE REVENIR EN ARRIERE"
630 PRINT "- (<C) RENVOIE A CE PETIT TABLEAU"
635 PRINT "- (<RETURN) CONSERVE LA DONNEE EXISTANTE"
640 VTAB 22: HTAB 5: INVERSE : PRINT " ENFONCEZ UNE TOUCHÉ QUELCONQUE ";: GET R$: PRINT : NORMAL : GOSUB 685: RETURN
645 :
650 REM *****
655 REM ** TRANSFERTS DES ECRANS **
660 REM *****
665 :
670 ES = 1024:LO = 60: GOSUB 680:ES = 2040:LO = 62: GOSUB 680:ES = 8192:LO = 66: GOSUB 680
675 POKE 768,160: POKE 769,0: POKE 770,76: POKE 771,44: POKE 772,254: CALL 768: RETURN
680 POKE LO,ES - INT (ES / 256) * 256: POKE LO + 1, INT (ES / 256): RETURN
685 ES = 8192:LO = 60: GOSUB 680:ES = 9200:LO = 62: GOSUB 680:ES = 1024:LO = 66: GOSUB 680: GOTO 675

```


Oui, il est facile, sur une imprimante ImageWriter (ou bien sur une DMP), d'éditer les pages graphiques de l'Apple.

POINT par POINT

LES imprimantes Apple sont accompagnées d'une documentation et d'une disquette TOOL KIT permettant d'éditer rapidement n'importe quel écran graphique, mais leur fonctionnement reste néanmoins mal connu.

Pourtant, quand on sait utiliser toutes les ressources de ces machines, réaliser de belles impressions, sortant de l'ordinaire, devient un jeu d'enfant... en y mettant le temps, comme on le verra plus loin.

L'ÉCRAN GRAPHIQUE DE L'APPLE : Les adresses de l'écran "TEXT" figurent dans de nombreux ouvrages, mais les adresses des écrans graphiques sont moins courantes. Une routine simple vous permettra de les afficher... et même de les imprimer (c'est de la documentation sûre !). La voici :

ADRESSES ECRAN

100	TEXT : HOME	0	\$0400	1024	\$2000	8192	\$4000	16384
110	FOR I = 768 TO 828: READ D: POK	1	\$0400	1152	\$2000	8320	\$4000	16512
	E I,D: NEXT	2	\$0500	1280	\$2100	8448	\$4100	16640
120	PRINT "TAPER I POUR IMPRIMER ";	3	\$0500	1408	\$2100	8576	\$4100	16768
	: GET R#: PRINT : HOME : IF R# =	4	\$0600	1536	\$2200	8704	\$4200	16896
	"I" THEN PR# 1	5	\$0600	1664	\$2200	8832	\$4200	17024
130	FOR I = 0 TO 23	6	\$0700	1792	\$2300	8960	\$4300	17152
140	POKE 6,I: IF I < 10 THEN PRINT	7	\$0700	1920	\$2300	9088	\$4300	17280
	" ";	8	\$0420	1064	\$2020	8232	\$4020	16424
150	PRINT I; TAB(4);: CALL 768	9	\$04A0	1192	\$20A0	8360	\$40A0	16552
160	IF I < 23 THEN PRINT	10	\$0520	1320	\$2120	8488	\$4120	16680
170	NEXT	11	\$05A0	1448	\$21A0	8616	\$41A0	16808
180	IF R# = "I" THEN PRINT : PR# 0	12	\$0620	1576	\$2220	8744	\$4220	16936
190	VTAB 3: POKE - 16368,0: WAIT	13	\$06A0	1704	\$22A0	8872	\$42A0	17064
	- 16384,128: HOME	14	\$0720	1832	\$2320	9000	\$4320	17192
200	DATA 165,6,32,193,251,165,41,13	15	\$07A0	1960	\$23A0	9128	\$43A0	17320
	3,6,32,33,3,165,41,24,105,28,133	16	\$0450	1104	\$2050	8272	\$4050	16464
	,6,32,33,3,165,41,24,105,60,133,	17	\$04D0	1232	\$20D0	8400	\$40D0	16592
	6,32,33,3,96	18	\$0550	1360	\$2150	8528	\$4150	16720
210	DATA 169,164,32,237,253,166,40,	19	\$05D0	1488	\$21D0	8656	\$41D0	16848
	165,6,32,65,249,32,55,3,166,40,1	20	\$0650	1616	\$2250	8784	\$4250	16976
	65,6,32,36,237,162,1,32,74,249,9	21	\$06D0	1744	\$22D0	8912	\$42D0	17104
	6	22	\$0750	1872	\$2350	9040	\$4350	17232
		23	\$07D0	2000	\$23D0	9168	\$43D0	17360

La structure de chaque écran graphique est apparemment la même que celle de l'écran TEXT, soit 24 lignes de 40 caractères, mais ce n'est pas tout. Chacune des lignes de notre tableau n'est que la première d'un groupe de 8.

Ainsi, si l'on considère l'écran HGR (\$4000), entre la ligne 0 et la ligne 1, on trouve 7 sous-lignes :

LIGNE 0 = 8192

0.1 = 8192 + 1024 = 9216
 0.2 = 8192 + 2048 = 10240
 0.3 = 8192 + 3072 = 11264
 0.4 = 8192 + 4096 = 12288
 0.5 = 8192 + 5120 = 13312
 0.6 = 8192 + 6144 = 14336
 0.7 = 8192 + 7168 = 15360

LIGNE 1 = 8320

1.1 = 8320 + 1024 = 9344
 1.2 = 8320 + 2048 = 10368
 1.3 = 8320 + 3072 = 11392
 1.4 = 8320 + 4096 = 12416
 1.5 = 8320 + 5120 = 13440
 1.6 = 8320 + 6144 = 14464
 1.7 = 8320 + 7168 = 15488

et ainsi de suite.

Terminé pour les rangées... mais il y a aussi les colonnes : 40 avons-nous dit. D'accord, mais 40 colonnes de 7 points (bits)... pour arriver au total de 280

UNE PAGE GRAPHIQUE EST DONC COMPOSÉE DE 280 X 192 (24X8), SOIT LA BAGATELLE DE 53760 POINTS (PIXELS).

On comprend pourquoi le programme ci-après (Basic + langage machine) exige plus de 3 minutes pour effectuer, POINT PAR POINT, une copie d'écran (le temps de traitement ne dépasse pas 1 minute... et c'est seulement la lenteur de l'imprimante qui est en cause).

Par contre, on obtient un résultat excellent... et même des ronds ronds (pas question de s'endormir pour autant !). Naturellement, il existe un autre moyen, plus rapide, mais nous en reparlerons.



LES FROMAGES PRÉFÉRÉS DES FRANÇAIS
D'APRÈS UNE ENQUÊTE DE LA SOFRES

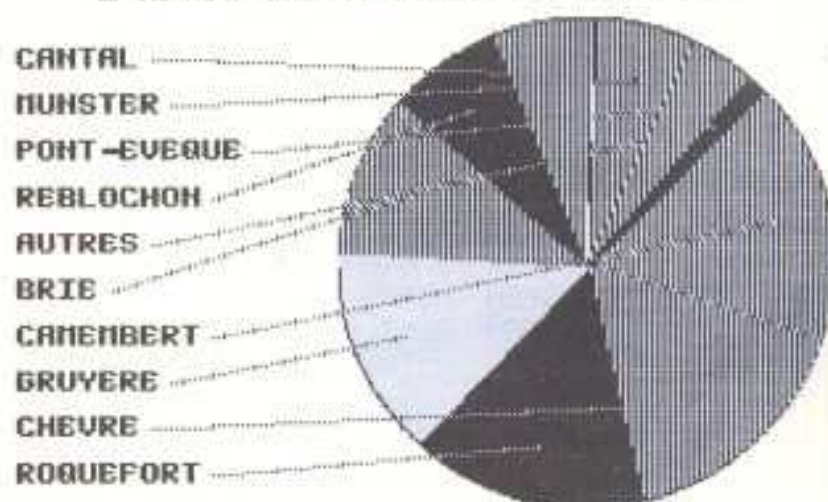
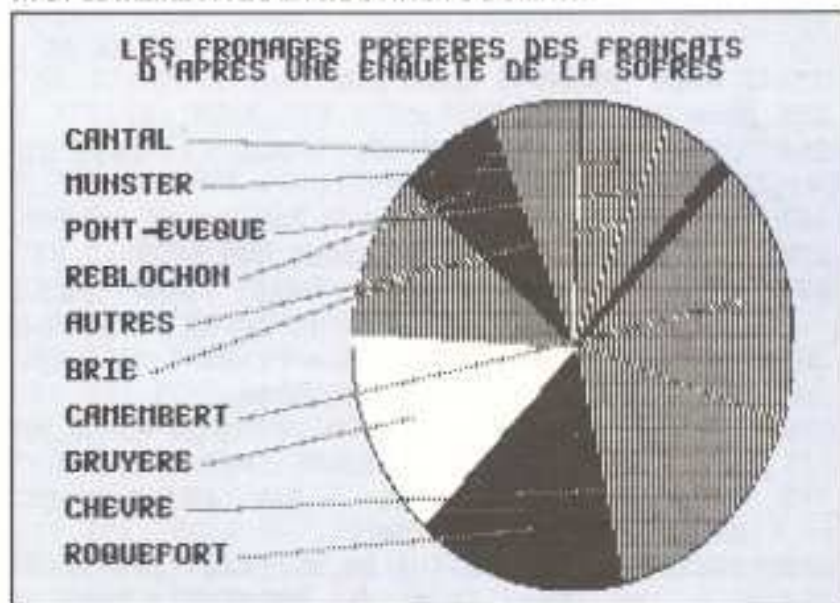


DIAGRAMME OBTENU AVEC LE PROGRAMME TREMLIN MICRO

... ET LE MÊME AVEC LA ROUTINE TOOL KIT...




```

100 TEXT : PRINT CHR$ (21): HOME : VTAB 22
105 D$ = CHR$ (4):E$ = CHR$ (27):SU = 0
110 GOSUB 400: GOTO 200
115 :
120 REM *****
125 REM ** AFFICHAGE DES PARAMETRES **
130 REM *****
135 :
140 PRINT : VTAB 10: CALL - 958
145 PRINT "PAGE ";; INVERSE : PRINT PG;; NORMAL
150 PRINT " - MODE ";; T$ = "BLANC SUR NOIR": IF M = 0 TH
EN T$ = "NOIR SUR BLANC"
155 PRINT T$
160 IF I$ < > "" THEN PRINT "IMAGE: "I$;; IF DR THEN
PRINT " - DRIVE "DR
165 IF SL THEN PRINT "IMPRIMANTE DANS LE SLOT ";SL
170 RETURN
175 :
180 REM *****
185 REM ** ENTREE DES PARAMETRES **
190 REM *****
195 :
200 VTAB 22: HTAB 1: CALL - 868: PRINT "<1> HGR <2> HG
R2 -> ";; GET P$:PG = VAL (P$): IF NOT PG THEN PG
= 1
205 IF PG = 1 THEN HGR
210 IF PG = 2 THEN HGR2 :SU = 8192
215 TEXT : HOME
220 GOSUB 140
225 VTAB 22: HTAB 1: CALL - 868: PRINT "<1> INVERSE <0>
NORMAL -> ";; GET R$
230 IF R$ = CHR$ (8) OR R$ = CHR$ (11) THEN 200
235 M = VAL (R$): IF M > 1 THEN M = 1
240 POKE 10,M
245 GOSUB 140
250 VTAB 22: HTAB 1: CALL - 868: PRINT "IMAGE -> ";; GE
T R$: IF R$ = "" OR R$ = CHR$ (8) OR R$ = CHR$ (11)
THEN PRINT : GOTO 200
255 I$ = R$
260 VTAB 22: HTAB 1: PRINT "IMAGE -> "I$;; HTAB 11: INPU
T "":R$:I$ = I$ + R$
265 IF I$ < > "?" AND (I$ < "A" OR I$ > "Z") THEN 250
270 IF I$ < > "?" THEN GOSUB 140
275 VTAB 22: HTAB 1: CALL - 868: PRINT "DRIVE 1-2 ? ->
";; GET R$: PRINT :DR = 1: IF R$ = "2" THEN DR = 2
280 IF R$ = CHR$ (8) OR R$ = CHR$ (11) THEN 250
285 IF I$ < > "?" THEN GOSUB 140
290 IF I$ = "?" THEN HOME : PRINT D$"CATALOG,D"DR: PRIN
T : GET R$: PRINT : PRINT : PRINT :I$ = "": GOTO 260
295 VTAB 22: HTAB 1: CALL - 868: PRINT "SLOT DE L'IMPRI
MANTE ? -> ";; GET R$
300 IF R$ = CHR$ (8) OR R$ = CHR$ (11) THEN 275
305 SL = VAL (R$): IF SL < 1 OR SL > 7 THEN 295

```

← Vers l'initialisation du système

← CALL—958 efface tout ce qui vient après le curseur (y compris au-dessous)

DR = Numéro de DRIVE
SL = Numéro de SLOT

PG = Numéro de page
(HGR=1 — HGR2=2)

SU = 8192 quand on choisit la page 2

M = Mode (1=Inverse, 2=Normal)

I\$ = Image à charger

Notez qu'il est possible d'obtenir le catalogue en tapant (?) et de revenir à l'item précédent en utilisant l'une ou l'autre des flèches de l'Apple IIe ou de l'Apple IIc: ←↑

DS = Densité (normal ou gras)

C'est par l'Ampersand (initialisé plus loin) que l'on transmet l'adresse du début de ligne à la routine en assembleur.

P = nombre de lignes à lire (1 principale + 7 sous-lignes)

D = nombre de lignes principales (24)

Si vous désirez en savoir plus sur ces 4 lignes (qui installent une courte routine en langage machine) reportez-vous à la page suivante.

```
310 GOSUB 140
315 VTAB 22: HTAB 1: CALL - 868: PRINT "<S>IMPLE DENSIT
E <D>DOUBLE ";: GET S$:DS = 0: IF S$ = "2" THEN DS = 1
320 P$ = ",A#2000": IF PG = 2 THEN P$ = ",A#4000"
325 PRINT : PRINT D$"BLOAD" I$:P$,D"DR
330 POKE 49235 + PG,0: POKE 49234,0: POKE 49239,0: POKE
49232,0
335 :
340 REM *****
345 REM ** IMPRESSION **
350 REM *****
355 :
360 PRINT CHR$(4)"PRÉ"SL
365 PRINT : PRINT CHR$(15)
370 PRINT CHR$(9)"80N"
375 IF NOT DS THEN 385
380 PRINT E$"!"; n. ut en caractères gras si DS=1
385 PRINT E$">"; impression unidirectionnelle
390 PRINT E$"n"; caractère choisi
395 PRINT E$"T01"
400 READ AD:AD = AD + SU:P = 0
405 & AD
410 PRINT E$"G0280";: CALL 779
415 PRINT : PRINT
420 P = P + 1: IF P < 8 THEN 410
425 D = D + 1: IF D < 24 THEN 400
430 PRINT
435 PRINT E$ + CHR$(34)
440 PRINT D$"PRÉ0"
445 X = 0: FOR I = 1013 TO 1015:X = X + 1: POKE I,A(X): N
EXT
450 TEXT : HOME : END
455 :
460 REM *****
465 REM ** ROUTINE LANGAGE MACHINE **
470 REM *****
475 :
480 POKE 768,32: POKE 769,123: POKE 770,221: POKE 771,32
: POKE 772,82: POKE 773,231: POKE 774,132: POKE 775,6
: POKE 776,133: POKE 777,7: POKE 778,96: POKE 779,160
: POKE 780,0: POKE 781,177: POKE 782,6: POKE 783,133
485 POKE 784,8: POKE 785,162: POKE 786,7: POKE 787,165:
POKE 788,8: POKE 789,24: POKE 790,74: POKE 791,133: P
OKE 792,8: POKE 793,169: POKE 794,0: POKE 795,105: PO
KE 796,0: POKE 797,24: POKE 798,101: POKE 799,10
490 POKE 800,201: POKE 801,2: POKE 802,208: POKE 803,3:
POKE 804,56: POKE 805,233: POKE 806,2: POKE 807,32: P
OKE 808,237: POKE 809,253: POKE 810,202: POKE 811,208
: POKE 812,230: POKE 813,200: POKE 814,192: POKE 815,
40
495 POKE 816,208: POKE 817,219: POKE 818,24: POKE 819,16
9: POKE 820,4: POKE 821,101: POKE 822,7: POKE 823,133
: POKE 824,7: POKE 825,96
```



```

500 :
505 X = 0: FOR I = 1013 TO 1015: X = X + 1: A(X) = PEEK (I
): NEXT
510 POKE 1013,76: POKE 1014,0: POKE 1015,03: RETURN
515 :
520 DATA 8192,8320,8448,8576,8704,8832,8960,9088,8232,83
60,8480,8616,8744,8872,9000,9128,8272,8400,8528,8656,
8784,8912,9040,9168

```

← Sauvegarde des valeurs pour le cas où elles seraient utilisées par ailleurs

← Adresses des 24 lignes principales

POUR EN SAVOIR PLUS SUR LES POKES (page 15)

0300-	20	7B	DD	JSR	\$DD7B] LIRE L'EXPLICATION (bas de page) (A)
0303-	20	52	E7	JSR	\$E752	
0306-	84	06		STY	\$06	
0308-	85	07		STA	\$07	
030A-	60			RTS		
030B-	A0	00		LDY	£\$00	On arrive ici par un CALL 779 (ligne 410)
030D-	B1	06		LDA	(<\$06), Y	LIRE EXPLICATION (B)
030F-	85	08		STA	\$08	Stockage à l'adresse 08, page zéro
0311-	A2	07		LDX	£\$07	X est mis à 7 (compteur de boucle)
0313-	A5	08		LDA	\$08	A chargé avec contenu de 08
0315-	18			CLC		Annulation de la retenue
0316-	4A			LSR		Décalage à droite (le bit 7 tombe dans la retenue)
0317-	85	08		STA	\$08	
0319-	A9	00		LDA	£\$00	
031B-	69	00		ADC	£\$00	On additionne 0 + la retenue à A
031D-	18			CLC		Nouvelle annulation
031E-	65	0A		ADC	\$0A	En \$0A, il y a 1 ou 0 (selon mode)
0320-	C9	02		CMP	£\$02	Obtient-on 2 ?
0322-	D0	03		BNE	\$0327	Si non, saut
0324-	38			SEC		Retenue à 1
0325-	E9	02		SBC	£\$02	Si oui — 2 (ainsi, 1 = 0 et 0 = 1)
0327-	20	ED	FD	JSR	\$FDED	Envoi à l'imprimante
032A-	CA			DEX		X = X - 1
032B-	D0	E6		BNE	\$0313	Si on n'est pas à 0, encore un tour
032D-	C8			INY		Y = Y + 1
032E-	C0	28		CPY	£\$28	Est-on à \$28 (40) ?
0330-	D0	DB		BNE	\$030D	Non : encore un tour
0332-	18			CLC		Annulation de la retenue
0333-	A9	04		LDA	£\$04	
0335-	65	07		ADC	\$07	
0337-	85	07		STA	\$07	On incrémente la partie haute de l'adresse de \$04
0339-	60			RTS		

(A) \$DD7B évalue la formule venant du Basic (ligne 405), via l'Ampersand et met le résultat dans FAC.

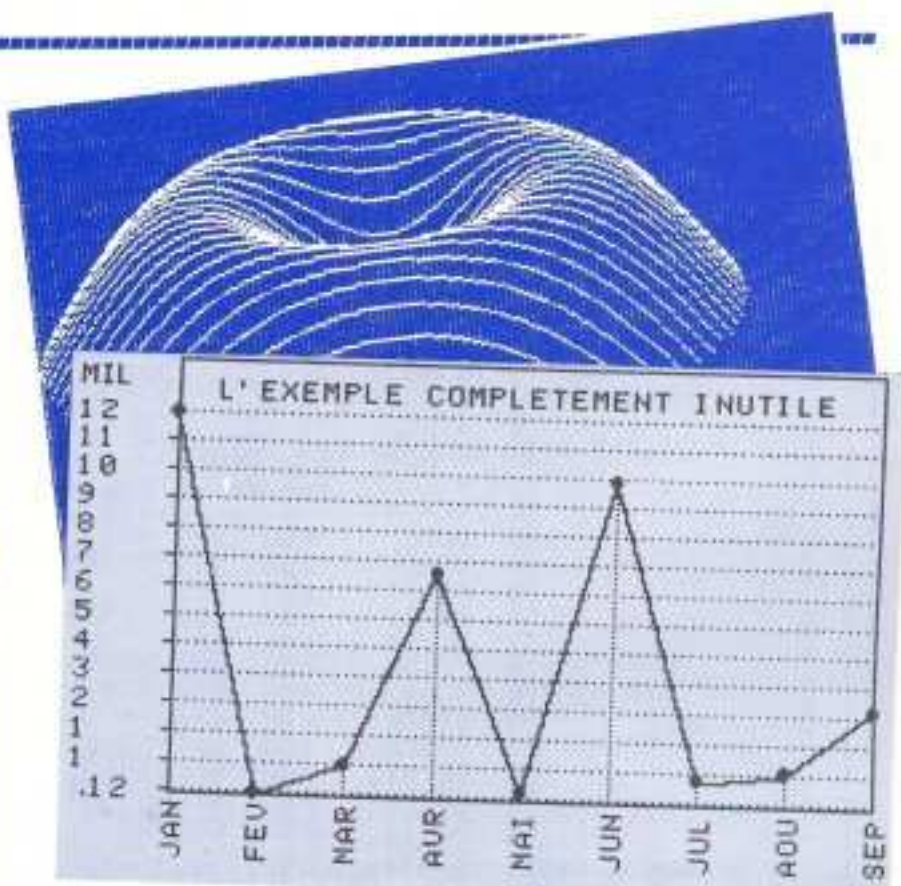
\$E752 va transformer ce résultat en un entier, sur 2 octets. On le récupère dans Y et A pour le stocker aux adresses \$6 et \$7.

(B) Ainsi, dans 6 et 7, on a l'adresse du point à lire sur l'écran (au départ : \$ 00 20... ou \$ 20 00, si l'on préfère).

LDA (\$06) Y lit la valeur qui se trouve à l'adresse 6+Y, mais une adresse comporte 2 octets et, en réalité, on lit à l'adresse 6—7 + Y... ou, dans le cas présent, \$2000 + Y.

Et voici le programme IMP.HGR.1

IL FONCTIONNE SUR
IMPRIMANTES DMP,
IMAGEWRITER, MANNESMANN
ET AUTRES COMPATIBLES.



EVIDEMMENT, il opère par groupe de 8 lignes, est entièrement en langage machine, et se révèle donc aussi rapide que les routines existant déjà.

Il est indispensable, au départ, d'initialiser les adresses 253 et 254 par POKE 253,00 et POKE 254,24 (ligne de départ et ligne d'arrivée). Si l'on sort plusieurs éditions du même graphique, il n'est pas nécessaire de se livrer à cette opération... sauf si l'on désire modifier le numéro de la première ou de la dernière ligne (ici, la notion de ligne est la même qu'en mode "TEXT" : 24 lignes, de 0 à 23).

Prenons un exemple précis : soit à imprimer un graphique commençant en VTAB 4 et se terminant en VTAB 12.

On tapera : POKE 253,3 (VTAB - 1) et, POKE 254,12. D'accord ?

MODE et PAGE

Par défaut, le logiciel vous fournira une copie d'écran de la page 1 (HGR), en mode normal. Attention ! mode normal signifie, sur le papier, blanc dans du noir (comme sur votre écran). C'est parfait pour les "photos", mais pas forcément joli pour les graphiques. Un POKE 6,1 permet d'obtenir l'image INVERSE (noir sur blanc).

Pour imprimer la page 2 (HGR2), taper un POKE 6,2.

RAPIDITÉ

Si vous utilisez \$4E (N) au lieu de \$6E (n) à l'adresse \$60D4, vous y gagnerez en rapidité, mais vos camemberts seront ovales. Pour impression en double densité, allumez votre imprimante avant de lancer l'impression et dites-lui gentiment que vous attendez du gras par la séquence suivante :

```
PR&1
PRINT CHR$(27 "I"
PR&0
```

EN CONCLUSION

Dans un programme Basic, une seule ligne fera le bonheur de IMP.HGR1 :

```
10 PRINT CHR$(4)"BLOAD IMP.HGR1" :
   POKE 6,1 : POKE 7,1 : POKE 253,00 :
   POKE 254,24 : PRINT CHR$(4)"BLOAD
   IMAGE, A$2000" : CALL 24576
```

Toutes les variantes sont possibles.

Le programme n'est pas relogeable, mais si vous disposez d'un assembleur, il suffit de modifier l'adresse ORG \$6000 (ligne 7) pour l'installer ailleurs. Attention ! il est un peu trop long pour être placé à l'adresse \$300.

Assemblage par ProCODE

```

0 *****
1 * COPIE HGR SUR IMAGEWRITER, OMP et MANESMANN MT 85/86 *
2 -----*
3 *          GUY-HACHETTE  TREMLIN MICRO  NOVEMBRE 85          *
4 *****
5          ORG $6000
6 -----*
7 * Pour utiliser, charger le programme par Bload CDPHGR *
8 * POKE 6,0 pour MODE NORMAL ou POKE 6,1 pour INVERSE *
9 * POKE 7,1 pour PAGE 1 ou 7,2 pour PAGE 2 *
10 * POKE 253,n0<n23 (ligne début) POKE 254,N>n<24 (Fin) *
11 * (ligne début = 0 par défaut et ligne fin = 23) *
12 -----*
13 MODE      EQU  $06      ; 0 pour NORMAL et 1 = Inverse
14 VPAGE     EQU  $07      ; $1C= Hgr - $3C= Hgr2
15 ABAS1     EQU  $08      ; Adresse de base de la ligne
16 ABAS2     EQU  $09      ; partie haute
17 NALIGN    EQU  $FD      ; Nombre de lignes (0 p/défaut)
18 FLIGN     EQU  $FE      ; Dernière ligne (23 p/défaut)
19 HTAB      EQU  $FF      ; Longueur de la ligne
20 STOCK     EQU  $18      ; Tampon de 7 octets
21 BASCALC   EQU  $FBC1    ; Calcule l'adresse de base
22 SETVID    EQU  $FE93    ; Equivalent de Pr#0
23 OUTPORT   EQU  $FE95    ; Pour Pr#1 si A contient 1
24 COUT      EQU  $FOED    ; Vers imprimante
25 *
26 *          INITIALISATION IMPRIMANTE ET DIVERS POINTEURS          *
27 *          -----*
28          LOA  Vpage
29          CLC
30          ADC  E$3A
31          CMP  E$3C
32          BEQ  Stp
33          LOA  E$1C
34          STA  Vpage
35          LDA  Nalign
36          CMP  E$17
37          BCC  Stld
38          LDA  E$0
39          STA  Nalign
40          LDA  Flign
41          CMP  E$18
42          BCC  Stlf
43          LOA  E$18
44          STA  Flign
45          LDA  E$01      ; Imprimante dans le Slot 1
46          JSR  Outport
47 *
48 *          INITIALISATION DE L'IMPRIMANTE (DEPART)          *
49 *          -----*
50          LOX  E$FF
51          INX  $B0
52          LOA  TAB1,X
53          JSR  Cout
54          CPX  E$17
55          BNE  $B0
56 *
57 *          "G0280" ENVOYE A L'IMPRIMANTE (QUI ATTEND UNE LIGNE) *
58 *          -----*
59          INIT  LOX  E$FF
60          INX  $B0
61          LOA  TAB2,X
62          JSR  Cout
63          CPX  E$07
64          BNE  $B0
65 *
66 *          ADRESSE DE BASE          *
67 *          -----*

```



```

6040: A9 00      68      LDA  E$00
6042: 85 FF      69      STA  Htab      ; Longueur de ligne à zéro
6044: A5 FD      70      DEBUT LDA  Nlign
6046: 20 C1 FB   71      JSR  Bascalo   ; L'adresse est dans $28 & $29
6049: A5 28      72      LDA  $28
604B: 65 FF      73      ADC  Htab      ; On y ajoute htab
604D: 85 08      74      STA  Abas1     ; Partie basse stockée
604F: A5 29      75      LDA  $29      ; Récupération partie haute
6051: 65 07      76      ADC  Vpage     ; à laquelle on ajoute $1C/$3C
6053: 85 09      77      STA  Abas2     ; avant de la sauvegarder
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *
133 *
134 *
135 *
136 *

```

ZONE DE STOCKAGE MISE A ZERO (\$18-\$1E)

```

LDX  E$07
LDA  E$00
STA  Stock-1,X
DEX
BNE  $80

```

PREPARATION DES 7 CODES DU CARAC GRAPHIQUE
Sept octets pour imprimer 1 caractère gra-
phique, le 1/40e de la ligne (280 / 7 = 40)

```

LDY  E$08
LDX  E$00
LDA  (Abas1,X) ; Lecture adressage indirect
LSR  ; Décalage à droite
ROR  Stock,X  ; La retenue entre à gauche
INX
CPX  E$07     ; Si 8 on a fait terminé!
BNE  Décal   ; sinon nouveau décalage
LDA  E$04     ; On va majorer la partie haute
CLC  ; de $04 (04/00=1024) pour pas-
ADC  Abas2   ; ser à la ligne suivante
STA  Abas2   ; immédiatement réinstallée
DEY
BNE  InitX   ; Si <> de zéro, encore un tour

```

IMPRESSION DU CARACTERE

```

LDX  E$FF
INX
LDA  Stock,X
LDY  Mode
CPY  E$01
BEQ  Filtre
EOR  E$FF

```

FILTRAGE DE \$0D \$8D \$1A ET \$9A (CTRL-13 ET CTRL-2)

```

CMP  E$0D
BEQ  Chang
CMP  E$8D
BEQ  Chang
CMP  E$1A
BEQ  Chang
CMP  E$9A
BEQ  Chang
BNE  Graph
LSR  ; Un zéro rentre à gauche...
LSR  ; ... et un second
ASL  ; Un zéro rentre à droite...
ASL  ; ... puis un second
JSR  Cout
CPX  E$6     ; Si $6, on a terminé
BNE  Bcarac
INC  Htab
LDA  Htab
CMP  E$28

```


IMP.HGR.1 (suite)

```

60A6: DD 9C      137      BNE  Debut
60A8: A9 DD      138      LOA  £$00      ; Positionnement marge gauche
60AA: 20 ED FD   139      JSR  Cout
60AD: E6 FD      140      INC  Nlign
60AF: A5 FD      141      LDA  Nlign
60B1: C5 FE      142      CMP  Flign
60B3: FD 03      143      BEQ  Fin
60B5: 20 33 60   144      JSR  Init
60B8: A9 1B      145      LDA  £$1B
60BA: 20 ED FD   146      JSR  Cout
60BD: A9 63      147      LDA  £$63      ; ASCII "c": imprimante normale
60BF: 20 ED FD   148      JSR  Cout
60C2: 20 93 FE   149      JSR  Setvid
60C5: 60         150      RTS
        151      *
        152      *
        153      *
        154      *
        155      *
        156      *
        157      *
        158      *
        159      *
        160      *
        161      *
        162      *
        163      *
        164      *
        165      *
        166      *
        167      *
        168      *
        169      *
        170      *
        171      *
        172      *
        173      *
        174      *
        175      *
        176      *
        177      *
        178      *
        179      *
        180      *
        181      *
        182      *
        183      *
        184      *
        185      *
        186      *
        187      *
        188      *
        189      *
        190      *
        191      *
        192      *
        193      *
        194      *
        195      *
        196      *
        197      *
        198      *
        199      *
        200      *
        201      *
        202      *
        203      *
        204      *
        205      *
        206      *
        207      *
        208      *
        209      *
        210      *
        211      *
        212      *
        213      *
        214      *
        215      *
        216      *
        217      *
        218      *
        219      *
        220      *
        221      *
        222      *
        223      *
        224      *
        225      *
        226      *
        227      *
        228      *
        229      *
        230      *
        231      *
        232      *
        233      *
        234      *
        235      *
        236      *
        237      *
        238      *
        239      *
        240      *
        241      *
        242      *
        243      *
        244      *
        245      *
        246      *
        247      *
        248      *
        249      *
        250      *
        251      *
        252      *
        253      *
        254      *
        255      *
        256      *
        257      *
        258      *
        259      *
        260      *
        261      *
        262      *
        263      *
        264      *
        265      *
        266      *
        267      *
        268      *
        269      *
        270      *
        271      *
        272      *
        273      *
        274      *
        275      *
        276      *
        277      *
        278      *
        279      *
        280      *
        281      *
        282      *
        283      *
        284      *
        285      *
        286      *
        287      *
        288      *
        289      *
        290      *
        291      *
        292      *
        293      *
        294      *
        295      *
        296      *
        297      *
        298      *
        299      *
        300      *
        301      *
        302      *
        303      *
        304      *
        305      *
        306      *
        307      *
        308      *
        309      *
        310      *
        311      *
        312      *
        313      *
        314      *
        315      *
        316      *
        317      *
        318      *
        319      *
        320      *
        321      *
        322      *
        323      *
        324      *
        325      *
        326      *
        327      *
        328      *
        329      *
        330      *
        331      *
        332      *
        333      *
        334      *
        335      *
        336      *
        337      *
        338      *
        339      *
        340      *
        341      *
        342      *
        343      *
        344      *
        345      *
        346      *
        347      *
        348      *
        349      *
        350      *
        351      *
        352      *
        353      *
        354      *
        355      *
        356      *
        357      *
        358      *
        359      *
        360      *
        361      *
        362      *
        363      *
        364      *
        365      *
        366      *
        367      *
        368      *
        369      *
        370      *
        371      *
        372      *
        373      *
        374      *
        375      *
        376      *
        377      *
        378      *
        379      *
        380      *
        381      *
        382      *
        383      *
        384      *
        385      *
        386      *
        387      *
        388      *
        389      *
        390      *
        391      *
        392      *
        393      *
        394      *
        395      *
        396      *
        397      *
        398      *
        399      *
        400      *
        401      *
        402      *
        403      *
        404      *
        405      *
        406      *
        407      *
        408      *
        409      *
        410      *
        411      *
        412      *
        413      *
        414      *
        415      *
        416      *
        417      *
        418      *
        419      *
        420      *
        421      *
        422      *
        423      *
        424      *
        425      *
        426      *
        427      *
        428      *
        429      *
        430      *
        431      *
        432      *
        433      *
        434      *
        435      *
        436      *
        437      *
        438      *
        439      *
        440      *
        441      *
        442      *
        443      *
        444      *
        445      *
        446      *
        447      *
        448      *
        449      *
        450      *
        451      *
        452      *
        453      *
        454      *
        455      *
        456      *
        457      *
        458      *
        459      *
        460      *
        461      *
        462      *
        463      *
        464      *
        465      *
        466      *
        467      *
        468      *
        469      *
        470      *
        471      *
        472      *
        473      *
        474      *
        475      *
        476      *
        477      *
        478      *
        479      *
        480      *
        481      *
        482      *
        483      *
        484      *
        485      *
        486      *
        487      *
        488      *
        489      *
        490      *
        491      *
        492      *
        493      *
        494      *
        495      *
        496      *
        497      *
        498      *
        499      *
        500      *
        501      *
        502      *
        503      *
        504      *
        505      *
        506      *
        507      *
        508      *
        509      *
        510      *
        511      *
        512      *
        513      *
        514      *
        515      *
        516      *
        517      *
        518      *
        519      *
        520      *
        521      *
        522      *
        523      *
        524      *
        525      *
        526      *
        527      *
        528      *
        529      *
        530      *
        531      *
        532      *
        533      *
        534      *
        535      *
        536      *
        537      *
        538      *
        539      *
        540      *
        541      *
        542      *
        543      *
        544      *
        545      *
        546      *
        547      *
        548      *
        549      *
        550      *
        551      *
        552      *
        553      *
        554      *
        555      *
        556      *
        557      *
        558      *
        559      *
        560      *
        561      *
        562      *
        563      *
        564      *
        565      *
        566      *
        567      *
        568      *
        569      *
        570      *
        571      *
        572      *
        573      *
        574      *
        575      *
        576      *
        577      *
        578      *
        579      *
        580      *
        581      *
        582      *
        583      *
        584      *
        585      *
        586      *
        587      *
        588      *
        589      *
        590      *
        591      *
        592      *
        593      *
        594      *
        595      *
        596      *
        597      *
        598      *
        599      *
        600      *
        601      *
        602      *
        603      *
        604      *
        605      *
        606      *
        607      *
        608      *
        609      *
        610      *
        611      *
        612      *
        613      *
        614      *
        615      *
        616      *
        617      *
        618      *
        619      *
        620      *
        621      *
        622      *
        623      *
        624      *
        625      *
        626      *
        627      *
        628      *
        629      *
        630      *
        631      *
        632      *
        633      *
        634      *
        635      *
        636      *
        637      *
        638      *
        639      *
        640      *
        641      *
        642      *
        643      *
        644      *
        645      *
        646      *
        647      *
        648      *
        649      *
        650      *
        651      *
        652      *
        653      *
        654      *
        655      *
        656      *
        657      *
        658      *
        659      *
        660      *
        661      *
        662      *
        663      *
        664      *
        665      *
        666      *
        667      *
        668      *
        669      *
        670      *
        671      *
        672      *
        673      *
        674      *
        675      *
        676      *
        677      *
        678      *
        679      *
        680      *
        681      *
        682      *
        683      *
        684      *
        685      *
        686      *
        687      *
        688      *
        689      *
        690      *
        691      *
        692      *
        693      *
        694      *
        695      *
        696      *
        697      *
        698      *
        699      *
        700      *
        701      *
        702      *
        703      *
        704      *
        705      *
        706      *
        707      *
        708      *
        709      *
        710      *
        711      *
        712      *
        713      *
        714      *
        715      *
        716      *
        717      *
        718      *
        719      *
        720      *
        721      *
        722      *
        723      *
        724      *
        725      *
        726      *
        727      *
        728      *
        729      *
        730      *
        731      *
        732      *
        733      *
        734      *
        735      *
        736      *
        737      *
        738      *
        739      *
        740      *
        741      *
        742      *
        743      *
        744      *
        745      *
        746      *
        747      *
        748      *
        749      *
        750      *
        751      *
        752      *
        753      *
        754      *
        755      *
        756      *
        757      *
        758      *
        759      *
        760      *
        761      *
        762      *
        763      *
        764      *
        765      *
        766      *
        767      *
        768      *
        769      *
        770      *
        771      *
        772      *
        773      *
        774      *
        775      *
        776      *
        777      *
        778      *
        779      *
        780      *
        781      *
        782      *
        783      *
        784      *
        785      *
        786      *
        787      *
        788      *
        789      *
        790      *
        791      *
        792      *
        793      *
        794      *
        795      *
        796      *
        797      *
        798      *
        799      *
        800      *
        801      *
        802      *
        803      *
        804      *
        805      *
        806      *
        807      *
        808      *
        809      *
        810      *
        811      *
        812      *
        813      *
        814      *
        815      *
        816      *
        817      *
        818      *
        819      *
        820      *
        821      *
        822      *
        823      *
        824      *
        825      *
        826      *
        827      *
        828      *
        829      *
        830      *
        831      *
        832      *
        833      *
        834      *
        835      *
        836      *
        837      *
        838      *
        839      *
        840      *
        841      *
        842      *
        843      *
        844      *
        845      *
        846      *
        847      *
        848      *
        849      *
        850      *
        851      *
        852      *
        853      *
        854      *
        855      *
        856      *
        857      *
        858      *
        859      *
        860      *
        861      *
        862      *
        863      *
        864      *
        865      *
        866      *
        867      *
        868      *
        869      *
        870      *
        871      *
        872      *
        873      *
        874      *
        875      *
        876      *
        877      *
        878      *
        879      *
        880      *
        881      *
        882      *
        883      *
        884      *
        885      *
        886      *
        887      *
        888      *
        889      *
        890      *
        891      *
        892      *
        893      *
        894      *
        895      *
        896      *
        897      *
        898      *
        899      *
        900      *
        901      *
        902      *
        903      *
        904      *
        905      *
        906      *
        907      *
        908      *
        909      *
        910      *
        911      *
        912      *
        913      *
        914      *
        915      *
        916      *
        917      *
        918      *
        919      *
        920      *
        921      *
        922      *
        923      *
        924      *
        925      *
        926      *
        927      *
        928      *
        929      *
        930      *
        931      *
        932      *
        933      *
        934      *
        935      *
        936      *
        937      *
        938      *
        939      *
        940      *
        941      *
        942      *
        943      *
        944      *
        945      *
        946      *
        947      *
        948      *
        949      *
        950      *
        951      *
        952      *
        953      *
        954      *
        955      *
        956      *
        957      *
        958      *
        959      *
        960      *
        961      *
        962      *
        963      *
        964      *
        965      *
        966      *
        967      *
        968      *
        969      *
        970      *
        971      *
        972      *
        973      *
        974      *
        975      *
        976      *
        977      *
        978      *
        979      *
        980      *
        981      *
        982      *
        983      *
        984      *
        985      *
        986      *
        987      *
        988      *
        989      *
        990      *
        991      *
        992      *
        993      *
        994      *
        995      *
        996      *
        997      *
        998      *
        999      *
        1000      *
    
```

TABLES POUR L'IMPRIMANTE

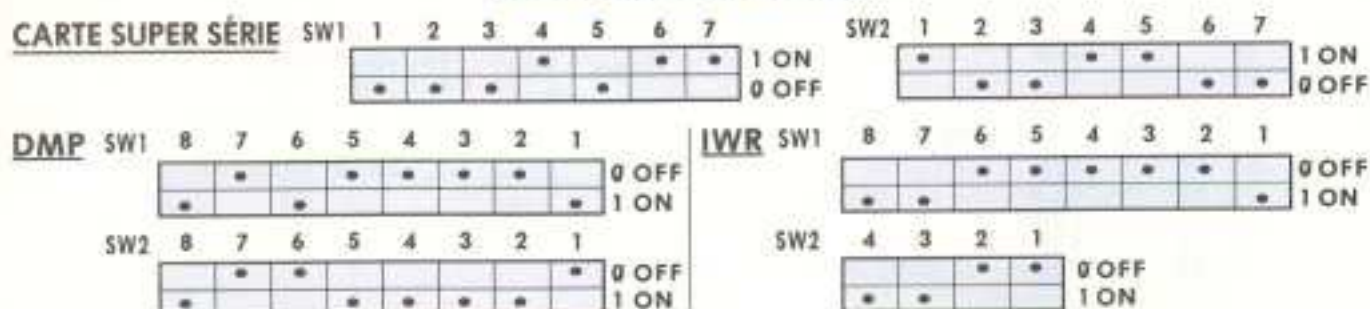
Table des symboles ordre alphabetique

-ABAS1.....\$0008	-ABAS2.....\$0009	-BASCALC...\$FBC1	-BCARAC....\$6078
-CHANG.....\$6095	-COUT.....\$FDED	-DEBUT....\$6044	-Décal....\$6064
-FILTRE....\$6083	-FIN.....\$6088	-FLIGN....\$00FE	-GRAPH....\$6099
-HTAB.....\$00FF	-INIT.....\$6033	-InitX....\$606D	-MODE.....\$0006
-NLIGN....\$00FD	-OUTPORT...\$FE95	-SETVID...\$FE93	-Std.....\$6015
-St1f.....\$601F	-STOCK....\$0018	-Stp.....\$600B	-TAB1.....\$60C6
-TAB2.....\$60DE	-VPAGE....\$0007		

Sauvez votre programme par un **BSAVE IMP.HGR.1, A24576,L230**

VERIFICATION: T = 0 : FOR I = 24576 TO 24575 + 230 : T = T + PEEK(I) : NEXT I : PRINT T Le résultat devra être : **28606**

RÉGLAGES CONSEILLÉS



CARACTÈRES PERSONNALISÉS

sur DMP et IMAGEWRITER

PROGRAMME BASIC : CARAC.1.0 (qui appelle CARAC.C, routine en L.M.)

OPTIONS : Après un RUN, le menu général s'affiche et propose plusieurs grandes options. Choisissons la première, et aussi la plus intéressante.

CREER UN CARACTERE

C'est la raison d'être de ce long programme, avec lequel vous pourrez créer un caractère... et même une police complète de caractères personnalisés (96 maximum, l'espace compris).

Lorsque vous choisirez l'option 1, une grille s'affichera sur votre écran et vous devrez indiquer la largeur du ou des caractères à créer.

Pour une fonte normale, cette largeur sera de 8 points, mais n'oubliez pas de réserver une colonne de blanc pour séparer vos caractères. Si vous tapez **CTRL-E**, un exemple de caractère s'inscrira sur votre écran.

Pendant toute la séquence de création, il vous suffira de presser la barre d'espacement pour transformer le point en astérisque ou vice versa, étant entendu que point = non impression et astérisque = impression. Pour vous déplacer, utilisez alors les flèches de l'Apple.

D'une manière générale, les caractères utilisent les aiguilles 1 à 8 de l'imprimante (lignes 1 à 8), mais les g, j, p, q et y, ainsi que les ç et certains signes de ponctuation font appel aux aiguilles 2/9.

Laissez le logiciel faire son travail et ne soyez pas surpris, après **ESC** si une lettre basse remonte d'un cran.

Si vous vouliez que tous vos caractères soient imprimés avec les aiguilles de 2 à 9, il vous suffirait de les dessiner à partir de la 2^e ligne — le programme ferait le reste.

FAUT-IL DEMANDER L'IMPRESSION DE CHAQUE CARACTÈRE, AU FUR ET À MESURE DE LA CRÉATION ?

Oui, au moins pour les dix ou vingt premiers, pour être sûr que tout se passe bien. Il serait inutile de perdre une heure... pour un résultat nul... ou décevant.

Ensuite, nous vous conseillons de n'imprimer les caractères déjà créés qu'une fois sur cinq.

Si la sortie imprimante devient soudainement mauvaise, c'est parce qu'un caractère a profondément déplu à votre chère machine.

Elle n'accepte pas les valeurs 13 et 141 (RETURN) qui la perturbent !

Il suffit de modifier légèrement le dessin pour repartir d'un bon pied, mais attention !

Pour redessiner un caractère existant, taper **CTRL-O** qui affiche les noms des lettres déjà créées, puis indique quel est celui à redéfinir. Si vous utilisez la touche **CTRL-F**, vous obtiendrez automatiquement l'affichage des caractères existants, après chaque création. Un simple **ESC** vous ramènera alors à l'option création.

En fait, **CTRL-F** étant une bascule, un nouvel appel à cette fonction... l'annulera.

IMPORTANT La lettre-clé désignant chaque caractère personnalisé doit être "affichable". Donc, pas question d'utiliser, du moins avec ce programme, de valeurs inférieures à 32 (ESPACE) ou supérieures à 127 (DELETE). Naturellement, cet utilitaire autorise bon nombre de fantaisies... graphiques, pourvu que celles-ci soient contenues dans les limites déjà indiquées : 16 pts de large pour 8 pts (utiles) de haut. Voici quelques exemples de caractères très spéciaux, réalisés de cette manière (16 pts... sans colonne de blanc... sur 8 pts). On les appelle par 1, 2, 3, 4, etc. ; mais on aurait évidemment pu choisir d'autres lettres-clés.



CONSEIL Lisez attentivement la liste des fonctions (au verso de cette fiche). Elle vous donnera de bonnes indications sur la manière d'utiliser CARAC.1.0. Ne manquez pas, par ailleurs, de nous indiquer les difficultés rencontrées. Nous mettrons au point, ensemble, une version 1.1 plus performante !

CRÉATION ET MODIFICATION DE CARACTÈRES (COMMANDES)

BARRE D'ESPACEMENT * si * et * si * (bascule)

- ➔ (ou CTRL-U) : Vers la droite
 - ➜ (ou CTRL-H) : Vers la gauche
 - ⬆ (ou CTRL-K) : Vers le haut
 - ⬇ (ou CTRL-J) : Vers le bas
- Un signal sonore indique quand la limite est atteinte

- CTRL-N : Permet de redéfinir la largeur
- CTRL-P : Ajoute une colonne à la grille (à droite)
- CTRL-L : Supprime une colonne (à droite)

RÉAFFICHAGE D'UN CARACTÈRE

- CTRL-R : Retracer le dernier caractère créé
- CTRL-A : Restitue le dernier écran de création
- CTRL-O : Affiche les noms des caractères déjà créés
- CTRL-F : Rend cet affichage automatique (mais c'est une bascule, et on l'annule par un second CTRL-F).

On sort de ces deux séquences en tapant la lettre-clé... ou par **ESCAPE**.

DÉPLACEMENT DU CARACTÈRE

Il est possible de mouvoir le caractère à l'intérieur de la grille (en cas d'erreur, récupération du dernier écran par CTRL-A).

- CTRL-D : Un point vers la droite
- CTRL-G : Un point vers la gauche
- CTRL-Z : Un point vers le haut
- CTRL-W : Un point vers le bas

EXEMPLE DE TRACÉ : CTRL-E

FIN DE CRÉATION : ESCAPE

Indiquer ensuite la lettre-clé (ou **ESCAPE** pour revenir)

CATALOG DE DISQUETTE : ?

MENU GÉNÉRAL : Taper CTRL-C

RÉCUPÉRATION DES FONTES GRAPHIQUES TOOL KIT ET CHAT MAUVE

Choisissez l'option 4 du menu, puis **T** pour **TOOL KIT** ou **C** pour **CHAT MAUVE**.

Attention ! le choix **P** ne vous autorise pas à charger une fonte graphique (même personnelle), mais une police de **caractères personnalisés** créée suivant les normes de **CARAC.1.0**.

Par contre, une **FONT**e comme **FAIRE.F1**, (n°1 de T.M.), peut être considérée comme un produit **TOOL KIT**. Pour l'utiliser, la renommer en lui ajoutant ".SET"... et ça marchera !

En cas d'erreur de lecture, tapez **GOTO 10000**, puis renouvelez votre tentative. Naturellement, les caractères créés à partir de la disquette **TOOL KIT** ne comportent pas de lettres accentuées, mais vous pourrez les modifier à volonté.

D'ailleurs, après transformation de la fonte, le programme choisit pour vous l'option "création-modification"... et vous vous trouvez en présence d'une grille de 8 pts de large.

Tapez **CTRL-O** et la liste des caractères disponibles apparaît. Vous pourrez les passer successivement en revue en répondant par la lettre-clé, puis par un nouveau **CTRL-O**, et ainsi de suite.

En cas de modification, la séquence se terminera par **ESCAPE**.

Il peut être intéressant, immédiatement après la création, de revenir au menu général (CTRL-C), puis de demander l'impression du fichier. En cas de plantage, n'accusez pas votre imprimante, mais les codes d'un caractère. Il faudra vous faire une raison et les passer tous, avec impression, caractère par caractère.

Si vous disposez d'une police de 95 ou 96 caractères, le programme vous imprime une page complète. Dans le cas contraire, il vous fournit les caractères disponibles, les uns à la suite des autres, en mode normal, puis expansé.

N'oubliez pas de sauver votre fonte (Menu général). Même si le titre apparaît sur l'écran, n'oubliez pas non plus de le valider en le retapant.

Et en cas de pépin : **GOTO 10000**.

CARACTÈRES PERSONNALISÉS

sur ImageWriter et DMP

Avez-vous déjà essayé de créer des caractères personnalisés, sur votre imprimante ImageWriter ou sur votre "vieille" DMP (notez que nos programmes fonctionnent aussi sur l'ImageWriter II, et probablement sur divers compatibles... mais sans aucune garantie) ? Ce n'est pas très facile... sauf quand on sait... bien sûr, et encore, certains problèmes subsistent.

Nous vous présentons ici une batterie de programmes qui devraient normalement répondre à votre attente. Ils vont vous paraître longs. Nous aurions certes pu les abréger, mais au détriment de la présentation. La plupart d'entre nous s'adonnent à la programmation par jeu plus que par nécessité. Pourquoi ne pas se faire plaisir en déplaçant par exemple des écrans ?

Personne n'interdit aux plus malins de se mettre à plusieurs pour "rentrer" un programme... ou encore d'utiliser les compétences de leur charmante épouse (ou époux... car nous connaissons aussi d'excellentes adeptes de la programmation). Et que l'on ne vienne pas nous accuser de publier des "trucs interminables faits pour obliger les bonnes gens à acheter nos disquettes" (dixit un lecteur pas content qui nous précise les avoir à l'œil... chez son dentiste. En voilà au moins un qui n'a pas la dent contre... excusez-moi, je divague !).

la syntaxe ou les termes choisis pour les commentaires, sauf si votre texte est PLUS COURT que le nôtre... car l'arrivée se fait dans un mouchoir de poche ! Consultez la fiche numéro 6 de *Tremplin Micro* avant de lancer le RUN décisif. Il est indispensable de connaître un certain nombre de "CTRL" pour tirer le meilleur parti possible de cet utilitaire. Nous l'avons longuement testé (trois fontes spéciales, dont une de gros caractères, figurent sur la disquette numéro 6, et ont FACILEMENT été réalisées avec ce programme), mais des erreurs peuvent subsister.

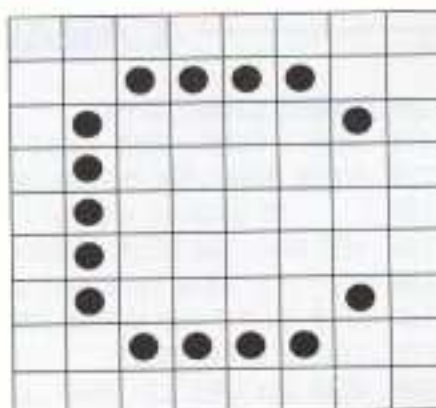
Dans tous les cas, un GOTO 10000 vous permettra de retrouver votre MENU.

2. CARAC.C

C'est la routine en langage machine. Le programme source est sur la disquette (encore elle, excusez-moi). Il n'était vraiment pas possible de le publier dans ce numéro, déjà très complet (nous avons dû ajouter 4 pages au tout dernier moment !). Inutile de préciser que chaque octet y est non seulement important, mais indispensable. CARAC.C accélère considérablement le programme en Basic. C'est sa raison d'être.

3. CARAC.CHARG

Quand vous aurez réalisé votre première fonte de caractères, vous utili-



serez CARAC.CHARG (Basic) et CARAC.CH (LM, page suivante) pour la... charger dans votre imprimante. Normalement, la batterie de programmes des "Caractères personnalisés" prend fin ici, mais nous avons voulu être plus complets avec :

CARAC.VISU Deux courtes routines (la première en Basic et la seconde en langage machine) vous permettant de visualiser une à une (peut-être avant de les transformer en caractères personnalisés), les polices de caractères figurant sur la disquette TOOL KIT APPLE (DOS 3.3.). Efficace et pratique.

CADRE SPÉCIAL (Application)

On peut assurément tracer un cadre... une grille, etc. en utilisant des caractères personnalisés, et sans aucune difficulté. Si vous en doutez, tapez les quelques lignes de la page 33... et un RUN saura définitivement vous convaincre.

DERNIÈRE RECOMMANDATION

Réglez éventuellement vos interrupteurs DIP en conformité avec les indications de la page 20 du présent numéro. Graphisme ou caractères personnalisés... c'est le même combat !

Et maintenant, à vous de jouer, et que l'on ne vienne pas me reprocher mon extrême concision ! G.-H.

1. CARACTÈRES PERSONNALISÉS (Basic)

Vérifiez minutieusement votre programme avant de le lancer. Tout y est important, et notamment la valeur de LOMEM. Ne modifiez pas



Ce texte est édité avec une fonte SPECIALE, sur la nouvelle imprimante ImageWriter II d'Apple en caractères semi-compressés.

CARACTÈRES PERSONNALISÉS 1

```

100 LOMEM: 22485
110 TEXT : PRINT CHR# (21): HOME
120 GOSUB 3890: REM INITIALISATION
130 :
140 REM *** EN GUISE DE MENU GENERAL ***
150 :
160 IF R < > 3 AND N THEN 790
170 TEXT : GOSUB 3560
180 VTAB 16: CALL LS: HTAB 13: CALL G: PRINT "
  C H O I X -> "; GET R#: IF R# < "1" OR R#
  > "8" THEN PRINT : GOTO 180
190 FLASH : PRINT R#: NORMAL
200 IF R# = "?" THEN GOSUB 3390: GOTO 180
210 ON VAL (R#) GOTO 790,460,480,2470,220,240
  ,3290,250
220 IF NOT BI AND NOT CF THEN ME# = "PAS DE
  FICHER BINAIRE EN MEMOIRE": GOSUB 340: GOS
  UB 290: GOTO 180
230 PI = 1: GOTO 1590
240 GOSUB 3390: GOTO 160
250 TEXT : HOME : GOSUB 350: PRINT "GOTO 1000
  0 POUR REDEMARRER": VTAB 20: END
260 :
270 REM *** BOUCLE D'ATTENTE & WAIT ***
280 :
290 FOR I = 1 TO 1500: NEXT : RETURN
300 CALL G: POKE -16368,0: WAIT -16384,120
  ,127: POKE -16368,0: PW = PEEK (-16384)
  : RETURN
310 :
320 REM *** AFFICHAGES REPETITIFS ***
330 :
340 VTAB 19: CALL G: CALL LS: HTAB INT (41 -
  LEN (ME#)) / 2: PRINT ME#: RETURN
350 VTAB 21: CALL FL: RETURN
360 FOR I = 5 TO 13: VTAB I: PRINT LEFT# (TL#
  ,N) + LEFT# (V#,17 - N): NEXT : RETURN
370 R# = ".": IF R = 12 THEN R# = " "
380 FOR I = 5 TO 13: VTAB I: HTAB N: PRINT R#:
  NEXT : RETURN
390 :
400 REM *** FENETRE EN BAS DE L'ECRAN ***
410 :
420 POKE 35,24: GOSUB 3570: VTAB 16: CALL LS:
  PRINT : PRINT T#: PRINT : RETURN
430 :
440 REM *** PLUS OU MOINS 1 CARACTERE ***
450 :
460 IF CF THEN CF = CF - 1: FS = FS - N - 2: CA
  LL G: CALL G: GOSUB 530
470 GOTO 180
480 IF CF < 96 THEN CF = CF + 1: FS = FS + N +
  2: CALL G: CALL G: GOSUB 530
490 GOTO 180
500 :
510 REM *** LE DERNIER CARACTERE EST: ***
520 :
530 VTAB 20: CALL FL: PRINT " DERNIER CARA
  CTERE : "; INVERSE : PRINT CF" "; NORMAL
  : IF C(CF,1) THEN PRINT "(" CHR# (C(CF,1))
  ")": GOTO 550
540 INVERSE : CALL G: PRINT "NON CREE": NORMAL
550 GOSUB 290: RETURN
560 :
570 REM *** AIGUILLES DU HAUT OU DU BAS ***
580 :
590 X = 0: Y = 0: HM = 0: PL = 1
600 FOR J = 0 TO N - 1
610 IF PEEK (A(1) + J) < > 174 THEN X = 1
620 IF PEEK (A(9) + J) < > 174 THEN Y = 1
630 NEXT
640 IF X + Y = 2 THEN VTAB 17: PRINT : CALL G
  : CALL FL: PRINT "T R O P": CALL FL: INVERS
  E : PRINT " HAUTE ": NORMAL : HM = 1: GOTO 9
  30
650 IF Y THEN PL = 2: POKE 6,N: POKE 24,0: CAL
  L 13114
660 GOTO 1240
670 :
680 REM *** DEPLACER LE CARACTERE ***
690 :
700 GOSUB 3510: POKE 6,NM: ES = 1
710 IF R = 4 THEN CALL 13194
720 IF R = 7 THEN CALL 13225
730 IF R = 23 THEN POKE 24,1: CALL 13114
740 IF R = 26 THEN POKE 24,0: CALL 13114
750 GOTO 930
760 :
770 REM *** TRACE DU CARACTERE ***
780 :
790 GOSUB 3750
800 IF NOT CF AND NOT N THEN 840
810 IF N THEN 900
820 N = ASC ( MID# (CO#,6,1)) - 64: IF N > 16
  THEN N = N - 32
830 GOTO 900
840 VTAB 15: HTAB 1: CALL G: PRINT "LARGEUR 7-
  16 "; R# = ""
850 GET B#: B = ASC (B#): IF B = 27 THEN N = 1
  6: GOTO 910
860 IF B > 47 AND B < 57 AND B < > 13 THEN P
  RINT B#: R# = R# + B#: GOTO 850
870 N = VAL (R#): L = LEN (CO#)

```


CARACTÈRES PERSONNALISÉS 2

```

880 IF L AND NOT N THEN N = L - 7
890 IF N < 7 OR N > 16 THEN 840
900 IF N < > 16 THEN FOR I = 5 TO 14: VTAB I
    : HTAB N + 1: PRINT LEFT$(V$,16 - N): NEX
    T
910 PRINT : VTAB 15: CALL FL: PRINT "L-> " : I
    NVERSE : PRINT N : NORMAL : PRINT " *AL$
920 IF F THEN GOSUB 2110
930 H = 1: V = 5
940 REM ONERRGOTO 4140
950 PRINT " " : VTAB V: HTAB H: GET R$
960 CALL 13474: R = ASC (R$): IF R = 32 THEN 9
    50
970 IF R = 21 AND H = N THEN 1200
980 IF R = 11 AND V = 5 THEN 1200
990 IF R = 10 AND V = 13 THEN 1200
1000 IF R = 8 AND H = 1 THEN 1200
1010 IF R = 21 THEN H = H + 1
1020 IF R = 8 THEN H = H - 1
1030 IF R = 10 THEN V = V + 1
1040 IF R = 11 THEN V = V - 1
1050 IF R = 1 AND ES THEN ES = 0: GOSUB 3520: G
    OTO 930
1060 IF R = 3 THEN ES = 1: GOSUB 3510: GOTO 160
1070 IF R = 4 THEN NM = N - 1: GOTO 700
1080 IF R = 5 THEN GOSUB 3510: GOSUB 3430: GOS
    UB 3520
1090 IF R = 6 THEN F = ABS (F - 1): IF F THEN
    R = 15
1100 IF R = 7 THEN NM = N: GOTO 700
1110 IF R = 12 AND N > 7 THEN GOSUB 370: N = N
    - 1: GOTO 910
1120 IF R = 14 THEN GOSUB 3540: GOTO 840
1130 IF R = 15 THEN GOSUB 2110
1140 IF R = 16 AND N < 16 THEN N = N + 1: GOSUB
    370: GOTO 910
1150 IF R = 18 THEN GOSUB 2220
1160 IF R = 23 OR R = 26 THEN NM = N: GOTO 700
1170 IF R = 27 THEN GOSUB 3510: ES = 1: GOTO 59
    0
1180 IF R = 63 THEN GOSUB 3390: GOTO 930
1190 GOTO 950
1200 POKE 6,50: POKE 7,1: CALL 13047: POKE 6,5:
    POKE 7,3: CALL 13047: GOTO 950: REM AVERTI
    SSEMENT SONORE (LIMITES DE LA GRILLE)
1210 :
1220 REM *** OU IL EST QUESTION DE 0101 ***
1230 :
1240 GOSUB 3510: REM SAUVEGARDE ECRAN
1250 FOR I = 5 TO 19: VTAB I: HTAB N + 2: CALL
    FL: NEXT
1260 FOR I = 1 TO N

```

Eliminer REM quand le programme est au point

```

1270 Z = USR (A(I)): POKE 8, PEEK (6): POKE 9,
    PEEK (7)
1280 POKE 6,1 - 1: POKE 7,20
1290 CALL 13070: VTAB 4 + I: HTAB 30
1300 B(I) = PEEK (24)
1310 IF B(I) = 13 THEN B(I) = 14
1320 PRINT " ) CHR$( "B(I) )"
1330 NEXT
1340 :
1350 REM *** QUEL SERA LE CODE D'APPEL? ***
1360 :
1370 POKE 34,16: POKE 33,17: POKE 35,22: VTAB 1
    8: HTAB 1: CALL L5: CALL G: PRINT "LETTRE-C
    LE -> " :
1380 GET L$: PRINT
1390 POKE 33,40: POKE 34,3
1400 IF L$ = CHR$(27) THEN GOSUB 3520: GOTO
    930
1410 IF L$ < CHR$(32) OR L$ > CHR$(159) THE
    N 1370
1420 AL$ = L$
1430 PRINT AL$ " CHR$( "ASC (L$) )" : GOSUB 290
1440 :
1450 REM *** ET LE CODE IW-DMP, DONC ? ***
1460 :
1470 IF PW = 27 THEN 1370
1480 IF N > 8 THEN 1500
1490 CO$ = ES$ + "-": GOTO 1510
1500 CO$ = IP$
1510 CO$ = CO$ + ES$ + CHR$(73) + L$
1520 CO$ = CO$ + CHR$(N + 32 + 32 * PL)
1530 FOR I = 1 TO N: CO$ = CO$ + CHR$(B(I)): N
    EXT : CO$ = CO$ + D$: X = 0
1540 IF ND < > 0 THEN GOSUB 2350
1550 CF = CF + 1: GOSUB 3650
1560 :
1570 REM *** FAISONS BONNE IMPRESSION! ***
1580 :
1590 PI = 0
1600 GOSUB 420
1610 VTAB 17: PRINT : INVERSE : PRINT "EDITION"
    : NORMAL
1620 GOSUB 350: CALL G: PRINT " (C)ARACTERE -
    (R)IEN - (T)OUT ? "
1630 GOSUB 380
1640 IF PW = 84 THEN PI = 1: GOTO 1670: REM EDI
    TION DU FICHER ACTUEL
1650 IF PW = 82 THEN 2060
1660 IF PW < > 67 THEN 1620
1670 GOSUB 3500
1680 IF PI THEN 1700
1690 :

```


CARACTÈRES PERSONNALISÉS 3

```

1700 PRINT : PRINT D$*PRE1": PRINT
1710 PRINT CHR$(9)*80N": PRINT CHR$(15);
1720 PRINT ES$*E";ES$"L010"; CHR$(9)"2"
1730 X = 0: REM POINTEUR
1740 IF NOT PI THEN 1990: REM IMPRESSION D'UN
    SEUL CARACTERE
1750 LG = FS + 1 - F1;X = 0
1760 Z = USR (LG): POKE 8, PEEK (6): POKE 9, PE
    EK (7)
1770 CALL 13006: PRINT D$: PRINT CHR$(24)
1780 :
1790 REM *** LA,TOUT LE FICHER Y PASSE! ***
1800 :
1810 PRINT CHR$(14);ES$*X": PRINT "FONTE "F$;
    ES$*Y": CHR$(15): PRINT
1820 IF CF > = 95 THEN 1870
1825 PRINT ES$"D 0";
1830 PRINT IP$;
1840 FOR I = 1 TO 95: IF C(I,1) THEN PRINT CH
    R$(C(I,1));
1850 NEXT : PRINT ;X = X + 1: IF X < 2 THEN PR
    INT CHR$(14): GOTO 1840
1860 X = 0: GOTO 1930
1870 H = CF / (6 - X * 2): IF H ( > ) INT (CF /
    6) THEN H = INT (H + 1)
1880 FOR I = 1 TO H:S = 31: FOR J = 1 TO (6 - X
    * 2)
1890 IF I + S > 126 THEN 1910
1900 PRINT IN$; CHR$(I + S);" -> ";IP$; CHR$(
    I + S);: IF J < (6 - X * 2) THEN PRINT "
    ";
1910 S = S + H: NEXT : PRINT : NEXT
1920 X = X + 1: IF X < 2 THEN PRINT CHR$(15);
    ES$"L005"; CHR$(14): GOTO 1870
1930 PRINT IN$: PRINT CHR$(15): PRINT D$*PRE0
    "
1940 :
1950 GOTO 2050
1960 :
1970 REM *** ICI, SEULEMENT 1 CARACTERE! ***
1980 :
1990 FOR I = 1 TO LEN (C0$): POKE 6, ASC ( MID
    $(C0$,I,1)): CALL 13000: NEXT
2000 FOR I = 1 TO 3
2010 IF L$ > CHR$(127) THEN PRINT ES$ + "*"
    ;L$;IN$;: GOTO 2030
2020 PRINT IP$;L$;IN$;
2030 PRINT " ";: NEXT
2040 X = X + 1: IF X < 2 THEN PRINT CHR$(14);
    : GOTO 2000
2050 PRINT : PRINT ES$"c": PRINT D$*PRE0"
2060 POKE 34,4: HOME :X = 0

2070 GOTO 160
2080 :
2090 REM *** CARACTERE ANCIEN ***
2100 :
2110 IF NOT CF THEN ME$ = "PAS DE CARACTERE EN
    MEMOIRE": GOSUB 340: GOSUB 290: RETURN
2120 VTAB 15: CALL 13415
2130 PRINT : VTAB 19: CALL G: INVERSE : PRINT "
    CARACTERE DESIRE";: NORMAL : PRINT ") ";: G
    ET R$: PRINT
2140 IF R$ = ES$ THEN POKE 35,22: VTAB 15: CAL
    L LS: RETURN
2150 IF R$ < " " THEN 2130
2160 FOR I = 1 TO CF: IF C(I,1) = ASC (R$) THE
    N ND = I: GOTO 2220
2170 NEXT
2180 VTAB 15: PRINT R$* EST INTROUVABLE": RETUR
    N
2190 :
2200 REM *** REAFFICHAGE DU CARACTERE ***
2210 :
2220 IF NOT ND THEN ND = CF
2230 P = PEEK (C(ND,0) + 1) - 64: IF P > 16 THE
    N P = P - 32
2240 IF N < > P THEN N = P: GOSUB 360
2250 FOR I = 1 TO P
2260 X = PEEK (C(ND,0) + I + 1):Y = INT (X / 1
    6):Z = X - Y * 16
2270 L$(I) = MID$(BI$,1 + (4 * Y),4) + MID$(
    BI$,1 + (4 * Z),4)
2280 NEXT
2290 VTAB 5: HTAB 1
2300 FOR I = 8 TO 1 STEP - 1: FOR J = 1 TO P:
    PRINT MID$(L$(J),I,1);: NEXT : PRINT : NE
    XT
2310 VTAB 14: CALL LS: RETURN
2320 :
2330 REM *** REMPLACEMENT D'UN CARACTERE ***
2340 :
2350 FOR I = 1 TO CF: IF C(I,1) = ASC (L$) THE
    N NE = I: GOTO 2370
2360 NEXT :NE = I: GOTO 2420
2370 GOSUB 420: PRINT : VTAB 18: CALL G: PRINT
    "Remplacement de "L$" (return) ? ";: GET R$
    : PRINT : IF R$ ( > ) CHR$(13) THEN POP :
    GOTO 1370
2380 L = LEN (C0$) - 1
2390 FOR I = 5 TO L
2400 POKE C(NE,0) + I - 5, ASC ( MID$(C0$,I,1)
    )
2410 NEXT : POP : GOTO 1590
2420 IF NE > CF THEN GOSUB 420: CALL G: PRINT

```


CARACTÈRES PERSONNALISÉS 4

```

: VTAB 18: PRINT " NOUVEAU CARACTERE: "L$
("NE") - RET=OUI "; GET R$: IF R$ = CHR$
(13) THEN RETURN
2430 POP : GOTO 1590
2440 :
2450 REM *** ATTENTION! VA-T-ON EFFACER? ***
2460 :
2470 IF NOT B1 THEN ME$ = "PAS DE FICHER BINA
IRE EN MEMOIRE": GOSUB 340: GOTO 2560
2480 ME$ = "ON EFFACE LE FICHER EN MEMOIRE (O/N
)": GOSUB 340
2490 GOSUB 300
2500 IF PW = 79 THEN 2560
2510 IF PW = 78 THEN 160
2520 CALL - 198: GOTO 2490
2530 :
2540 REM *** CHARGEMENT D'UNE FONTE ***
2550 :
2560 POKE 35,24
2570 GOSUB 350: PRINT "<T>OOL KIT <C>HAT MAUVE
<P>ERSONNEL -> "; INVERSE : HTAB 2: PRINT
"T"; HTAB 13: PRINT "C"; HTAB 26: PRINT "
P"; NORMAL : HTAB 39
2580 CALL G: GET R$: PRINT ""
2590 IF R$ = "?" THEN GOSUB 3390: GOTO 2570
2600 CM = 0: FT = 0: FP = 0: IF R$ = "C" THEN CM
= 1: GOTO 2640
2610 IF R$ = "T" THEN FT = 1: GOTO 2640
2620 IF R$ = "P" THEN FP = 1: GOTO 2640
2630 GOTO 160
2640 PRINT : GOSUB 350: CALL G: INPUT "FONTE DE
SIREE ? "; F$
2650 IF F$ = "" THEN 2570
2660 GOSUB 2770: IF R$ < > CHR$ (13) THEN 257
0
2670 A$ = ",A$3500": IF FP THEN A$ = ",A$4000"
2680 DE = 1: ONERR GOTO 4140
2690 POKE 34,22: VTAB 20: PRINT : PRINT D$"BLOA
D*F$ + A$:BI = 1
2700 POKE 216,0: POKE 206,0: POKE 207,53: REM A
DRESSE FONTE
2710 IF CM THEN DF = 32: GOTO 2900
2720 DF = 0: IF FT THEN 2890
2730 IF FP THEN 3110: REM MATERIALISATION
2740 GOTO 160
2750 :
2760 REM *** SUFFIXE <SET> <FONT> <FP> ***
2770 :
2780 IF CM AND RIGHT$ (F$,5) < > ".FONT" THEN
F$ = F$ + ".FONT": REM CHAT MAUVE :GOTO500
8
2790 IF (FT) AND RIGHT$ (F$,4) < > ".SET" THE
N F$ = F$ + ".SET": REM TOOL KIT APPLE
2800 GOSUB 350: CALL G: PRINT "CHOIX: "; INVER
SE : PRINT F$: NORMAL : PRINT " (RETURN=DU
I)": GET R$: PRINT : RETURN
2810 :
2820 IF RIGHT$ (F$,5) = ".FONT" THEN F$ = LEF
T$ (F$, LEN (F$) - 5) + ".CM"
2830 IF RIGHT$ (F$,4) = ".SET" THEN F$ = LEFT
$ (F$, LEN (F$) - 4) + ".FT"
2840 IF (FP) AND RIGHT$ (F$,3) = ".FP" THEN F$
= LEFT$ (F$, LEN (F$) - 3) + ".FP"
2850 GOTO 2800
2860 :
2870 REM *** FONTES D'APRES CHAT MAUVE ***
2880 :
2890 POKE 13344,102: CALL 13310: POKE 13344,30:
GOTO 2910
2900 CALL 13310
2910 :
2920 GOSUB 420: INVERSE : PRINT "* APPLE TRAVAI
LLE POUR VOUS: PATIENCE *": NORMAL : PRINT
2930 A1 = 13567 + 8 * DF: C = 32: L = 8: LI = 14560
: IF FT THEN LI = 14319
2940 POKE AP,27: POKE AP + 1,45: POKE AP + 2,27
: POKE AP + 3,73: A2 = AP + 4
2950 POKE A2,C: POKE A2 + 1,64 + L: FOR I = A2
+ 2 TO A2 + 9: POKE I,0: NEXT
2960 P = A2 + 1: Z = USR (P): POKE 8, PEEK (6):
PEKE 9, PEEK (7)
2970 Z = USR (A1): POKE 24,L: POKE 25,L + 2
2980 H = 64: C = C + 1: A2 = A2 + 10: A1 = A1 + L
2990 CALL 13260
3000 IF PEEK (26) > 127 THEN H = 96
3010 POKE A2,C: POKE A2 + 1,H + L
3020 IF A1 < LI THEN 2980
3030 POKE A2 + 10,4: REM FIN DE FICHER
3040 VTAB 22: PRINT " FONTE PRETE A EDITER & S
AUVEGARDER"
3050 CF = C - 31: L = A2 + 10 - (AP - 1): Z = USR
(L): POKE LB, PEEK (6): POKE LB + 1, PEEK
(7)
3060 CM = 0: FT = 0: FP = 1: F1 = AP
3070 GOTO 3100
3080 :
3090 REM *** PETITE ANALYSE DU FICHER ***
3100 :
3110 LG = PEEK (LB) + PEEK (LB + 1) * 256: FS =
F1 - 1 + LG
3120 IF PEEK (FS) = 4 THEN FS = FS - 1
3130 N1 = F1 + 4: X = 0: CF = 0
3140 CF = CF + 1: C(CF,0) = N1: C(CF,1) = PEEK (N
1)

```


CARACTÈRES PERSONNALISÉS 5

```

3150 R = PEEK (N1 + 1):R = R - 64: IF R > 16 TH
EN R = R - 32
3160 N1 = N1 + R + 2: IF N1 < FS - 1 THEN 3140
3170 VTAB 17: CALL LS: PRINT : PRINT "FICHER "
;: INVERSE : PRINT F$:; NORMAL : PRINT " AV
EC ";: INVERSE : PRINT CF:; NORMAL : PRINT
" CARACTERES"
3180 N = R: GOSUB 290: GOTO 160
3190 :
3200 REM *** ANNULATION D'UN CARACTERE ***
3210 :
3220 FOR I = C(AN,0) + 2 TO FS: POKE I,0: NEXT
: IF AN = CF THEN FS = C(AN,0) - 1:CF = CF
- 1: GOTO 3240
3230 RETURN
3240 IF NOT CF THEN FS = F1 - 1
3250 RETURN
3260 :
3270 REM *** COPIE SUR UNE DISQUETTE ***
3280 :
3290 GOSUB 2820:FM$ = F$: IF R$ ( ) CHR$ (13)
THEN 160
3300 GOSUB 350: PRINT "TITRE (?=CAT) ";FM$:; HT
AB 15: CALL G: INPUT "":F$: IF F$ = "" THEN
160
3310 IF F$ = "?" THEN GOSUB 3390: GOTO 3290
3320 IF PEEK (FS) ( ) 4 THEN FS = FS + 1: POK
E FS,4
3330 GOSUB 350: PRINT "BSAVE "F$,"A$4000,L":;L
= FS + 1 - F1: PRINT L
3340 PRINT : PRINT CHR$ (4)"BSAVE"F$,"A$4000,L
"L
3350 GOTO 180
3360 :
3370 REM *** CATALOGUE DE DISQUETTE SUP? ***
3380 :
3390 TEXT : GOSUB 3570: HOME : PRINT : PRINT D$
"CAT": PRINT : INVERSE : PRINT "RETURN SUP"
;: NORMAL : INPUT R$: GOSUB 3580: RETURN
3400 : REM "CATALOG" SOUS DOS 3.3
3410 E1 = M1:E2 = M2:E3 = 17400: GOTO 3590
3420 :
3430 VA = 4: FOR I = 1 TO LEN (E$) STEP 8:VA =
VA + 1: VTAB VA: HTAB 20: CALL FL: PRINT M
ID$ (E$,I,8): NEXT
3440 PRINT : GOSUB 3470: RETURN
3450 VTAB 16: HTAB 9: PRINT "PRESSEZ UNE TOUCHE
SUP ";
3460 CALL G: GET R$: PRINT : VTAB 16: CALL FL:
RETURN
3470 GOSUB 300: RETURN
3480 :
3490 REM *** TRANSFERT DES DIVERS ECRANS ***
3500 :
3510 E1 = M1:E2 = M2:E3 = 21468: GOTO 3590
3520 E1 = 21468:E2 = 22484:E3 = M1: GOTO 3590
3530 E1 = M1:E2 = M2:E3 = 18417: GOTO 3590
3540 E1 = 18417:E2 = 19433:E3 = M1: GOTO 3590
3550 E1 = M1:E2 = M2:E3 = 19434: GOTO 3590
3560 E1 = 19434:E2 = 20450:E3 = M1: GOTO 3590
3570 E1 = M1:E2 = M2:E3 = 20451: GOTO 3590
3580 E1 = 20451:E2 = 21467:E3 = M1
3590 E = E1:L0 = 60: GOSUB 3600:E = E2:L0 = 62:
GOSUB 3600:E = E3:L0 = 66: GOSUB 3600: CALL
13517: RETURN
3600 Z =USR (E): POKE L0, PEEK (6): POKE L0 +
1, PEEK (7): RETURN
3610 :
3620 REM *** CONSTITUTION DU FICHER ***
3630 :
3640 F1 = AP:FS = F1 - 1: RETURN
3650 C(CF,0) = FS + 1
3660 IF CF = 1 THEN FOR I = 1 TO 4:FS = FS + 1
: POKE FS, ASC ( MID$ (IM$,I,1)): NEXT :C(C
F,0) = C(CF,0) + 4
3670 FOR I = 5 TO 6:FS = FS + 1: POKE FS, ASC (
MID$ (CO$,I,1)): NEXT
3680 FOR I = 1 TO N:FS = FS + 1: POKE FS,B(I):
NEXT
3690 C(CF,1) = ASC ( MID$ (CO$,5,1))
3700 RETURN
3710 :
3720 :
3730 REM *** PRESENTATION ***
3740 :
3750 TEXT : HOME : IF Z THEN GOSUB 3540: RETU
RN
3760 PRINT T$: PRINT T$
3770 VTAB 2: PRINT "* CREATION DE CARACTERES PE
RSONNALISES *"
3780 IF BI THEN GOSUB 360: GOTO 3800
3790 N = 16: GOSUB 360:N = 8
3800 VTAB 23: PRINT T$"DEPLACEMENTS AVEC LES FL
ECHES DE L'APPL":; VTAB 15: POKE 2039,133
3810 POKE 34,4: POKE 35,22
3820 VTAB 6: HTAB 20: PRINT "LA BARRE D'ESPACEM
ENT":; HTAB 20: PRINT "TRACE OU EFFACE LES
*":; HTAB 20: PRINT "FIGURANT LE CARACTERE"
3830 VTAB 10: HTAB 20: PRINT "LARGEUR NORMALE:8
PTS":; HTAB 20: PRINT "LARGEUR MAX:16 POIN
TS":; HTAB 20: PRINT "CTRL-E POUR EXEMPLES
":; HTAB 20: PRINT LEFT$ (T$,21)
3840 IF NOT Z THEN GOSUB 3530:Z = 1
3850 RETURN

```


CARACTÈRES PERSONNALISÉS 6

```

3860 :
3870 REM *** INITIALISATIONS GENERALES ***
3880 :
3890 DIM B(16),L$(16),S$(9),C(128,1),A(16)
3900 D$ = CHR$(4):E$ = CHR$(27)
3910 M1 = 1024:M2 = 2048:AP = 16384: REM ADR POL
    ICE
3920 :
3930 PRINT D$"BLOAD CARAC.C"
3940 CALL 13394: REM INITUSR
3950 IN$ = E$ + "$":IP$ = E$ + "'":IM$ = E$ +
    "+" + E$ + "I"
3960 G = -198:FL = -868:LS = -958
3970 MN$ = "015263748":LB = 48848
3980 MT$ = "NU-MENU-MENU-MENU-MENU-MENU-MENU-MEN
    U-ME"
3990 M$ = "1 CREER CARACTERE 5 EDITER LE FICH
    IER2 MOINS 1 CARACTERE 6 VOIR LE CATALOGUE
    3 PLUS 1 CARACTERE 7 SAUVER LE FICHIER4 C
    HARGER FICHIER 8 FIN DE TRAVAIL"
4000 BI$ = ".....*.....**.....*.*.....*****
    *..**.*.*.....**.....**.....**.....**....."
4010 E$ = ".*.....*.....**.....*.....**
    *...*...*.....**.....*.....**....."
4020 FOR I = 1 TO 16: READ A(I): NEXT I
4030 FOR I = 1 TO 10:T$ = T$ + "-----":V$ = V$ +
    * " :TL$ = TL$ + "....": NEXT :V$ = V$ + *
    *
4040 GOSUB 3640: REM INIT FICHER TRAVAIL
4050 PRINT T$MT$: POKE 34,3
4060 X = 5: FOR I = 1 TO 160 STEP 40:X = X + 2:
    VTAB X: PRINT MID$(M$,I,40): NEXT
4070 INVERSE : FOR I = 1 TO 4: VTAB 5 + I * 2:
    PRINT MID$(MN$,I + 2,1);: HTAB 22: PRINT
    MID$(MN$,1 + I * 2,1): NEXT : NORMAL
4080 GOSUB 3550: REM STOCKAGE ECRAN
4090 GOTO 160
4100 DATA 1536,1664,1792,1920,1064,1192,1320,14
    40,1576,1704,1832,1960,1104,1232,1360,1480
4110 :
4120 REM *** EN CAS D'ERREUR ***
4130 :
4140 ER = PEEK(222): CALL LS: CALL G: INVERSE
    : PRINT "ERREUR "ER;: NORMAL : PRINT " ";:
    GET R$: PRINT : TEXT : HOME : PRINT D$"CATA
    LOG"
4150 PRINT : PRINT T$: INPUT "FONTE DESIREE ? "
    ;F$: IF F$ = "" THEN POKE 216,0: GOSUB 356
    0: GOTO 160
4160 GOSUB 3560: VTAB 19: RESUME
4170 POKE 216,0: GOTO 930
10000 POKE 51,0: GOTO 160

```

CARAC.C

522 octets

Le programme en Basic fait fréquemment appel à diverses routines en langage machine. Celles-ci sont regroupées à partir de l'adresse \$32C8 (13000), sur 522 octets.

Publier le programme source aurait monopolisé 9 pages de la revue d'où son absence (par contre, il figure sur la disquette *Tremplin Micro n°6...* ainsi que plusieurs fontes de caractères personnalisés).

Rappelons que, pour rentrer un programme en langage machine, il suffit de taper CALL—151, suivi de RETURN, puis chacune des lignes ci-contre, un RETURN marquant la fin de chaque ligne.

Pour sortir du moniteur, taper CTRL-C (Contrôle C) et RETURN

SAUVEGARDE : BSAVE CARAC.C, A\$32C8, L522

```

32C8: A5 06 20 ED FD 60 A9 FF
32D0: 85 06 A9 3F 85 07 E6 06
32D8: A5 06 D0 02 E6 07 C6 08
32E0: A5 08 C9 FF D0 06 A5 09
32E8: F0 0C C6 09 A0 00 B1 06
32F0: 20 ED FD 4C D6 32 60 A4
32F8: 06 A6 06 A9 04 20 A8 FC
3300: AD 30 C0 E8 D8 FD 08 D0
3308: F0 C6 07 D0 EA 60 A9 00
3310: 85 18 A9 0C 85 25 20 C1
3318: FB A4 06 B1 28 C9 AA F0
3320: 05 A9 00 18 90 03 38 A9
3328: 31 A4 07 91 08 26 18 C6
3330: 25 E6 07 A5 25 C9 03 D0
3338: D0 60 A5 18 F0 0F A9 0C
3340: 8D 4E 33 A9 C6 8D 5C 33
3348: A9 03 8D 77 33 A9 04 85
3350: 25 20 C1 FB A5 28 05 08
3358: A5 29 85 09 E6 25 A5 25
3360: 20 C1 FB A0 00 B1 28 C9
3368: A0 D0 02 A9 AE 91 08 C8
3370: C4 06 D0 F1 A5 25 C9 0D

```

(suite page 30)

CARAC.C

Suite

```

3378: D8 DA A9 04 8D 4E 33 A9
3380: E6 8D 5C 33 A9 0D 8D 77
3388: 33 60 A9 04 85 25 20 C1
3390: FB A4 06 88 B1 28 C8 91
3398: 28 88 D8 F7 A9 AE 91 28
33A0: E6 25 A5 25 C9 0D D8 E6
33A8: 68 A9 04 85 25 20 C1 FB
33B0: A0 00 C8 B1 28 88 91 28
33B8: C8 C4 06 D8 F5 A9 AE 88
33C0: 91 28 C8 E6 25 A5 25 C9
33C8: 0D D8 E2 68 A5 06 18 65
33D0: 18 85 06 A5 07 69 08 85
33D8: 07 A5 08 18 65 19 85 08
33E0: A5 09 69 08 85 09 A8 08
33E8: 84 1A C8 B1 06 C9 0D D8
33F0: 02 A9 0C 91 08 29 88 85
33F8: 1A C8 08 D8 ED 68 A5 CE
3400: 85 06 A5 CF 85 07 A9 04
3408: 85 08 A8 08 A2 08 B1 06
3410: 95 0A C8 E8 E8 08 D8 38
3418: 98 48 A8 08 A2 08 56 0A
3420: 26 09 E8 E8 08 D8 F7 A5
3428: 09 99 12 08 C8 C8 07 D8
3430: EB A9 08 85 19 68 38 E9
3438: 08 A8 A2 08 85 12 91 06
3440: C8 E8 E8 08 D8 F6 A2 08
3448: C8 08 D8 C2 E6 07 C6 08
3450: D8 BC A9 4C 85 0A A9 5F
3458: 85 08 A9 34 85 0C 68 28
3460: 52 E7 85 07 84 06 68 A9
3468: 04 85 06 A9 48 85 07 A8
3470: 08 B1 06 C9 28 98 2A C9
3478: 7F B8 26 48 A8 01 B1 06
3480: 38 E9 48 C9 11 98 03 38
3488: E9 28 18 69 02 65 06 85
3490: 06 A5 07 69 08 85 07 68
3498: 18 69 88 28 ED FD 4C 6F
34A0: 34 68 AD 08 C8 C9 28 F8
34A8: 01 68 A5 25 20 C1 FB A5
34B0: 28 18 65 24 85 28 A5 29
34B8: 69 08 85 29 A8 08 B1 28
34C0: C9 AA F8 04 A9 AA D8 82
34C8: A9 AE 91 28 68 A8 08 4C
34D0: 2C FE
    
```

VÉRIFICATION :

```
T = 0 : FOR I = 13000 TO 12999 +
522 : T = T + PEEK(I) : NEXT : PRINT T
```

Le résultat doit être **58641**

● CARAC.CHARG

```

100 TEXT : NORMAL : PRINT CHR$(21): HOME
110 D$ = CHR$(4):E$ = CHR$(27)
120 PRINT D$"BLOAD CARAC.CH"
130 VTAB 12: PRINT "TITRE DE LA FONTE A CH
ARGER (?=CATALOG)"
140 VTAB 14: INPUT "-> ",F$: IF F$ = "?" T
HEN HOME : PRINT D$"CATALOG": GET R$:
PRINT : RUN
150 ONERR GOTO 270
160 PRINT : PRINT D$"BLOAD"F$,A$4000"
170 VTAB 12: PRINT "ALLUMEZ L'IMPRIMANTE (
DMP/IMAGE WRITER)"
180 ONERR GOTO 280
190 PRINT D$"PR#1": PRINT E$"N": PRINT C
HR$(9)"80N": PRINT CHR$(15), CHR$(9
)"Z"
200 CALL 768
210 PRINT D$: PRINT E$"":
220 FOR I = 33 TO 127: PRINT CHR$(I): N
EXT : PRINT
230 X = X + 1: IF X < 2 THEN PRINT CHR$(
14): GOTO 220
    
```

Chargement d'une police de caractères personnalisés

BASIC :
CARAC.CHARG
L.M. :
CARAC.CH

```

240 PRINT CHR$(15),E$"": PRINT : PRINT
D$"PR#0"
250 HOME : VTAB 12: PRINT "VOTRE FONTE ",:
INVERSE : PRINT F$: NORMAL : PRINT "
EST INSTALLEE"
260 CALL - 198: POKE - 16368,0: WAIT -
16384,128: HOME : END
270 POKE 216,0: VTAB 22: INVERSE : PRINT "
A REJOUER! ": NORMAL : CALL - 198: GO
TO 140
280 POKE 216,0: VTAB 22: INVERSE : PRINT "
IL Y A UN PROBLEME ": NORMAL
290 GOTO 260
    
```


CARACTÈRES PERSONNALISÉS 8

ASSEMBLAGE PAR ProCODE

Si vous ne possédez pas d'assembleur, seule cette partie vous concerne.

0300:	A9 FF
0302:	85 06
0304:	A9 3F
0306:	85 07
0308:	AD 60 AA
030B:	85 08
030D:	AD 61 AA
0310:	85 09
0312:	E6 06
0314:	A5 06
0316:	D0 02
0318:	E6 07
031A:	C6 08
031C:	A5 08
031E:	C9 FF
0320:	D0 06
0322:	A5 09
0324:	F0 0C
0326:	C6 09
0328:	A0 00
032A:	B1 06
032C:	20 ED FD
032F:	4C 12 03
0332:	60

```

0 *****
1 * CHARGEMENT D'UNE POLICE DE CARACTÈRES PERSONNALISÉS *
2 *-----*
3 * FONTES ISSUES DU PROGRAMME CARAC DE TREMLIN MICRO *
4 * Imprimantes DMP et Image Writer Apple *
5 *****
6 *
7 ORG $300
8 *
9 LBL EQU $AA60 ; Contient la long du fic B
10 * $BEC8-BEC9 avec ProDos
11 COUT EQU $FDED
12 *
13 LDA £$FF ; Adresse de la fonte moins 1
14 STA $06
15 LDA £$3F
16 STA $07
17 *
18 LDA LBL ; Longueur partie basse
19 STA $08
20 LDA LBL+1 ; Longueur partie haute
21 STA $09
22 *
23 INC INC $06 ; Plus 1 pour l'adresse
24 LDA $06
25 BNE Dec ; Si ce n'est pas 0
26 INC $07 ; Si c'était 0, par contre
27 DEC DEC $08
28 LDA $08
29 CMP £$FF ; Est-ce $FF ou non ?
30 BNE Non
31 LDA $09
32 BEQ Duf ; Si 0, terminé
33 DEC $09 ; sinon on décrémente
34 NON LDY £$00 ; Pour adressage indirect
35 LDA ($06),Y
36 JSR COUT ; Transmission imprimante
37 JMP Inc ; Si ce n'était pas fini
38 OUF RTS
39 *-----*
    
```

● Pour terminer : BSAVE CARAC.CH, A\$300, L\$33

Table des symboles ordre alphabétique

-COUT.....\$FDED	-DEC.....\$031A	-INC.....\$0312	-LBL.....\$AA60
-NON.....\$0328	-OUF.....\$0332		

Pour vérifier l'exactitude des données, tapez en direct la ligne suivante (sans oublier un RETURN, bien entendu) :

T = 0 : FOR I = 768 TO 767 + 51 : T = T + PEEK(I) : PRINT T Le résultat devra être : 5691

CARAC.VISU

BASIC : CARAC.VISU

L.M. : CARAC.VIS

CE mini-programme (n'oubliez pas les 126 octets de la routine en langage machine) vous permettra de visualiser les polices de caractères de votre disquette TOOL KIT (Applesoft Programmer's assistant). Si vous possédez une version récente, confrontez les titres de son catalogue et ceux indiqués dans notre ligne de données (250).

```

100 TEXT : NORMAL : PRINT CHR# (21): HOME
110 PRINT CHR# (4)"BLORD CARAC.VIS"
120 ONERR GOTO 310
130 VTAB 12: INVERSE : PRINT " VISU.FONTE "; NORMAL
140 VTAB 15: PRINT " Pour visualiser les fontes du Tool Kit"
150 PRINT : PRINT "Taper <R> pour visualisation automatique"
160 GOSUB 240
170 IF PEEK (49152) = 65 THEN 260
180 VTAB 23: PRINT : PRINT : INPUT "FONTE ? ";F#
190 IF F# = "" THEN 230
200 IF F# = "?" THEN TEXT : HOME : PRINT CHR# (4)"CATALOG": PRINT : PRINT
    : GOTO 160
210 PRINT : PRINT CHR# (4)"BLORD"F#",R#4000"
220 CALL 768: GOTO 180
230 TEXT : HOME : END
240 VTAB 21: HTAB 5: INVERSE : PRINT " PRESSER UNE TOUCHE QUELCONQUE ";: NO
    RMAL : CALL 887: RETURN
250 DATA BLIPPO, BLACK, BYTE, COLOSSAL, COUNT, CYRILLIC, ESPERANTO, FLOW, GOTHIC, GR
    APHIC, GREEK, KATAKANA, MIRROR, OUTLINE, PINOCCHIO, PUDGY, ROMAN, SHADOW, SLANT, S
    TOP, UPSIDE DOWN
260 FOR I = 1 TO 20: READ F#:F# = F# + ".SET"
270 PRINT : PRINT CHR# (4)"BLORD"F#",R#4000": CALL 768
280 VTAB 23: HTAB (41 - LEN (F#)) / 2: PRINT F#;
290 GOSUB 240: PRINT : NEXT
300 GOTO 230
310 HOME : INVERSE : PRINT " N'OUBLIEZ PAS LA DISQUETTE TOOL.KIT !": NORMAL
    : GOSUB 240
320 POKE 216,0: GOTO 120
    
```

CARAC.VIS

```

0300:20 58 FC 20 E2 F3 A9 03 85 19 A9 00
    85 06 85 08 A9 40 85 07 6A 85 09 A9 28
    85 18 A0 00 A5 08 85 0A A5 09 85 0B B1 0
    6 48 C0 00 F0 10 84 0C A6 0C A5 0B 18 69
    04 CA D0 FA 85 0B A4 0C 68 81 0A C8 C0
    08 90 D9 A5 06 18 69 08 85 06 A5 07 69 0
    0 85 07
0351:E6 08 D0 02 E6 09 C6 18 D0 C0 C6 19
    F0 17 A5 08 18 69 58 85 08 A5 09 69 00
    85 09 A5 19 C9 01 D0 A5 A9 0F D0 A3 60 2
    0 E4 FB 20 1B FD 60
    
```

Tapez exactement ces deux lignes (après un CALL—151 et RETURN).

VÉRIFICATION :

```

T = 0 : FOR I = 768 TO
893 : T = T + PEEK(I) :
NEXT : PRINT T
    
```

Résultat : 12697

BSAVE CARAC.VIS, A\$300, L\$7E

CARACTÈRES PERSONNALISÉS

Ce cadre en est une première application

A = Γ
B = -
C = 7
D = |
E = |
F = L
G = -
H = J

En créant
ces huit
caractères
spéciaux,
tous les
encadrés
deviennent
possibles

On peut certes imprimer des cadres en utilisant d'une part le souligné, et d'autre part le traditionnel point d'exclamation, mais le résultat n'est pas folichon. Par contre, avec quelques caractères personnalisés, créés en cinq minutes (ici, ils vous sont fournis aux lignes 280 et 290), on obtient réellement de bons résultats. Le cadre ci-dessus en est un exemple. Vous obtiendrez le même, sur ImageWriter ou DMP.

Rappelons que l'instruction CHR\$(27) "R00nC" imprime n fois le caractère C, ce qui permet l'économie d'une boucle.

Cette partie du programme installe la police spéciale dans l'imprimante.

```
100 GOSUB 230
110 PRINT CHR$(27)"/"; CHR$(27)"L020"
120 PRINT CHR$(27)"E"
130 PRINT CHR$(27)">"
140 PRINT CHR$(27)"T16"
150 PRINT "A"; CHR$(27)"R050B";"C"
160 FOR I = 1 TO 22: PRINT "D";: PRINT CHR$(27)"R05
0";" ";: PRINT "E": NEXT
170 PRINT "F"; CHR$(27)"R050G";"H"
180 PRINT
190 PRINT CHR$(27)"T22"; CHR$(27)"<"
200 FOR I = 1 TO 8: PRINT CHR$(27)"$"; CHR$(I + 64
)" = "; CHR$(27)"/"; CHR$(I + 64): PRINT : NEXT

210 PRINT CHR$(27)"$": PRINT CHR$(9)"60N": PRINT
CHR$(27)"L010": LIST
220 PRINT CHR$(4)"PR#0": HOME : END
230 FOR I = 768 TO 773: READ C: POKE I,C: NEXT
240 DATA 165,6,32,237,253,96
250 PRINT CHR$(4)"PR#1": PRINT : PRINT CHR$(27)"I
"; CHR$(9)"60N": PRINT CHR$(27)"Z"
260 FOR I = 1 TO 84: READ C: POKE 6,C: CALL 768: NEXT
: PRINT
270 RETURN
280 DATA 27,43,27,73,65,72,255,1,1,1,1,1,1,1,66,72,1,
1,1,1,1,1,1,67,72,1,1,1,1,1,1,1,255,68,72,255,0
,0,0,0,0,0
290 DATA 0,69,72,0,0,0,0,0,0,255,70,72,255,128,128,
128,128,128,128,128,71,72,128,128,128,128,128,128
,128,128,72,72,128,128,128,128,128,128,255,4
```


Votre bibliothèque INFORMATIQUE

par Clément RENARD

● *Guide du PC (Modulo, Belin)*

Je sais : nous sommes ici entre utilisateurs d'Apple, mais qui, parmi nous, n'utilise pas — dans le cadre de son entreprise, notamment —, ou ne sera pas amené à le faire, un IBM PC ? Vous ? alors, passez votre chemin car ce *GUIDE DU PC*, un gros bouquin de Lon Poole (traduction de Michèle Brière et André Cantin), est entièrement dédié à l'ordinateur personnel IBM... que l'auteur vous propose d'apprivoiser. Ah ! comme j'aimerais pouvoir disposer, sur mon 65C02, du Basic PC ! Evidemment, personne ne m'empêche d'acheter un Mac, mais je suis à la fois têtu et inconditionnel de l'Apple... alors,

N'empêche, que j'ai feuilleté le *Guide du PC* avec beaucoup d'intérêt. Je ne suis guère doué pour les langues étrangères, mais j'avoue que la langue parlée chez IBM m'est plutôt familière, sauf que, chez eux, certains mots sont directement compris par la machine, tandis que, chez nous, il faut les lui expliquer en lui fabriquant des routines appropriées.

Mais revenons au livre : super. Bien présenté, bien rédigé, apparemment complet, avec des exemples et encore des exemples. Bravo ! Lon Poole !

Guide du PC



MODULO

Poole

BELIN

Cantin

Brière

● *La télé-informatique personnelle (Hachette informatique)*

Joël de Rosnay a préfacé le livre de Jacques Bessières, lequel travaille chez Apple... d'où ces *Applications aux ordinateurs Apple*. Là, nous sommes bien en terrain connu. Lisez, braves gens, surtout si la télé-informatique personnelle ne vous laisse pas indifférents.

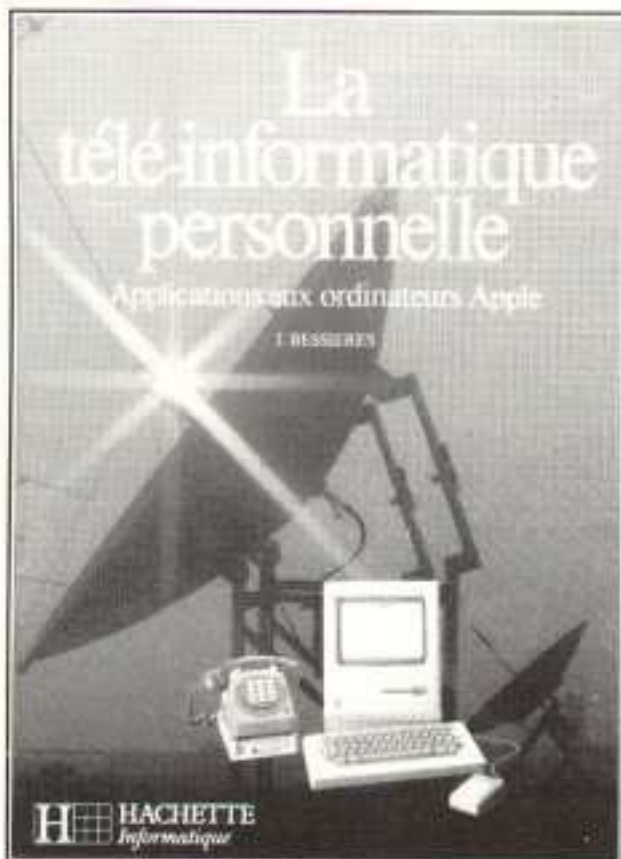
Que faut-il pour communiquer ? Les secrets de la communication ? Transpac... La pratique de quelques serveurs... L'Apple et les connexions... les émulations IBM... le vidéotex...

J'aurai énuméré les titres des chapitres de cet ouvrage quand j'aurai précisé que les deux derniers sont consacrés à l'Apple // micro-serveur et à un guide pratique des communications.

Ainsi, après avoir lu — ou dévoré (il n'est pas indigeste, croyez-moi !) — le texte de Jacques Bessières, non seulement vous saurez comment vous équiper pour communiquer (via Apple), mais vous serez en mesure de créer votre propre micro-serveur.

Une invitation à la communication qui devrait assurément figurer en bonne place sur l'un des nombreux rayons de votre bibliothèque informatique !

Clément RENARD.



I GROSSIR ?

Une simple formalité, avec le programme que vous présente Bernard GOURC.

POUR en comprendre le fonctionnement, commencez par suivre le déroulement de la routine en langage machine (elle n'est pas commentée). Utilisez votre assembleur si possible, mais ne vous découragez surtout pas. Vous pourrez très bien vous contenter du CALL—151 habituel, suivi des valeurs. De toute façon, de \$6106 à \$62F4, il n'y a plus rien à comprendre... puisqu'il s'agit de la fonte de caractères que vous fournit Bernard GOURC... en prime, bien sûr. Résumons-nous :

GROSSIR

se compose donc de :

1. GROS (programme de \$6000 à \$6105, puis fonte)
2. DEMO1, en Basic
3. DEMO2, en Basic également (200 M sur un dessin).

GROS		0	ORG \$6000	601E: E6 FF	29	INC CPTL
		1	CODE EQU \$06	6020: E6 EE	30	INC LIGB
		2	CODE1 EQU \$FC	6022: C6 FB	31	DEC CPTL
		3	CODE2 EQU \$FD	6024: D0 EA	32	BNE AFFICHE
		4	COLONNE EQU \$08	6026: 68	33	RTS
		5	LIGNE EQU \$09		34	*****
		6	CPTL EQU \$FF		35	* DOUBLE LARGEUR *
		7	MEM EQU \$07		36	*****
		8	CPTL EQU \$FB	6027: A9 07	37	DOL LDA \$07
		9	LD EQU \$FE	6029: 85 FF	38	STA CPTL
		10	LIGB EQU \$EE	602B: A5 09	39	DEBUT LDA LIGNE
		11	COLB EQU \$EF	602D: 20 11 F4	40	JSR \$F411
		12	*****	6030: A4 08	41	LDY COLONNE
		13	* AFFICHE UN CARACTERE *	6032: 81 26	42	LDA (\$26),Y
		14	*****	6034: 85 06	43	STA CODE
6000: A9 00	15	AF1 LDA \$00	6036: 20 4B 68	44	JSR DOUBLE	
6002: 85 FF	16	STA CPTL	6039: A4 08	45	LDY COLONNE	
6004: A9 07	17	LDA \$07	603B: A5 FC	46	LDA CODE1	
6006: 85 FB	18	STA CPTL	603D: 91 26	47	STA (\$26),Y	
6008: A5 EE	19	LDA LIGB	603F: C8	48	INY	
600A: 85 09	20	STA LIGNE	6040: A5 FD	49	LDA CODE2	
600C: A5 EF	21	LDA COLB	6042: 91 26	50	STA (\$26),Y	
600E: 85 08	22	STA COLONNE	6044: C6 09	51	DEC LIGNE	
6010: A5 EE	23	AFFICHE LDA LIGB	6046: C6 FF	52	DEC CPTL	
6012: 20 11 F4	24	JSR \$F411	6048: D0 E1	53	BNE DEBUT	
6015: A4 EF	25	LDY COLB	604A: 68	54	RTS	
6017: A6 FF	26	LDX CPTL	604B: A5 06	55	DOUBLE LDA CODE	
6019: 8D 06 61	27	LDA CARACT,X	604D: 29 0F	56	AND \$0F	
601C: 91 26	28	STA (\$26),Y	604F: 85 07	57	STA MEM	

GROS (suite)

```

6851: A9 08      58      LDA  E$00
6853: 85 FB      59      STA  CPTÉ
6855: 85 FC      60      STA  CODE1
6857: 85 FD      61      STA  CODE2
6859: A5 07      62  PREM  LDA  MEM
685B: 4A         63      LSR
685C: 85 07      64      STA  MEM
685E: 20 1F      65      BCS  TOT1
6860: E6 FB      66  A1    INC  CPTÉ
6862: A5 FB      67      LDA  CPTÉ
6864: C9 04      68      CMP  E$04
6866: D0 F1      69      BNE  PREM
6868: A5 06      70      LDA  CODE
686A: 29 78      71      AND  E$78
686C: 0A         72      ASL
686D: 85 07      73      STA  MEM
686F: A9 04      74      LDA  E$04
6871: 85 FB      75      STA  CPTÉ
6873: A5 07      76  DEUX  LDA  MEM
6875: 0A         77      ASL
6876: 85 07      78      STA  MEM
6878: B0 13      79      BCS  TOT2
687A: C6 FB      80  A2    DEC  CPTÉ
687C: D0 F5      81      BNE  DEUX
687E: 60         82      RTS
687F: A6 FB      83  TOT1  LDX  CPTÉ
6881: A5 FC      84      LDA  CODE1
6883: 10         85      CLC
6884: 7D 9B 60   86      ADC  TABLE1,X
6887: 85 FC      87      STA  CODE1
6889: 10         88      CLC
688A: 90 D4      89      BCC  A1
688C: 60         90      RTS
688D: A6 FB      91  TOT2  LDX  CPTÉ
688F: A5 FD      92      LDA  CODE2
6891: 10         93      CLC
6892: 7D 9F 60   94      ADC  TABLE2,X
6895: 85 FD      95      STA  CODE2
6897: 10         96      CLC
6898: 90 E0      97      BCC  A2
689A: 60         98      RTS
689B: 03 0C 30   99  TABLE1  HEX  030C3040
      40
689F: 00 01 06  100 TABLE2  HEX  0001061060
      10 60
      101 *****
      102 * DOUBLE HAUTEUR *
      103 *****
68A4: A9 07      104  DHAUT  LDA  E$07
68A6: 85 FF      105      STA  CPTÉL
68A8: A5 09      106      LDA  LIGNE

```

```

68AA: 30         107      SEC
68AB: E9 06      108      SBC  E$06
68AD: 85 07      109      STA  MEM
68AF: 30         110      SEC
68B0: E9 07      111      SBC  E$07
68B2: 85 FE      112      STA  LD
68B4: A5 07      113  AUTRE  LDA  MEM
68B6: 20 11 F4   114      JSR  $F411
68B9: A4 00      115      LDY  COLONNE
68BB: B1 26      116      LDA  ($26),Y
68BD: 85 06      117      STA  CODE
68BF: A5 FE      118      LDA  LD
68C1: 20 11 F4   119      JSR  $F411
68C4: A4 00      120      LDY  COLONNE
68C6: A5 06      121      LDA  CODE
68C8: 91 26      122      STA  ($26),Y
68CA: E6 FE      123      INC  LD
68CC: A5 FE      124      LDA  LD
68CE: 20 11 F4   125      JSR  $F411
68D1: A4 00      126      LDY  COLONNE
68D3: A5 06      127      LDA  CODE
68D5: 91 26      128      STA  ($26),Y
68D7: E6 07      129      INC  MEM
68D9: E6 FE      130      INC  LD
68DB: C6 FF      131      DEC  CPTÉL
68DD: A5 FF      132      LDA  CPTÉL
68DF: D0 D3      133      BNE  AUTRE
68E1: 60         134      RTS
      135 *****
      136 * DOUBLE HAUTEUR+LARGEUR *
      137 *****
68E2: A5 09      138  DLDH   LDA  LIGNE
68E4: 85 EE      139      STA  LIGB
68E6: A5 00      140      LDA  COLONNE
68E8: 85 EF      141      STA  COLB
68EA: 20 27 60   142      JSR  DOL
68ED: A5 EE      143      LDA  LIGB
68EF: 85 09      144      STA  LIGNE
68F1: A5 EF      145      LDA  COLB
68F3: 85 00      146      STA  COLONNE
68F5: 20 A4 60   147      JSR  DHAUT
68F8: A5 EE      148      LDA  LIGB
68FA: 05 09      149      STA  LIGNE
68FC: E6 EF      150      INC  COLB
68FE: A5 EF      151      LDA  COLB
6100: 85 00      152      STA  COLONNE
6102: 20 A4 60   153      JSR  DHAUT
6105: 60         154      RTS
6106:             155  CARACT

```

ATTENTION ! il y a aussi les caractères, page suivante...

GROS

(suite et fin)

*6106.62F2 POLICE DE CARACTÈRES

6106- 00 00
6108- 00 00 00 00 00 00 00 00
6110- 00 00 00 00 14 14 14 00
6118- 00 00 00 1C 22 02 0C 06
6120- 24 1E 00 3C 0A 1E 28 28
6128- 1E 00 26 16 00 34 32 00
6130- 00 14 14 00 2C 12 2C 00
6138- 00 00 00 00 00 00 00 04
6140- 02 02 02 04 00 00 10 20
6148- 20 20 10 00 00 2A 1C 3E
6150- 1C 2A 00 00 00 00 00 3E
6158- 00 00 00 00 00 00 00 00
6160- 04 00 00 00 00 3E 00 00
6168- 00 00 00 00 00 04 00 00
6170- 20 10 00 04 02 00 1C 22
6178- 32 2A 26 22 1C 00 0C 0A
6180- 00 00 00 00 1C 22 20 10
6188- 0C 02 3E 1C 22 20 18 20
6190- 22 1C 10 18 14 12 3E 10
6198- 10 3E 02 1E 20 20 22 1C
61A0- 30 00 04 1E 22 22 1C 3E
61A8- 22 20 10 00 04 02 1C 22
61B0- 22 1C 22 22 1C 1C 22 22
61B8- 3C 10 00 06 00 00 00 00
61C0- 00 00 00 00 00 00 00 00
61C8- 00 04 10 00 04 02 04 00
61D0- 10 00 00 00 3E 00 3E 00
61D8- 04 00 10 20 10 00 04 1C
61E0- 22 20 10 00 00 00 00 3E

61E8- 22 22 22 3E 00 00 14 22
61F0- 22 3E 22 22 1E 22 22 1E
61F8- 22 22 1E 1C 22 02 02 02
6200- 22 1C 1E 22 22 22 22 22
6208- 1E 3E 02 02 1E 02 02 3E
6210- 3E 02 02 1E 02 02 02 1C
6218- 22 02 02 3A 22 1C 22 22
6220- 22 3E 22 22 22 1C 00 00
6228- 00 00 00 1C 20 20 20 20
6230- 20 22 1C 22 12 0A 06 0A
6238- 12 22 02 02 02 02 02 02
6240- 3E 22 36 2A 2A 22 22 22
6248- 22 22 26 2A 32 22 22 1C
6250- 22 22 22 22 22 1C 1E 22
6258- 22 1E 02 02 02 1C 22 22
6260- 22 2A 12 2C 1E 22 22 1E
6268- 0A 12 22 1C 22 02 1C 20
6270- 22 1C 3E 00 00 00 00 00
6278- 00 22 22 22 22 22 22 1C
6280- 22 22 22 22 22 14 00 22
6288- 22 22 2A 2A 36 22 22 22
6290- 14 00 14 22 22 22 22 14
6298- 00 00 00 00 3E 20 10 00
62A0- 04 02 3E 18 00 36 3F 1F
62A8- 3F 36 18 00 36 21 11 29
62B0- 36 00 04 06 7F 06 04 00
62B8- 00 10 30 7F 30 10 00 00
62C0- 00 00 00 3E 1C 00 00 1C
62C8- 3E 00 00 00 00 7F 7F 7F
62D0- 7F 7F 7F 7F 0F 0F 0F 0F
62D8- 0F 0F 0F 78 78 78 78 78
62E0- 78 78 7F 7F 7F 7F 00 00
62E8- 00 00 00 00 7F 7F 7F 7F
62F0- 00 FF FF

Vous sauvez le tout par BSAVE GROS, A\$6000, L\$2F3

CONTRÔLE : T = 0 : FOR I = 24576 TO 25330 : T = T + PEEK(I) : NEXT I : PRINT T Le résultat doit être 47339

UTILISATION

On peut employer une autre fonte de caractères (provenant de TOOL KIT, par exemple). Il suffit de la charger à partir de l'adresse \$6106. Toutefois, le programme de Bernard GOURC est prévu pour travailler avec des caractères de 7 bits. Or, ceux de TOOL KIT sont sur 8 bits. Il conviendra donc de remplacer la valeur 07 (ligne 17, page 35) par 08. On obtiendra le même résultat par POKE 24581,08.

COMMENT AFFICHER SIMPLEMENT UN CARACTÈRE ?

Placer en \$EF (POKE 239,...) le numéro de la colonne où on désire afficher (de 0 à 39) et en \$EE (POKE 238,...) le numéro de sa ligne de base (de 7 à 19). Ensuite, il faut placer l'adresse du caractère dans la table en \$601A (partie basse) et \$601B (partie haute). Ainsi, pour afficher le premier caractère de la table, on pourra soit (moniteur) écrire : * 601A : 0661, soit (en basic) envoyer les pokes suivants : 24602,06 et 24603,97 (97 = \$61).

AFFICHAGE NORMAL : CALL 24576 — **DOUBLE HAUTEUR :** CALL 24740 — **DOUBLE LARGEUR :** CALL 24615
DOUBLE HAUTEUR ET LARGEUR : CALL 24802

Attention ! après affichage normal, ne choisissez que l'une des 3 options et non les trois successivement !

GROSSIR.DEMO1

```
100 PRINT CHR$(4);"BLOAD GROS"
110 DEF FN H(ADR) = INT (ADR / 256)
120 DEF FN L(ADR) = ((ADR / 256) - FN
    H(ADR)) * 256
130 HGR : HCOLOR= 3: POKE - 16302,0
140 REM *****
150 REM *      ECRIT      *
160 REM *****
170 C = 0:L = 21:A$ = "GOURC BERNARD"
180 FOR F = 1 TO LEN (A$): GOSUB 660
    :C = C + 1: NEXT
190 REM *****
200 REM *ECRIT+DOUBLE HAUTEUR *
210 REM *****
220 C = 0:L = 21
230 FOR F = 1 TO LEN (A$)
240 GOSUB 660: CALL 24740
250 C = C + 1: NEXT F
260 REM *****
270 REM *ECRIT+DOUBLE LARGEUR *
280 REM *****
290 C = 20:L = 45
300 A$ = "PRESENTE"
310 FOR F = 1 TO LEN (A$)
320 GOSUB 660
330 CALL 24615
340 C = C + 2: NEXT F
350 REM *****
360 REM *ECRIT+2*LARGE+2*HAUT *
370 REM *****
380 C = 12:L = 84
390 A$ = "GROSSIT"
400 FOR F = 1 TO LEN (A$)
410 GOSUB 660
420 CALL 24802
430 C = C + 2: NEXT F
440 HPLLOT 0,110 TO 279,110
450 REM *****
460 REM * TOUS LES CARACTERES *
470 REM *****
480 C = 2:L = 171
490 FOR A = 0 TO 70: GOSUB 700
500 C = C + 1: IF C < 38 THEN 520
510 C = 2:L = L + 14
520 NEXT A
530 HPLLOT 0,154 TO 279,154
540 REM *****
550 REM * ANIMATION .... *
560 REM *****
```

```
570 FOR A = 0 TO 70
580 S = PEEK ( - 16336)
590 C = 5:L = 134: GOSUB 700
600 C = 15:L = 134: GOSUB 700: CALL
    24740
610 C = 25:L = 134: GOSUB 700: CALL
    24615
620 C = 35:L = 134: GOSUB 700: CALL
    24802
630 S = PEEK ( - 16336)
640 FOR T = 1 TO 100: NEXT T,A
650 GOTO 570
660 REM *****
670 REM *AFFICHE UN CARACTERE *
680 REM *****
690 B$ = MID$(A$,F,1):A = ASC (B$)
    - 32
700 ADR = 24838 + (A * 7)
710 POKE 24602, FN L(ADR): POKE 2460
    3, FN H(ADR)
720 POKE 239,C: POKE 238,L: CALL 245
    76
730 RETURN
```

GROSSIR.DEMO2

```
10 HOME : HGR
15 PRINT CHR$(4);"BLOAD GROS"
20 PRINT CHR$(4);"BLOAD DESSIN"
25 HCOLOR= 3: POKE - 16302,0
30 REM *****
35 REM *   DOUBLE HAUTEUR   *
40 REM *****
45 L = 182: REM LIGNE BASSE DE LA P
    ARTIE AGRANDIE
50 A = 91: REM NOMBRE DE LIGNES AGR
    ANDIES,SOIT GRANDEUR DU CARACTER
    E
55 POKE 24741,A: POKE 24748,A - 1:
    POKE 24753,A
60 FOR T = 1 TO 2000: NEXT T: REM A
    TTENTE
65 FOR C = 4 TO 25: REM NOMBRE DE C
    OLLONNES TEXT AGRANDIES
70 POKE 8,C: POKE 9,L: CALL 24740:
    REM INITIALISATION + APPEL ROUTI
    NE
75 NEXT C
```


Un excellent traitement de texte :

• LE JONGLEUR DE MOTS



ONNAISSEZ-VOUS *Le Jongleur de Mots* ? Joli, non ? Et poétique avec cela. Ça sent bon la vieille France, celle que l'on retrouve avec tant de plaisir quand on a le privilège de séjourner au Québec.

Car *Le Jongleur de Mots* est canadien (en langue anglaise, son titre — non moins évocateur — est *World Juggler*). J'avoue avoir un faible pour tout ce qui vient de là-bas. Peut-être parce que mon Poitou natal compte bon nombre de cousins des Canadiens et Acadiens. Eux et moi avons les mêmes racines terriennes. Aussi, amie Lectrice, ami Lecteur, ne compte pas sur moi pour "démolir" ce traitement de texte en français (au Québec, on adopte volontiers le tutoiement, et c'est sympa). S'il était mauvais, je n'en parlerais pas, et voilà tout !

Première constatation : *Le Jongleur de Mots* est bavard. Dommage que l'auteur de son mode d'emploi ne soit pas l'un de nos distingués rédacteurs : *Tremplin Micro* déborderait d'explications. Le livret, 136 vraies pages, bien remplies — et rédigées dans un style agréable — explique réellement comment utiliser au mieux le produit. C'est de l'excellente doc, classique dans la présentation, mais bien imprimée.

Les menus se révèlent au moins aussi bavards que le bouquin... dont il est certainement possible de se passer, les commentaires du logiciel se suffisent à eux-mêmes. Que l'on ne se méprenne pas : lorsque j'écris "bavards", ce n'est pas un reproche, au contraire.

Faut-il aborder les questions techniques ? Je suppose que vous n'attendez pas autre chose de moi. Sachez donc que *Le Jongleur de Mots* est utilisable sous ProDOS (version 1.1.1, c'est-à-dire la bonne... celle qui est compatible avec les nouveaux lecteurs Apple). Il est écrit en assembleur, d'où une rapidité qui n'a rien à envier aux autres produits de cette catégorie (en mode "insertion" notamment). Au moment où je rédige ces quelques lignes, mon Apple IIe (avec son 65C02) vaut une bonne machine à écrire, et je retrouve bien, sur les 80 colonnes de mon écran, tous les caractères rentrés au clavier (je vous signale au passage que je ne tape qu'avec deux ou trois doigts, mais avec une rapidité satisfaisante...).

On se promène dans le texte — pour insérer ou effacer — par le truchement des flèches et je vous assure que la souris ne me manque absolument pas. Pour ne rien vous cacher, si j'en suis partisan dans des utilitaires de type "Mouse Desk" (fameux !) ou "Version Calc", j'apprécie ici son absence.

Un détail : pas de programme de coupure de mots dans *Le Jongleur de Mots* (cette commodité est l'apanage de Gribouille). En fait, aucun mot n'est coupé : vous frappez votre texte au kilomètre et le logiciel se charge de faire passer le curseur à la ligne suivante, sans transformer votre correspondant en cher "con citoyen".

Un autre détail : les accents circonflexes sont sur les lettres. Comme je vous le dis. On est Canadien ou on ne l'est pas. Quand on l'est, on le prouve en traitant sa langue natale... mieux que les Cousins de France. Vous en déduirez naturellement que l'écran est graphique.

(suite page 40).



LE JONGLEUR DE MOTS (suite)

Il est bon de noter que tous les déplacements du curseur sont prévus, mais je vous fais grâce des vingt possibilités que vous offrent les flèches, ainsi que les diverses combinaisons avec Pomme Ouverte. Ce serait inutilement fastidieux.

Les touches d'édition sont plus nombreuses encore et permettent de réaliser des tas de choses : changer, copier, déplacer, effacer, insérer, trouver, etc. Au premier abord, un traitement de texte paraît toujours difficile à assimiler, mais on a tôt fait, quand on l'utilise régulièrement, d'en mémoriser les principaux codes. Il reste que les futurs "jongleurs de mots" applaudiront certainement l'initiative de l'éditeur : la fourniture, avec les disquettes et le mode d'emploi, d'une petite bande adhésive pour identifier certaines des touches de l'Apple.

Et l'impression ? Alors là, sachez qu'avec une bonne imprimante (et notamment avec Image-Writer), la touche ESCAPE et une quarantaine de touches de contrôle, on peut réaliser la plupart des présentations et impressions : alinéa, cen-

trage, interlignage, justification, longueur de page et de texte, marge, soulignement, gras, indigage supérieur ou inférieur, etc. C'est réellement très complet.

Une précision : tous les codes sont répertoriés sur une "carte de références"... que j'aurais personnellement souhaitée en bristol léger plutôt qu'en papier (de bonne qualité).

Peut-on utiliser des fichiers de données, des variables, PFS ou Quick File pour les lettres types, des fichiers de texte Dos 3.3 ou Pascal ?... la réponse à ces questions est oui.

Ultime constatation : la longueur maximum d'un document ne pourra jamais dépasser 785 lignes (en 80 colonnes), avec 128 K de mémoire... et à condition de ne pas abuser des variables, bien entendu.

Il me reste à émettre un vœu : que de nombreux Lecteurs (je n'oublie pas les Lectrices) de *Tremplin Micro* demandent une démonstration du *Jongleur de Mots* à leur magasin "micro" préféré...

Guy-HACHETTE.

Editions TURGEON, 89 avenue de Wagram, Paris — Tél. 16 (1) 43.80.86.69.

TRUC DOS 3.3

Ce court programme de démonstration vous montrera, mieux que de longs discours, comment vous passer, dans un programme, des instructions **INVERSE** et **NORMAL**, remplacées par **CTRL-I** et **CTRL-N**. Cela fonctionne aussi en mode direct. On en sort par **RESET** ou **CALL—1438** (pseudo — **RESET**). On retrouve la fonction par **Call 768**.

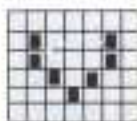
```
10 TEXT : NORMAL : PRINT CHR$(12); CHR$(21); HOME : GOSUB 130
20 INPUT "COMBIEN DE MOTS ? ";N: PRINT : IF N < 1 THEN 80
30 FOR I = 1 TO N: PRINT "MOT "I" -> "; INPUT " ";M$: PRINT : PRINT "
  "M$"; PRINT : NEXT
40 REM UN CTRL-I PRECEDE LES MOTS A METTRE EN INVERSE, ET UN CTRL-N RAMENE
  AU MODE NORMAL
50 REM LES VALEURS NORMALES DES ADRESSES 54 ET 55 SONT 189 ET 158 SOIT $9E
  BD, ADRESSE DE LA ROUTINE DE SORTIE DE CARACTERES ($36-37)
60 REM CTRL-RESET RESTAURE LES PARAMETRES D’AFFICHAGE
70 CALL - 198: POKE - 16368,0: WAIT - 16384,128: POKE - 16368,0
80 HOME : LIST 10,60
90 UTAB 23: PRINT "MENU DE DISQUETTE FINI DE JOUER! "; GET R$: POKE - 1
  6368,0
100 UTAB 22: PRINT : IF R$ = "M" THEN PRINT D$"RUN MENU"
110 HOME : END
130 FOR I = 768 TO 800: READ R: POKE I,R: NEXT : CALL 768: RETURN
140 DATA 169,11,133,54,169,3,133,55,76,234,3,201,137,208,5,169,63,133,50,96
  ,201,142,208,5,169,255,133,50,96,32,240,253,96
```

Inverse-normal

VISU-PUCE

par Maurice CHAVELLI

Long, mais très intéressant !



OICI un programme qui devrait faire plaisir au débutant. Que celui-ci ne s'effraie pas devant sa longueur. Il pourra l'utiliser sans nécessairement savoir comment il fonctionne ; c'est d'ailleurs ce qu'il fait avec le BASIC ! J'y ai toutefois mis suffisamment de commentaires pour que les «courageux» puissent s'y retrouver. Quant aux habitués de l'assembleur, ils doivent posséder un débogueur comme Bug Byter, bien plus performant... mais pas toujours plus pratique.

Visu-Puce, comme son nom l'indique, permet de voir ce qui se passe dans votre Apple lorsque «tourne» un programme, avec la possibilité de le suivre pas à pas, en affichant l'état des registres et de 4 mémoires (définissables selon les besoins, en binaire, hexadécimal, décimal, ASCII). Il permet aussi de modifier ces valeurs à tout moment. La pile est également affichée et peut être facilement éditée.

INSTALLATION

Le programme BAS.VISU permet d'installer VISU-PUCE à l'adresse \$9000 (36864) et les deux écrans MASQUE et HELP dans la carte langage en MEV2 (\$D000-\$D3F7 et \$D400-\$D7F7). Le premier écran est indispensable et ne doit pas être modifié ! Le second est facultatif, puisqu'il sert uniquement d'aide-mémoire (il est appelé par «?» et peut être supprimé à condition de ne plus utiliser la commande ?). D'autre part, l'Ampersand est vectorisé, ce qui veut dire que vous pouvez appeler *Visu-Puce* à partir du Basic, directement en tapant &, suivi bien sûr de l'indispensable RETURN. Du moniteur, par contre, il vous faudra taper 9000G et RETURN.

CREATION DES ECRANS

Commencez donc par créer les deux écrans à l'aide de FAIRE.ECRAN, paru dans le numéro 1 de *Tremplin Micro* ou de tout autre programme de création de masque. N'oubliez pas de les sauver sur disquette ! Entrez ensuite le gros morceau VISU.OBJ0 et le programme en Basic VISU.CH. Un RUN VISU.CH lancera la routine.

DEMARRAGE

Vous avez maintenant MASQUE sur votre écran et un «!» en haut à gauche (qui a été accompagné d'un bip bien connu). Si ce n'est pas le cas, c'est que vous avez commis une erreur de frappe quelque part... alors GOTO au début !

Si tout a fonctionné normalement, bravo ! Le plus dur est passé. Tapez maintenant un «?» (pas les guillemets : juste le point d'interrogation !) suivi de RETURN (je ne le dirai plus). Vous devez retrouver avec un plaisir bien compréhensible votre écran HELP. Tapez une touche pour revenir. Vous savez maintenant comment combler un trou de mémoire à tout moment.

PREMIER ESSAI

Entrez «A=56» (toujours sans les guillemets : je ne le dirai plus non plus); avez-vous vu changer la valeur de A ? Essayez avec une autre valeur (la valeur étant en HEXA, il faut vous limiter à deux chiffres). Si tout va bien, essayez aussi avec X,Y et P, par exemple «P=7».

Le bas de l'écran est réservé à l'affichage de 4 mémoires dont vous pouvez déterminer les adresses à tout moment.

Essayez «M1=9000», essayez de même M2, M3 et M4 (ici aussi la valeur est en HEXA et doit donc être limitée à 4 chiffres).

Il y a quelques limitations en page 0 : ça ne marchera pas pour les adresses de \$18 à \$1F, car elles sont utilisées par le programme. Par contre, elles sont sauvegardées puis restaurées et un programme qui les utiliserait ne serait donc pas perturbé. Certaines adresses sont modifiées par *Visu-Puce* grâce aux routines du moniteur qu'il utilise.

Important : évitez la zone des entrées/sorties qui démarre en \$C000 et qui contient des commutateurs logiciels. De toute façon, si vous vous retrouvez coincé, un RESET vous ramènera en principe au BASIC.

Tapez maintenant «S», vous voyez un curseur clignotant au sommet de la pile. Tapez la touche flèche à gauche et tout est remonté d'un cran : vous avez dépilé une valeur, celle-ci est maintenant

perdue. Pour empiler une valeur, il suffit de l'entrer au clavier : essayez «3A» (pour une fois sans RETURN). Habituez-vous un moment à ces opérations (dont vous sortirez par ESC).

SUITE

Vous voici prêt pour les choses sérieuses. Tapez «E6FB», vous avez la première instruction du programme qui démarre en \$E6FB désassemblée sur la deuxième ligne de l'écran. Vous pouvez alors taper :

1. Flèche à droite pour sauter à l'instruction suivante sans exécuter la première.
2. La barre d'ESPACE pour exécuter l'instruction et voir son action sur les registres. Il vous faudra taper à nouveau la barre d'ESPACE pour afficher l'instruction suivante. Cette procédure en deux temps permet un meilleur «suivi» du programme.
3. RETURN pour exécuter le programme sans interruption ; vous pouvez l'arrêter à tout moment par la barre d'ESPACE.

Entraînez-vous à ces différentes manipulations ; en \$E6FB se trouve juste une routine basic sans risque, et relativement longue.

Vous risquez d'avoir quelques surprises dans les routines du moniteur gérant les entrées/sorties car *Visu-Puce* les utilise aussi !

Essayez par exemple l'adresse \$FDED qui contient la routine COUT de sortie des caractères. Vous pouvez aussi essayer avec *Visu-Puce* qui démarre en \$9000. Sortez par RESET si ça coince, puis tapez «&» pour revenir.

Vous reviendrez à tout moment au mode commande (bip et point d'exclamation) en tapant la touche ESC. De là, vous accéderez au moniteur en tapant «Q», ce qui peut être nécessaire pour changer la valeur d'une adresse par exemple, ou déplacer une zone mémoire par la routine MOVE, ou même utiliser le mini-assembleur du BASIC INTEGER.

Pour retrouver le programme où vous l'avez laissé (si vous n'avez pas touché à PCL et PCH), il suffit de taper 9000G pour récupérer *Visu-Puce* et, ensuite, simplement RETURN derrière le point d'exclamation.

Après ce tour d'horizon des commandes de *Visu-Puce*, il ne me reste plus qu'à vous souhaiter une fructueuse exploration en profondeur de votre Apple et une meilleure compréhension des programmes en assembleur qui paraissent dans *Tremplin Micro*.

Maurice Chavelli.

MASQUE

Réalisez cet écran avec **Faire-Ecran** (*Tremplin Micro n°1*), puis sauvez-le sur disquette sous le nom de **MASQUE**.

Il va de la 4^e à la 24^e ligne

PILE	BINAIRE	HEX	DEC	CAR
->	A	\$		
	X	\$		
	Y	\$		
	P	\$		
	N U X B D I Z C			
	\$	\$		
	\$	\$		
	\$	\$		
	\$	\$		

VISU-PUCE

Version 1.0 • Maurice Chavelli pour Tremplin Micro • (Octobre 85)

SI VOUS NE
POSSÉDEZ PAS
D'ASSEMBLEUR,
CETTE
COLONNE
EST LA SEULE
À RECOPIER.

9000 : BA
9001 : B6 49
9003 : A9 A1
9005 : B5 33

9007 : A9 D0
9009 : 8D C7 92
900C : A9 D3

900E : 8D CF 92
9011 : 20 BF 92
9014 : 20 9D 91

9017 : A9 03
9019 : 8D 83 93
901C : 85 23
901E : 20 58 FC

0				
1				
2				
3				
4				
5				
6				
7				
8	WNCBTM	EQU	\$23	: Bas de la fenêtre
9	CH	EQU	\$24	: Positions horizontales...
10	CV	EQU	\$25	: ...et verticales du curseur
11	LENGTH	EQU	\$2F	: Longueur de l'instruction désassemblée
12	ADD	EQU	\$18	: Pointeur pour M1, M2, M3 et M4
13	PROMPT	EQU	\$33	: Visu-Puce "!" au lieu de "+"
14	PCL	EQU	\$3A	: Sauvegarde du...
15	PCH	EQU	\$3B	: ...compteur ordinal
16	A2L	EQU	\$3E	: Mémoire de travail...
17	A2H	EQU	\$3F	: ...pour GETNUM
18	ACC	EQU	\$45	
19	XREG	EQU	\$46	: Sauvegarde...
20	YREG	EQU	\$47	: ...des ...
21	STATUS	EQU	\$48	: ...registres
22	SPNT	EQU	\$49	
23	IN	EQU	\$200	: Buffer d'entrée
24	KBD	EQU	\$C000	: Reçoit le caractère du clavier
25	STROBE	EQU	\$C010	: Remet à 0 le bit 7 de KBD pour nouvelle touche
26	LINPRT	EQU	\$ED24	: Affiche la valeur déc. de A,X
27	INSTDSP	EQU	\$F8D0	: Désassemble l'instruction pointée par PCL,PCH
28	PRNTAX	EQU	\$F941	: Affiche le contenu de A,X en 4 chiffres HEXA
29	PRBLNK	EQU	\$F948	: 3 espaces
30	PCADJ2	EQU	\$F954	: Mise à jour PCH et PCL...
31	PCADJ3	EQU	\$F956	: ...pour instruction suivante
32	UP	EQU	\$FC1A	: Monte le curseur d'une ligne
33	VTAB	EQU	\$FC22	: Met le curseur sur la ligne A
34	HOME	EQU	\$FC58	: Nettoie l'écran
35	LF	EQU	\$FC66	: Descend le curseur d'une ligne
36	RDKEY	EQU	\$FD0C	: Attend une touche avec curseur clignotant
37	GETLN	EQU	\$FD6A	: Attend une ligne de commande, fin par Return
38	PRBYTE	EQU	\$FDDA	: Affiche le contenu de A en 2 chiffres HEXA
39	PRHEX	EQU	\$FDE3	: Affiche le poids faible de A en 1 chiffre HEXA
40	COUT	EQU	\$FDED	: Affiche le car. contenu dans A
41	BELL	EQU	\$FF3A	: Emet un charmant BIP !
42	RESTORE	EQU	\$FF3F	: Récupère A,X,Y et P
43	SAVE	EQU	\$FF4A	: Sauvegarde de A,X,Y,P et S
44	MON	EQU	\$FF65	: Entrée dans le Moniteur
45	GETNUM	EQU	\$FFA7	: Récupère un nombre HEXA & le met dans A2L,A2H
46		ORG	\$9000	: Adresse de début de programme
47		TSX		: On sauve le pointeur...
48		STX	SPNT	: ...de pile
49		LDA	£"!"	: On change...
50		STA	PROMPT	: ...le PROMPT

RÉCUPÈRE LE MASQUE

54	MASQUE	LDA	£SD0	
55		STA	CGA+1	
56		LDA	£SD3	
57		STA	CGB+1	
58		JSR	CHARGE	
59		JSR	AFFICH	

INITIALISATION

63	INIT	LDA	£\$3	
64		STA	MODE	: Si MODE = 0 : Pas à Pas, MODE = 1 : Passe sans exéc
65		STA	WNCBTM	: Réduit la fenêtre aux 3 premières lignes
66		JSR	HOME	

9021	:	20	3A	FF	67	JSR	BELL		
9024	:	20	6A	FD	68	JSR	GETLN		
9027	:	E0	00		69	CPX	£\$00	:	Si RETURN direct...
9029	:	F0	38		70	BEQ	INIT2	:	...reprise au point d'arrêt
902B	:	A2	00		71	LDX	£\$00		
902D	:	BD	00	02	72	LDA	IN,X	:	Premier caractère dans A
9030	:	C9	D1		73	CMP	£"Q"	:	Si ce n'est pas "Q"...
9032	:	D0	0A		74	BNE	INIT1	:	...on saute la sortie vers Moniteur
9034	:	A9	18		75	LDA	£\$18	:	On remet une...
9036	:	85	23		76	STA	WNDBTM	:	...fenêtre normale
9038	:	20	58	FC	77	JSR	HOME		
903B	:	4C	65	FF	78	JMP	MON	:	On passe la main au moniteur

TEST DU PREMIER CARACTÈRE

903E	:	C9	8F		82	INIT1	CMP	£"?"	
9040	:	F0	50		83		BEQ	HELP	
9042	:	C9	D3		84		CMP	£"S"	
9044	:	F0	64		85		BEQ	INITS	
9046	:	C9	C1		86		CMP	£"A"	
9048	:	F0	1C		87		BEQ	INITA	
904A	:	C9	D8		88		CMP	£"X"	
904C	:	F0	23		89		BEQ	INITX	
904E	:	C9	D9		90		CMP	£"Y"	
9050	:	F0	2A		91		BEQ	INITY	
9052	:	C9	D0		92		CMP	£"P"	
9054	:	F0	31		93		BEQ	INITP	
9056	:	C9	CD		94		CMP	£"M"	
9058	:	F0	53		95		BEQ	INITM	

96 On récupère l'adresse entrée au clavier...
97 ...pour la mettre en PCL,PCH

905A	:	A0	00		98		LDY	£\$0	
905C	:	20	E4	92	99		JSR	BUFF1	
905F	:	85	3A		100		STA	PCL	
9061	:	86	3B		101		STX	PCH	
9063	:	4C	FA	90	102	INIT2	JMP	DEBUT	

MISE À JOUR DES REGISTRES PAR ENTRÉE AU CLAVIER

9066	:	20	E2	92	106	INITA	JSR	BUFF	
9069	:	85	45		107		STA	ACC	
906B	:	20	9D	91	108		JSR	AFFICH	
906E	:	4C	17	90	109		JMP	INIT	
9071	:	20	E2	92	110	INITX	JSR	BUFF	
9074	:	85	46		111		STA	XREG	
9076	:	20	9D	91	112		JSR	AFFICH	
9079	:	4C	17	90	113		JMP	INIT	
907C	:	20	E2	92	114	INITY	JSR	BUFF	
907F	:	85	47		115		STA	YREG	
9081	:	20	9D	91	116		JSR	AFFICH	
9084	:	4C	17	90	117		JMP	INIT	
9087	:	20	E2	92	118	INITP	JSR	BUFF	
908A	:	85	48		119		STA	STATUS	
908C	:	20	9D	91	120		JSR	AFFICH	
908F	:	4C	17	90	121		JMP	INIT	

TRAITEMENT DU "?" → ÉCRAN DE RAPPEL DES COMMANDES

9092	:	A9	D4		125	HELP	LDA	£\$D4	:	Ajuste les adresses....
9094	:	8D	C7	92	126		STA	CGA+1		
9097	:	A9	D7		127		LDA	£\$D7		
9099	:	8D	CF	92	128		STA	CGB+1		
909C	:	20	BF	92	129		JSR	CHARGE	:	...et charge l'écran.
909F	:	AD	00	C0	130	HELP1	LDA	KBD	:	Attend une...
90A2	:	10	FB		131		BPL	HELP1	:	...touche
90A4	:	8D	10	C0	132		STA	STROBE		
90A7	:	4C	07	90	133		JMP	MASQUE	:	On repart au début pour récupérer le masque
90AA	:	4C	28	93	134	INITS	JMP	PILE	:	Edition de la pile

135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202

MISE À JOUR DES MÉMOIRES M1 M2 M3 et M4

90AD : AD 02 04
90B0 : A0 03
90B2 : C9 B1
90B4 : F0 17
90B6 : C9 B2
90B8 : F0 22
90BA : C9 B3
90BC : F0 2D
90BE : 20 E4 92
90C1 : 8D 92 93
90C4 : 8E 93 93
90C7 : 20 9D 91
90CA : 4C 17 90
90CD : 20 E4 92
90D0 : 8D 8C 93
90D3 : 8E 8D 93
90D6 : 20 9D 91
90D9 : 4C 17 90
90DC : 20 E4 92
90DF : 8D 8E 93
90E2 : 8E 8F 93
90E5 : 20 9D 91
90E8 : 4C 17 90
90EB : 20 E4 92
90EE : 8D 90 93
90F1 : 8E 91 93
90F4 : 20 9D 91
90F7 : 4C 17 90

INITM LDA \$402 ; On récupère le 2^e caractère
LDY ££3
CMP £"1"
BEQ INITM1
CMP £"2"
BEQ INITM2
CMP £"3"
BEQ INITM3
JSR BUFF1 ; On récupère la valeur entrée au clavier
STA SV2+6
STX SV2+7
JSR AFFICH
JMP INIT
INITM1 JSR BUFF1
STA SV2
STX SV2+1
JSR AFFICH
JMP INIT
INITM2 JSR BUFF1
STA SV2+2
STX SV2+3
JSR AFFICH
JMP INIT
INITM3 JSR BUFF1
STA SV2+4
STX SV2+5
JSR AFFICH
JMP INIT

CETTE PARTIE REPREND PRATIQUEMENT

LA ROUTINE STEP DES PREMIERS APPLE

90FA : 20 58 FC
90FD : 20 D0 F8
9100 : 20 EC 92
9103 : C9 01
9105 : F0 61
9107 : A9 EA
9109 : 8D 41 91
910C : 8D 42 91
910F : 8D 43 91
9112 : A2 00
9114 : A1 3A
9116 : F0 43
9118 : A4 2F
911A : C9 20
911C : F0 54
911E : C9 60
9120 : F0 40
9122 : C9 4C
9124 : F0 57
9126 : C9 6C
9128 : F0 54
912A : C9 40
912C : F0 30
912E : 29 1F
9130 : 49 14
9132 : C9 04
9134 : F0 02
9136 : B1 3A
9138 : 99 41 91
913B : 88
913C : 10 F8
913E : 20 3F FF

DEBUT JSR HOME
JSR INSTDSP ; Désassemble l'instruction
JSR KEY ; Attend une touche : ESCAPE, RETURN ou —>
CMP £\$1 ; Si A = 1 on passe sans exécuter
BEQ PCINC3
LDA £\$EA ; On met des NOP
STA INSTR
STA INSTR+1
STA INSTR+2
LDX £00
LDA (PCL,X) ; On met l'instruction dans A
BEQ BREAK ; Si 00 —> BRK
LDY LENGTH ; Longueur de l'instruction dans Y
Série de tests pour l'instruction
CMP £\$20
BEQ JSP ; Si JMP
CMP £\$60
BEQ RETOUR ; Si RTS
CMP £\$4C
BEQ JUMP ; Si JMP
CMP £\$6C
BEQ INJUMP ; Si JMP indirect
CMP £\$40
BEQ XRT1 ; Si RTI
AND £\$1F ; Test de branchement relatif
EOR £\$14 ; 10 30 50 70 90 B0 D0 F0
CMP £\$04
BEQ RELAT
ECIN LDA (PCL),Y ; On place l'instr. dans INSTR
RELAT STA INSTR,Y
DEY
BPL ECIN
JSR RESTORE ; On récupère les registres...

9141	: EA		203	INSTR	NOP	:	...on exécute l'instruction...
9142	: EA		204		NOP	:	...que l'on a déposée ici
9143	: EA		205		NOP		
9144	: 4C 55 91		206		JMP NBREL	:	Si pas branchement relatif
9147	: 18		207		CLC	:	Si branchement relatif...
9148	: A0 01		208		LDY £\$01	:	...on calcule...
914A	: B1 3A		209		LDA (PCL),Y	:	...la nouvelle...
914C	: 20 56 F9		210		JSR PCADJ3	:	...adresse
914F	: 85 3A		211		STA PCL		
9151	: 98		212		TYA		
9152	: 38		213		SEC		
9153	: B0 11		214		BCS PCINC2	:	toujours !
9155	: 20 4A FF		215	NBREL	JSR SAVE	:	On sauvegarde les registres pour la prochaine
9158	: 38		216		SEC		
9159	: B0 0D		217		BCS PCINC3	:	toujours !
915B	: 4C 63 90		218	BREAK	JMP INIT2	:	En cas de BRK, retour à INIT
915E	: 18		219	XRTI	CLC	:	En cas de RTI...
915F	: 68		220		PLA	:	...on récupère l'adresse...
9160	: 85 48		221		STA STATUS	:	...de retour
9162	: 68		222	RETOUR	PLA	:	Idem pour RTS
9163	: 85 3A		223		STA PCL		
9165	: 68		224		PLA		
9166	: 85 3B		225	PCINC2	STA PCH	:	On stocke la partie haute
9168	: A5 2F		226	PCINC3	LDA LENGTH	:	Longueur instruction dans A
916A	: 20 56 F9		227		JSR PCADJ3	:	On calcule la nouvelle adresse
916D	: 84 3B		228		STY PCH		
916F	: 18		229		CLC		
9170	: 90 14		230		BCC NEWPCL	:	toujours !
9172	: 18		231	JSP	CLC	:	Cas du JSR
9173	: 20 54 F9		232		JSR PCADJ2		
9176	: AA		233		TAX	:	On empile...
9177	: 98		234		TYA	:	...l'adresse...
9178	: 48		235		PHA	:	...du retour
9179	: 8A		236		TXA		
917A	: 48		237		PHA		
917B	: A0 02		238		LDY £\$02		
917D	: 18		239	JUMP	CLC		
917E	: B1 3A		240	INJUMP	LDA (PCL),Y	:	Cas du JMP indirect
9180	: AA		241		TAX		
9181	: 88		242		DEY		
9182	: B1 3A		243		LDA (PCL),Y		
9184	: 86 3B		244		STX PCH		
9186	: 85 3A		245	NEWPCL	STA PCL		
9188	: B0 F3		246		BCS JUMP	:	Un 2 ^e tour pour l'indirect !
918A	: BA		247		TSX	:	On ajuste le pointeur...
918B	: 86 49		248		STX SPNT	:	...de pile
918D	: AD 83 93		249		LDA MODE	:	Est-on en mode passage...
9190	: C9 01		250		CMP £\$1	:	...sans exécution ?
9192	: F0 06		251		BEQ NEW1		
9194	: 20 9D 91		252		JSR AFFICH		
9197	: 20 EC 92		253		JSR KEY		
919A	: 4C FA 90		254	NEW1	JMP DEBUT		

ROUTINES D'AFFICHAGE

919D	: A9 18		258	AFFICH	LDA £\$18	:	On rétablit une...
919F	: 85 23		259		STA WNDBTM	:	...fenêtre normale
91A1	: A9 07		260		LDA £\$7	:	On va à la ligne 6 (ça commence à zéro !)
91A3	: 85 25		261		STA CV		
91A5	: 20 22 FC		262		JSR VTAB		
91A8	: A5 45		263		LDA ACC	:	On récupère la valeur de A...
91AA	: 20 48 92		264		JSR BHD1	:	...et on l'affiche en BIN, HEXA, DECI, ASCI
91AD	: A5 46		265		LDA XREG	:	Idem pour X,Y,P
91AF	: 20 48 92		266		JSR BHD1		
91B2	: A5 47		267		LDA YREG		
91B4	: 20 48 92		268		JSR BHD1		
91B7	: A5 48		269		LDA STATUS		
91B9	: 20 48 92		270		JSR BHD1		

91BC	:	20	66	FC	271		JSR	LF		; On descend de 2 lignes
91BF	:	20	66	FC	272		JSR	LF		
					273					On sauvegarde les adresses \$18 à \$1F...
					274					...qui vont être écrasées
91C2	:	A2	07		275		LDX	£\$07		
91C4	:	B5	18		276	BCL1	LDA	ADD,X		
91C6	:	9D	84	93	277		STA	SV1,X		
91C9	:	CA			278		DEX			
91CA	:	10	F8		279		BPL	BCL1		
91CC	:	A2	07		280		LDX	£\$07		
					281					On récupère M1,M2,M3,M4 pour...
					282					...les mettre de \$18 à \$1F
91CE	:	BD	8C	93	283	BCL2	LDA	SV2,X		
91D1	:	95	18		284		STA	ADD,X		
91D3	:	CA			285		DEX			
91D4	:	10	F8		286		BPL	BCL2		
					287					On affiche ces mémoires et leurs valeurs
91D6	:	A0	00		288		LDY	£\$0		
91D8	:	8C	82	93	289		STY	IND		
91DB	:	20	36	92	290	AFF1	JSR	BHD		
91DE	:	EE	82	93	291		INC	IND		
91E1	:	EE	82	93	292		INC	IND		
91E4	:	AC	82	93	293		LDY	IND		
91E7	:	C0	08		294		CPY	£\$8		
91E9	:	F0	0A		295		BEQ	AFF2		; Fin de l'affichage des mémoires
91EB	:	C0	04		296		CPY	£\$4		
91ED	:	F0	EC		297		BEQ	AFF1		
91EF	:	20	1A	FC	298		JSR	UP		
91F2	:	4C	DB	91	299		JMP	AFF1		
					300					On restaure les adresses \$18 à \$1F
91F5	:	A2	07		301	AFF2	LDX	£\$07		
91F7	:	BD	84	93	302	BCL3	LDA	SV1,X		
91FA	:	95	18		303		STA	ADD,X		
91FC	:	CA			304		DEX			
91FD	:	10	F8		305		BPL	BCL3		
					306					Affichage de la pile
91FF	:	EA	EA		307		NOP	NOP		; Ligne "bidon"
9201	:	A9	08		308		LDA	£\$08		
9203	:	85	25		309		STA	CV		
9205	:	20	22	FC	310		JSR	VTAB		
9208	:	A6	49		311		LDX	SPNT		
920A	:	E8			312		INX			
920B	:	A0	0E		313		LDY	£\$E		; Seulement 14 valeurs
920D	:	E0	00		314	AFF3	CPX	£\$00		; Test de débordement
920F	:	F0	16		315		BEQ	AFF5		
9211	:	A9	03		316		LDA	£\$3		
9213	:	85	24		317		STA	CH		
9215	:	BD	00	01	318		LDA	\$100,X		
9218	:	20	DA	FD	319		JSR	PRBYTE		
921B	:	E8			320		INX			
921C	:	20	66	FC	321		JSR	LF		
921F	:	88			322		DEY			
9220	:	D0	EB		323		BNE	AFF3		
					324					Réajustement fenêtre et retour
9222	:	A9	03		325	AFF4	LDA	£\$3		
9224	:	85	23		326		STA	WNOBTM		
9226	:	60			327		RTS			
					328					Met des "blancs" pour finir la pile
9227	:	A9	03		329	AFF5	LDA	£\$3		
9229	:	85	24		330		STA	CH		
922B	:	20	48	F9	331		JSR	PRBLNK		
922E	:	20	66	FC	332		JSR	LF		
9231	:	88			333		DEY			
9232	:	D0	F3		334		BNE	AFF5		
9234	:	F0	EC		335		BEQ	AFF4		
					336					
					337					
					338					

ROUTINES DE CONVERSION BIN, HEXA, DECI, ASCII

9236	:	A9	09	339	BHD	LDA	£\$9	:	Pour M1,M2,M3,M4...
9238	:	85	24	340		STA	CH	:	...affiche l'adresse
923A	:	B9	18 00	341		LDA	ADD,Y		
923D	:	AA		342		TAX			
923E	:	B9	19 00	343		LDA	ADD+1,Y		
9241	:	20	41 F9	344		JSR	PRNTAX		
9244	:	98		345		TYA			
9245	:	AA		346		TAX			
9246	:	A1	18	347		LDA	(ADD,X)		
9248	:	8D	81 93	348	BHD1	STA	REG	:	REG sert de mémoire de travail
924B	:	48		349		PHA		:	On sauvegarde A
				350		Affichage en binaire			
924C	:	A9	0E	351		LDA	£\$E	:	On se positionne colonne 15
924E	:	85	24	352		STA	CH		
9250	:	A2	08	353		LDX	£\$8	:	C'est parti pour 8 tours !
9252	:	A9	00	354	BHD2	LDA	£\$0		
9254	:	0E	81 93	355		ASL	REG	:	On décale un coup à gauche
9257	:	2A		356		ROL		:	On récupère dans A...
9258	:	20	E3 FD	357		JSR	PRHEX	:	...et on affiche
925B	:	CA		358		DEX			
925C	:	D0	F4	359		BNE	BHD2		
				360		Affichage en HEXA			
925E	:	A9	1A	361		LDA	£\$1A		
9260	:	85	24	362		STA	CH		
9262	:	68		363		PLA		:	On récupère la valeur...
9263	:	8D	81 93	364		STA	REG	:	...pour la mettre en A...
9266	:	20	DA FD	365		JSR	PRBYTE	:	...et l'afficher !
				366		Affichage ASCII			
9269	:	A2	25	367		LDX	£\$25		
926B	:	86	24	368		STX	CH		
926D	:	AD	81 93	369		LDA	REG		
9270	:	C9	8A	370		CMP	£\$8A		
9272	:	F0	21	371		BEQ	BHD4		
9274	:	C9	8D	372		CMP	£\$8D		
9276	:	F0	1D	373		BEQ	BHD4		
9278	:	20	ED FD	374	BHD3	JSR	COUT		
				375		Affichage en décimal			
927B	:	A9	1F	376		LDA	£\$1F		
927D	:	85	24	377		STA	CH		
927F	:	20	48 F9	378		JSR	PRBLNK	:	On commence par effacer ce qu'il y avait...
9282	:	A9	1F	379		LDA	£\$1F		
9284	:	85	24	380		STA	CH		
9286	:	A9	00	381		LDA	£\$0		
9288	:	AE	81 93	382		LDX	REG		
928B	:	20	24 ED	383		JSR	LINPRT	:	...et on affiche !
928E	:	20	66 FC	384		JSR	LF		
9291	:	20	66 FC	385		JSR	LF		
9294	:	60		386		RTS			
9295	:	A9	A0	387	BHD4	LDA	£\$A0		
9297	:	D0	DF	388		BNE	BHD3		
				389					
				390		ROUTINE DE SAUVEGARDE & RESTITUTION (fiche TREMLIN 3)			
				391					
9299	:	AD	81 C0	392	SAUVE	LDA	\$C081		
929C	:	AD	81 C0	393		LDA	\$C081		
929F	:	A9	00	394		LDA	£\$0		
92A1	:	85	3C	395		STA	\$3C		
92A3	:	A9	40	396		LDA	£\$40		
92A5	:	85	3D	397		STA	\$3D		
92A7	:	A9	F7	398		LDA	£\$F7	:	Ici profitez-en pour corriger la fiche
92A9	:	85	3E	399		STA	\$3E		
92AB	:	A9	43	400		LDA	£\$43		
92AD	:	85	3F	401		STA	\$3F		
92AF	:	A9	00	402		LDA	£\$0		
92B1	:	85	42	403		STA	\$42		
92B3	:	A9	00	404	SV	LDA	£\$00	:	Mémoire ajustée suivant masque
92B5	:	85	43	405		STA	\$43		
92B7	:	38		406		SEC			


```

92B8 : 20 11 C3 407 JSR $C311
92BB : AD 82 C0 408 LDA $C082
92BE : 60 409 RTS
92BF : AD 80 C0 410 CHARGE LDA $C080
92C2 : A9 00 411 LDA £$0
92C4 : 85 3C 412 STA $3C
92C6 : A9 00 413 CGA LDA £$00
92C8 : 85 3D 414 STA $3D
92CA : A9 F7 415 LDA £$F7 ; Idem correction
92CC : 85 3E 416 STA $3E
92CE : A9 00 417 CGB LDA £$00
92D0 : 85 3F 418 STA $3F
92D2 : A9 00 419 LDA £$0
92D4 : 85 42 420 STA $42
92D6 : A9 04 421 LDA £$4
92D8 : 85 43 422 STA $43
92DA : 18 423 CLC
92DB : 20 11 C3 424 JSR $C311
92DE : AD 82 C0 425 LDA $C082
92E1 : 60 426 RTS
427

```

ROUTINE DE SAISIE DE VALEUR HEXA

```

428
429
92E2 : A0 02 430 BUFF LDY £$2
92E4 : 20 A7 FF 431 BUFF1 JSR GETNUM
92E7 : A5 3E 432 LDA A2L
92E9 : A6 3F 433 LDX A2H
92EB : 60 434 RTS
435

```

ROUTINE D'ATTENTE DE CARACTÈRE ET D'AJUSTEMENT DU MODE

```

436
437
438
92EC : AD 83 93 439 KEY LDA MODE
92EF : F0 27 440 BEQ KEY6
92F1 : AD 00 C0 441 KEY1 LDA KBD
92F4 : 10 FB 442 BPL KEY1
92F6 : 8D 10 C0 443 KEY2 STA STROBE
92F9 : C9 9B 444 CMP £$9B ; Si ESC on sort
92FB : F0 10 445 BEQ KEY3
92FD : C9 8D 446 CMP £$8D ; Si RETURN —» MODE = 0, trace sans arrêt
92FF : F0 11 447 BEQ KEY4
9301 : C9 95 448 CMP £$95 ; Si FLÈCHE à DROITE —» MODE = 1, passe sans exéc.
9303 : F0 1D 449 BEQ KEY7
9305 : C9 A0 450 CMP £$A0 ; Si ESCAPE —» MODE quelconque —» Pas à pas
9307 : D0 E8 451 BNE KEY1
9309 : 8D 83 93 452 STA MODE
930C : 60 453 RTS
930D : 68 454 KEY3 PLA
930E : 68 455 PLA
930F : 4C 17 90 456 JMP INIT
9312 : A9 00 457 KEY4 LDA £$00
9314 : 8D 83 93 458 STA MODE
9317 : 60 459 KEY5 RTS
9318 : AD 00 C0 460 KEY6 LDA KBD
931B : 10 FA 461 BPL KEY5
931D : 8D 83 93 462 STA MODE
9320 : 30 D4 463 BMI KEY2
9322 : A9 01 464 KEY7 LDA £$1
9324 : 8D 83 93 465 STA MODE
9327 : 60 466 RTS
467

```

ROUTINE D'ÉDITION DE LA PILE

```

468
469
9328 : 20 9D 91 470 PILE JSR AFFICH
932B : A9 18 471 LDA £$18 ; On rétablit la fenêtre
932D : 85 23 472 STA WNDBTM
932F : A9 07 473 LDA £$7 ; On se place au sommet
9331 : 85 25 474 STA CV

```


9333	:	20	22	FC	475	JSR	VTAB			
9336	:	A9	03		476	LDA	£\$3			
9338	:	85	24		477	STA	CH			
933A	:	20	48	F9	478	JSR	PRBLNK			
933D	:	A9	03		479	LDA	£\$3			
933F	:	85	24		480	STA	CH			
9341	:	A2	00		481	LDX	£\$0			
9343	:	20	0C	FD	482	JSR	RDKEY	:	On attend 1 caractère	
9346	:	9D	00	02	483	STA	IN,X	:	On le stocke dans le buffer	
9349	:	20	ED	FD	484	JSR	COUT	:	On l'affiche	
934C	:	C9	9B		485	CMP	£\$9B	:	Est-ce ESC ?	
934E	:	F0	1C		486	BEQ	PILE2	:	si oui on sort	
9350	:	C9	88		487	CMP	£\$88	:	Est-ce flèche à gauche ?	
9352	:	F0	22		488	BEQ	PILE3	:	si oui on dépile	
9354	:	E8			489	INX				
9355	:	E0	02		490	CPX	£\$2	:	2 caractères ?	
9357	:	D0	EA		491	BNE	PILE1	:	non —> on va chercher le 2°	
9359	:	A9	8D		492	LDA	£\$8D	:	oui —> RETURN pour "marquer" la fin	
935B	:	9D	00	02	493	STA	IN,X			
935E	:	A0	00		494	LDY	£\$0			
9360	:	20	A7	FF	495	JSR	GETNUM	:	On récupère la valeur...	
9363	:	A5	3E		496	LDA	A2L	:	...dans A...	
9365	:	48			497	PHA		:	...et on l'empile	
9366	:	BA			498	TSX				
9367	:	86	49		499	STX	SPNT			
9369	:	4C	28	93	500	JMP	PILE			
936C	:	A9	03		501	LDA	£\$3	:	Sortie de l'édition	
936E	:	85	24		502	STA	CH			
9370	:	20	48	F9	503	JSR	PRBLNK			
9373	:	4C	17	90	504	JMP	INIT			
9376	:	BA			505	TSX		:	Ici on dépile	
9377	:	E0	FF		506	CPX	£\$FF			
9379	:	F0	AD		507	BEQ	PILE			
937B	:	E8			508	INX				
937C	:	9A			509	TXS				
937D	:	86	49		510	STX	SPNT			
937F	:	D0	A7		511	BNE	PILE			
9381	:	00			512	REG	DFB	00	:	Mémoires de travail
9382	:	00			513	IND	DFB	00		
9383	:	00			514	MODE	DFB	00		
9384	:	00	00	00						
	:	00	00	00						
	:	00	00		515	SV1	DFB	00, 00, 00, 00, 00, 00, 00, 00	:	Sauvegarde de \$18 à \$1F
938C	:	00	00	00						
	:	00	00	00						
	:	00	00		516	SV2	DFB	00, 00, 00, 00, 00, 00, 00, 00	:	Sauvegarde M1, M2, M3, M4
	:				517					

TABLE DES SYMBOLES ORDRE ALPHABÉTIQUE

A2H	\$003F	PCADJ2	..	\$F954	INIT	\$9017	ACC	\$0045	PCH	\$003B	INIT2	...	\$9063
AFF1	\$91DB	PCINC3	..	\$9168	INITM	\$90AD	AFF3	\$920D	PILE	\$9328	INITM2	...	\$90DC
AFF5	\$9227	PILE2	\$936C	INITP	\$90B7	BCL1	\$91C4	PRBLNK	..	\$F948	INITX	\$9071
BCL3	\$91F7	PRHEX	\$FDE3	INJUMP	\$917E	BHD	\$9236	PROMPT	\$0033	INSTR	\$9141
BHD2	\$9252	REG	\$9381	JUMP	\$917D	BHD4	\$9295	RESTORE	..	\$FF3F	KEY	\$92EC
BUFF	\$92E2	SAUVE	\$9299	KEY2	\$92F6	CGA	\$92C6	SPNT	\$0049	KEY4	\$9312
CH	\$0024	STROBE	\$C010	KEY6	\$9318	COUT	\$FDED	SV1	\$9384	LENGTH	..	\$002F
DEBUT	\$90FA	UP	\$FC1A	LINPRT	\$ED24	GETLN	\$FD6A	WNBDM	\$0023	MODE	\$9383
HELP	\$9092	XRT1	\$915E	NBREL	\$9155	HOME	\$FC58	ADD	\$0018	NEWPCL	\$9186
IND	\$9382	A2L	\$003E	PCADJ3	\$F956	INIT1	\$903E	AFF4	\$9222	PCINC2	\$9166
INITA	\$9066	AFF2	\$91F5	PCL	\$003A	INITM1	\$90CD	BCL2	\$91CE	PILE1	\$9343
INITM3	\$90EB	AFFICH	\$919D	PILE3	\$9376	INITS	\$90AA	BHD1	\$9248	PRBYTE	\$FD0A
INITY	\$907C	BELL	\$FF3A	PRNTAX	\$F941	INSTDSP	\$F8D0	BREAK	\$915B	RDKEY	\$FD0C
JSP	\$9172	BHD3	\$9278	RELAT	\$9138	KBD	\$C000	CGB	\$92CE	RETOUR	\$9162
KEY1	\$92F1	BUFF1	\$92E4	SAVE	\$FF4A	KEY3	\$930D	CV	\$0025	STATUS	\$0048
KEY5	\$9317	CHARGE	\$928F	SV	\$9283	KEY7	\$9322	GETNUM	\$FFA7	SV2	\$938C
LF	\$FC66	ECIN	\$9136	VTAB	\$FC22	MASQUE	\$9007	IN	\$0200	XREG	\$0046
MON	\$FF65	HELP1	\$909F	YREG	\$0047	NEW1	\$919A						

VISU.CH

Ce court programme en Basic vous permettra de charger **VISU.PUCE** dans de bonnes conditions. Rappelons que ce programme ne tourne que sous DOS 3.3.

```

100 HIMEM: 36863: HOME
110 REM -----
120 REM Chargement de VISU-PUCE
130 REM -----
140 PRINT CHR$(4)"BLOADVISU.OBJ0"
150 PRINT CHR$(4)"BLOADMASQUE,A$4000"
160 POKE 36864 + 690,208: CALL 36864 + 663
170 PRINT CHR$(4)"BLOADHELP,A$4000"
180 POKE 36864 + 690,212: CALL 36864 + 663
190 POKE 1013,76: POKE 1014,0: POKE 1015,144
200 CALL 36864

```

MODE COMMANDE

```

RETURN : Reprise au point d'arrêt
A=45 : Place la valeur 45 dans A
X=78 : Place la valeur 78 dans X
Y=1A : Place la valeur 1A dans Y
P=FE : Place la valeur FE dans P
M1=1222 : M1 donne le contenu de la
memoire $1222, idem pour
M2,M3, et M4...
Q : Retour au moniteur, le
retour a VISU-PUCE se fait
par *90000 «RETURN»
S : Edition de la pile:
«- : depile une valeur
34: Empile 34 - ESC: Mode commande

```

MODE EXECUTION

```

ESPACE : Exécute instr. et avance
RETURN : Exécute instr. sans arrêt
-> passe sans execution - ESC:retour

```

Pour terminer (également en utilisant le **FAIRE.ECRAN** de **TREPLIN MICRO** n°1), préparez-vous un écran ressemblant à celui-ci.

Vous le mémoriserez sous le nom de **HELP**, prévu dans la routine en Basic de Maurice Chavelli (voir ci-dessus).

SAISIE DE MOTS

INITIATION

Si vous n'êtes plus néophyte dans l'art de la programmation en Basic, passez votre chemin, amie Lectrice, ami Lecteur, et abandonnez cette courte routine aux débutants.

NESTOR

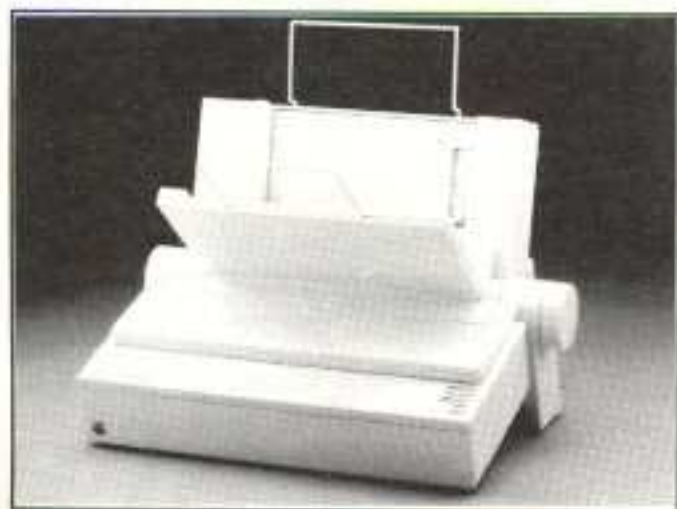
```

100 TEXT : PRINT CHR$(21): HOME : DIM M$(50)
110 INVERSE : PRINT " SAISIE EN BASIC " : NORMAL : PRINT TAB(
30)"INITIATION": POKE 34,3
120 VTAB 22: PRINT "* POUR CORRECTION - RETURN POUR FINIR"
130 N = N + 1
140 V = N + 3: IF V > 20 THEN POKE 35,21: CALL - 912: POKE 35,
23:V = 20
150 V$ = " ": IF N > 9 THEN V$ = ""
160 IF M$ = "*" AND V = 20 THEN VTAB 4: FOR I = N - 16 TO N: P
RINT I" " TAB(5)M$(I): CALL - 868:V = 20: NEXT
170 VTAB V: PRINT V$:N": "M$(N): HTAB 5: INPUT "":M$
180 IF M$ = "*" AND N > 1 THEN 240
190 IF M$ = "*" THEN 140
200 IF M$ < > "*" THEN 230
210 IF M$(N) = "" THEN VTAB V: CALL - 868:N = N - 1: GOTO 260
220 GOTO 240
230 M$(N) = M$
240 VTAB V: HTAB 5: CALL - 868: PRINT M$(N): IF M$ = "*" THEN
N = N - 1: GOTO 140
250 GOTO 130
260 VTAB 22: HTAB 21: CALL - 868: INVERSE : PRINT "CTRL-C POUR
FIN " : NORMAL : CALL - 198: GET M$: PRINT ""
270 IF M$ = "*" THEN N = N + 1: GOTO 240
280 IF M$ < > CHR$(3) THEN 120
290 TEXT : HOME

```


Faites-vous plaisir :

offrez-vous les nouveaux produits !



■ IMAGEWRITER II

Je l'avais vue au Sicob. Elle m'avait tapé dans l'œil. Je l'ai connectée sur mon Apple IIe (avec 65C02) ; elle m'a complètement séduit.

D'abord, elle est réellement rapide. Ce n'est pas une vue de l'esprit : 250 cps pour la qualité dite "brouillon" (acceptable pour tout ce qui est listage), 180 cps pour la qualité standard (bonne) et 45 cps pour le traitement du courrier (lent... mais proche de la marguerite).

Introduction du papier, installation du ruban (on peut faire de la couleur), tout est agréable sur cette *IMAGEWRITER II*, entièrement compatible (au stade où en sont mes essais) avec l'ImageWriter normale et la DMP.

Vous ai-je parlé du bruit ? non. Alors, sachez qu'il n'est pas gênant ; l'insonorisation me paraît excellente.

Autre point fort : la documentation est claire, précise, parlante. On sait comment effectuer chacun des réglages, à condition de prendre le temps de tout lire, bien sûr. Un modèle du genre. Toutefois, je déplore que l'on y escamote le problème des caractères personnalisés. Pourquoi ne pas joindre le second tome du *Manuel de l'utilisateur...* au premier ?

Apple paraît ignorer que des milliers de gens ne se contentent pas des logiciels du commerce et désirent programmer eux-mêmes certaines applications...

■ UNIDISK 3.5

Avec 800 Ko sur une seule disquette, nous sommes enfin en présence d'un vrai lecteur permettant réellement de travailler avec un Apple... et sans faire obligatoirement appel à un disque dur.

Attention ! si vous achetez un *UNIDISK 3.5*, adressez-vous à un revendeur qualifié car :

- le kit d'extension Apple IIc est indispensable pour faire fonctionner le lecteur *Unidisk 3.5*. Toutefois, ce kit apporte divers avantages : mini-assembleur, routines assembleurs de debugging (STEP et TRACE), un "Protocol Converter" pour les entrées-sorties de périphériques sous ProDOS et PASCAL 1.3... ainsi que des routines de pilotage du réseau AppleTalk.

Comment savoir si cette ROM existe ? En tapant "PR&7" sous Basic. Si votre Apple IIc répond "AppleTalk Offline", il est fin prêt pour recevoir un lecteur *Unidisk 3.5*.

Dans le cas contraire, faites-le opérer chez votre concessionnaire.

Sur un Apple IIe avec 65C02, rien à faire, sinon acheter une carte contrôleur spéciale (675 FH.T).

A noter que l'*Unidisk* exige 64 Ko de mémoire pour fonctionner. Sous Pascal, il faut disposer de la version 1.3.

Tenez ! sur une seule disquette, j'ai rentré, dans les dossiers TM1, TM2, etc. tous les programmes parus dans les cinq premiers numéros de *Tremplin Micro...* et ce n'est pas terminé... bien que je me sois aussi offert le luxe d'y placer *ProDOS* et *Basic-System* ! Un détail : Avec un *Unidisk* et *Mouse Desk*, on se régale ! **NESTOR.**



CONVERSION

DÉCIMAL, HEXADÉCIMAL, BINAIRE

C E programme en BASIC fait néanmoins appel à une routine en langage machine. Il vous permettra de convertir des valeurs dans les modes de numération les plus utilisés : DÉCIMAL, HEXADÉCIMAL et BINAIRE.

PRINT CHR\$(21) vous fait passer en mode 40 colonnes, si l'Apple était en 80 colonnes.

CLEAR (ligne 110) annule toutes les variables du programme. Notez qu'un **RUN** a le même effet.

Les **POKES** de la ligne 180 écrivent directement sur l'écran les caractères en mode FLASH, puis NORMAL. Le premier caractère de la ligne est à l'adresse 1024 et il est donc facile de connaître les adresses du **D**, du **H**, et du **B** (que l'on a mises en DATA, à la ligne 760).

Essayez de comprendre vous-même le pourquoi des diverses interdictions, au cours de la saisie des valeurs (290 à 360) ! C'est fort intéressant...

```

100 TEXT : NORMAL : PRINT CHR$ (21): GOSUB 720
110 CLEAR : HOME
120 :
130 REM *** AFFICHAGE DU TITRE ***
140 :
150 INVERSE : PRINT "CONVERSION: DECIMAL HEXADECIMAL BI
NAIRE": NORMAL
160 FOR I = 1 TO 10:T$ = T$ + "____": NEXT
170 POKE - 16368,0: READ P,V: IF P = 0 THEN RESTORE :
GOTO 170
180 POKE P,V + 64
190 FOR I = 1 TO 15:B = PEEK (49200)
200 S = PEEK ( - 16384): IF S = V + 64 THEN 260
210 NEXT
220 POKE P,V: GOTO 170
230 :
240 REM *** SAISIE DU NOMBRE ***
250 :
260 PRINT : PRINT : CALL - 868
270 VTAB 5: PRINT "NOMBRE ";; INVERSE : PRINT CHR$ (S):
: NORMAL : INPUT " ";N$
280 L = LEN (N$)
290 IF (S = 66 AND (L < 4 OR L > 16)) OR (S = 66 AND ( I
NT (L / 4) < > L / 4)) THEN 270
300 IF S = 66 THEN 320
310 IF L > 4 AND S = 72 OR L > 5 AND S < > 72 THEN 270
320 FOR I = 1 TO L:M = ASC ( MID$ (N$,I,1))
330 IF S = 66 AND (M < > 48 AND M < > 49) THEN 260
340 IF M > 47 AND M < 58 THEN 370
350 IF M > 64 AND M < 71 AND S = 72 THEN 370
360 GOTO 260
370 NEXT
380 :
390 REM *** BINAIRE EN DECIMAL ***
400 :
410 IF S = 66 THEN P = 0: FOR I = L TO 1 STEP - 1:V =
VAL ( MID$ (N$,I,1)):N = N + (V * 2 ^ P):P = P + 1: N

```



```

EXT :N$ = STR$(N): GOTO 440
420 IF S = 72 THEN GOSUB 540: GOTO 440
430 N = VAL(N$)
440 B = INT(N / 256):A = N - B * 256: POKE 6,B: POKE 7,
  A
450 VTAB 15: PRINT T$T$: VTAB 7: PRINT T$T$: INVERSE : F
  OR I = 8 TO 16: VTAB I: PRINT " *": HTAB 40: PRINT "
  ": NEXT : NORMAL : POKE 32,3: POKE 33,36
460 :
470 REM *** DECIMAL EN HEXADECIMAL ***
480 :
490 VTAB 12: PRINT "HEXADECIMAL: $": IF S = 72 THEN PR
  INT N$: GOTO 570
500 CALL 768: GOTO 570
510 :
520 REM *** HEXADECIMAL EN DECIMAL ***
530 :
540 N$ = RIGHT$("000" + N$,4):P = 0
550 FOR I = 4 TO 1 STEP -1:V = ASC(MID$(N$,I,1)) -
  48: IF V > 9 THEN V = V - 7
560 N = N + (V * 16 ^ P):P = P + 1: NEXT : RETURN
570 PRINT : VTAB 10: PRINT "  DECIMAL: ";
580 CALL 775
590 :
600 REM *** HEXADECIMAL EN BINAIRE ***
610 :
620 PRINT : VTAB 14: PRINT "  BINAIRE: " : CALL 782
630 :
640 REM *** ON CONTINUE OU PAS ? ***
650 :
660 POKE 32,0: POKE 33,40: VTAB 21: PRINT
670 PRINT : VTAB 22: PRINT "<1> ENCORE <2> MENU DISQUETT
  E <3> FIN " : POKE 1745,49: POKE 1756,50: POKE 1775,5
  1: GET R$: PRINT
680 ON VAL(R$) GOTO 110,690,700: GOTO 670
690 PRINT CHR$(4)"RUN MENU"
700 HOME : END
710 :
720 POKE 768,165: POKE 769,6: POKE 770,166: POKE 771,7:
  POKE 772,76: POKE 773,65: POKE 774,249: POKE 775,165:
  POKE 776,6: POKE 777,166: POKE 778,7: POKE 779,76: P
  OKE 780,36: POKE 781,237: POKE 782,165: POKE 783,6: P
  OKE 784,240
730 POKE 785,3: POKE 786,32: POKE 787,23: POKE 788,3: PO
  KE 789,165: POKE 790,7: POKE 791,160: POKE 792,8: POK
  E 793,10: POKE 794,72: POKE 795,169: POKE 796,176: PO
  KE 797,144: POKE 798,2: POKE 799,169
740 POKE 800,177: POKE 801,32: POKE 802,237: POKE 803,25
  3: POKE 804,104: POKE 805,136: POKE 806,208: POKE 807
  ,241: POKE 808,169: POKE 809,160: POKE 810,32: POKE 8
  11,237: POKE 812,253: POKE 813,96
750 RETURN
760 DATA 1037,4,1045,8,1057,2,0,0

```

Notez (ligne 440) comment installer une valeur décimale sur 2 octets... (ces valeurs sont reprises par la routine en langage machine).

POKE 32,3 : POKE 33,36 (ligne 450) protège le cadre de présentation. Quand vous utilisez ces pokes, il ne faut jamais que leur total dépasse 40 (en 40 colonnes) ou 80 (en 80 colonnes).

Les **POKES** de la ligne 670 font la même chose que ceux de la ligne 180 : ils affichent les chiffres 1, 2 et 3 en mode **INVERSE**. C'est plus court que d'indiquer toutes les tabulations et changements de mode.

Les lignes 720 à 740 installent une courte routine en langage machine. Pour la lister, après un premier **RUN**, tapez **CALL -151 RETURN**, puis **300L...** et vous obtiendrez les 20 premières lignes, désassemblées. Un autre **L**, suivi de **RETURN**, vous permettra de lire les suivantes. Si vous avez d'abord tapé **PRE1**, vous les imprimerez au fur et à mesure.

LA CONVERSION ?

c'est dans la poche !

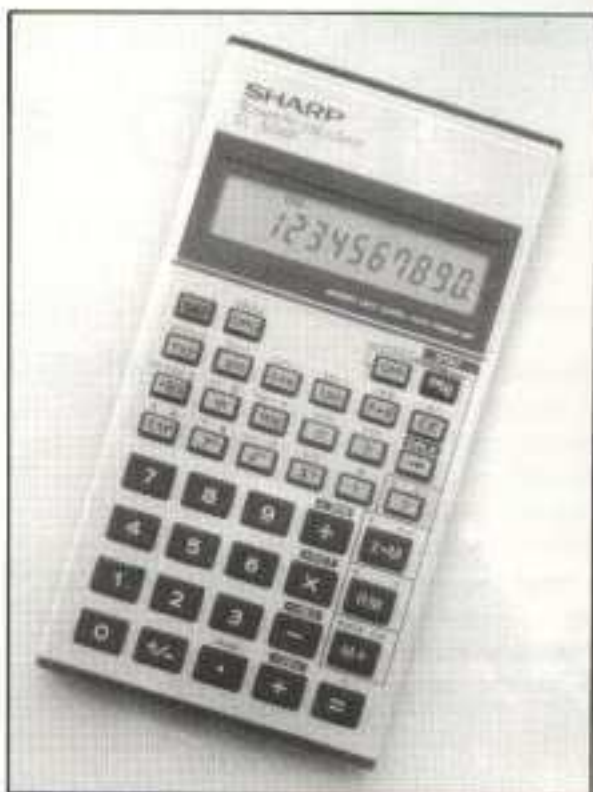
CRÉER des routines personnelles de conversion, soit en Basic, soit en langage machine (ou en associant les deux, comme nous venons de le faire dans "HDEC.BAS"), paraît évidemment constituer, pour des mordus de la programmation, la démarche idéale, mais l'utilisation de telles routines pose parfois des problèmes. En fait, il arrive toujours un moment où il est impossible de les installer à l'adresse prévue... à cause du programme en cours de réalisation, par exemple. Est-on coincé pour autant ?

Ne serait-il pas plus simple d'utiliser une calculatrice ? Ou quelque PC mis au rancart ? Qui ne possède pas, dans un placard, avec une pile en train de rendre l'âme, un PC 1211 (Sharp) ou une FX-702P (Casio) ? Vrai, vous êtes dans ce cas-là ? alors, vous n'avez jamais connu la fébrilité des débutants que nous fûmes, anxieux d'obtenir, de ces merveilleuses petites machines, des programmes évidemment géniaux ! Dommage... pour vous.

Les fabricants n'insistent pas assez, dans les annonces publicitaires concernant leurs calculatrices scientifiques, sur le fait, pourtant très important (du moins pour les utilisateurs d'ordinateurs personnels), que certaines d'entre elles autorisent, et d'une manière on ne peut plus simple, toutes les conversions : binaire, octale, décimale et hexadécimale.

Chez Sharp, c'est le cas de la EL-525, une machine solaire semi-portative très performante, de la EL-545 (toujours avec le soleil, mais en format de poche) et, enfin, de la EL-506P, que j'ai essayée... et adoptée. On peut se la mettre dans la poche sans problème. Vous vous l'offrirez pour une petite paire de billets de 100 F, mais sa pile ne sera pas inusable.

Comment ça marche ? Ne me dites pas que vous ne lirez pas la documentation que Sharp, prévoyant, vous présente dans cinq langues (468



pages... dont 1/5° d'utiles... quel gaspillage !). Sachez pourtant qu'elle répondra clairement à toutes vos questions (parce que, entre nous, la Sharp EL-506P, elle est capable de calculs autrement plus compliqués que vos banales conversions). Mais encore ? Je vous livre un secret. Soit \$6152 à convertir en valeur décimale, puis en binaire ; j'enfonce la touche 2ndF (c'est la seule qui soit orange), puis HEX... mais zyeutez plutôt :
2ndF HEX 6152 2ndF DEC 24914 2ndF BIN 0
...et un E, à gauche, qui m'indique — et il a bien raison — que ma merveilleuse machine ne peut afficher que... 10 chiffres. Mais entre nous, que voulez-vous faire de seize 1 ou 0 ?

Un nouvel essai, mais cette fois avec une valeur raisonnable, prouve que, effectivement, ça marche :

2ndF HEX F7 2ndF DEC 247 2ndF BIN 11110111
...c'est bel et bien gagné ! Me voilà équipé pour poquer à loisir, et sans jamais me tromper dans mes conversions !

Ceci étant, cet outil génère aussi des nombres aléatoires, possède les fonctions trigonométriques, trigonométriques inverses, logarithmiques, exponentielles, permet les calculs de pourcentages, les conversions d'angle/temps, des calculs statistiques, etc. Bref, c'est tout beau !

NESTOR

CONVERSION

mais en langage machine

Si vous ne possédez pas d'assembleur, tapez seulement ces 59 octets, après un CALL—151.

```
0300: 20 30 03
0303: 20 42 FC
0306: 20 30 03
0309: A9 A4
030B: 20 ED FD
030E: A2 00
0310: 20 75 FD
0313: 20 C7 FF
0316: 20 A7 FF
0319: A5 3E
031B: 18
031C: 65 3F
031E: F0 0D
0320: A6 3E
0322: A5 3F
0324: 20 24 ED
0327: 20 42 FC
032A: 4C 06 03
032D: 4C 62 FA
0330: A9 13
0332: 85 25
0334: 20 C1 FB
0337: 20 62 FC
033A: 60
```

```
0 *-----*
1 * CONVERSION D'UN NOMBRE HEXADÉCIMAL EN NOMBRE DECIMAL *
2 *-----*
3 *
4
5 ORG $300
6 LINPTR EQU $ED24 ; Affiche X et A en hexadécimal
7 BASCALC EQU $FBC1 ; Calcule l'adresse de base
8 CLREOP EQU $FC42 ; Effacement fin de la page
9 CR EQU $FC62 ; Envoie un retour chariot
10 NXTCHAR EQU $FD75 ; Lire le prochain caractère
11 COUT EQU $FDED ; Sortie de caractère
12 GETNUM EQU $FFA7 ; Récupère un nombre hexa
13 ZMODE EQU $FFC7 ; Met à Zéro-Mode ($31)
14 RESET EQU $FA62 ; Retour Applesoft avec Bip
15 *-----*
16
17 JSR Position ; Première utilisation
18 INIT JSR Clreop
19 LDA £A4 ; $A4 = dollar
20 JSR Cout ; Affichage du dollar
21 LDX £00 ; Registre X mis à zéro
22 JSR Nxtchar ; Saisie du nombre hexa
23 JSR Zmode ; Mise à zéro du Mode
24 JSR Getnum ; et récupération du nbre hexa
25 LDA $3E ; contenu dans $3E et
26 CLC ; (attention, en additionne)
27 ADC $3F ; $3F va grossir (pu pas) A
28 BEQ Sortie ; (si 3E+3F <> 0 on continue)
29 LDX $3E ; X est chargé avec $3E
30 LDA $3F ; et A avec $3F pour aller à
31 JSR Linptr ; qui affiche le nombre hexa
32 JSR Clreop ; Effacement (Clreol convenait)
33 JMP INIT ; Et ça recommence
34 SORTIE JMP Reset
35 POSITION LDA £13 ; $13 = ligne 19
36 STA $25
37 JSR Bascalc ; Calcul indispensable
38 JSR Cr ; Pour aller ligne suivante
39 RTS ; Retour à l'envoyeur
```

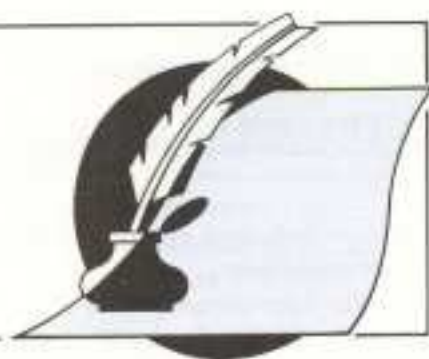
SAUVEGARDE :

BSAVE HDEC,
A\$300,L\$3B

MODE D'EMPLOI :

Pour une utilisation avec & (Ampersand) taper POKE 1013,76, POKE 1014,0 et POKE 1015,3. Ensuite, avant chaque utilisation, il suffira de taper "&"
Un résultat identique sera obtenu avec un simple CALL 768

Vous avez écrit à TREMPLIN MICRO



* Je vous signale un "bug" dans le programme **CALENDRIER.IW** de T.M. n°5 : la ligne de tirets s'imprime sur les noms abrégés des jours de la semaine. Il faut, aux lignes 315 et 320, supprimer les "IF... etc." Il y a aussi un autre "bug" qui, si l'on demande plusieurs éditions du calendrier (4 mois par ligne), indique un mauvais saut de page à l'imprimante. Là, je n'ai pas trouvé le remède.

Claude R. (13009 MARSEILLE)

TM Le programme fonctionne normalement, mais avec une certaine configuration de l'imprimante. Dans un cas comme celui-ci, il est plus facile de modifier le programme que de reconfigurer l'imprimante. Ce que vous avez fait. Par contre, le saut de page est effectivement mauvais. Remplacez le CHR\$(12) de la ligne 255 par le nombre de PRINT correspondant à votre format de papier.

* Bravo pour votre revue qui veut se mettre à la portée de tous, débutants compris. Mais dites, pour vous, un débutant, c'est quoi ?

CALL du droit au menton, PEEK du gauche au foie, série de POKES à la face et ce "chien" d'Apple qui m'insulte, ma femme qui me bat (ou le contraire), Nestor qui rit en disant fastoche...

S'il vous plaît, une petite page pour débutant... débutant, et ainsi éclairer notre machine de son langage...

Jean-Pierre C. (78250 MÉZY)

TM Vous nous dites aussi posséder TOUS les numéros de **TREMPLIN MICRO**. Alors, de grâce, comptez les pages réservées aux débutants... et dites-nous quelle autre revue (consacrée à Apple) en a autant fait... en si peu de numéros.

Sans ses CALLS, PEEKS et POKES, le Basic Applesoft perd de son intérêt... et n'y gagne certes pas en rapidité ! Il faut absolument les connaître et s'en dresser une petite liste (ils ne sont pas nombreux d'ailleurs). Notez que nous rappelons souvent leur fonction.

Ne tuez pas le chien pour autant. S'il aboie, cela prouve qu'il manque encore d'affection !

* Impossible d'imprimer avec ma SEIKOSHA-GP 100A et un Apple IIe, mais uniquement quand je suis en 80 colonnes (je possède la carte 80 colonnes normale et la carte Chat Mauve). Avec un PRE1 et un programme en 40 colonnes tout va bien. Par contre, en 80 colonnes et même en prenant la précaution de taper un CHR\$(21), ça se plante !

Précision : mon interface est une APPLE.

Jean S. (04860 PIERREVERT)

TM Si l'un de nos Lecteurs, équipé du même matériel, connaît une solution, qu'il se mette en rapport avec **TREMPLIN MICRO**.

* Je possède la version ProDOS B.5, mais je n'ai pas la doc. Or, sur cette disquette, il y a un éditeur-assembleur (EDASM) plus ou moins auto-documenté. Malheureusement, je ne sais pas m'en servir et, quand je tape vos programmes, je préfère pourtant taper les commandes... plutôt qu'une suite inintéressante de codes, après un CALL—151. J'espère que vous ne me répondrez pas que je suis un méchant pirate, car j'ai acheté cette disquette, en même temps que mon APPLE, à un particulier qui n'avait pas la doc.

Est-il possible, pour votre programme **MOUSE-TIMER**, de substituer le joystick à la souris ?

Eric V. (75014 PARIS)

TM La version B5 (DOS 3.4) n'est pas compatible avec EDASM 1.0 et la documentation compte 240 pages. C'est dire qu'elle est indispensable ! Essayez donc votre EDASM sous ProDOS 1.0 et, si cela ne fonctionne pas, il ne vous restera plus qu'à vous procurer (contre espèces sonnantes et trébuchantes) "ProDOS ASSEMBLER TOOLS" d'Apple (prix raisonnable). **MOUSE-TIMER** utilise la carte souris, mais surtout pas la souris (laissez-la dormir si vous vous servez de ce petit utilitaire). Oui, les manettes de jeux sont utilisables à la place de la souris et nous vous le prouvons avec un programme, dans ce numéro de **TREMPLIN MICRO**.

SUITE DU COURRIER (PAGE 68)

Vous avez écrit à TREMPLIN MICRO



• Je suis débutant en micro-informatique. Voici plusieurs questions :

— Si j'ai bien compris, une valeur PEEK correspond à chaque touche du clavier. Laquelle ? Quelles sont celles des deux POMMES, de la barre d'espacement, etc. ?

— Quand ferez-vous paraître un programme transformant le clavier de mon Apple en clavier d'orgue électronique ?

— Comment réaliser l'impression des graphiques de vos programmes HISTOGRAMME, VISUBLOCS, etc. ?

TM • Ne pas confondre PEEK et valeur ASCII d'un caractère. Quand on tape PRINT PEEK(768), on demande à l'Apple (plein de bonne volonté, non ?) d'afficher la valeur décimale contenue dans la case n°768 de la mémoire. Des comme cela, il y en a 65536.

Pour connaître la valeur ASCII (décimale) de chaque caractère, taper les lignes suivantes et faire RUN

```
10 HOME : VTAB 12 : PRINT "LETTRE → " : GET L$
20 PRINT ASC(L$) : " " : GET L$ : PRINT : GOTO 10
```

On en sort par RESET.

• La musique, il y en a déjà eu... et nous y reviendrons. La place manque.

• Avec ce numéro, côté copie HGR, nous répondons à la question !

• Programmeur débutant, je dispose depuis peu d'un Apple IIe avec 65C02 + carte féline (ext. 80 col. + 128 K — couleur).

1° Toutes les indications figurant dans le livre "Clefs pour Apple IIc" de Mme Bréaud-Pouliquen (Ed. du PSI) sont-elles applicables à un Apple IIe équipé du processeur 65C02 ?

2° Pourriez-vous publier dans votre revue un programme qui avec ProDOS permette de lister le catalogue des fichiers d'une disquette avec la fonction CAT (ou une commande annexe à intituler CAT.2 ou CAT.3) non pas sur un seul groupe de colonnes mais (en mode 80 colonnes bien sûr) sur 2 ou 3 groupes de colonnes afin d'avoir sur une seule page écran, 3 fois plus de titres de fichiers, chaque fichier ne comportant que son nom (15 caractères), le type de langage (3 caractères) et le nombre de blocs utilisés par fichier (3 caractères) + les espaces entre

chaque rubrique, soit $3 \times 26 = 78$ colonnes.

Je crois que ce type de programme (que je suis encore incapable d'écrire), intéresserait de nombreux utilisateurs non chevronnés comme moi.

3° Lorsque je copie certains programmes personnels d'une disquette sur une autre, à l'aide de la disquette utilisateur ProDOS ou via la mémoire vive par la commande SAVE et que je vérifie par la sous commande "Comparer Fichiers" de la disquette utilisateur ProDOS si ces 2 fichiers sont identiques, la réponse que j'obtiens est "Fichiers non identiques" et ceci quelles que soient la disquette source et la disquette destination (j'ai fait de nombreux essais différents). J'ai bien sûr évité d'utiliser des noms de fichiers interdits. D'autre part, il m'arrive, toujours dans le cas ci-dessus, si, par exemple, un programme à copier occupe 17 blocs, de voir après copie le nombre de blocs utilisés augmenter non pas de 17 blocs mais de 2 fois 17 blocs alors que je n'ai copié le fichier qu'une fois ?

Gilles P. (58000 NEVERS)

TM • Tous les IIc et IIe avec 65C02 sont compatibles (Apple dixit), y compris l'affichage en 80 colonnes... quand on s'offre la carte 80 colonnes étendue.

• CATALOGUE : voici un exemple de catalogue sur 3 colonnes. On peut trouver mieux, bien entendu, mais en y mettant le prix.

```
10 TEXT : D$ = CHR$(4) : PRINT
D$ "PRÉ3" : H = 1 : V = 1 : DIM F$(
100), BL$(100) : PRINT D$ "PRÉFI
X" : INPUT C$ : PRINT D$ "OPEN" C
$, "TDIR" : PRINT D$ "READ" C$ : O
NERR GOTO 17
```

```
15 FOR I = 1 TO 3 : INPUT BOF$ :
NEXT : FOR I = 1 TO 52 : INPUT
F$(I) : F$(I) = LEFT$(F$(I),
28) : BL$(I) = RIGHT$(F$(I), 3
) : NEXT
```

```
17 PRINT D$ "CLOSE" : I = I - 3 : DI
= INT(I / 3) : IF DI * 3 <
I THEN DI = DI + 1
```

```
25 FOR J = 1 TO I : POKE 1403, H :
VTAB V : PRINT LEFT$(F$(J),
28) + BL$(J) : V = V + 1 : IF (J
/ DI) = INT(J / DI) THEN
IF H < 59 THEN H = H + 26 : V =
1
```

```
30 NEXT : VTAB 20 : END
```

• Votre troisième question nous laisse perplexes. Nous ne comprenons pas. Ah ! Si nous avions vos disquettes, vos programmes... et votre ProDOS !



• Est-il possible de récupérer la valeur des 2 pointeurs remis à zéro avec l'instruction NEW... et comment ? Si vous avez le temps, pourriez-vous m'aider ?

Alain F. (27000 EVREUX)

TM Réponse ci-contre. N'en profitez pas pour vous offrir une centaine de NEW successifs ; nous n'avons pas été aussi loin dans nos tests !

• Comment peut-on lister le catalogue d'une disquette formatée DOS 3.3 ou PASCAL... quand on est sous ProDOS, sans passer par les utilitaires SYSTEME ? Merci de bien vouloir me trouver une solution.

Qui se cache sous le nom de NESTOR ?

Patrick C. (27520 BOURG-THEROULDE)

TM • Une idée de programme (à réaliser)... pour ceux de nos Lecteurs qui maîtrisent parfaitement le DOS 3.3, PASCAL... et ProDOS !
• NESTOR, c'est LE SAGE, en deux mots, qu'on se le dise !

• J'ai acheté un Apple II Plus, et je n'ai pas pu trouver un manuel de références. Par contre, je peux acheter celui du IIe. Puis-je le faire, ou avez-vous une autre solution ?

Louis B. (90100 DELLE)

TM Oui, mais vous aurez des surprises. Il est probable que de vieux manuels II+ traînent en France. Peut-être pourriez-vous fouiller dans les rayons LIBRAIRIE de magasins comme SIVEA, ILLEL, etc.

• Comment connaître le nombre de secteurs disponibles sur une disquette ProDOS, pendant le fonctionnement d'un programme (avant un BSAVE fichier, par exemple) ? De plus, après un HGR2, comment, en mode direct, voir ce que l'on tape au clavier (pas commode, dans le noir absolu, de faire un BLOAD FICHER, A\$4000 !) ?

N. F. (85160 ST-JEAN-DE-MONTS)

Après un NEW malencontreux

SAUVÉ

Tout n'est pas perdu... si, immédiatement après le NEW malheureux, vous tapez un heureux BRUN SAUVÉ (qui vous fera retrouver votre programme).

1	*		
2		ORG	\$300
3	*		
4	INDEXL	EQU	\$5E
5	INDEXH	EQU	\$5F
6	TXTTABL	EQU	\$67
7	TXTTABH	EQU	\$68
8	LOMEML	EQU	\$69
9	LOMEMH	EQU	\$6A
10	BELL	EQU	\$FF3A
11	*		
12	PROG	LDA	TXTTABL
13		LDY	TXTTABH
14		STA	INDEXL
15		STY	INDEXH
16		JSR	RECPRG
17		LDA	\$3
18		CLC	
19		ADC	INDEXL
20		STA	LOMEML
21		LDA	INDEXH
22		ADC	\$0
23		STA	LOMEMH
24		JSR	BELL
25		JMP	\$D4F5
26	*		
27	RECPRG	LDA	\$1
28		TAY	
29		STA	(INDEXL),Y
30	BCLA	LDY	\$5
31	BCL1	LDA	(INDEXL),Y
32		INY	
33		BNE	BCL1
34		INY	
35		TYA	
36		CLC	
37		ADC	INDEXL
38		STA	INDEXL
39		BCC	BCL2
40		INC	INDEXH
41	BCL2	LDY	\$1
42		LDA	(INDEXL),Y
43		BNE	BCLA
44		RTS	

Si vous n'avez pas d'assembleur, tapez seulement les 60 octets ci-dessous.

```

300: A5 67
302: A4 68
304: 85 5E
306: 84 5F
308: 20 1E 03
30B: A9 03
30D: 18
30E: 65 5E
310: 85 69
312: A5 5F
314: 69 00
316: 85 6A
318: 20 3A FF
31B: 4C F5 D4

31E: A9 01
320: A8
321: 91 5E
323: A0 05
325: B1 5E
327: C8
328: D0 FB
32A: C8
32B: 98
32C: 18
32D: 65 5E
32F: 85 5E
331: 90 02
333: E6 5F
335: A0 01
337: B1 5E
339: D0 E8
33B: 60
  
```

BSAVE SAUVÉ, A\$300, L\$3C

Vous avez écrit à TREMPLIN MICRO



- TM** 1° Le nombre de BLOCS disponibles vous est donné par :
- $$L = 280 - (\text{PEEK}(48828) + \text{PEEK}(48829) * 256)$$
- 2° Processus : HGR2 : TEXT : BLOADFICHIER, A\$4000 : POKE 49237,0 : POKE 49232,0 (de cette manière, vous retrouverez intact votre écran HGR2).

• Comment sauver à la fois programme et réponses pour tout conserver et retravailler dessus par la suite (sous ProDOS) ?

Autre question : si je réussis à faire bouger (par effacement couleur) points et lignes, comment animer une figure plus complexe ? Existe-t-il un logiciel adéquat ?

- TM** • Le SED (ou DOS... ou ProDOS) se charge de conserver les variables avec l'instruction STORE, utilisée comme SAVE, sous ProDOS. RESTORE permettra de les récupérer. Avec le DOS 3.3, il faut utiliser un fichier binaire... et connaître l'adresse de début et de fin des variables, ce qui est plus complexe.

SAISIE

```
10 HOME
20 FOR I = 1 TO 5
30 INPUT "DONNEE —> ";MS(I)
40 NEXT
50 PRINT CHR$(4) "STORE ESSAI"
```

Taper un RUN, et saisir les données.

RESTORE

Après mémorisation, taper NEW et ceci :

```
10 HOME
20 PRINT CHR$(4) "RESTORE ESSAI"
30 FOR J = 1 TO I - 1 : PRINT MS(J) : NEXT
```

• Seul le langage machine assure de bonnes animations par utilisation des transferts d'octets. VIDEO WORKS est un logiciel complet en la matière (Hayden).

• J'ai parfois un "Syntax error in 65028", après un simple SAVE ou CAT. Pourquoi ? Comment y remédier sans éteindre mon Apple ?

B. L. (37000 TOURS)

- TM** Cette erreur résulte souvent de quelque POKE malencontreux. Il faut entrer dans le moniteur par "CALL—151", puis relancer le système avec : * 3EAG (DOS 3.3) * 3D0G (ProDOS)

• Comment transférer une page graphique HGR2 en HGR, par une routine en Basic ? J'ai essayé une boucle, mais c'est interminable ! Pourquoi ne pas insérer des petites annonces dans T.M. ?

- TM** • Essayez le (court) programme ci-après :
- ```
10 HGR
20 PRINT CHR$(4) "BLOAD WORLD MAP.PIC,
A$4000"
30 A$ = "2000 > 4000.5FFEM D823G"
40 FOR I = 1 TO LEN(A$) : POKE 511 + I,
ASC(MID$(A$,I,1)) + 128 : NEXT : POKE
72,0 : CALL — 144
50 REM ET VOILA L'IMAGE
```

• Pour les petites annonces... la question reste en suspens. Il faudrait, pour les rendre intéressantes que T.M. paraisse tous les mois.

• Lorsque vous présentez des programmes de gestion d'affichage, vous le faites toujours en 40 colonnes, quitte à faire ensuite un nouvel article pour les 80 colonnes. Je sais bien que c'est économiquement intéressant, car le même corps de programme sert deux fois, mais qui, à l'heure actuelle ne dispose que de 40 colonnes ? Je pense que la grande majorité des anciens possesseurs d'Apple II+ ont déjà résolu ce genre de problème... et vous visez manifestement un public de néophytes.

Pierre G. (42170 ST-JUST-ST-RAMBERT)

- TM** "Economiquement intéressant" ? Et vous connaissez les différences qui existent entre les gestions 40 et 80 colonnes ! Vous nous prêtez des calculs qui nous chagrinent, chez Lecteur. De plus, vous vous trompez. Il n'est pas forcément agréable, pour bon nombre de gens, de travailler en 80 colonnes, à cause d'une lisibilité qui est loin d'être excellente. Et pourquoi se fatiguer les yeux sur des caractères minuscules quand on peut réaliser la même lecture (ou saisie) en 40 colonnes ? Enfin, je vous signale que tous les possesseurs d'Apple n'ont pas forcément les moyens de s'offrir, du moins la première année, une carte 80 colonnes.

• Peut-on effacer la /RAM par un simple ordre BASIC sans passer par des (PRINT CHR\$(4) : "DELETE/RAM/....."). Autrement dit existe-t-il un "CLEAR" bête et brutal de la /RAM qui conserverait intactes les autres zones (mémoire et programme) ?

La Notice de l'imagewriter pêche beaucoup par son manque d'exemples d'application. Aussi je n'ai toujours pas compris (entre autres) comment fixer la



## Vous avez écrit à TREMLIN MICRO



longueur de page et les marges hautes et basses. Si vous pouviez m'indiquer les caractères de commande à envoyer pour avoir par exemple :

- interligne : 6/inch
- longueur papier : 10.7 inch
- marge haute : 1 inch
- marge basse : 2 inch

... je vous promets ma reconnaissance éternelle.

APPLEMENT VOTRE.

Guy V. (92370 CHAVILLE)

**TM** La /RAM est un volume ProDOS qui se comporte comme tout autre volume et il n'existe pas de fonctions pour l'initialiser. Le ProDOS est un Système structuré comme l'UCSD du PASCAL et il faut des utilitaires sur disquettes pour le gérer. Il est véritablement la version professionnelle du DOS 3.3 et peut être utilisé avec un disque dur. Imaginez les suites malheureuses d'un INIT DISQUE DUR accidentel ! Le programme ci-dessous, qui est inédit, peut éventuellement vous rendre service.

```
100 REM EFFACE /RAM
110 :
120 DS = CHR$(4): GS = CHR$(7): DIM
 FS(60): TEXT : HOME : PRINT DS"PRE-
 FIX/RAM": PRINT DS"CAT"
135 :
136 REM LECTURE FICHIER DIRECTEUR /RAM
137 :
140 ONERR GOTO 190
150 PRINT DS"OPEN/RAM,TDIR": PRINT
 DS"READ/RAM": FOR I = 1 TO 51: INPUT
 FS(I): NEXT
190 PRINT DS"CLOSE"
195 IF I = 4 THEN PRINT GS GS: INVERSE :
 PRINT "/RAM VIDE !": NORMAL : GOTO
 500
200 :
201 REM TRI
202 :
210 FOR J = 4 TO 1 — 1: FS(J) = LEFT$
 (FS(J),17): IF LEFT$ (FS(J),1) = CHR$(32)
 OR LEFT$ (FS(J),1) = CHR$(42) THEN FS(J)
 = RIGHT$ (FS(J), LEN (FS(J)) — 1): PRINT
 DS"UNLOCK"FS(J): PRINT DS"DELE-
 TE"FS(J): PRINT GS:: NEXT : HOME :
 PRINT DS"CAT": PRINT "TERMINE !":
 PRINT GS
```

```
TM 300 INVERSE : PRINT "FICHIERS": NORMAL :
 PRINT " : ": FOR J = 4 TO 1 — 1: HTAB
 20: PRINT FS(J): NEXT : PRINT : HTAB 20:
 PRINT "..... effacés !"
500 END
```

• A quoi sert un "CHR\$(17)", trouvé après un "PRE3" ?

**TM** CHR\$(17) fait passer l'Apple en mode 40 colonnes, mais sans désactiver la carte 80 colonnes... que l'on retrouve intacte par un CHR\$(18).

• J'ai bien aimé votre (court) article sur les imprimantes ; j'aurais apprécié ce conseil il y a cinq mois lors de l'achat de ma "SCRIBLE" ; car bonjour la "doc", après 30 pages de déchiffrement d'américain, l'on arrive tout juste à placer correctement le papier et le ruban...

Quant aux possibilités graphiques ou caractères spéciaux, j'en suis toujours à "la case départ". Il y avait pourtant bien "vaguement égarée" à la fin du manuel une feuille explicative... mais DUR ! DUR ! l'APPLE... Conclusion «à vendre» !

Il serait bon que vous mettiez en garde les éventuels acheteurs de matériel APPLE des possibilités et du prix... car à Strasbourg une IMAGEWRITER coûte 1500 Frs plus cher qu'à Paris. Un "simple" APPLE WORKS coûte 1200 Frs de plus (prix Strasbourg 3000 Frs T.T.C.).

Vous pourriez aussi nous parler davantage des logiciels. En continuant mon exemple de l'imprimante, il est impossible de mettre un accent circonflexe même avec APPLE WORKS... En "bidouillant" j'ai trouvé, mais CAMBRONNE fut de la saisie...

Je crois en vérité que beaucoup de lecteurs cherchent des trucs simples (c'est mon cas) par exemple un utilitaire pour faire des "MENU" sur un APPLE IIc. Je lance ça (?) juste faire 1 ou 2 pour lancer un programme du "CAT"... Au poil ! quel temps de gagné pour faire d'autres petits trucs !

Cela existe-t-il ?

Michel C. (HAGUENAU)

**TM** Pour ce qui est de la première partie de votre lettre, nous sommes ouverts à toutes les suggestions. Les imprimantes Apple sont livrées avec une doc en Français et, d'après NESTOR, celle de la nouvelle Imagewriter II est extra. Faut-il réellement publier, dans Tremplin Micro, des programmes de 10 ou 20 lignes, semblables à un iota près, aux centaines que l'on trouve dans tous les bouquins d'initiation ? La question est posée. A vous d'y répondre, amie Lectrice, ami Lecteur... et seule raison d'être de la revue !

Guy-HACHETTE



# LA MULTIPLICATION

Ce court programme de démonstration nous montre, sur écran, comment traiter le problème de la multiplication... en langage machine. Attention ! le résultat doit tenir sur un octet et ne doit donc pas dépasser \$FF (255). Le multiplicande est stocké à l'adresse \$6 et le multiplicateur à l'adresse \$7. Le résultat sera lu... à cette même adresse \$7.

```

100 TEXT : PRINT CHR$(4)*PR#3: HOME : GOSUB 430
110 FOR I = 1 TO 10:T$ = T$ + "_____": NEXT
120 PRINT T$: INVERSE : VTAB 1: PRINT "LA MULTIPLICATION EN LANGAGE MACHINE"
 ;; POKE 1403,79: PRINT "1": NORMAL
130 PRINT : PRINT "Soit la multiplication:": PRINT "-----"
140 PRINT : PRINT " 1 2": PRINT "X 1 3": PRINT "-----": PRINT "
 3 6": PRINT "+ 1 2 ": PRINT "-----": PRINT "= 1 5 6"
150 INVERSE : HTAB 14: VTAB 7: PRINT "$C": HTAB 14: PRINT "$D": NORMAL
160 H = 16: VTAB 7: HTAB H
170 L$ = " 1100 1101 ----- 1100 0000 1100
 1100 ----- 10011100"
180 FOR I = 1 TO 99 STEP 11: HTAB H: PRINT MID$(L$,I,11): NEXT
190 PRINT : PRINT : PRINT : PRINT "FINALEMENT, LE RESULTAT EST:": PRINT : PR
 INT "1001 1100 soit ";; INVERSE : PRINT "9C";; NORMAL : PRINT "= 156"
200 VTAB 23: PRINT "PRESSER UNE TOUCHE SVP ";; GET R$: PRINT "
210 INVERSE : FOR I = 4 TO 21: VTAB I: HTAB 32: PRINT " ": NEXT : NORMAL
220 POKE 32,35: POKE 33,45: POKE 34,3: POKE 35,23: PRINT : HOME
230 PRINT "MEME TECHNIQUE EN LANGAGE MACHINE:": PRINT "-----"
 -----"
240 PRINT : PRINT "A9 0C LDA #0C : Multiplicande dans A"
250 PRINT "85 06 STA #06 : et stocké en 6, page 0"
260 PRINT "A9 0D LDA #0D : Multiplicateur dans A"
270 PRINT "85 07 STA #07 : et stocké en 7, page 0"
280 PRINT "A9 00 LDA #00 : Initialisation de A"
290 PRINT "A2 08 LDX #08 : Init. X (compteur)"
300 PRINT "46 07 LSR #07 : Premier décalage"
310 PRINT "90 03 RET BCC DEC : Si la retenue est 0"
320 PRINT "18 CLC : Annulation de retenue"
330 PRINT "65 06 ADC #06 : Ac=Ac + Multiplicande"
340 PRINT "6A DEC ROR : Décalage de l'Accum."
350 PRINT "66 07 ROR #07 : puis de #07"
360 PRINT "CA DEX : X = X moins 1"
370 PRINT "D0 F5 BNE RET : Encore si pas zéro"
380 PRINT "60 RTS : Fini!"
390 POKE 32,0: POKE 33,80: VTAB 23: HTAB 1: PRINT " POUR OBTENIR LE MENU D
 E CETTE DISQUETTE, PRESSER LA TOUCHE <M> COMME MENU ";; GET R$
400 TEXT : HOME : IF R$ < > M$ AND R$ < > "m" THEN END
410 PRINT : PRINT CHR$(4)*RUN MENU"
420 REM CES DEUX LIGNES ONT PLACE LE PROGRAMME A L'ADRESSE $300
430 FOR I = 768 TO 793: READ A: POKE I,A: NEXT : RETURN
440 DATA 169,12,133,6,169,13,133,7,169,0,162,8,70,7,144,3,24,101,6,106,102,7
 .202,208,245,96

```

Vous lirez, dans un prochain numéro de *Tremplin Micro*, les réponses aux questions que suscitera cette page, volontairement succincte quant aux explications. Réfléchissez... et interrogez !



Un outil indispensable

# MOUSE DESK

ProDOS  
PLUS  
SOURIS

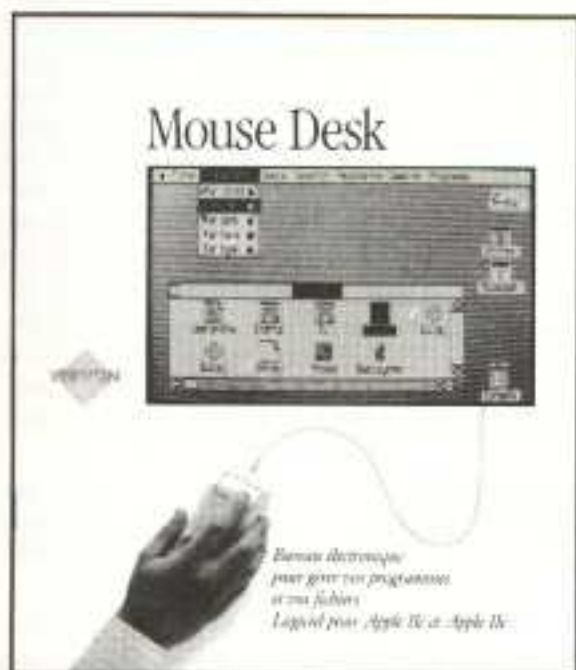
Je suis conquis par *MOUSE DESK*, l'un des récents produits de *Version Soft*. Pourtant, il ne m'avait pas réellement enthousiasmé lors d'un premier essai, sans souris, sur *Apple IIe*. C'était génial, beau... mais je n'en comprenais pas tellement l'utilité.

J'étais persuadé que *Version Soft* et *Apple* réunis s'ingénieraient, avec *Mouse Desk*, à nous faire avaler la philosophie du Mac... pour nous amener à acheter cette machine. J'aurais pu rester sur cette idée. Le logiciel conservait son caractère génial.

Mais je me suis offert l'*UniDisk* (parfaitement : offert). On lira par ailleurs ce que Nestor pense de ce nouveau lecteur. Je partage son opinion : c'est extra, mais avec *Mouse Desk*, cela devient tout à fait pratique.

Adieu les autres utilitaires ProDOS ! Pour ce qui est de transférer des programmes et d'organiser une disquette, *Mouse Desk* comblera vos plus chers désirs.

Je ne suis pas un très chaud partisan de la souris et il m'arrive de choisir tel logiciel... simplement parce qu'il fonctionne SANS LA SOURIS. Côté *MOUSE DESK*, je suis bien obligé de reconnaître que la souris joue parfaitement son rôle. Aucun doute, dans peu de temps, toute la France possédera une version de cet excellent produit !



TREMLIN MICRO



## Le nouvel anglais sans peine...

### et sur votre Apple !

Qui ne connaît pas, au moins de nom, la fameuse méthode *ASSIMIL* ? Qui n'a pas essayé d'apprendre, avec elle, une langue étrangère ? L'anglais, par exemple. *ASSIMIL* est vraiment l'une des méthodes les plus répandues... et dans le monde entier.

Il est agréable de constater que ses promoteurs vivent avec leur temps. Ils ont utilisé le disque, la cassette... et nous retrouvons les exercices d'auto-contrôle sur disquette, à l'intention des utilisateurs d'*Apple II+*, *IIc* et *IIe*.

What is it ? Un logiciel — vous vous en doutiez un peu ? — dont le but est de vous faire assimiler (le terme est doublement bien choisi) une centaine de points de grammaire, orthographe ou vocabulaire de l'anglais courant. Comment ? En vous présentant la règle correspondante et en vous proposant de faire des exercices d'application (plus de 500 !).

Les auteurs ont prévu les principaux cas d'erreurs généralement rencontrés et la plupart du temps, le logiciel vous corrigera en vous indiquant votre faute... ou alors — on ne saurait tout mettre sur une disquette — il vous renverra à la règle... et au livre.

Ce livre, c'est précisément *Le nouvel anglais sans peine* (580 pages)... auquel correspondent déjà, on le sait, disques et cassettes. Quel est l'intérêt d'exercices sur disquette ? D'abord, l'aspect visuel. Alors qu'une page de livre présente les informations de façon statique, sans relief, l'ordinateur permet une exploitation plus rationnelle du graphisme, de la nature des données... et du temps de présentation.

L'aspect ludique n'est pas à négliger. On joue davantage avec un Apple qu'avec un bouquin, même excellent. Toutefois, il est certain que c'est surtout la détection des erreurs qui donne tout son intérêt à ce nouveau volet de la méthode *Assimil*. L'erreur devient source d'enseignement.

Bref : une bonne utilisation d'une méthode qui a déjà fait ses preuves, sur un matériel qui n'a plus rien à prouver : votre Apple chéri !

G.-H.



# Votre bibliothèque INFORMATIQUE

• par NESTOR



## • LE LOGO SUR APPLE (Cedic/Nathan)

On sait que le LOGO a été conçu par Seymour Papert et ses collaborateurs. Il en existe une version française sur Apple, et Harold Abelson, dans une traduction de Louis-Philippe Hébert, se propose de nous la faire découvrir.

Nous sommes donc en présence d'une vraie méthode, abondamment et judicieusement illustrée, avec un texte qu'on ne peut pas qualifier de concis, mais qui a le mérite de rester clair et agréable à lire.

J'avoue avoir une expérience fort limitée de la pratique du Logo, une pratique qui remonte à quelques années. La tortue ne m'inspire guère... et ma doc personnelle (en anglais) ne m'avait guère incité à persévérer dans un apprentissage que j'avais estimé futile. Depuis, un peu partout, j'ai vu des enfants jouer avec cette fameuse tortue... et j'ai compris que Seymour Papert avait de bonnes raisons pour défendre son langage.

C'est vrai que l'on peut apprendre à aimer l'informatique en dessinant et en manipulant des objets que l'on crée

soi-même. Or, il est évident que, malgré l'apparente simplicité du Logo, on a besoin d'un guide, voire d'un excellent guide : *Le Logo sur Apple* remplira cette fonction.



## • L'ALGORITHMIQUE, de la pratique à la théorie (Cedic/Nathan)

Attention ! C'est du sérieux. Encore une méthode ? Mieux : un cours avec ses exercices, celui de G. Chaty et J. Vicard, un professeur de l'Université et un maître-assistant, qui nous livrent le produit d'une longue expérience pédagogique.

Si, comme moi, vous succomez à la fâcheuse tentation de ne point trop chercher à structurer vos "programmes" (grand mot pour désigner de modestes "essais" !), il est sûr que ce cours saura vous donner de meilleures idées... en Pascal, que les auteurs ont précisément adopté parce qu'il "prépare l'acquisition de bonnes habitudes de programmation" (je les cite).

Un détail qui a son importance : j'ai évoqué des "exercices", mais j'ai omis de préciser que ceux-ci sont largement développés. Ce ne sont certes pas les listages et solutions qui manquent

dans cet ouvrage, réalisé avec sérieux, à l'intention de lecteurs désirant "étudier" l'informatique et non la "survoler"...

Vous savez, comme moi, que les Français ont la regrettable habitude de ne point lire les documentations accompagnant leurs appareils. On branche d'abord et on regarde ensuite pourquoi le moteur fume. Explication : il n'y a pas si longtemps, tout était formulé en anglais.

## • LIRE PASCAL (Cedic/Nathan)

Il y a aussi un sous-titre : "le langage de la programmation structurée", un point sur lequel le manuel de Dominique Piot rejoint le précédent, consacré à l'Algorithmique. Moi, quand je lis quelques chapitres de bouquins vantant les mérites du Pascal, j'éprouve toujours une irrésistible envie de bondir sur mon Apple chéri, de lancer mes disquettes Pascal, de sortir mes méthodes... et de poursuivre ma formation. Je suis même allé jusque-là, et notamment grâce à *Lire Pascal*, mais les compilations obligatoires m'agacent, et aussi le fait que j'ai l'impression d'être moins libre qu'en Basic.

Aïe ! où ai-je mis les pieds ! Je suis, M. le Professeur : quand j'y aurai enfin pris goût, je ne pourrai plus m'en passer. Le pire c'est que vous avez sûrement raison... Peut-être aurais-je besoin de leçons particulières... mais revenons au livre pour encourager les gens sérieux, capables d'oublier leur Basic (s'ils n'en ont jamais fait, c'est encore mieux), à étudier. C'est un manuel d'utilisation du Pascal, je l'ai souligné d'emblée. Il est mince et pas cher (120 pages environ), mais on en a pour son argent, ce qui n'est pas toujours le cas en matière d'édition, dans le domaine de l'informatique.

Les auteurs de Cedic/Nathan ont bien travaillé. Pas question d'avoir la dent dure. Il serait navrant que tant d'efforts ne se révèlent pas payants !

NESTOR.



# Version-Tel : le graphisme Minitel sur l'écran de votre Apple !

**Nouveau venu parmi les logiciels de communication, Version-Tel présente une originalité exclusive : il est le seul qui transforme vraiment un Apple en Minitel intelligent. Car il lui permet de mémoriser, d'imprimer et de traiter les écrans du Minitel, tout en conservant le graphisme des pages Teletel. Il fonctionne sur Apple IIe mais aussi, et c'est plus nouveau, sur IIc. Voyez plutôt...**

L'amateur désireux de «communiquer» dispose actuellement de deux possibilités :

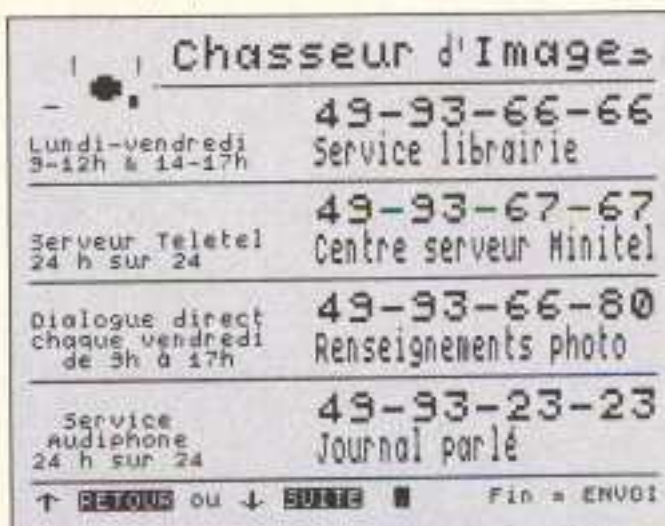
— le mode ASCII, pour transmettre du texte : la liaison s'établit à la vitesse 300 ou 1200 bauds et les graphismes sont exclus. C'est le mode standard des micro-ordinateurs utilisant l'interface RS-232 et des modems. Selon les machines, les caractères sont codés sur 7 ou 8 bits, avec 1 ou 2 bits de stop. Le rôle des logiciels consiste à accorder les protocoles des deux machines, à gérer la liaison en assurant affichage, mémorisation et impression ;

— le mode Teletel : c'est le standard adopté par les PTT pour près d'un million de terminaux Minitel actuellement en service. La liaison s'établit à 1200 bauds côté serveur (l'appelé) et à 75 bauds côté appelé. On dispose de plusieurs jeux de caractères, dont un pseudo-graphisme constitué d'une mosaïque de carrés joints ou disjoints. Chaque caractère peut être précédé d'attributs définissant sa couleur, la couleur du fond, le clignotement, la taille (double hauteur, double largeur, double taille) et quantité d'autres choses. L'objectif des PTT est clair : transmettre les plus beaux écrans possibles avec un minimum de signes. Pour les auteurs de logiciels, c'est un véritable casse-tête : il faut interpréter ces codes pour les afficher... ou les éliminer, solution la plus retenue jusqu'alors. De plus, gérer le mode 1200/75 bauds sur des machines comme l'Apple IIc était, jusqu'à maintenant, considéré comme quasi-impossible.

Version-Tel, de Version-Soft, résout ces problèmes et transforme l'Apple en Minitel. Vendu moins de 1 000 F, il nécessite les configurations suivantes :

— Apple IIc et modem Apple-Sectrad 1200/75, Digitelec, DTL +/V 23 ou boîtier Convertel associé à un Minitel (cette dernière solution permettant d'utiliser le modem du terminal PTT). Si l'on souhaite imprimer les graphismes Teletel, une ImageWriter est indispensable ;

— Apple IIe équipé, en slot 2, d'une carte série SSC Apple (de préférence), de la carte Digitelec ou de la Supercom MID. Mêmes modems qu'avec le IIc.



Un exemple d'écran Minitel imprimé avec Version-Tel utilisé sur un Apple IIc et une ImageWriter. Certaines zones de couleur du graphisme Teletel (fond bleu) ont disparu et le soulignement n'est pas restitué, mais l'écran ainsi obtenu reste tout à fait exploitable.

Pour une fois, les utilisateurs d'Apple IIc sont favorisés : c'est avec leur machine que Version-Tel est le plus facile à utiliser. Glissez la disquette, branchez le modem, allumez... vous avez un Minitel !

Avec l'Apple IIe, les réglages sont plus longs car il faut d'abord paramétrer la carte RS-232. Nous avons testé une version préliminaire de Version-Tel qui ne comportait aucune indication précise relative aux réglages des fameux «switches» de la carte. En fait, le logiciel gère tout lui-même, pourvu que l'interrupteur SW2 soit sur ON et le petit triangle orienté vers Terminal. Il n'y a donc aucun souci à se faire.

L'utilisation de Version-Tel ne pose guère de problèmes. Les écrans sont peu documentés et, au début, on patauge un peu. Après une petite demi-heure d'essais, tout devient évident.

On dispose de deux modes : mémoire et terminal. Les communications se font toujours en **mode terminal** : l'Apple dispose de 20 K pour mémoriser les pages Teletel reçues ; quand cette capacité est atteinte, un bip signale qu'il faut sauvegarder tout cela sur disquette. Les serveurs Teletel ont le chic pour transmettre des pages inutiles ; avec Version-Tel, on les fait défiler à grande vitesse et on trie APRES, une fois la communication terminée.

En **mode mémoire**, on peut revivre la

communication précédente ou tout appel mémorisé sur le disque et imprimer les pages (en graphique ou texte), les mémoriser (en binaire, texte ou graphique), ou «restructures» en éliminant les pages indésirables.

Le **mode apprentissage**, quand il est activé, mémorise les opérations effectuées et pourra les renouveler, automatiquement, lors d'un appel futur. Cette fonction est supérieure à celle de la carte Apple-Tel : Version-Tel tient compte des temps de réponse du serveur, qui peuvent donc être variables d'un appel à l'autre, et raccourcit les séquences, sans risque de perte d'information.

Pourquoi transformer un Apple en Minitel ? Pour enregistrer les écrans, les imprimer ou en extraire les informations essentielles ! On peut ainsi réduire les

temps de communication en faisant défiler les pages à toute vitesse pour les relire une fois le téléphone raccroché.

Version-Tel fait tout cela, vite et bien. Les graphismes Teletel sont respectés, à quelques nuances près : certains attributs des écrans Minitel sont oubliés, comme le soulignement, certaines couleurs de fond (bleu notamment) ou le clignotement. Certaines pages Teletel seront donc un peu moins belles sur Apple que sur Minitel. Même chose évidemment à l'imprimante... mais Version-Tel étant le premier logiciel pour Apple capable d'imprimer les écrans du Minitel en mode texte ou graphique, nous pardonnons volontiers cette lacune. Mieux : Version-Tel fait des économies et va même jusqu'à gérer des remontées sur l'ImageWriter pour mettre deux écrans côte à côte et consommer moins de papier !

Pour moins de 1000 F, Version-Tel va donc donner des ailes aux Apple IIc et IIe qui, jusqu'à maintenant, ignoraient les graphismes Teletel. C'est un outil facile à utiliser (après un petit apprentissage toutefois) et très bien conçu. En un mot, bravo !

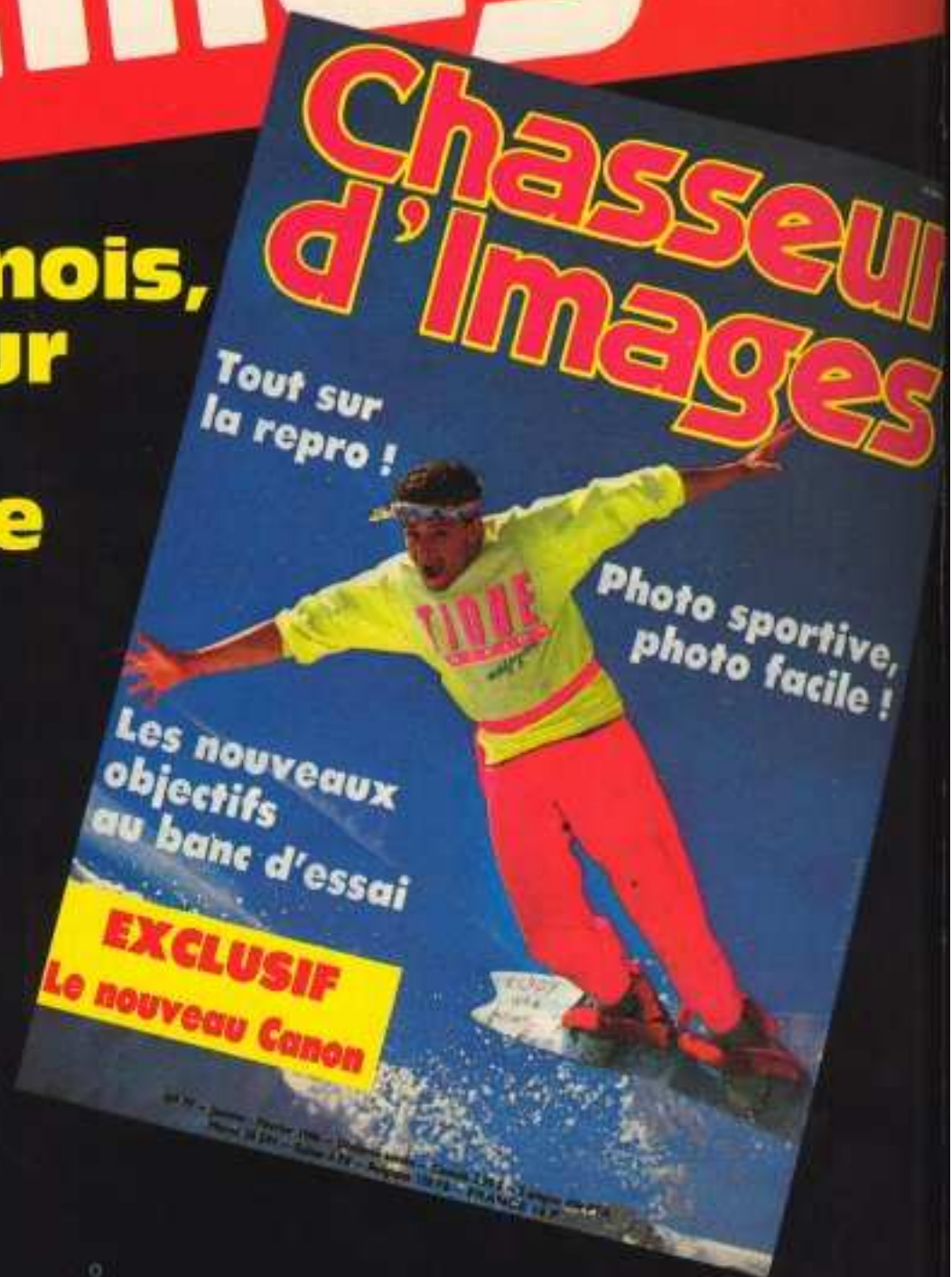
Côté lacunes, regrettons l'absence de numérotation automatique (il faut faire son numéro soi-même), malheureusement expliquée en partie par les lacunes des modems les plus en vogue.

Guy-Michel Cogné



# Chasseur d'Images

**Chaque mois,  
le meilleur  
de la  
technique  
et de la  
pratique  
photo !**



**Chez votre  
marchand de journaux !**