

# **tremplin micro**

**Chiffres en  
dents de scie**

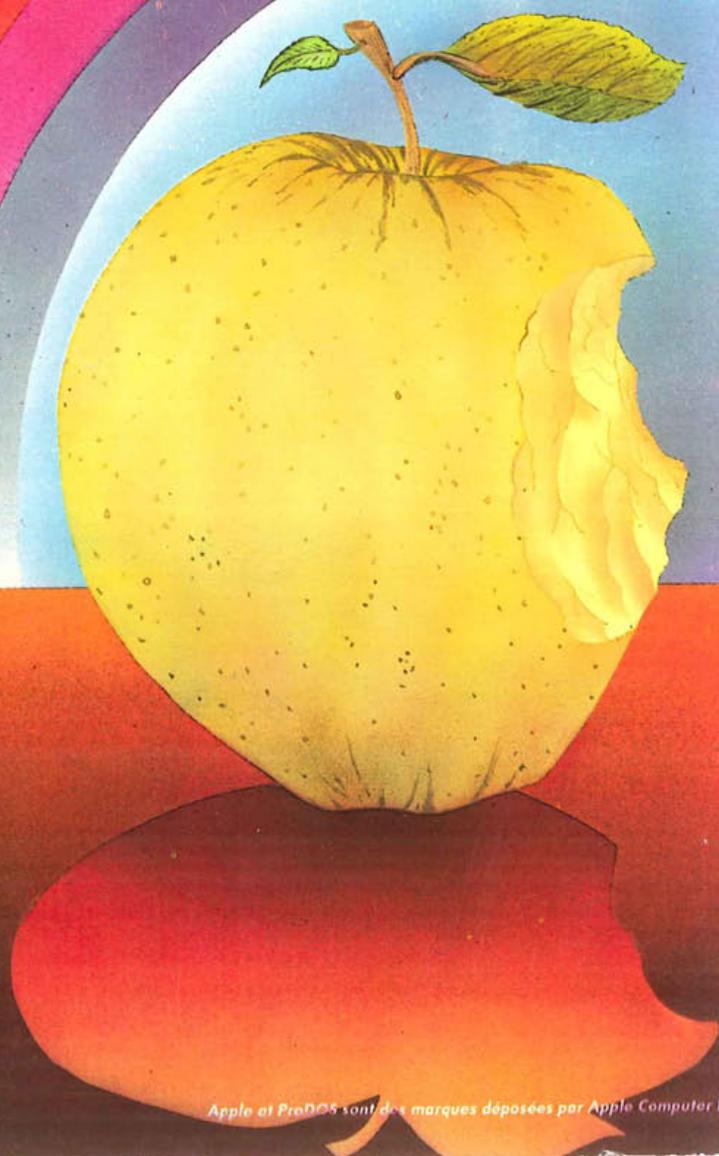
**Fonte spéciale  
pour vos titres**

**Un tas de  
petites routines**

**Flirtez  
avec ProDOS !**

**TOUJOURS PLUS  
POUR VOTRE  
APPLE !**

Bimestriel — Première année  
OCTOBRE-NOVEMBRE 1985  
Numéro 4 — 30 F



# tremplin micro

# 4

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

## **BIMESTRIEL**

Le numéro : 30 F  
Abonnement d'un an 150 F  
(5 numéros)

## **EDITIONS JIBENA**

### **Direction-Rédaction :**

Editions JIBENA

Guy-HACHETTE

La Petite Motte — Senillé  
86100 CHÂTELLERAULT.

### **Téléphone :**

(49) 93-66-66

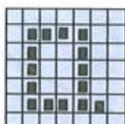
### **PUBLICITÉ :**

Joelle (même numéro)

### **Commission paritaire :**

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.

## Un tas de petites routines !



U'ATTENDEZ-VOUS de *Tremplin Micro* ? qu'il vous parle des nouveautés que, chaque semaine, nous annoncent certains constructeurs... mais que nous ne verrons peut-être jamais en France ? N'y comptez pas. Si l'avenir de l'informatique personnelle ne nous laisse pas indifférents, notre objectif immédiat se situe dans le présent, et vous l'avez d'ailleurs parfaitement compris. Il est vrai qu'il vous concerne directement puisqu'il se résume en quelques mots : *vous aider à tirer le meilleur parti possible de votre Apple actuel.*

Celui-ci a le mérite d'exister et vous avez la chance de pouvoir en disposer. Malheureusement, bon nombre de possesseurs d'Apple en connaissent encore mal le fonctionnement et, à plus forte raison, les innombrables possibilités.

*Tremplin Micro* vous propose, dans chacun de ses numéros, un tas de petites routines, mais elles exigent un minimum de connaissances de base. Hélas ! trop d'amateurs laissent de côté la documentation qui accompagne chaque machine et qui en constitue pourtant le meilleur abc. Y a-t-il là faute (ou négligence) du réseau de distribution ?

Ce n'est pas certain. L'ordinateur, personnel ou non, est encore considéré comme une sorte de boîte magique, capable de se suffire à elle-même, et rendant inutile tout effort intellectuel.

C'est évidemment faux. Il est certes permis de rêver, mais en restant réaliste. Non, monsieur, votre imprimante "graphique" ne va pas vous fabriquer des fromages à la suite d'un quelconque RUN et si d'aventure elle réalise cette opération, c'est évidemment parce qu'un logiciel lui fournit, un à un, les bits des précieux octets qui, de l'écran de votre Apple, vont passer sur le papier. Il arrive qu'une interface réalise cette opération, ce qui revient au même.

Ne vous laissez jamais abuser par la technicité verbale d'un vendeur. Exigez une démonstration.

Et puis, quand patiemment vous déchiffrez les programmes de *Tremplin Micro*, ne capitulez pas devant les apparentes (et parfois réelles) difficultés.

Plus la moisson sera longue... et plus la récolte sera bonne !

**GUY-HACHETTE.**

# Sommaire

# 4

## Jouer sur un Apple

- Pour 7 ronds ..... 15
- Logogrille  
(la phrase cachée) ..... 50

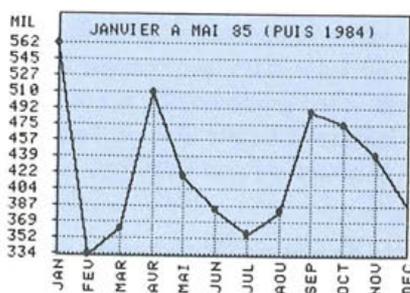
## Des trucs

- Calls ..... 31
- Est-ce une anagramme ? ..... 33
- Echange (swap) de variables  
(accélérez vos programmes de tri) . 35
- Petits trucs d'assembleur ..... 37
- Afficher un texte, empiler et  
dépiler ..... 38
- Cadres express ..... 55

## ProDOS et vous

- Préfixe — Créer des sous-volumes —  
Adresse et longueur d'un programme  
..... 36

## Chiffres en dents de scie



Les Lectrices et Lecteurs de **Tremplin Micro**  
ont beaucoup aimé les HISTOGRAMMES,  
DIAGRAMMES et VISUBLOCS.  
Voici les GRAPHES ..... 3

## PROGRAMMES UTILITAIRES

- POKE... d'accord,  
mais à quelle adresse ? ..... 17
- Désassemblage pratique ..... 18
- Catalogue thématique ..... 19
- Ouvrez donc une fenêtre ! ..... 43

## FONTE SPÉCIALE

Caractères pour vos titres ..... 22

## GRAPHISME

- Les fruits de la passion ..... 11
- Chiffres HGR ..... 12
- Plus loin avec la fonte GRAPH.CAR .. 13

## DÉMO

- Goto acrobatique par le  
truchement de l'ampersand .. 46
- Utilisation de la page 2 (text)  
pour stocker un écran ..... 48

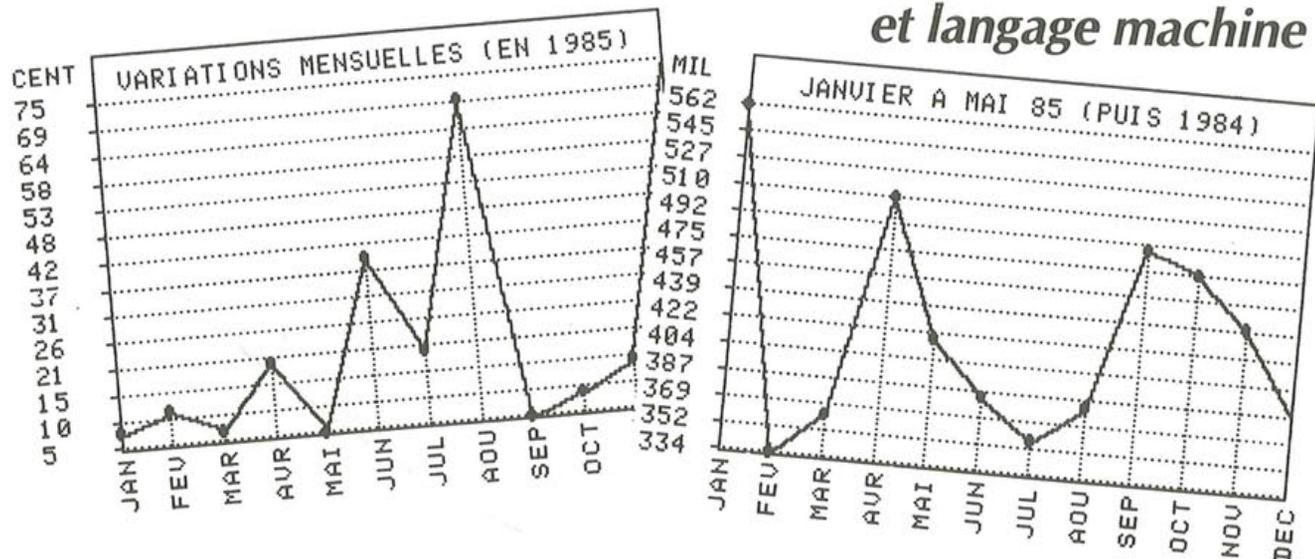
## LISEZ AUSSI :

- Complétez votre bibliothèque  
informatique ..... 14 — 21 — 32
- Questions et réponses ..... 31
- Les problèmes d'imprimante ..... 51

- Votre fiche n°4 ..... 41
- Le courrier des lecteurs ..... 57
- Index ..... 61
- Bulletin de commande  
et d'abonnement ..... 62

## TREMPIN MICRO

Le numéro 5 paraîtra le 3 décembre.



## Chiffres en dents de scie

Vous avez déjà apprécié nos HISTOGRAMMES, DIAGRAMMES et VISUBLOCS (respectivement présentés dans les numéros 1, 2 et 3 de *Tremplin Micro*). Voici les GRAPHEs, non moins "parlants", mais plus classiques.

**QU'EST-CE QU'UN GRAPHE ?** Reprenons paresseusement la très bonne définition qu'en donne le *Flammarion* : «Un schéma constitué par un ensemble de points (dits *sommets*) réunis par des lignes (dites *arcs* ou *arêtes*)».

**NOMBRE DE DONNÉES** Nous l'avons limité à 12 (comme les mois de l'année), mais il peut être modifié suivant les besoins. Toutefois, il n'est pas souhaitable de charger démesurément ce schéma.

**RÉSULTATS** Suivant la plus grande valeur rentrée, les chiffres sont divisés par 1000, par 100... ou utilisés tels quels. Il n'a pas été prévu de valeurs négatives. Les nombres apparaissant en abscisses sont obtenus en divisant la fourchette (plus grand moins plus petit) par 13. Le premier et le dernier correspondent seuls à des valeurs exactes. On n'ac-

corde que 3 caractères à la nature des données (ici, des mois), mais le texte peut être plus long lors de la saisie.

**MENU** Il est en langage machine, ce qui permet un affichage instantané. Le tracé du graphe pourrait être effectué à l'insu de l'utilisateur, en prolongeant l'affichage du menu, mais il nous a semblé amusant de montrer comment se déroule l'opération... assez rapidement d'ailleurs.

**FONTE DE CARACTÈRES** Elle est limitée à l'apostrophe, au point, aux nombres... et aux 26 lettres capitales.

**IMPRESSION** Attention ! elle est ici prévue avec une interface graphique de chez BIP. Vous devrez adapter cette partie du programme à votre cas personnel.

*Ce programme fonctionne avec ou sans carte 80 colonnes.*

# GRAPHE

## BASIC 1

```
10 TEXT : NORMAL : PRINT CHR$(12); CHR$(21): HOME
   : SCALE= 1: ROT= 0: HCOLOR= 3
20 DIM V$(12,1),H(12)
30 ONERR GOTO 1690
40 PRINT CHR$(4)"BLOAD GRAPH.MEN": GOSUB 1150
50 TR$ = " ": FOR I = 1 TO 10:TR$ = TR$ + "____": NEXT
60 :
70 REM *****
80 REM **      MENU EXPRESS      **
90 REM *****
100 :
110 TEXT : CALL 768
120 GOSUB 200: IF R < 1 OR R > 6 THEN 120
130 HOME
140 ON R GOTO 300,870,1320,1410,1610,1740
150 :
160 REM *****
170 REM **      PETITES ROUTINES      **
180 REM *****
190 :
200 POKE - 16368,0: WAIT - 16384,128,127: POKE - 16
   368,0:R = PEEK ( - 16384) - 48: RETURN
210 HOME : INVERSE : HTAB (41 - LEN (T$)) / 2: PRINT
   T$: NORMAL : POKE 34,3: PRINT : RETURN
220 HTAB 1: CALL - 868: CALL - 198: HTAB (41 - LEN
   (T$)) / 2: PRINT T$: RETURN
230 PRINT : PRINT CHR$(4)"CATALOG": PRINT : PRINT :
   PRINT : PRINT : VTAB 20: PRINT TR$: RETURN
240 POKE 49232,0: POKE 49234,0: POKE 49237,0: POKE 492
   39,0: RETURN
250 :
260 REM *****
270 REM **      1. SAISIE DES DONNEES      **
280 REM *****
290 :
300 PRINT : VTAB 22: CALL - 198: PRINT "DESIREZ-VOUS
   LIRE UN FICHER (O/N) "; GET R$: POKE - 16368,0:
   IF R$ = "N" THEN 350
310 HOME : IF R$ < > "0" THEN 300
320 GOSUB 230
330 VTAB 22: CALL - 868: PRINT "TITRE A LIRE GRD0";:
   HTAB 14: INPUT " ";F$: IF F$ = "" THEN 300
340 VTAB 21: PRINT : PRINT CHR$(4)"OPEN"F$: PRINT C
   HR$(4)"READ"F$: INPUT N: FOR I = 1 TO N: INPUT V$(
   I,0): INPUT V$(I,1): NEXT : INPUT TI$: PRINT CHR$(
   4)"CLOSE"F$
350 T$ = " SAISIE DES DONNEES ": GOSUB 210
360 VTAB 5: PRINT "<?> RENVOIE AU MENU SANS CHANGER LE
   NOM- BRE DES DONNEES DEJA SAISIES": PRINT : PRI
   NT "</> RENVOIE A L'ITEM PRECEDENT"
```

**SCALE :** Echelle utilisée pour tracer avec DRAW (ou XDRAW). (Il est déconseillé d'agrandir des caractères).

**ROT :** Angle de rotation de la forme (ici, un caractère). Valeur normale : 0.

**LOMEM :** On n'a pas jugé utile de modifier la valeur de LOMEM (fin de programme). Ici, les variables peuvent s'étendre sans problème, jusqu'à l'adresse \$4000 (HGR2). Si vous fixez LOMEM, créez une ligne 5 ainsi conçue :

**5 LOMEM : 25290**

← Attente sans curseur. Il faut noter qu'un CTRL-C provoquerait un BREAK... gênant.

← Avec ProDOS, remplacer "CATALOG" par "CAT"  
← PAGE2 visualisée sans effacement du graphe.

← CALL-198 = CTRL-G = Bell... la cloche. Vous pouvez inclure un CTRL-G dans un programme en tapant un CTRL-G entre des guillemets... ou à la suite d'un caractère lui-même placé entre des guillemets. Il n'apparaîtra ni à l'écran, ni sur le papier, mais vous l'entendrez si vous listez votre programme... ainsi qu'en exécutant celui-ci.

Le rappel des fonctions est fait lors de chaque appel à la routine "Saisie des données".

Il convient de noter → que cette routine permet de :

- lire un fichier
- saisir des données
- modifier des données

NM sauvegarde la → valeur primitive de N.

Le point d'interrogation → renvoie ici au menu. Le / renvoie à l'INPUT précédent.

#### Pourquoi des variables alphanumériques ?

Simplement parce que la saisie est plus facile, notamment quand l'une des données est une chaîne de caractères.

Le programme pouvait certainement traiter les nombres d'une autre manière (et même de plusieurs autres). Si vous en trouvez de plus performantes... à vos claviers !

## GRAPHE

## BASIC 2

```
370 PRINT : PRINT "<RET> TERMINE UNE SAISIE NORMALE":
PRINT : PRINT "<CTRL-C> ABREGE UNE SAISIE EN ELIMINANT
      LES VALEURS QUI SUIVAIENT."
380 GOSUB 200: HOME
390 IF N < > 0 THEN FOR I = 1 TO N: VTAB 5 + I: HTAB
      1: PRINT RIGHT$(("< " + STR$(I)),2) + ". "V$(I,
      0);: HTAB 25 - LEN (V$(I,1)): PRINT V$(I,1): NEXT
      :SV = 1: GOSUB 590: GOSUB 630:SV = 0: GOTO 660
400 VTAB 23:T$ = " PAS PLUS DE 12 DONNEES S.V.P ": GOSUB
      220
410 VTAB 21: PRINT TR$: VTAB 4: POKE 35,20
420 NM = N:N = 0
430 N = N + 1
440 VTAB 4: HTAB 1: CALL - 868: IF N = 1 THEN PRINT
      "EX. JANVIER      12000"
450 GOSUB 780
460 HTAB 5: INPUT " ";V$: IF V$ = "" AND V$(N,0) < > "
      " THEN 510
470 IF V$ = "?" THEN 800
480 IF V$ = "/" AND N > 1 THEN GOSUB 780:N = N - 1: GOTO
      510
490 IF V$ = "" THEN 560
500 V$(N,0) = V$
510 GOSUB 780
520 HTAB 25 - LEN (V$(N,1)): IF V$(N,1) = "" THEN HTAB
      18
530 INPUT " ";V$: IF V$ = "" AND V$(N,1) < > "" THEN 580
540 IF V$ = "/" THEN 450
550 IF V$ = "?" THEN 800
560 IF V$ = "" THEN VTAB 5 + N: CALL - 958:N = N - 1:
      GOTO 630
570 V$(N,1) = V$
580 GOSUB 780
590 T = 0: FOR K = 1 TO N:T = T + VAL (V$(K,1)): NEXT
600 FOR I = 1 TO N:P = INT (( VAL (V$(I,1)) * 10000)
      / T):P$ = STR$(P): IF LEN (P$) < 3 THEN P$ = RIGHT$(
      ("000" + P$),3)
610 P$ = LEFT$(P$, LEN (P$) - 2) + "." + RIGHT$(P$,
      2) + "%": HTAB 33: VTAB 5 + I: PRINT "      ";: HTAB
      40 - LEN (P$): PRINT P$: NEXT : IF SV = 1 THEN RETURN
620 IF N < 12 THEN 430
630 VTAB 19: PRINT "TITRE ";TI$;: IF SV = 1 THEN PRINT
      T : RETURN
640 HTAB 7: INPUT " ";R$: IF R$ < > "" THEN TI$ = R$
650 VTAB 19: HTAB 7: PRINT TI$
660 POKE 35,24:T$ = " <1> SAISIE TERMINEE <2> CORRECTI
      ONS "
670 VTAB 23: GOSUB 220
```

(SUITE PAGE 6)

# GRAPHE

## BASIC 3

```
680 GOSUB 200
690 IF R = 1 THEN NM = N: GOTO 720
700 IF R < > 2 THEN 680
710 GOTO 400
720 PP = VAL (V$(1,1)):PG = 0: FOR I = 1 TO N: IF VAL
  (V$(I,1)) > PG THEN PG = VAL (V$(I,1))
730 IF VAL (V$(I,1)) < PP THEN PP = VAL (V$(I,1))
740 NEXT
750 FOURCHETTE = PG + 1 - PP:F0 = F0 / 130
760 FOR I = 1 TO N:H(I) = ( VAL (V$(I,1)) - PP) / F0:H
  (I) = 149 - H(I): NEXT
770 GOTO 110
780 VTAB 5 + N: HTAB 1: CALL - 868: IF N < 10 THEN H
  TAB 2
790 PRINT N". ";V$(N,0); TAB( 25 - LEN (V$(N,1)));V$(
  N,1);: RETURN
800 IF NM > N THEN N = NM
810 GOTO 720
820 :
830 REM *****
840 REM ** 2. TRACE DU GRAPHIQUE **
850 REM *****
860 :
870 IF N = 0 THEN 110
880 X = 0: IF G < > 1 THEN 910
890 HTAB 5: VTAB 22: PRINT "EFFACER GRAPH ACTUEL (0/N)
  ";: GET R$: POKE - 16368,0: IF R$ = "0" THEN 910
900 GOSUB 240: GOTO 1070
910 HGR2
920 DI = 1: IF PG > 9999 THEN DRAW 24 AT 2,3: DRAW 20
  AT 9,3: DRAW 23 AT 14,3:DI = 1000: GOTO 940
930 IF PG > 999 THEN DRAW 14 AT 2,3: DRAW 16 AT 9,3:
  DRAW 25 AT 16,3: DRAW 31 AT 23,3:DI = 100
940 E = (PG - PP) / 13:V = PG
950 HPLLOT 35,0 TO 274,0 TO 274,149 TO 35,149 TO 35,0
960 FOR I = 18 TO 148 STEP 10: HPLLOT 32,I TO 35,I: FOR
  J = 37 TO 272 STEP 3: HPLLOT J,I: NEXT
970 IF I = 148 THEN V = PP
980 V$ = STR$ ( INT (V / DI)): IF V$ = "0" THEN V$ = V
  $ + MID$ (( STR$ (V / DI)),2,2)
990 FOR K = 1 TO LEN (V$): IF K = 1 AND ASC (V$) = 4
  8 THEN DRAW 43 AT 5,I - 3: GOTO 1010
1000 DRAW ASC ( MID$ (V$,K,1)) - 46 AT (K * 7) - 5,I -
  3: NEXT :V = PG - (E * INT (I / 10))
1010 NEXT
1020 D = N - 1: FOR I = 35 TO 274 STEP (239 / D):X = X +
  1: FOR J = 148 TO H(X) STEP - 2: HPLLOT I,J: NEXT
1030 DRAW 42 AT I - 3,H(X) - 4: GOSUB 1090
1040 IF X > 1 THEN HPLLOT M,H(X - 1) TO I,J
1050 M = I: NEXT : IF X < N THEN X = N: HPLLOT M,H(X - 1)
```

Si ?, on récupère la valeur primitive de N dans NM.

← Quelle est la plus grande valeur... et quelle est la plus petite ? Réponse ici.

← Détermination des ordonnées (mais pas désordonnée pour autant !).

← Affichage des données

← S'il n'y a pas de données, retour au menu et si G est différent de 1, inutile de poser la question : on va directement à HGR2... pour tracer.

← DI est une variable qui divise par 1, par 100 ou par 1000, suivant les besoins.

← Ça, c'est le cadre, puis viennent les lignes en pointillé (horizontalement).

← Ecriture des nombres (si le premier chiffre est un 0, on le remplace par un point).

Puis vient la suite du graphe... tout un programme !

# GRAPHE

## BASIC 4

Écriture de la nature  
des données chacune  
d'elles étant limitée à  
ses trois premiers caractères.

### ÉCRITURE du titre

On ne dispose que d'une  
fonte réduite, composée  
des chiffres, des lettres  
capitales, de l'apostrophe,  
du point et des  
parenthèses.

Mémorisation intéressante  
si vous n'imprimez pas  
immédiatement votre GRAPHE  
(il peut être inséré dans un  
texte réalisé par GRI-BOUILLE,  
par exemple).

Mémorisez vos données :  
cela vous évitera de les  
retaper... et elles occupent  
peu de place sur la disquette.

```
TO I,H(X): DRAW 42 AT I - 3,H(X) - 4: GOSUB 1090
1060 GOTO 1160
1070 PRINT : CALL - 198: GOSUB 200:G = 1: IF PEEK ( -
    16384) = 9 THEN 1610
1080 GOTO 110
1090 ROT= 48:V$ = LEFT$ (V$(X,0),3): FOR K = LEN (V$)
    TO 1 STEP - 1: DRAW ( ASC ( MID$ (V$,K,1)) - 53)
    AT I - 2,178 - (K * 7): NEXT K: ROT= 0: RETURN
1100 :
1110 REM *****
1120 REM ** GRAPH.CAR (A$6000,L762) **
1130 REM *****
1140 :
1150 PRINT CHR$ (4)"BLOAD GRAPH.CAR": POKE 232,0: POKE
    233,96: RETURN
1160 X = 2:Y = 7: IF TI$ = "" THEN 1070
1170 L = LEN (TI$):X = 37 + ((249 - L * 7) / 2):Y = 7
1180 FOR I = 1 TO L:R = ASC ( MID$ (TI$,I,1)): GOSUB 1
    200: NEXT
1190 GOTO 1070
1200 IF R > 64 AND R < 91 THEN R = R - 53: GOTO 1250
1210 IF R > 47 AND R < 58 THEN R = R - 46: GOTO 1250
1220 IF R = 39 THEN R = 1: GOTO 1250
1230 IF R = 40 OR R = 41 THEN R = R - 2: GOTO 1250
1240 GOTO 1260
1250 DRAW R AT X - 7,Y
1260 X = X + 7: RETURN
1270 :
1280 REM *****
1290 REM ** 3. MEMORISATION DU GRAPH **
1300 REM *****
1310 :
1320 IF G < > 1 THEN 110
1330 FB$ = "GR0": GOSUB 1500
1340 VTAB 21: PRINT : PRINT CHR$ (4)"BSAVE"F$,A16384,
    L8184"
1350 GOTO 110
1360 :
1370 REM *****
1380 REM ** 4. MEMO FIC DE DONNEES **
1390 REM *****
1400 :
1410 IF N = 0 THEN 110
1420 FB$ = "GRD0": GOSUB 1500
1430 VTAB 21: PRINT : PRINT CHR$ (4)"OPEN"F$: PRINT C
    HR$ (4)"WRITE"F$: PRINT N: FOR I = 1 TO N: PRINT V$
    (I,0): PRINT V$(I,1): NEXT : PRINT TI$: PRINT CHR$
    (4)"CLOSE"F$
1440 GOTO 1350
1450 :
```

(SUITE PAGE 8)

# GRAPHE

## BASIC 5

```
1460 REM *****
1470 REM ** SAISIE-CONTROLE TITRE FIC *
1480 REM *****
1490 :
1500 PRINT : CALL - 198: VTAB 22: HTAB 1: CALL - 868:
      PRINT "TITRE: "FB$;: HTAB 8: INPUT " ";F$: IF F$ =
      "" THEN POP : GOTO 110
1510 IF F$ = "?" THEN HOME : GOSUB 230: GOTO 1500
1520 IF ASC (F$) < 65 OR ASC (F$) > 90 THEN 1500
1530 VTAB 21: PRINT CHR$ (4)"UNLOCK"F$: VTAB 22: CALL
      - 958: CALL - 198: PRINT "CE FICHER VA ETRE ECRA
      SE (O/N) ";: GET R$: POKE - 16378,0: IF R$ = "N"
      THEN 1500
1540 IF R$ < > "0" THEN 1530
1550 VTAB 21: PRINT : PRINT CHR$ (4)"DELETE"F$: RETURN

1560 :
1570 REM *****
1580 REM ** 5. IMPRESSION DU GRAPH **
1590 REM *****
1600 :
1610 IF G < > 1 THEN 110
1620 GOSUB 240
1630 PRINT : PRINT CHR$ (4)"PR£1": PRINT CHR$ (9)"GE2
      ": PRINT : PRINT CHR$ (4)"PR£0": GOTO 110

1640 :
1650 REM *****
1660 REM ** TRAITEMENT DES ERREURS **
1670 REM *****
1680 :
1690 CALL 948:ER = PEEK (222)
1700 IF ER = 255 THEN V$ = "": GOTO 560
1710 IF ER = 6 THEN RETURN
1720 IF ER = 9 THEN VTAB 21: CALL - 958: INVERSE : HT
      AB 12: PRINT "DISQUETTE PLEINE": NORMAL : FOR I = 1
      TO 2000: NEXT
1730 GOTO 110
1740 TEXT : HOME : END
1750 POKE 51,0: GOTO 110
```

En cas de fausse manoeuvre,  
GOTO 1750 renvoie au menu.

← Cette partie du programme utilise le traitement des erreurs.

← A revoir suivant la marque de votre imprimante... et l'interface que vous utilisez.

← Une routine classique de traitement des erreurs est installée à la fin du programme GRAPH.MEN, à partir de l'adresse 948.

← Sous ProDOS, transformez ainsi la ligne 1740 :  
TEXTE : HOME : PRINT CHR\$(4) "OPEN" : END

Le programme **GRAPHE** se compose de :

- GRAPH (Basic, pages 4/8)
- GRAPH.MEN (langage machine, page 9)
- GRAPH.CAR (fonte, page 10)

Ces titres sont ceux qui figurent sur la disquette de *Tremplin Micro* n°4

# GRAPH.MEN

Si vous manquez d'expérience, contentez-vous de taper la version présentée page 56.

0300-	20 58 FC	JSR	\$FC58
0303-	A2 2C	LDX	£\$2C
0305-	A9 06	LDA	£\$06
0307-	85 25	STA	\$25
0309-	A9 0A	LDA	£\$0A
030B-	85 24	STA	\$24
030D-	20 22 FC	JSR	\$FC22
0310-	BD 00 03	LDA	\$0300,X
0313-	C9 00	CMP	£\$00
0315-	F0 06	BEQ	\$031D
0317-	20 F0 FD	JSR	\$FDF0
031A-	E8	INX	
031B-	D0 F3	BNE	\$0310
031D-	E0 B2	CPX	£\$B2
031F-	F0 07	BEQ	\$0328
0321-	E6 25	INC	\$25
0323-	E6 25	INC	\$25
0325-	E8	INX	
0326-	D0 E1	BNE	\$0309
0328-	20 E4 FB	JSR	\$FBE4
032B-	60	RTS	

\$FC58 vide l'écran (HOME)

Le registre X est chargé avec \$2C (44 octets)

Autrement dit : POKE 37,6 (position verticale du curseur, \$25) et POKE 36,10 (position horizontale, \$24)... \$FC22 qui envoie le curseur à l'emplacement ainsi préparé.

On lit le caractère qui est à l'adresse \$300 + X

Est-ce un 0 ?

Si oui a-t-on terminé (\$31D) ?

\$FDF0 affiche le caractère

X est incrémenté de 1

Si ce n'était pas un zéro, encore un tour !

Est-ce la fin ?

Si oui, direction \$328

+ 1 verticalement

+ 1 autre

Et plus 1 pour X aussi

Et encore une petite tournée !

\$FBE4 vous réveille avec un joli petit bip...

... puis retour au Basic

032C-	A0 A0 A0 A0
0330-	A0 20 0D 20 05 20 0E 20
0338-	15 20 00 31 A0 D3 C1 C9
0340-	D3 C9 C5 A0 C4 C5 D3 A0
0348-	C4 CF CE CE C5 C5 D3 00
0350-	32 A0 C1 C6 C6 C9 C3 C8
0358-	C1 C7 C5 A0 C4 D5 A0 C7
0360-	D2 C1 D0 C8 00 33 A0 CD
0368-	C5 CD CF D2 C9 D3 C1 D4
0370-	C9 CF CE A0 C7 D2 C1 D0
0378-	C8 00 34 A0 CD C5 CD CF
0380-	A0 C4 C5 D3 A0 C4 CF CE
0388-	CE C5 C5 D3 00 35 A0 C5
0390-	C4 C9 D4 C9 CF CE A0 C4
0398-	D5 A0 C7 D2 C1 D0 C8 00
03A0-	36 A0 C6 C9 CE A0 C4 C5
03A8-	A0 D0 D2 CF C7 D2 C1 CD
03B0-	CD C5 00 00

M E N  
U 1 SAI  
SIE DES  
DONNEES  
2 AFFICH  
AGE DU G  
RAPH 3 M  
EMORISAT  
ION GRAP  
H 4 MEMO  
DES DON  
NEES 5 E  
DITION D  
U GRAPH  
6 FIN DE  
PROGRAM  
ME

De \$32C à \$3B3, sont mémorisées les six lignes du menu (les deux derniers octets sont inutilisés).

Les \$A0 sont des espaces... de même que les \$20 (mais ces derniers sont affichés en mode inverse).

Les chiffres sont également en mode inverse.

03B4-	68	PLA	
03B5-	A8	TAY	
03B6-	68	PLA	
03B7-	A6 DF	LDX	\$DF
03B9-	9A	TXS	
03BA-	48	PHA	
03BB-	98	TYA	
03BC-	48	PHA	
03BD-	60	RTS	

Sous-programme de récupération des erreurs, évitant de perdre le fil des opérations (GOTO et GOSUB).

Lorsque vous aurez tout tapé, après le CALL-151 de rigueur, faites un :

**BSAVE GRAPH.MEN,  
A\$300, L\$BE.**

# GRAPH.CAR

## SAISIE

CALL-151, puis 6000 : 2B 01 69 00, etc. Ensuite, taper CTRL-C,  
puis BSAVE GRAPH.CAR, A\$6000, L\$2C8

\*6000.62C7

6000- 2B 01 69 00 6E 00 80 00  
6008- 89 00 97 00 A4 00 B1 00  
6010- C0 00 D1 00 DC 00 ED 00  
6018- FE 00 10 01 21 01 2F 01  
6020- 3E 01 4D 01 5B 01 69 01  
6028- 7B 01 85 01 90 01 9F 01  
6030- A7 01 B9 01 CB 01 DC 01  
6038- EA 01 FC 01 0D 02 1D 02  
6040- 27 02 38 02 48 02 5A 02  
6048- 6A 02 76 02 82 02 8B 02  
6050- 94 02 AC 02 BC 02 62 00  
6058- 00 00 00 00 00 00 00 00  
6060- 00 92 92 8A F6 00 00 00  
6068- 2D 36 96 52 01 00 29 AD  
6070- DF 33 4D 35 1F 1F 2E 4D  
6078- FE 1B 0E 2D 4E 01 00 17  
6080- 31 77 36 BE 2D 4E 00 00  
6088- 69 29 AD DF 73 49 1E BF  
6090- 1E 2E 2D 75 01 00 4E 2D  
6098- 2D BE 3E 4E FE 1B 0E 2D  
60A0- 4E 01 00 71 49 3E 17 0D  
60A8- FE 33 2D 2D 1E 76 09 00  
60B0- 4E 2D 2D DE 1B 2E 2D 15  
60B8- 36 DF 73 2D 4E 01 00 2D  
60C0- 09 2D DE BB 4D 3A 3F 6E  
60C8- 09 FE 1B 0E 2D 4E 01 00  
60D0- 15 2D 2D BE 1E 17 36 4E  
60D8- 49 00 00 15 29 AD DF 33  
60E0- 4D F1 3F 17 4D 31 DF 73  
60E8- 2D 4E 01 00 17 29 AD DF  
60F0- 33 4D 31 3F 77 09 1E DF  
60F8- 2A 75 49 00 00 FF 09 15  
6100- 1F 17 4D 31 DF 33 2D 2D  
6108- FE 1B 6E 09 4E 00 00 37  
6110- 2D AD DF 33 4D F1 3F 37  
6118- 4D 31 DF 33 2D 75 09 00  
6120- 92 29 AD DF 33 36 4D 11  
6128- DF 73 2D 4E 01 00 00 2D  
6130- AD DF 6E 31 DF 6E 31 DF  
6138- 17 2D 75 09 00 09 2D 2D  
6140- DE 1B 6E 11 3F 37 36 2D  
6148- 2D 4E 00 00 33 2D 2D DE  
6150- 1B 6E 11 3F 37 36 4E 49  
6158- 01 00 2D 29 AD DF 33 36

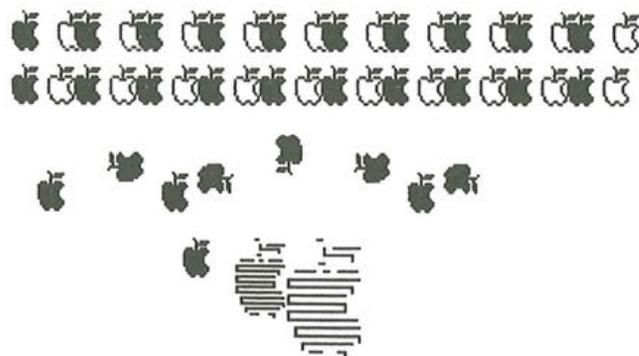
6160- 4D 35 DF 73 2D 75 01 00  
6168- 49 4D 31 DF 33 4D 31 3F  
6170- 3F 6E 09 FE 1B 6E 09 4E  
6178- 00 00 3E 2D 1E 36 36 17  
6180- 2D 4E 00 00 4E 49 31 36  
6188- 36 DF 73 2D 4E 01 00 09  
6190- 4D F1 DF 6E 1E 37 0D 15  
6198- DF 6E 09 4E 00 00 11 36  
61A0- 36 36 2D 75 01 00 6E 4D  
61A8- 31 FF 37 0D 0D FE 1F 6E  
61B0- 09 FE 1B 6E 09 4E 00 00  
61B8- 3F 4D 31 DF 33 6D 31 1F  
61C0- 1F 6E 29 FE 1B 6E 09 4E  
61C8- 00 00 29 29 AD DF 33 4D  
61D0- 31 DF 33 4D 31 DF 73 2D  
61D8- 4E 01 00 52 2D AD DF 33  
61E0- 4D F1 3F 37 36 4E 49 01  
61E8- 00 49 29 AD DF 33 4D 31  
61F0- DF 33 0D 0D 1E DF 0E 6D  
61F8- 4E 00 00 2D 2D AD DF 33  
6200- 4D F1 3F 37 0D 15 DF 6E  
6208- 09 4E 00 00 2D 29 AD DF  
6210- 33 4D 3A 77 09 FE 1B 0E  
6218- 2D 4E 01 00 1E 2D 2D DE  
6220- 36 36 76 49 00 00 00 4D  
6228- 31 DF 33 4D 31 DF 33 4D  
6230- 31 DF 73 2D 4E 01 00 0D  
6238- 4D 31 DF 33 4D 31 DF 33  
6240- 4D F1 1F 0E 4E 09 00 00  
6248- 4D 31 DF 33 4D 31 1F 1F  
6250- 6E 0D 3E 1F 37 4D 71 01  
6258- 00 09 4D 31 DF 73 0D 1E  
6260- 17 0D 15 DF 33 4D 71 01  
6268- 00 00 4D 31 DF 73 0D 1E  
6270- 36 76 49 00 00 01 2D 2D  
6278- BE 1E 17 1E 2E 2D 75 01  
6280- 00 75 09 1E 36 36 0E 4E  
6288- 00 00 36 09 15 36 36 17  
6290- 4E 01 00 36 2D 2D 35 3F  
6298- 3F 37 2D 2D 35 3F 3F 37  
62A0- 2D 2D 35 3F 3F 37 2D 2D  
62A8- 75 01 00 09 49 11 3F 37  
62B0- 2D 35 3F 37 2D 35 3F B7  
62B8- 49 09 00 00 49 3A 17 2D  
62C0- 35 3F 37 2D F5 B7 00 00

### VÉRIFICATION (en mode direct)

ST=0:FORI=24576T024576+711:T=T+PEEK(I):NEXT:?:?"TOTAL-CONTROLE=" T

LE RÉSULTAT DOIT ÊTRE : TOTAL-CONTROLE= 49488

# Les fruits de la passion



Peut-être aurez-vous besoin, dans l'un de vos programmes ludiques, de figurer les touches *Pomme ouverte* et *Pomme fermée* de votre Apple... les voici !

Sur un Apple IIc, ces deux illustres pommes figurent en tête des caractères "souris". Sur ce sujet, revoir la page 34 du n°2 de *Tremplin Micro*.

```

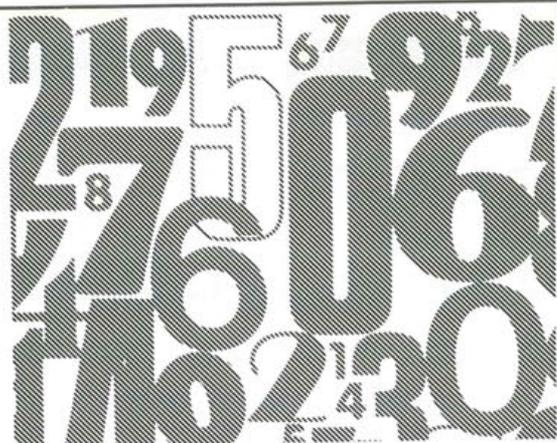
10 REM 138 OCTETS POUR DEUX POMMES
20 SCALE= 1: ROT= 0: HCOLOR= 3: HGR
30 TEXT : HOME : T$ = "139 OCTETS POUR DEUX POMMES": PRINT T$: FOR I = 1 TO
  LEN (T$): PRINT "-";: NEXT : PRINT : PRINT
40 INPUT "ADRESSE DECIMALE DE LA TABLE ";AD
50 GOSUB 180: HGR
60 X = 3: FOR I = 1 TO 10: DRAW 1 AT X,30:X = X + 16: DRAW 2 AT X,30:X = X
  + 10: NEXT
70 X = 3: FOR I = 1 TO 10: DRAW 1 AT X,50:X = X + 12: DRAW 2 AT X,50:X = X
  + 14: NEXT
80 REM ROTATION 16 A 64 (90, 180, 270, 360 DEGRES)
90 R = 48:X = 14: FOR I = 1 TO 8: DRAW 1 AT X,90: ROT= R:X = X + 26:R = R +
  16: IF R = 80 THEN R = 16
100 NEXT : ROT= 0
110 REM SCALE
120 X = 25: FOR I = 1 TO 3: SCALE= I: DRAW 1 AT 50 + X * I,115: NEXT : SCALE
  = 1
130 GOSUB 170
140 TEXT : HOME : LIST
150 PRINT : FOR I = 1 TO 10: PRINT "____";: NEXT : PRINT : PRINT "<1> MENU
  DE DISQUETTE <2> NON MERCI ": GOSUB 170: IF PEEK ( - 16384) = 49 THEN
  PRINT : PRINT CHR$ (4)"RUN MENU"
160 TEXT : HOME : END
170 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: POKE - 16368,0: R
  ETURN
180 POKE 232,AD - INT (AD / 256) * 256: POKE 233, INT (AD / 256)
190 FOR I = AD TO AD + 137: READ R: POKE I,R: NEXT : RETURN
200 REM PARAMETRES (2 FORMES, PREMIERE AU 8e OCTET, SECONDE AU 74e)
210 DATA 2,0,8,0,74,0,0,0
220 REM POMME FERMEE
230 DATA 105,73,17,63,255,46,45,245,219,223,19,45,13,13,45,21,63,63,63,63,6
  3,46,45,45,45,45,30,63,63,63,63,46,45,45,45,53,63,63,63,63,46,45,45,45,4
  5,21,63,63,63,63,55,45,45,45,45,245,63,63,63,119,109,41,78,9,0
240 REM POMME OUVERTE
250 DATA 73,73,105,73,17,63,255,46,45,245,219,223,19,45,13,13,45,21,223,59,
  255,27,110,73,73,241,223,219,27,110,73,73,254,219,219,110,73,73,169,223,
  219,219,110,73,73,241,223,255,115,109,41,78,9,0,64,192,64,192,64,192,64,
  8,0,25,0
  
```

• Agrandir une telle forme n'est pas très heureux !

• 2 indique le nombre de formes, 8 la position du premier octet de la première, 74 celle du début de la seconde.

# I Chiffres HGR

Ce court programme de démonstration vous montre comment utiliser les chiffres d'une (courte) table de formes. POKE 232,0 et POKE 233,96\* (\$E8 et \$E9) fournissent à votre Apple l'adresse du premier octet de cette fameuse table.



```

1000 TEXT : NORMAL : PRINT CHR$(12); CHR$(21): HOME
1010 GOSUB 1160
1020 GOSUB 1240: HOME
1030 HGR : HCOLOR= 3: SCALE= 1: ROT= 1:X = 2:Y = 7
1040 VTAB 22: HTAB 20: INVERSE : PRINT "CTRL-C=FIN": NORMAL
1050 VTAB 22: HTAB 1: GET R$:R = ASC (R$): IF R = 3 THEN HOME : TEXT : END

1060 IF R = 11 AND Y > 7 THEN Y = Y - 7: GOTO 1050
1070 IF R = 10 AND Y < 140 THEN Y = Y + 7: GOTO 1050
1080 IF R = 8 AND X > 9 THEN X = X - 7: GOTO 1050
1090 IF R = 21 AND X < 282 THEN X = X + 7: GOTO 1050
1100 IF R = 4 THEN D = R: GOTO 1050
1110 POKE - 16368,0:X = X + 7: IF X = 289 THEN X = 9:Y = Y + 7
1120 GOSUB 1130: GOTO 1050
1130 IF D = 4 THEN D = 0: XDRAW VAL (R$) + 1 AT X - 7,Y: RETURN
1140 DRAW VAL (R$) + 1 AT X - 7,Y: RETURN
1150 HTAB (41 - LEN (T$)) / 2: PRINT T$: RETURN
1160 T$ = "PROGRAMME DEMO D'ECRIURE": VTAB 6: GOSUB 1150
1170 PRINT :T$ = "EN PAGE GRAPHIQUE HGR OU HGR2": GOSUB 1150
1180 PRINT :T$ = "AVEC DES CHIFFRES EN DATA": GOSUB 1150
1190 PRINT :T$ = "SE DEPLACER AVEC LES FLECHES APPLE": GOSUB 1150
1200 T$ = "POUR EFFACER UN CHIFFRE:REVENIR": GOSUB 1150
1210 T$ = "SUR LE CARACTERE, TAPER CTRL-D": GOSUB 1150
1220 T$ = "PUIS LE CHIFFRE INCRIMINE": GOSUB 1150
1230 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: POKE - 16368,0: R
      ETURN
1240 POKE 232,0: POKE 233,96: FOR I = 24576 TO 24705: READ J: POKE I,J: NEXT
      : RETURN
1250 DATA 10,0,22,0,36,0,44,0,54,0,66,0,76,0,88,0,99,0,108,0,119,0
1260 DATA 12,37,28,63,23,54,46,30,14,45,44,36,4,0,36,188,150,18,45,28,36,0
1270 DATA 101,228,63,23,150,241,46,45,37,0,12,12,60,63,183,146,21,45,12,228,
      7,0
1280 DATA 58,39,12,12,12,54,174,55,62,0,56,39,44,45,245,170,54,23,63,28,4,0
1290 DATA 117,246,63,28,36,229,12,12,45,6,0,12,12,60,63,183,82,30,46,0
1300 DATA 231,100,45,21,246,14,246,63,28,36,0,231,100,45,21,54,119,30,30,63,
      4,0
  
```

\* **POQUER UNE ADRESSE HEXA.**— Exemple \$03/00 = POKE X,0 : POKE X+1,3 ou encore \$60/00 = POKE X,0 : POKE X+1,96 (\$60 en HEXA = 96 en numération décimale).

Si la question vous intéresse, tapez ce court programme de démonstration. Il vous aidera à bien comprendre comment lire (et même modifier) une fonte de caractères telle que GRAPH.CAR (p.10).

# Plus loin avec la fonte GRAPH.CAR

```

10 PRINT CHR$(12); CHR$(21)
20 AD = 16385 + 16 + 44: REM PROGRAMME + NOMBRE DE CARACTERES
30 GOSUB 190: GOTO 50
40 H = 30:V = V + 15: FOR X = 1 TO LEN(A$): POKE 25, ASC ( MID$(A$,X,1)
): CALL CA: DRAW PEEK (25) AT (H - 7) + (X * 7),V: NEXT : RETURN
50 HOME : HGR :A$ = "CARACTERES HGR (DEMO)": GOSUB 40
60 A$ = "ABCDEFGHJKLMNOPQRSTUVWXYZ": GOSUB 40
70 A$ = "(0123456789)-?-0": GOSUB 40: REM SIGNES INCONNUS TRANSFORMES EN E
SPACES
80 V = V + 25:Y = 0: FOR X = 38 TO 43:Y = Y + 1: DRAW X AT Y * 7 + 23,V: N
EXT
90 A$ = "(<>)*,": GOSUB 40
100 A$ = "LA FONTE GRAPH.CAR NE COMPORTE PAS": GOSUB 40
110 A$ = "L'ESPACE MAIS ELLE EST FACILEMENT":V = V - 5: GOSUB 40
120 A$ = "MODIFIABLE GRACE A SES VALEURS 00 ":V = V - 5: GOSUB 40
130 GOSUB 170: TEXT : PRINT CHR$(4)"PR£3": PRINT : LIST 10,30: LIST 190,
240
140 PRINT : GOSUB 170: HOME : LIST 30,120: GOSUB 170: HOME
150 VTAB 22: PRINT "<1> MENU DE DISQUETTE <2> FIN DE PROGRAMME": GOSUB 180
: IF PEEK ( - 16384) = 49 THEN VTAB 20: PRINT : PRINT CHR$(4)"RUN M
ENU"
160 HOME : END
170 VTAB 22: HTAB 8: INVERSE : PRINT "PRESSEZ UNE TOUCHE S.V.P": NORMAL
180 CALL - 198: POKE - 16368,0: WAIT - 16384,128: POKE - 16368,0: PRIN
T : RETURN
190 : PRINT : PRINT CHR$(4)"BLOADGRAPH.CAR,A"AD: REM FONTE SANS ESPACE
200 POKE AD,44: POKE AD + 88,95: REM NOMBRE DE FORMES PORTE DE 43 A 44 & A
DRESSE DE L'ESPACE
210 POKE 6,AD - INT (AD / 256) * 256: POKE 7, INT (AD / 256): POKE 232, P
EEK (6): POKE 233, PEEK (7): SCALE= 1: ROT= 0: HCOLOR= 3
220 A$ = "'0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ(<>)*,":CA = AD - LEN (A$)
- 1:Y = CA: FOR X = 1 TO LEN (A$):Y = Y + 1: POKE Y, ASC ( MID$(A$,X
,1)): NEXT : POKE 8,CA - INT (CA / 256) * 256: POKE 9, INT (CA / 256)
230 CA = CA - 16: FOR X = CA TO CA + 15: READ Y: POKE X,Y: NEXT : RETURN
240 DATA 160,44,177,8,197,25,240,5,136,208,247,160,44,132,25,96
    
```

4000- A0 2C  
4002- B1 08  
4004- C5 19  
4006- F0 05  
4008- 88  
4009- D0 F7  
400B- A0 2C  
400D- 84 19  
400F- 60



LDY £\$2C  
LDA (\$08),Y  
CMP \$19  
BEQ \$400D  
DEY  
BNE \$4002  
LDY £\$2C  
STY \$19  
RTS

Initialisation avec nombre de caractères (\$2C=44)  
Lecture indirecte (adresse donnée en \$8 - 9 + Y)  
Comparaison avec le contenu de la mémoire \$19 (25)  
Si égalité (Z=1), c'est bon !  
Sinon, moins 1 pour Y...  
... et retour au début de boucle pour lire  
Si le caractère est inconnu : c'est un espace  
Mémo du numéro de caractère à l'adresse 25  
Retour au Basic.

# Votre bibliothèque INFORMATIQUE

par Clément RENARD

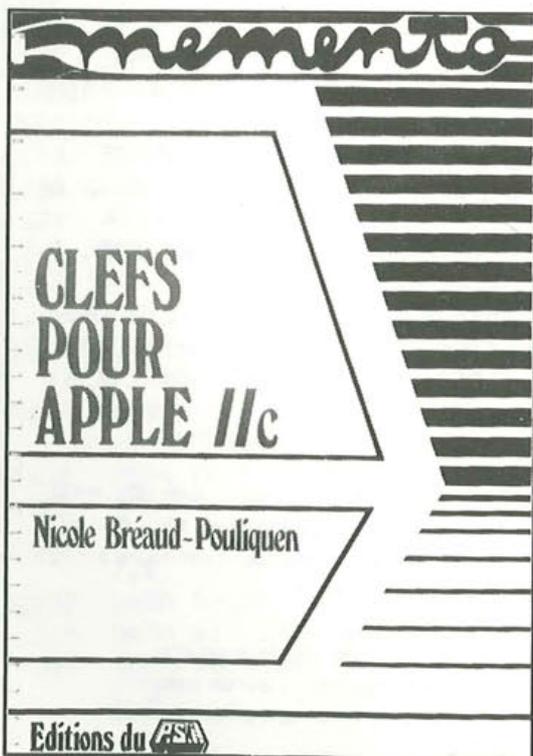
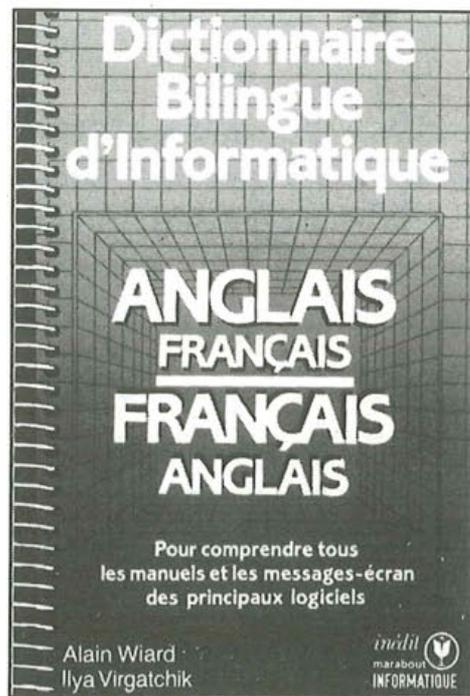
## ● *Il ne s'agit pas d'apprendre l'Anglais, mais bien de le COMPRENDRE*

La Collection MARABOUT INFORMATIQUE nous propose, sous la double signature d'Alain Wiard et d'Ilya Virgatchik, un DICTIONNAIRE BILINGUE D'INFORMATIQUE qui devrait rendre d'éminents services à celles et à ceux qui, taquinant le Basic ou les logiciels du commerce, ont parfois — en Anglais s'entend — la comprenette difficile.

Bon ! c'est mon cas, mais évitons de faire du nombrilisme; et revenons au "poche" de Marabout (de ficelles, au pluriel, mais oui, car cet ouvrage est en effet bourré de trucs fort intéressants).

Je l'aurais considéré comme un honnête aide-mémoire, sans plus, semblable — en plus concis — à beaucoup d'autres, s'il ne comportait pas, en supplément, la traduction des principaux messages d'erreurs et cela pour plusieurs systèmes d'exploitation... dont celui du *business basic apple*.

Une idée généreuse, traitée en quelques pages, mais dont devraient bien s'inspirer les auteurs des prochains dictionnaires du genre...



## ● *De vraies clefs pour Apple IIc*

Nicole Bréaud-Pouliquen est une récidiviste (mais absolument pas dangereuse... sauf pour les bogues, bien entendu). Nos lecteurs connaissent déjà son *Langage machine et assembleur du 6502* (en collaboration avec Daniel-Jean David).

Par ailleurs, on ne compte plus les croqueurs de Pomme à la recherche, dans les *Clefs pour l'Apple II* de l'érudite Nicole (conseil en informatique individuelle, et spécialiste de l'enseignement de la programmation), de quelque pépin égaré.

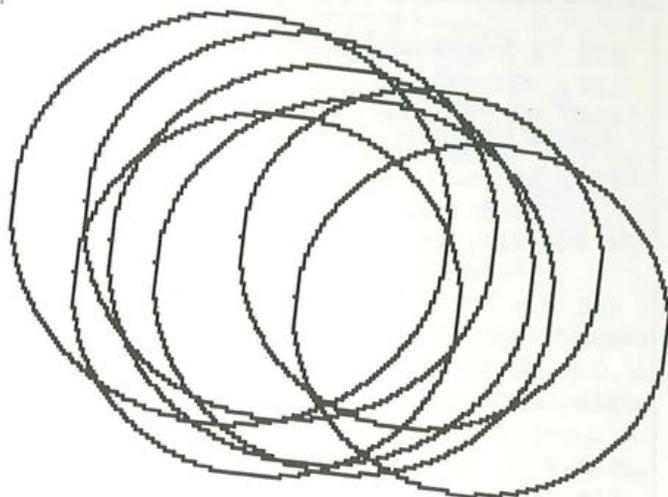
*Clefs pour Apple IIc* reprend les grands thèmes du précédent, mais fournit au néophyte toutes les adresses dont il a (ou aura) besoin. C'est l'ouvrage de référence par excellence... et on peut même se demander pourquoi il n'est pas systématiquement offert à tous les acheteurs d'un Apple IIc (ou d'un nouvel Apple IIe... c).

Je suis persuadé que les Editions du P.S.I. n'y verraient aucun inconvénient !

C.R

# Pour 7 ronds !

JEU graphique essentiellement visuel. Son apparente simplicité ne lui enlève rien : il se révèle capable, à l'usage, de développer l'esprit d'observation et la mémoire. Parfaitement ! vous constaterez que, sur l'écran, les ronds peuvent passer pour des ellipses : c'est ainsi que l'on obtient des ronds ronds sur une imprimante !



La fonction **RND** est bien pratique... quand il s'agit de mêler aléatoirement des valeurs. ➔

Ronds ou ellipses ? Je vous laisse juge ! ➔

**P1** et **P0** sont utilisés pour comparer les dimensions et déterminer quelle est la plus grande ellipse et quelle est la plus petite (lignes 360 et 370). ➔

**CALL-198** émet un bip. ➔

```

100 TEXT : PRINT CHR$(12); CHR$(21): HOME
110 HGR : HCOLOR= 3:ST = .1
120 :
130 REM *****
140 REM **      CALCUL ALEATOIRE      **
150 REM **      DES SEPT RAYONS      **
160 REM *****
170 :
180 FOR I = 1 TO 7:A(I) = 55 + I * 2: NEXT : FOR I =
    7 TO 1 STEP - 1:J = INT ( RND (1) * I + 1):R(I)
    = A(J):A(J) = A(I): NEXT
190 :
200 REM *****
210 REM ** ET LE JEU PEUT COMMENCER **
220 REM *****
230 :
240 VTAB 22: HTAB 1: PRINT "Voici sept ellipses: Dite
    s les NUMEROS de la plus grande et de la plus pet
    ite": VTAB 1
250 P1 = 0:P0 = 80
260 READ YC: READ XC:R = R + 1:RA = R(R):FL = 0
270 POKE 1635,48 + R
280 FOR I = 0 TO 6.3 STEP ST
290 X = RAD * COS (I) + XC
300 Y = (RAD / 4) * SIN (I) / .3 + YC
310 IF X > 279 OR X < 0 OR Y > 191 OR Y < 0 THEN FL =
    0: GOTO 340
320 IF NOT FL THEN HPLOT X,Y:FL = 1
330 HPLOT TO X,Y
340 NEXT
350 :
360 IF R(R) > P1 THEN P1 = R(R):PG = R
370 IF R(R) < P0 THEN P0 = R(R):PP = R
380 CALL - 198
390 IF R < 7 THEN 260
400 GOSUB 660

```

SUITE PAGE 16

```

420 REM *****
430 REM ** LA QUESTION EST POSEE **
440 REM *****
450 :
460 HOME : VTAB 21: CALL - 198: PRINT "RANG DE LA PL
US GRANDE --> ";; GOSUB 560
470 PRINT R$;: IF VAL (R$) = PG THEN GOSUB 590: GOT
O 490
480 P = PG: GOSUB 600
490 PRINT : VTAB 22: HTAB 1: CALL - 198: PRINT "NUME
RO DE LA PLUS PETITE --> ";; GOSUB 560
500 PRINT R$;: IF VAL (R$) = PP THEN GOSUB 590: GOT
O 520
510 P = PP: GOSUB 600
520 GOSUB 660
530 HOME : VTAB 21: PRINT "<1> UN AUTRE TOUR <2> MENU
<3> FIN ? ";; GET R$;: IF R$ = "1" THEN RUN
540 IF R$ = "2" THEN PRINT : PRINT CHR$(4)"RUN MEN
U"
550 TEXT : HOME : END
560 GET R$;: IF ASC (R$) < 49 OR ASC (R$) > 55 THEN
560
570 RETURN
580 :
590 PRINT " ";: INVERSE : PRINT "EXACT": NORMAL : RET
URN
600 PRINT " ";: INVERSE : PRINT "FAUX";: NORMAL : PRI
NT " "P: RETURN
610 :
620 REM *****
630 REM ** CENTRE DES SEPT RONDS **
640 REM *****
650 :
660 FOR I = 1 TO 2000: NEXT : RETURN
670 DATA 70,144,60, 92,80,174,75,113,65,133,80,155,6
0,160

```

← Notez que l'on n'utilise qu'un seul GET pour les lignes 460 et 490... par mesure d'économie. Ici, c'est sans importance, car le logiciel est court, mais parfois il s'agit d'une économie nécessaire.

Naturellement, ce petit jeu est modifiable à volonté, suivant votre inspiration.

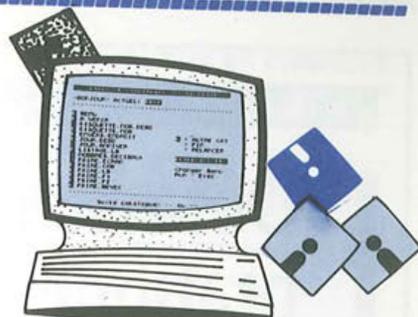
← Centres des sept ellipses.

Comme toutes les revues TRÈS SPÉCIALISÉES, TREMPLIN MICRO ne peut s'offrir le luxe d'une campagne de publicité, même modeste.

## TREMPLIN MICRO

ne compte que sur le soutien effectif de ses Lectrices et de ses Lecteurs. Parlez de lui à vos amis et vous contribuerez, en aidant à sa diffusion, à en améliorer le contenu. **MERCI !**

# POKE... D'ACCORD, mais à quelle adresse ?



**V** OUS est-il déjà arrivé, pendant l'écriture d'un programme en Basic, d'avoir besoin de connaître l'adresse d'un point particulier de votre écran ? Je suis sûr que oui. Il y a évidemment la solution consistant à calculer cette adresse d'après les paramètres de la fiche n°4 de *Tremplin Micro*, mais vous pouvez aussi utiliser la petite routine ci-après. Notez qu'elle est relogeable (autrement dit : vous pouvez la charger à partir de n'importe quelle adresse, puis y avoir accès par un CALL ADRESSE).

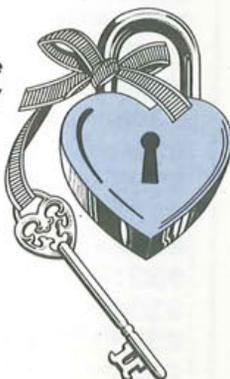
**Attention !** Elle utilise, sans les sauvegarder, les adresses \$6, \$7, \$8 de la page 0.

0300-	20 F5 E6	JSR	\$E6F5
0303-	86 06	STX	\$06
0305-	20 F5 E6	JSR	\$E6F5
0308-	8A	TXA	
0309-	48	PHA	
030A-	4A	LSR	
030B-	29 03	AND	£\$03
030D-	09 04	ORA	£\$04
030F-	85 08	STA	\$08
0311-	68	PLA	
0312-	29 18	AND	£\$18
0314-	90 02	BCC	\$0318
0316-	69 7F	ADC	£\$7F
0318-	85 07	STA	\$07
031A-	0A	ASL	
031B-	0A	ASL	
031C-	05 07	ORA	\$07
031E-	18	CLC	
031F-	65 06	ADC	\$06
0321-	85 07	STA	\$07
0323-	AA	TAX	
0324-	A5 08	LDA	\$08
0326-	48	PHA	
0327-	20 24 ED	JSR	\$ED24
032A-	20 48 F9	JSR	\$F948
032D-	A6 07	LDX	\$07
032F-	68	PLA	
0330-	4C 41 F9	JMP	\$F941

Accepte l'expression venant après la virgule...  
... on trouve le résultat dans X et on le sauve en \$06  
Deuxième expression, suivant le CALL 768  
... également récupérée dans X et passée dans A.

*Cette partie "calcul" est empruntée à la routine du moniteur \$FBC1 (-1087). Celle-ci calcule l'adresse du DÉBUT (ou de BASE, d'où son nom de BASCALC) de la ligne d'écran en cours. On trouve habituellement le résultat en \$28 - \$29, que nous avons remplacées par \$7 et \$8. Pourquoi ne pas appeler directement \$FBC1 ? Tout simplement parce que cette routine modifie l'affichage. Finalement, la solution adoptée ici simplifie les choses.*

Annulation de la retenue  
On ajoute HTAB à l'adresse de base  
Le résultat est sauvegardé dans \$07  
Puis passé dans X  
Lecture de \$08  
Sauvegarde au sommet de la pile  
Affiche la valeur décimale de X,A  
Affiche trois espaces  
Lecture de \$07  
Récupération de 08 sur la pile  
Affiche les registres A, X en hexa



**UTILISATION :** CALL 768, H, V (H=0-39 — V=0-23) en mode direct

**Avec l'Ampersand :** Taper POKE 1014,0 : POKE 1015,3. Il suffira alors de taper &,H,V... pour obtenir le résultat.

**Sauvegarde du programme :** BSAVE ADRECR, A\$300, L\$33

**Chargement :** BLOAD ADRECR

# Désassemblage pratique

Le programme de Clément Renard vous donne l'adresse décimale d'une routine dont vous ne possédez que l'entrée hexa... et désassemble, sur votre écran, autant d'instructions que vous le désirez.

```

10 TEXT : PRINT CHR$(12); CHR$(2
1): PRINT CHR$(4)"BLOAD EXAROUT
.LM": HOME
20 INVERSE : HTAB 4: PRINT " LISTAG
E COMMODE DES ROUTINES LM ": NORM
AL : POKE 34,2
30 VTAB 3: INPUT "ADRESSE -> ";AD$
: IF AD$ = "" THEN 60
40 CALL - 998: HTAB 20: PRINT "DEC
=";
50 CALL 768,AD$
60 VTAB 23: PRINT : VTAB 24: PRINT
"<1> SUITE <2> AUTRE <3> FIN DE T
RAVAIL "; GET R$: POKE - 16368,
0: IF R$ < "1" OR R$ > "3" THEN 6
0
70 PRINT : IF R$ = "1" THEN HOME :
CALL 823: GOTO 60
80 IF R$ = "2" THEN HOME : RUN 30
90 TEXT : HOME : END
    
```

0300-	20 BE DE	JSR	\$DEBE	CHKOPN : cherche la virgule
0303-	20 7B DD	JSR	\$DD7B	FRMEVL : évalue la formule et place le résultat dans FAC
0306-	20 FD E5	JSR	\$E5FD	FRESTR : est-ce une chaîne ?
0309-	A0 FF	LDY	£\$FF	Initialisation (on va avoir \$FF + 1 = \$0 )
030B-	C8	INY		Incréméntation de Y
030C-	B1 5E	LDA	(\$5E), Y	On trouve en 5E-5F l'adresse de la chaîne
030E-	F0 03	BEQ	\$0313	Si 0, direction 313
0310-	18	CLC		Annulation de la retenue
0311-	69 80	ADC	£\$80	Plus \$80 (\$31 = \$B1... \$41 = \$C1, etc.)
0313-	99 00 02	STA	\$0200, Y	Stockage dans le buffer d'entrée
0316-	D0 F3	BNE	\$030B	Encore un tour !
0318-	20 C7 FF	JSR	\$FFC7	ZMODE : 0 mode
031B-	20 A7 FF	JSR	\$FFA7	GETNUM : Lit un nombre sur 2 octets
031E-	A6 3E	LDX	\$3E	Résultat dans A2L...
0320-	A5 3F	LDA	\$3F	... et A2L + 1 (\$3E et \$3F)
0322-	C0 03	CPY	£\$03	
0324-	B0 07	BCS	\$032D	Le nombre est-il sur un ou deux octets ?
0326-	C0 01	CPY	£\$01	S'il est sur 1, il faut transférer X dans A, puis mettre 0 dans X.
0328-	D0 03	BNE	\$032D	S'il est sur deux octets, on saute à \$32D
032A-	8A	TXA		
032B-	A2 00	LDX	£\$00	
032D-	86 3A	STX	\$3A	
032F-	85 3B	STA	\$3B	Préparation de l'affichage des instructions
0331-	20 24 ED	JSR	\$ED24	LINPRT : Affiche les 2 octets de X, A (décimaux)
0334-	20 8E FD	JSR	\$FD8E	CROUT : Envoie le caractère RET
0337-	A9 12	LDA	£\$12	Nombre d'instructions à désassembler.
0339-	4C 63 FE	JMP	\$FE63	LIST2 : Désassemble A instructions.

```

*300:20 BE DE 20 7B DD 20 FD E5 A0 FF C8
B1 5E F0 03 18 69 80 99 00 02 D0 F3 20
C7 FF 20 A7 FF A6 3E A5 3F C0 03 B0 07 C
0 01 D0 03 8A A2 00 86 3A 85 3B 20 24 ED
20 8E FD A9 12 4C 63 FE
    
```

**SAUVEGARDE SUR DISQUETTE :**

BSAVE EXAROUT.LM, A\$300, L\$3C

**VÉRIFICATION (EN MODE DIRECT) :**

T = 0 : FOR I = 768 TO 827 : T = T + PEEK (I) :  
NEXT : PRINT ——— Résultat : 7536.

# CATALOGUE THÉMATIQUE

**I** NUTILE d'essayer ce programme si vous utilisez ProDOS : il ne fonctionne qu'avec le DOS 3.3 d'origine, mais vous donnera, après quelques secondes d'attente, le catalogue complet de votre disquette, dans l'ordre alphabétique, et d'une manière aussi "thématique" que possible (le tout sur 80 colonnes).

Pourquoi les guillemets ? parce que le classement par grands thèmes ne sera effectif que dans la mesure où vous aurez adopté, pour titrer vos programmes, une certaine méthode.

Quand un programme comportera plusieurs sous-programmes (et notamment des routines en langage machine, ou bien des fichiers), vous devrez procéder de la manière suivante (le nom PROGRAMME est ici choisi arbitrairement) :

- Basic = PROGRAMME
- Langage machine = PROGRAMME.LM
- Fichier = PROGRAMME.FIC

l'essentiel étant que tous les éléments commencent par le même préfixe et soient tous plus grands (alphabétiquement parlant) que le programme Basic, moyennant quoi, sur votre écran, cela donnera :



## PROGRAMME

- PROGRAMME.FIC
- PROGRAMME.LM

Si vous désirez supprimer le classement alphabétique, il vous suffira d'éliminer les lignes 70 à 90, puis de réécrire la ligne 70 :

```
70 N = N + 1 : C$(N) = C$(I) : GOTO 50
```

Et puis, dans la foulée, éliminez aussi la ligne 130. Vous disposerez alors d'un catalogue (rapide) sur 80 colonnes, capable de vous présenter (sur un seul écran), plus de 50 titres (51 exactement). Modifiez aussi la ligne 280... puisqu'il n'y aura plus de tri.

```
10 GOSUB 380
20 TEXT : PRINT CHR$(4)"PR&3": PRINT : HOME : DIM C$(107)
30 PRINT "INSERER LA DISQUETTE, ET RETURN": CALL - 198: POKE - 16368,0: W
AIT - 16384,128: POKE - 16368,0: PRINT : GOSUB 280
40 & C$(0):N = 0:I = 0
50 I = I + 1: IF C$(I) = "" THEN 100
60 IF MID$(C$(I),2,1) = CHR$(8) THEN 50
70 N = N + 1: IF C$(I) > C$(N - 1) THEN C$(N) = C$(I): GOTO 50
80 FOR K = N - 1 TO 1 STEP - 1: IF C$(I) < C$(K) THEN C$(K + 1) = C$(K): N
EXT
90 C$(K + 1) = C$(I): GOTO 50
```

(SUITE PAGE 20)

## CATALOGUE THÉMATIQUE (suite et fin)

```
100 HOME : INVERSE : PRINT "CATALOGUE COMPLET DE LA DISQUETTE " ; NORMAL : F
OR I = 1 TO 20 : TI$ = TI$ + "----" : NEXT : PRINT LEFT$ (TI$,46) : POKE 34,
2
110 V = 2 : H = 0 : HOME : GOSUB 270
120 FOR I = 1 TO N
130 V$ = " - " : IF LEFT$ (C$(I),3) < > LEFT$ (C$(I - 1),3) THEN V$ = "" : I
F V < > 2 THEN V = V + 1
140 V = V + 1 : IF V < 20 THEN 180
150 IF H = 0 THEN V = 3 : H = 27 : GOTO 180
160 IF H = 27 THEN V = 3 : H = 54 : GOTO 180
170 GOSUB 210 : HOME : V = 3 : H = 0 : GOSUB 270
180 POKE 1403,H : VTAB V
190 PRINT V$ + LEFT$ (C$(I),20)
200 NEXT : GOTO 300
210 VTAB 22 : POKE 1403,0 : PRINT "ENFONCEZ UNE TOUCHE POUR CONTINUER " ; GET
R$ : POKE - 16368,0 : IF ASC (R$) < > 18 THEN RETURN
220 PRINT : VTAB 22 : CALL - 868 : INPUT "TITRE DU PROGRAMME A LANCER : " ; RP$
: IF RP$ = "" THEN RETURN
230 FOR S = 1 TO N : IF C$(S) = RP$ THEN VTAB 21 : PRINT : GOTO 250
240 NEXT S : POKE 216,0 : VTAB 22 : CALL - 868 : CALL - 198 : INVERSE : PRINT "
CE PROGRAMME N'EXISTE PAS " ; NORMAL : GET R$ : PRINT R$ : GOTO 110
250 ONERR GOTO 370
260 PRINT CHR$ (4) "RUN" RP$
270 VTAB 21 : POKE 1403,0 : PRINT TI$ : RETURN
280 HOME : VTAB 12 : INVERSE : PRINT "PATIENCE SVP : LECTURE ET TRI" : NORMAL
290 : PRINT : PRINT "Si vous désirez lancer un programme par un RUN, tapez un
Ctrl-R, puis le titre choisi (qui doit évidemment correspondre à un pro
gramme écrit en BASIC)" : RETURN
300 VTAB 22 : POKE 1403,0 : CALL - 198 : PRINT " LE CATALOGUE DE DISQUETTE CO
MPTTE " ; INVERSE : PRINT I - 1 ; NORMAL : PRINT " RUBRIQUES -- " ;
310 R$ = "Début Fin Autre --> " : CALL 870,R$ : POKE 1406,40 : GET R$
320 IF ASC (R$) = 18 THEN GOSUB 220 : GOTO 300
330 PRINT R$ : IF R$ = "D" THEN 110
340 IF R$ = "A" THEN RUN 20
350 IF R$ < > "F" THEN 300
360 TEXT : HOME : END
370 :
380 FOR I = 768 TO 905 : READ R : POKE I,R : NEXT : POKE 1014,0 : POKE 1015,3 : R
ETURN
390 DATA 32,227,223,32,247,175,24,32,17,176,176,84,162,0,142,156,179,189,198
,180,240,74,48,65,32,48,178,169,160,202,202,202,221,198,180,240,250,232,1
38,72,56,237,156,179
400 DATA 233,3,32,221,227,168,104,170,136,202,189,198,180,41,127,145,158,152
,208,244,165,157,145,131,200,165,158,145,131,200,165,159,145,131,24,165,1
31,105,3,133,131,144,2,230,132,32,48,178,144,176,176,167,160,0,152,145,13
1,96
410 DATA 32,190,222,32,123,221,32,253,229,133,96,160,0,177,94,201,65,144,6,2
01,90,176,2,144,2,9,128,32,237,253,200,196,96,144,234,96
```

Programme de Bernard GEDENNE

# Votre bibliothèque INFORMATIQUE

## ● *Vers l'ordinateur intelligent ?*

Qu'est-ce que l'Intelligence Artificielle (I.A. pour les initiés) ? Est-il possible de concevoir, sur un ordinateur personnel — et notamment sur un Apple — des programmes relevant de l'I.A. ?

L'ouvrage de Mike James (Eyrolles), traduit par Edith Schomberg, répond non seulement à ces questions, mais à plusieurs autres, non moins intéressantes.

Vous lirez le chapitre I comme un roman, et peut-être serez-vous d'accord avec l'auteur lorsqu'il écrit : "Le cerveau humain est-il simplement un ordinateur très compliqué ou applique-t-il un principe encore inconnu ? Si nous ne sommes rien de plus que des ordinateurs (ce qui est mon opinion personnelle), les limites de l'ordinateur sont alors également les nôtres". J'avoue partager cette opinion, mais être incapable d'expliquer pourquoi... ce qui ne doit pas vous paraître très logique...

Revenons à cette "introduction à l'Intelligence Artificielle" (sur micro-ordinateur) pour souligner que les exemples sont

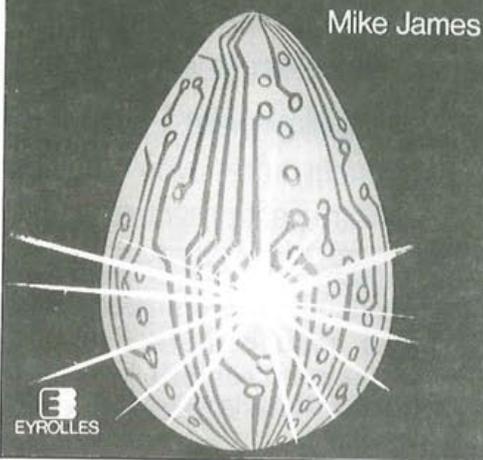
donnés en Basic (MICROSOFT) et seront facilement testés sur Apple, sans modifications importantes. Vous apprendrez ainsi comment écrire un programme de simulation du fonctionnement de la mémoire.

**Conclusion :** une bonne approche d'un sujet qui a déjà fait couler beaucoup d'encre, mais demeure assez mal connu.

par Guy-HACHETTE

## Introduction à l'INTELLIGENCE ARTIFICIELLE sur micro-ordinateur

Mike James



## ● *Guide pratique pour Apple IIe*

Nombreux, parmi les utilisateurs d'Apple IIe sont ceux qui disposent, sur leur machine, du système CP/M.

Jacques Neel et Jean-Claude Hillion ont tiré parti, pour réaliser leur ouvrage, d'une brochure qui a servi de manuel de base à l'Institut d'Etudes Politiques de Paris, dans le cadre des travaux pratiques d'informatique !

Ils ont voulu réaliser un guide pratique aussi clair que possible et il est certain que, à travers leurs explications et exemples, la gestion de fichiers personnels paraîtra plus facile à l'amateur.

Le matériel utilisé par les auteurs est un Apple IIe, mais tout ordinateur personnel travaillant sous CP/M donnera les mêmes résultats.

Plusieurs annexes : mémento des procédures usuelles, procédures à respecter en cas d'arrêt, tableau synoptique des commandes MS-DOS, tableau synoptique des commandes PROLOGUE, etc.

# FONTE SPÉCIALE

Est-elle inédite ? Oui

Peut-on la copier ? Oui

Comment l'utiliser ?

Facilement, après avoir compris et essayé  
notre petit programme DEMO.

**D**ANS notre mini-programme, il n'est pas tenu compte des différences de chasse (encombrement latéral de la lettre) existant entre les caractères minuscules et majuscules... ou encore entre les lettres dites normales et les caractères étroits ("!" ou "i", par exemple).

L'utilisation — avec espacement proportionnel — de FONTE.SPECIALE exige une petite routine facile à programmer, mais comportant un certain nombre d'IF.

Si l'application que vous en tirerez vous paraît très intéressante, n'hésitez pas à nous la soumettre. Nous la publierons pour en faire profiter les lecteurs de *Tremplin Micro*.

Ne manquez pas de relire les programmes des pages 12 et 13 : ils traitent le même problème, mais d'une manière un peu différente.

Notez que votre FONTE.SPECIALE pourra être chargée à une autre adresse que celle choisie ici. Un simple *BLOAD FONTE.SPECIALE, A adresse* permet de le faire, mais n'oubliez pas de poquer les nouvelles valeurs aux adresses 232 et 233, en vous inspirant des conseils donnés par ailleurs (notamment page 12).

Sur votre clavier (mode AZERTY), les caractères de la FONTE.SPECIALE correspondent exactement aux caractères ci-après de l'Apple :

!"£\$%&'()\*+,-./0123456789:;<=>?àABCDEFGHIJKLMNO  
PQRSTUVWXYZ"ç§^\_`abcdefghijklmnopqrstuvwxyzéùè"



# FONTE DÉMO

Ne pas oublier de protéger votre fonte par LOMEM (ici : adresse du premier octet + longueur).

Ces valeurs concernent l'adresse \$6000 (\$00/60 = 00/96 en mode décimal).

POKE 49233,0 = mode TEXT.

L'écran graphique haute résolution mixte se compose de 280 points horizontaux et de 160 points verticaux (lignes 1 à 20 de l'écran texte).

Si barre d'espacement H = H + 7

POKE 49232,0 permet de retrouver la page graphique - mais sans effacement, au contraire de HGR.

```
10 TEXT : NORMAL : PRINT CHR$(12); CHR$(21): HO
ME
20 PRINT "UTILISATION DE LA FONTE SPECIALE DE TM 2
";
30 FOR I = 1 TO 10:T$ = T$ + "----": NEXT : PRINT
T$: POKE 34,3
40 LIST 50,100
50 AD = 24576:L = 4498: REM AD=ADRESSE FONTE - L=N
OMBRE D'OCTETS
60 LOMEM: AD + L
70 PRINT CHR$(4)"BLOAD FONTE.SPECIALE,A$6000"
80 POKE 232,0: POKE 233,96
90 HGR : HCOLOR= 3: SCALE= 1: ROT= 0
100 F$ = "!" + CHR$(34) + "£$%&'()*+,-./0123456789
;<=>?àABCDEFGHIJKLMN O PQRSTU VWXYZ"ç$^`_`abcdefghijklmnopqrstuvwxyzè"
110 POKE 49233,0
120 GOSUB 320
130 HOME : PRINT "VOICI LES CARACTERES A TAPER,
ANS LE MEME ORDRE QUE CEUX DE LA FONTE"
140 PRINT : PRINT F$: PRINT
150 PRINT : PRINT : PRINT "Pomme ouverte = &": PRIN
T "Pomme fermée = "": PRINT
160 PRINT "CTRL-F vous pour revoir cette page": PRI
NT "CTRL-C pour terminer"
170 GOSUB 320
180 H = - 15:V = 15
190 VTAB 21: CALL - 958: PRINT : VTAB 22
200 CALL - 198: PRINT "CATACTERE DESIRE ? -> "": G
ET A$
210 IF A$ = CHR$(6) THEN POKE 49233,0: GOSUB 320
: GOTO 190
220 IF A$ = CHR$(3) THEN 350
230 IF A$ = CHR$(32) THEN H = H + 7: GOTO 190
240 FOR I = 1 TO 95: IF A$ = MID$(F$,I,1) THEN 26
0
250 NEXT : GOTO 190
260 H = H + 15: IF H > 263 THEN H = 0:V = V + 20
270 IF V > 160 THEN HGR
280 POKE 49232,0
290 DRAW I + 1 AT H,V
300 GOTO 190
320 VTAB 22: HTAB 4: PRINT "PRESSEZ UNE TOUCHE POUR
CONTINUER"
330 CALL - 198: POKE - 16368,0: WAIT - 16384,128
,127: POKE - 16368,0
340 RETURN
350 TEXT : HOME : END
```

Et maintenant, tapez tranquillement, en plusieurs fois, les pages 24 à 29 en vérifiant, à la fin de chaque colonne et en sauvegardant à chaque fois la totalité par un **BSAVE FONTE.SPECIALE, A24576, L4498**

# ■ FONTE SPÉCIALE

```

6000- 64 02 D0 00 D9 00 F2 00
6008- 07 01 3C 01 67 01 95 01
6010- D3 01 E0 01 F3 01 07 02
6018- 32 02 4E 02 5B 02 69 02
6020- 76 02 8C 02 CD 02 E9 02
6028- 1A 03 4F 03 7D 03 AD 03
6030- E9 03 0F 04 4E 04 8A 04
6038- 9B 04 AD 04 CD 04 E6 04
6040- 05 05 2A 05 5D 05 9B 05
6048- DB 05 0D 06 4F 06 7D 06
6050- A4 06 E1 06 1D 07 36 07
6058- 5E 07 98 07 B8 07 06 08
6060- 45 08 86 08 BD 08 FF 08
6068- 41 09 77 09 98 09 D7 09
6070- 0D 0A 56 0A 92 0A C4 0A
6078- F7 0A 2A 0B 55 0B 86 0B
6080- BB 0B D2 0B 04 0C 34 0C
6088- 6B 0C 94 0C C8 0C F4 0C
6090- 17 0D 4C 0D 7D 0D 94 0D
6098- AE 0D E0 0D F9 0D 2F 0E
60A0- 5C 0E 89 0E B9 0E E4 0E
60A8- FE 0E 27 0F 46 0F 71 0F
60B0- 94 0F C9 0F F1 0F 24 10
60B8- 4C 10 7B 10 AB 10 DA 10
60C0- 20 11 61 11 61 11 61 11
60C8- 6A 11 52 00 00 00 00 00
60D0- 49 89 92 92 92 92 12 00
60D8- 00 2D 3E 37 2D 3E 37 2D
60E0- 3E 37 2D 3E 37 2D 3E 77
60E8- 11 3F 2E 35 3F 56 49 00
60F0- 00 00 2D 0D 2D 3E FF 3F
60F8- 4E 4D 31 3F 1F 3F 96 92
6100- 92 92 49 49 09 00 29 09
6108- 6D 29 3E DF 37 6D 29 3E
6110- DF FF 2A 2D 2D 2D 2D 3E
6118- 3F 3F 3F 3F 4E 6D 29 3E
6120- DF FF 2A 2D 2D 2D 2D 3E
6128- 3F 3F 3F 3F 4E 6D 29 3E
6130- DF 37 6D 29 3E DF B7 49
6138- 49 09 00 49 49 35 FF 2A
6140- 2D 2D 35 3F 3F 3F 37 6D
6148- 35 FF 37 2D 2D 2D 15 3F
6150- 3F 3F 4E 6D 35 FF FF 13
6158- 2D 2D 2D F5 3F 3F 3F 4E
6160- 29 3E 56 49 01 00 29 29
6168- 6D 09 35 FF 3B 3F 37 6D
6170- 6D 35 FF 3F 3F 0E 2D 0D
6178- 35 BF 35 BF 6D 89 3F 1F

```

```

6180- BF 6D 2D 2D 3E 1F FF BF
6188- 6D 29 2D F5 3F DF 3B 56
6190- 49 49 09 00 09 49 49 29
6198- 35 DF 6E 1E FF 9B 2D 0D
61A0- 0D 2D 15 3F 3F 3F 3F 3F
61A8- 17 6D 29 6D 29 1E FF DB
61B0- 1B 37 6D 49 09 AD FF DB
61B8- DB 37 6D 49 49 29 3E DF
61C0- 3F FF 3B 0E 2D 2D 2D 2D
61C8- F5 3F DF 3F 56 49 49 49
61D0- 01 00 37 2D 3E 77 31 3F
61D8- 96 92 92 92 49 01 00 52
61E0- 29 3E 37 35 37 35 37 35
61E8- 37 35 37 35 37 2D 3E 56
61F0- 09 00 00 AD 3F 0E 35 37
61F8- 35 37 35 37 35 37 35 BF
6200- 2D 1E B7 49 01 00 49 52
6208- 49 11 FF 9B 6D 29 4D F5
6210- FF FF 77 2D 2D F5 3F FF
6218- 13 2D 2D 2D 2D F5 1B 3F
6220- BF 2D 2D AD FF FF BF 6D
6228- 29 4D F5 DB B7 49 49 00
6230- 00 92 92 4A 09 35 37 35
6238- FF 9B 2D 2D 2D 2D 35 3F
6240- 3F 3F 3F 77 49 35 37 35
6248- B7 49 49 00 00 0A 92 92
6250- 92 92 2D 3E 77 31 3F 4D
6258- 01 00 6F 92 92 12 2D 2D
6260- 3E 3F B7 92 4A 49 00 00
6268- 92 92 92 92 4A 3A 37 2D
6270- 3E B7 49 00 00 49 49 49
6278- 35 BF 35 BF 35 BF 35 BF
6280- 35 BF 35 BF 35 B7 49 49
6288- 49 00 00 09 29 2D 2D 2D
6290- 15 3F 3F 3F 3F 37 2D 4D
6298- 49 35 FF DB 3B 37 2D 4D
62A0- 49 35 FF DB 3B 37 2D 4D
62A8- 49 35 FF DB 3B 37 2D 4D
62B0- 49 35 FF DB 3B 37 2D 4D
62B8- 49 35 FF DB 3B 37 2D 2D
62C0- 2D 2D F5 3F 3F 3F B7 49
62C8- 49 49 01 00 3F 09 2D 3E
62D0- 3F 17 2D 2D 3E 37 2D 3E
62D8- 37 2D 3E 37 2D 3E 37 2D
62E0- 3E 37 2D 3E B7 49 01 00
62E8- 29 29 2D 2D 2D 15 3F 3F
62F0- 3F 3F 37 2D 4D 49 35 37
62F8- 35 FF DB 2A 2D 2D F5 3F
6300- 3F 3F 17 2D F5 3F 37 2D
6308- 3E 37 2D 2D 2D 2D 35 3F
6310- 3F 3F 3F 56 49 49 49 00

```

VÉRIFICATION : T = 0 : FOR I = 24576 TO 24959 : T = T + PEEK (I) : NEXT : PRINT T  
RÉSULTAT : 27812

VÉRIFICATION : T = 0 : FOR I = 24960 TO 25367 : T = T + PEEK (I) : NEXT : PRINT T  
RÉSULTAT : 33940

6318- 00 49 29 2D 2D 2D 15 3F  
6320- 3F 3F 3F 37 6D 49 29 35  
6328- 3F 2E F5 FF DB 2A 2D 2D  
6330- 2D 3E 3F 3F 3F 4E 49 2D  
6338- 3E 37 2D 3E FF DB 3B 2E  
6340- 2D 2D 2D 2D 1E 3F 3F 3F  
6348- B7 49 49 49 01 00 3F 49  
6350- 49 2D 3E 3F 17 2D 2D 3E  
6358- FF BF 6D 29 35 3F DF 3B  
6360- 17 6D 49 2D 3E FF DB 37  
6368- 2D 2D 2D 2D 35 3F 3F 3F  
6370- 3F 4E 49 2D 3E 37 2D 3E  
6378- B7 49 09 00 29 29 2D 2D  
6380- 2D 3E 3F 3F 3F 37 35 37  
6388- 35 37 2D 2D 2D 2D 15 3F  
6390- 3F 3F 3F 4E 49 29 35 3F  
6398- 2E 35 3F DF DB 37 2D 2D  
63A0- 2D 2D F5 3F 3F 3F B7 49  
63A8- 49 49 01 00 01 29 2D 2D  
63B0- 2D 15 3F 3F 3F 3F 37 6D  
63B8- 49 09 F5 DB DB 3B 2E 3E  
63C0- 2E 2D 2D 2D AD 3F 3F 3F  
63C8- 3F 37 6D 49 29 35 3F DF  
63D0- DB 37 6D 49 29 35 3F DF  
63D8- DB 37 2D 2D 2D 2D F5 3F  
63E0- 3F 3F B7 49 49 49 01 00  
63E8- 2D 2D 2D 2D 2D 35 3F 3F  
63F0- 3F 3F 37 2D 4D 09 2D 3E  
63F8- 37 2D 3E BF 2D 1E 3F 17  
6400- 2D 1E 3F 2E 35 3F 2E 35  
6408- 3F 56 49 49 01 00 09 09  
6410- 2D 2D AD 3F 3F 3F 37 2D  
6418- 4D 29 3E DF 3B 37 2D 4D  
6420- 29 3E DF 3B BF 2D 2D 2D  
6428- 2D 15 3F 3F 3F 3F 37 2D  
6430- 4D 49 35 FF DB 3B 37 2D  
6438- 4D 49 35 FF DB 3B 37 2D  
6440- 2D 2D 2D F5 3F 3F 3F B7  
6448- 49 49 49 01 00 2D 29 2D  
6450- 2D 2D 15 3F 3F 3F 3F 37  
6458- 2D 4D 49 35 FF DB 3B 37  
6460- 2D 4D 49 35 FF DB 3B 37  
6468- 2D 2D 2D 2D 35 3F 3F 3F  
6470- 3F 4E 49 09 35 37 35 FF  
6478- DB 1B 37 2D 2D 2D 2D F5  
6480- 3F 3F 3F B7 49 49 49 01  
6488- 00 B7 92 4A 3A 37 2D 3E  
6490- B7 4A 3A 37 2D 3E B7 49  
6498- 00 00 92 92 4A 3A 37 2D  
64A0- 3E B7 4A 3A 37 2D 3E 77  
64A8- F5 6F 01 00 92 92 4A 09

64B0- 3E 17 2D 4D 89 3F 3F 3F  
64B8- 3F 17 2D 2D 2D 2D 35 3F  
64C0- 3F 3F 3F 0E 2D 3E 0E 96  
64C8- 49 49 00 00 92 92 92 2D  
64D0- 2D 2D 35 3F 3F 3F B7 2A  
64D8- 2D 2D 2D 3E 3F 3F 3F 96  
64E0- 4A 49 49 00 00 92 92 4A  
64E8- 49 15 37 2D 15 3F 3F 3F  
64F0- 3F 2E 2D 2D 2D 2D 1E 3F  
64F8- 3F 3F 3F 4E 49 2D 1E 37  
6500- 96 49 09 00 09 29 2D 2D  
6508- 2D 15 3F 3F 3F 3F 37 6D  
6510- 49 29 35 3F 2E 35 3F DF  
6518- 2A 2D 2D 1E 3F 3F 37 35  
6520- 37 F5 2A 3E 56 49 49 01  
6528- 00 2D 49 11 77 F5 DB 2A  
6530- 2D 2D AD 3F 3F 3F 2E  
6538- 4D 49 35 FF DB 13 2D 2D  
6540- 2D 35 3F 3F 3F 2E 6D  
6548- 49 35 FF DB 3F 2E 2D 2D  
6550- 2D 2D 3E 3F 3F 3F B7 49  
6558- 49 49 01 00 DB 49 2D 2D  
6560- AD 3F 3F 3F BF 2D 4D 09  
6568- 35 FF DB 3F 2E 6D 49 35  
6570- FF DB 3F 17 2D 2D 2D 2D  
6578- 2D 3E 3F 3F 3F 3F 77 2D  
6580- 4D 09 35 FF DB 3F 2E 6D  
6588- 49 35 FF DB 3F 2E 6D 49  
6590- 35 FF DB 3F 56 49 49 49  
6598- 01 00 09 2D 2D 2D AD 3F  
65A0- 3F 3F 3F 2E 6D 49 35 FF  
65A8- DB 3F 2E 6D 49 35 FF DB  
65B0- 3F 2E 2D 2D 2D AD 3F 3F  
65B8- 3F 3F 37 2D 4D 49 35 FF  
65C0- DB 3B 37 2D 4D 49 35 FF  
65C8- DB 3B 37 2D 2D 2D 2D F5  
65D0- 3F 3F 3F 3F 56 49 49 49  
65D8- 01 00 29 29 2D 2D 2D 15  
65E0- 3F 3F 3F 3F 37 2D 4D 49  
65E8- 35 FF DB 3B 37 2D 3E 37  
65F0- 2D 3E 37 2D 3E 37 2D 4D  
65F8- 49 35 FF DB 3B 37 2D 2D  
6600- 2D 2D F5 3F 3F 3F B7 49  
6608- 49 49 01 00 00 2D 2D 2D  
6610- 2D 15 3F 3F 3F 3F 37 2D  
6618- 4D 49 35 FF DB 3B 37 2D  
6620- 4D 49 35 FF DB 3B 37 2D  
6628- 4D 49 35 FF DB 3B 37 2D  
6630- 4D 49 35 FF DB 3B 37 2D  
6638- 4D 49 35 FF DB 3B 37 2D  
6640- 2D 2D 2D F5 3F 3F 3F 3F

VÉRIFICATION : T = 0 : FOR I = 25368 TO 25775 : T  
= T + PEEK (I) : NEXT : PRINT T  
RÉSULTAT : 30425

VÉRIFICATION : T = 0 : FOR I = 25776 TO 26183 : T  
= T + PEEK (I) : NEXT : PRINT T  
RÉSULTAT : 32034

# I FONTE SPÉCIALE

6648- 56 49 49 49 01 00 2D 29  
 6650- 2D 2D 2D 35 3F 3F 3F 3F  
 6658- 37 2D 3E 37 2D 3E 37 2D  
 6660- 2D 2D 35 3F 3F 3F 37 2D  
 6668- 3E 37 2D 3E 37 2D 2D 2D  
 6670- 2D 35 3F 3F 3F 3F 56 49  
 6678- 49 49 00 00 00 29 2D 2D  
 6680- 2D 35 3F 3F 3F 3F 37 2D  
 6688- 3E 37 2D 3E 37 2D 2D 2D  
 6690- 35 3F 3F 3F 37 2D 3E 37  
 6698- 2D 3E 37 2D 3E B7 49 49  
 66A0- 49 09 00 49 29 2D 2D 2D  
 66A8- 15 3F 3F 3F 3F 37 2D 4D  
 66B0- 49 F5 DB DB 3F 2E 35 3F  
 66B8- 2E 6D 09 2D 35 3F FF 1B  
 66C0- 3F 2E 6D 49 29 3E DF DB  
 66C8- 3F 2E 6D 49 29 3E DF DB  
 66D0- 3F 2E 2D 2D 2D 2D 1E 3F  
 66D8- 3F 3F B7 49 49 49 01 00  
 66E0- 6D 29 6D 49 35 FF DB 3F  
 66E8- 2E 6D 49 35 FF DB 3F 2E  
 66F0- 6D 49 35 FF DB 3F 17 2D  
 66F8- 2D 2D 2D 35 3F 3F 3F 3F  
 6700- 77 2D 4D 09 35 FF DB 3F  
 6708- 2E 6D 49 35 FF DB 3F 2E  
 6710- 6D 49 35 FF DB 3F 56 49  
 6718- 49 49 00 00 49 2D 3E 37  
 6720- 2D 3E 37 2D 3E 37 2D 3E  
 6728- 37 2D 3E 37 2D 3E 37 2D  
 6730- 3E B7 49 01 00 00 49 49  
 6738- 2D 3E 37 2D 3E 37 2D 3E  
 6740- 37 2D 3E 37 2D 3E FF DB  
 6748- 13 6D 49 2D 3E FF DB 37  
 6750- 2D 2D 2D 2D 1E 3F 3F 3F  
 6758- 56 49 49 09 00 00 2D 4D  
 6760- 09 35 FF DB 3F 2E 6D 49  
 6768- 35 FF DB 3F 2E 6D 09 2D  
 6770- 1E 3F DF 3F 2E 2D 2D F5  
 6778- 3F 3F 37 2D 2D AD 3F 1F  
 6780- 3F 2E 6D 29 AD 3F DF 3B  
 6788- 37 2D 4D 09 2D 3E DF DB  
 6790- 3F 56 49 49 49 01 00 35  
 6798- 2D 3E 37 2D 3E 37 2D 3E  
 67A0- 37 2D 3E 37 2D 3E 37 2D  
 67A8- 3E 37 2D 2D 2D 2D 3E 3F  
 67B0- 3F 3F 56 49 49 09 00 DB  
 67B8- 29 2D 4D 2D AD 3F 3F 3F  
 67C0- 3F 3F 37 2D 4D 6D 09 35

67C8- FF 1B FF 3B 37 2D 4D 6D  
 67D0- 09 35 FF 1B FF 3B 37 2D  
 67D8- 4D 6D 09 35 FF 1B FF 3B  
 67E0- 37 2D 4D 6D 09 35 FF 1B  
 67E8- FF 3B 37 2D 4D 6D 09 35  
 67F0- FF 1B FF 3B 37 2D 4D 6D  
 67F8- 09 35 FF 1B FF 3B B7 49  
 6800- 49 49 49 01 00 00 2D 4D  
 6808- 2D AD 3F 3F 3F 3F 37 2D  
 6810- 4D 49 35 FF DB 3B 37 2D  
 6818- 4D 49 35 FF DB 3B 37 2D  
 6820- 4D 49 35 FF DB 3B 37 2D  
 6828- 4D 49 35 FF DB 3B 37 2D  
 6830- 4D 49 35 FF DB 3B 37 2D  
 6838- 4D 49 35 FF DB 3B B7 49  
 6840- 49 49 09 00 00 29 2D 2D  
 6848- 2D 15 3F 3F 3F 3F 37 2D  
 6850- 4D 49 35 FF DB 3B 37 2D  
 6858- 4D 49 35 FF DB 3B 37 2D  
 6860- 4D 49 35 FF DB 3B 37 2D  
 6868- 4D 49 35 FF DB 3B 37 2D  
 6870- 4D 49 35 FF DB 3B 37 2D  
 6878- 2D 2D 2D F5 3F 3F 3F B7  
 6880- 49 49 49 01 00 AD 29 2D  
 6888- 2D 2D 15 3F 3F 3F 3F 37  
 6890- 2D 4D 49 35 FF DB 3B 37  
 6898- 2D 4D 49 35 FF DB 3B 37  
 68A0- 2D 4D 49 35 3F 3F 3F 3F  
 68A8- 37 2D 2D 2D 2D DE DB 3B  
 68B0- 37 2D 3E 37 2D 3E B7 49  
 68B8- 49 49 09 00 09 29 2D 2D  
 68C0- 2D 15 3F 3F 3F 3F 37 2D  
 68C8- 4D 49 35 FF DB 3B 37 2D  
 68D0- 4D 49 35 FF DB 3B 37 2D  
 68D8- 4D 49 35 FF DB 3B 37 2D  
 68E0- 4D 49 35 FF DB 3B 37 2D  
 68E8- 4D 6D 35 3F FF 1B 3F 2E  
 68F0- 2D 2D 2D 2D 3E 3F 3F 3F  
 68F8- B7 49 49 49 01 00 2D 29  
 6900- 2D 2D 2D 15 3F 3F 3F 3F  
 6908- 37 2D 4D 49 35 FF DB 3B  
 6910- 37 2D 4D 49 35 3F DF 1B  
 6918- 3F 2E 2D 2D 2D 35 3F 3F  
 6920- 3F 3F 2E 6D 49 2D 3E DF  
 6928- DB 3F 2E 6D 49 29 3E DF  
 6930- DB 3F 2E 6D 49 29 3E DF  
 6938- DB 3F 56 49 49 49 01 00  
 6940- 2D 29 2D 2D 2D 15 3F 3F  
 6948- 3F 3F 37 6D 49 09 F5 DB  
 6950- DB 3B 2E 3E 2E 2D 2D 2D  
 6958- AD 3F 3F 3F 3F 4E 49 29

VÉRIFICATION : T = 0 : FOR I = 26184 TO 26567 : T  
 = T + PEEK (I) : NEXT : PRINT T  
 RÉSULTAT : 27756

VÉRIFICATION : T = 0 : FOR I = 26568 TO 26975 : T  
 = T + PEEK (I) : NEXT : PRINT T  
 RÉSULTAT : 34961

```

6960- 35 3F 2E 35 3F DF DB 37
6968- 2D 2D 2D 2D F5 3F 3F 3F
6970- B7 49 49 49 01 00 00 2D
6978- 2D 2D 2D 3E 3F 3F 3F 77
6980- 09 2D 3E 37 2D 3E 37 2D
6988- 3E 37 2D 3E 37 2D 3E 37
6990- 2D 3E B7 49 49 01 00 00
6998- 2D 4D 49 35 FF DB 3B 37
69A0- 2D 4D 49 35 FF DB 3B 37
69A8- 2D 4D 49 35 FF DB 3B 37
69B0- 2D 4D 49 35 FF DB 3B 37
69B8- 2D 4D 49 35 FF DB 3B 37
69C0- 2D 4D 49 35 FF DB 3B 37
69C8- 2D 2D 2D 2D F5 3F 3F 3F
69D0- B7 49 49 49 01 00 49 2D
69D8- 4D 49 35 FF DB 3B 37 2D
69E0- 4D 49 35 FF DB 3B 37 2D
69E8- 4D 49 35 FF DB 3B 37 2D
69F0- 4D 49 35 FF DB 3B 37 2D
69F8- 4D 49 35 3F DF 3B 77 2D
6A00- 0D 2D 1E 3F 3F 0E 2D 1E
6A08- 56 49 09 00 00 2D 4D 49
6A10- 09 35 FF DB DB 3F 2E 6D
6A18- 49 49 35 FF DB DB 3F 2E
6A20- 6D 09 6D 29 3E DF FF 1B
6A28- 3F 2E 6D 09 6D 29 3E DF
6A30- FF 1B 3F 2E 6D 09 6D 29
6A38- 3E DF FF 1B 3F 2E 6D 09
6A40- 6D 29 3E 3F 3F 3F 3F 77
6A48- 2D 6D 29 2D 1E FF DB B7
6A50- 49 49 49 01 00 00 2D 4D
6A58- 49 35 FF DB 3B 37 2D 4D
6A60- 49 35 FF DB 3B 37 2D 4D
6A68- 49 F5 FF 1B 3F 0E 2D 2D
6A70- 35 3F 3F BF 2D 4D 29 15
6A78- FF DB 3B 37 2D 4D 49 35
6A80- FF DB 3B 37 2D 4D 49 35
6A88- FF DB 3B B7 49 49 49 09
6A90- 00 49 2D 4D 49 35 FF DB
6A98- 3B 37 2D 4D 49 35 FF DB
6AA0- 3B 37 2D 4D 49 35 FF DB
6AA8- 3B 37 2D 4D 49 F5 FF 1B
6AB0- 3F 0E 2D 0D F5 3F 37 2D
6AB8- 35 3F 37 2D 35 3F B7 49
6AC0- 49 09 00 00 29 2D 2D 2D
6AC8- 15 3F 3F 3F 3F 37 6D 49
6AD0- 29 F5 3F 17 2D 1E 3F 17
6AD8- 2D 1E 3F 17 2D 1E 3F 2E
6AE0- 6D 49 89 FF DB 3B 37 2D
6AE8- 2D 2D 2D F5 3F 3F 3F 3F
6AF0- 56 49 49 49 01 00 49 49

```

VÉRIFICATION : T = 0 : FOR I = 26976 TO 27383 : T = T + PEEK (I) : NEXT : PRINT T  
RÉSULTAT : 34972

```

6AF8- 49 3A 1F 37 6D F5 DB 13
6B00- 2D 2D 2D 15 3F 3F 3F 3F
6B08- 2E 6D 49 35 FF DB 3F 2E
6B10- 2D 2D 2D 35 3F 3F 3F 3F
6B18- 2E 35 3F 2E 2D 2D 2D 35
6B20- 3F 3F 3F B7 49 49 49 00
6B28- 00 2D 92 0A 2D 2D 2D 15
6B30- 3F 3F 3F 3F 2E 6D 49 F5
6B38- DB 1B 3F 2E 35 3F 2E 6D
6B40- 49 11 FF DB 3F 2E 2D 2D
6B48- 2D F5 3F 3F 3F 4E 29 1E
6B50- 6F 49 09 00 49 49 89 BF
6B58- 2D F5 DB 13 2D 2D 2D 15
6B60- 3F 3F 3F 3F 2E 6D 49 35
6B68- FF DB 3F 2E 2D 2D 2D 35
6B70- 3F 3F 3F 3F 2E 35 3F 2E
6B78- 2D 2D 2D 35 3F 3F 3F B7
6B80- 49 49 49 00 00 FF 49 89
6B88- BF 2D F5 DB 13 2D 2D 2D
6B90- 15 3F 3F 3F 3F 2E 4D 49
6B98- 35 FF DB 13 2D 2D 2D 35
6BA0- 3F 3F 3F 3F 2E 6D 49 35
6BA8- FF DB 3F 2E 2D 2D 2D 2D
6BB0- 3E 3F 3F 3F B7 49 49 49
6BB8- 01 00 35 92 92 92 92 12
6BC0- 2D 2D 2D 2D 2D 35 3F 3F
6BC8- 3F 3F 3F 6F 49 49 49 09
6BD0- 00 BF 49 89 BF 2D F5 DB
6BD8- 13 2D 2D 2D 15 3F 3F 3F
6BE0- 3F 2E 6D 49 35 FF DB 3F
6BE8- 2E 6D 49 35 FF DB 3F 2E
6BF0- 6D 49 35 FF DB 3F 2E 2D
6BF8- 2D 2D F5 3F 3F 3F 56 49
6C00- 49 09 00 2D 92 0A 2D 2D
6C08- 2D 15 3F 3F 3F 3F 2E 4D
6C10- 49 35 FF DB 13 2D 2D 2D
6C18- 35 3F 3F 3F 3F 2E 6D 49
6C20- 35 FF DB 3F 2E 2D 2D 2D
6C28- 2D 3E 3F 3F 3F B7 49 49
6C30- 49 01 00 35 2D 3E 37 2D
6C38- 3E 37 2D 2D 2D 2D 15 3F
6C40- 3F 3F 3F 37 2D 4D 49 35
6C48- FF DB 3B 37 2D 4D 49 35
6C50- FF DB 3B 37 2D 4D 49 35
6C58- FF DB 3B 37 2D 2D 2D 2D
6C60- F5 3F 3F 3F B7 49 49 49
6C68- 01 00 0A 92 0A 2D 2D 2D
6C70- 15 3F 3F 3F 3F 2E 6D 49
6C78- F5 DB 1B 3F 2E 35 3F 2E
6C80- 6D 49 11 FF DB 3F 2E 2D
6C88- 2D 2D F5 3F 3F 3F 56 49

```

VÉRIFICATION : T = 0 : FOR I = 27384 TO 27791 : T = T + PEEK (I) : NEXT : PRINT T  
RÉSULTAT : 32371

# FONTES SPÉCIALES

6C90- 49 09 00 00 49 49 29 3E  
 6C98- 2E 3E DF DB 2A 2D 2D 2D  
 6CA0- 3E 3F 3F 3F 37 2D 4D 09  
 6CA8- 35 FF DB 3F 2E 6D 49 35  
 6CB0- FF DB 3F 2E 6D 49 35 FF  
 6CB8- DB 3F 2E 2D 2D 2D F5 3F  
 6CC0- 3F 3F 56 49 49 09 00 09  
 6CC8- 92 0A 2D 2D 2D 15 3F 3F  
 6CD0- 3F 3F 2E 6D 49 35 FF DB  
 6CD8- 3F 2E 2D 2D 2D 35 3F 3F  
 6CE0- 3F 3F 2E 35 3F 2E 2D 2D  
 6CE8- 2D 35 3F 3F 3F B7 49 49  
 6CF0- 49 00 00 00 49 2D 2D 3E  
 6CF8- 3F 3F 2E 35 3F 2E 6D 3A  
 6D00- 3F 3F 37 2D 2D 2D DE 3F  
 6D08- 2E 35 3F 2E 35 3F 2E 35  
 6D10- 3F 56 49 49 00 00 92 92  
 6D18- 0A 2D 2D 2D 15 3F 3F 3F  
 6D20- 3F 2E 6D 49 35 FF DB 3F  
 6D28- 2E 6D 49 35 FF DB 3F 2E  
 6D30- 2D 2D 2D 35 3F 3F 3F 77  
 6D38- 49 09 35 FF DB 9B 2D 2D  
 6D40- 2D 2D 1E 3F 3F 3F 6F 49  
 6D48- 49 09 00 00 2D 3E 37 2D  
 6D50- 3E 37 2D 2D 2D AD 3F 3F  
 6D58- 3F 3F 2E 6D 49 35 FF DB  
 6D60- 3F 2E 6D 49 35 FF DB 3F  
 6D68- 2E 6D 49 35 FF DB 3F 2E  
 6D70- 6D 49 35 FF DB 3F 56 49  
 6D78- 49 49 00 00 3E 2D 3E B7  
 6D80- 2A 35 3F 2E 35 3F 2E 35  
 6D88- 3F 2E 35 3F 2E 35 3F 56  
 6D90- 49 00 00 00 29 35 3F 96  
 6D98- 2D 3E 37 2D 3E 37 2D 3E  
 6DA0- 37 2D 3E 77 35 FF 2A 2D  
 6DA8- 1E 3F 4D 09 00 35 2D 3E  
 6DB0- 37 2D 3E 37 2D 4D 09 35  
 6DB8- 3F DF 3B 37 2D 4D 2D 1E  
 6DC0- 3F 3F 3F 2E 2D 2D AD 3F  
 6DC8- DF 3F 2E 6D 09 2D 3E DF  
 6DD0- 1B 3F 2E 6D 49 35 FF DB  
 6DD8- 3F 56 49 49 49 00 00 37  
 6DE0- 2D 3E 37 2D 3E 37 2D 3E  
 6DE8- 37 2D 3E 37 2D 3E 37 2D  
 6DF0- 3E 37 2D 3E B7 49 01 00  
 6DF8- 01 92 0A 2D 6D 29 AD 3F  
 6E00- 3F 3F 3F 3F 2E 6D 29 4D  
 6E08- 35 FF 3B DF 3F 2E 6D 29

6E10- 4D 35 FF 3B DF 3F 2E 6D  
 6E18- 29 4D 35 FF 3B DF 3F 2E  
 6E20- 6D 29 4D 35 FF 3B DF 3F  
 6E28- 56 49 49 49 09 00 49 92  
 6E30- 0A 2D 2D 2D 15 3F 3F 3F  
 6E38- 3F 2E 6D 49 35 FF DB 3F  
 6E40- 2E 6D 49 35 FF DB 3F 2E  
 6E48- 6D 49 35 FF DB 3F 2E 6D  
 6E50- 49 35 FF DB 3F 56 49 49  
 6E58- 49 00 00 00 92 0A 2D 2D  
 6E60- 2D 15 3F 3F 3F 3F 2E 6D  
 6E68- 49 35 FF DB 3F 2E 6D 49  
 6E70- 35 FF DB 3F 2E 6D 49 35  
 6E78- FF DB 3F 2E 2D 2D 2D F5  
 6E80- 3F 3F 3F 56 49 49 09 00  
 6E88- 00 92 0A 2D 2D 2D 15 3F  
 6E90- 3F 3F 3F 2E 6D 49 35 FF  
 6E98- DB 3F 2E 6D 49 35 FF DB  
 6EA0- 3F 2E 6D 49 35 3F 3F 3F  
 6EA8- 3F 2E 2D 2D 2D DE DB 3F  
 6EB0- 2E 35 3F 4D 49 49 01 00  
 6EB8- 2D 92 0A 2D 2D 2D 35 3F  
 6EC0- 3F 3F 3F 2E 6D 49 35 FF  
 6EC8- DB 3F 2E 6D 49 35 FF DB  
 6ED0- 3F 2E 6D 49 35 3F 3F 3F  
 6ED8- 3F 0E 2D 2D 2D 35 37 35  
 6EE0- 6F 01 00 00 92 0A 2D 2D  
 6EE8- 35 3F 3F 3F 2E 35 3F 2E  
 6EF0- 35 3F 2E 35 3F 2E 35 3F  
 6EF8- 56 49 49 01 00 00 92 0A  
 6F00- 2D 2D AD 3F 3F 3F 37 6D  
 6F08- 49 F5 DB 1B 37 2D 2D 2D  
 6F10- 15 3F 3F 3F 4E 09 2D 3E  
 6F18- FF 1B 37 2D 2D 2D F5 3F  
 6F20- 3F B7 49 49 09 00 3E 49  
 6F28- 11 3F 2E 35 3F 9F 2D 2D  
 6F30- 35 3F 3F 77 29 35 3F 2E  
 6F38- 35 3F 2E AD 3F 37 2D 35  
 6F40- 3F 56 49 00 00 4D 92 2A  
 6F48- 6D 49 35 FF DB 3F 2E 6D  
 6F50- 49 35 FF DB 3F 2E 6D 49  
 6F58- 35 FF DB 3F 2E 6D 49 35  
 6F60- FF DB 3F 2E 2D 2D 2D F5  
 6F68- 3F 3F 3F 56 49 49 09 00  
 6F70- 09 92 2A 6D 49 35 FF DB  
 6F78- 3F 2E 6D 49 35 FF DB 3F  
 6F80- 2E 6D 49 F5 FF 3B 77 2D  
 6F88- 2D 1E 3F 2E 35 3F 56 49  
 6F90- 49 00 00 B7 92 2A 6D 49  
 6F98- 09 35 FF DB 1B 3F 2E 6D  
 6FA0- 29 4D 35 FF 3B DF 3F 2E

VÉRIFICATION : T = 0 : FOR I = 27792 TO 28175 : T  
 = T + PEEK (I) : NEXT : PRINT T  
 RÉSULTAT : 28239

VÉRIFICATION : T = 0 : FOR I = 28176 TO 28583 : T  
 = T + PEEK (I) : NEXT : PRINT T  
 RÉSULTAT : 34168

6FA8-	6D	29	4D	35	FF	3B	DF	3F	70A0-	2D	F5	3F	3F	3F	56	49	49
6FB0-	2E	6D	29	4D	35	FF	3B	DF	70A8-	09	00	52	49	11	77	F5	DB
6FB8-	3F	2E	2D	2D	2D	2D	F5	3F	70B0-	2A	2D	2D	AD	3F	3F	3F	3F
6FC0-	DF	3F	B7	49	49	49	09	00	70B8-	2E	6D	49	35	FF	DB	3F	2E
6FC8-	49	92	2A	6D	49	35	FF	DB	70C0-	2D	2D	2D	35	3F	3F	3F	3F
6FD0-	3F	2E	6D	49	F5	FF	3B	77	70C8-	2E	35	3F	2E	2D	2D	2D	35
6FD8-	2D	2D	3E	3F	BF	2D	4D	AD	70D0-	3F	3F	3F	B7	49	49	49	00
6FE0-	FF	DB	3F	2E	6D	49	35	FF	70D8-	00	1E	49	49	29	35	3F	37
6FE8-	DB	3F	56	49	49	49	00	00	70E0-	2D	1E	FF	9B	2D	0D	0D	2D
6FF0-	B7	92	2A	6D	49	35	FF	DB	70E8-	15	3F	3F	3F	3F	3F	17	2D
6FF8-	3F	2E	6D	49	35	FF	DB	3F	70F0-	2D	2D	2D	2D	1E	3F	3F	3F
7000-	2E	6D	49	35	FF	DB	3F	2E	70F8-	3F	37	2D	2D	2D	2D	AD	3F
7008-	2D	2D	2D	35	3F	3F	3F	77	7100-	3F	3F	3F	3F	2E	2D	2D	2D
7010-	49	09	35	FF	DB	9B	2D	2D	7108-	2D	2D	3E	3F	3F	3F	3F	3F
7018-	2D	2D	1E	3F	3F	3F	6F	49	7110-	0E	2D	2D	2D	2D	F5	3F	DF
7020-	49	09	00	09	92	2A	2D	2D	7118-	3F	56	49	49	49	01	00	2D
7028-	2D	3E	3F	3F	3F	2E	4D	29	7120-	52	49	49	89	3F	3F	3F	3F
7030-	F5	3F	17	2D	1E	3F	17	2D	7128-	37	2D	2D	2D	2D	35	3F	3F
7038-	4D	89	FF	DB	3F	2E	2D	2D	7130-	3F	3F	37	2D	2D	2D	2D	35
7040-	2D	F5	3F	3F	3F	B7	49	49	7138-	3F	3F	3F	3F	37	2D	2D	2D
7048-	49	01	00	3F	49	89	BF	F5	7140-	2D	35	3F	3F	3F	3F	37	2D
7050-	9B	2D	2D	2D	15	3F	3F	3F	7148-	2D	2D	2D	35	3F	3F	3F	3F
7058-	3F	2E	6D	49	35	FF	DB	3F	7150-	37	2D	2D	2D	2D	35	3F	3F
7060-	2E	2D	2D	2D	35	3F	3F	3F	7158-	3F	3F	B7	49	49	49	01	00
7068-	3F	2E	35	3F	2E	2D	2D	2D	7160-	00	40	C0	40	C0	40	C0	40
7070-	35	3F	3F	3F	B7	49	49	49	7168-	08	00	19	00	00	00	00	00
7078-	00	00	52	49	11	77	F5	DB	7170-	00	00	00	00	00	B2	0D	0D
7080-	13	2D	4D	09	35	FF	DB	3F	7178-	0D	0D	4D	45	4D	4F	52	59
7088-	2E	6D	49	35	FF	DB	3F	2E	7180-	20	4C	41	4E	45	0D	3D	3D
7090-	6D	49	35	FF	DB	3F	2E	6D	7188-	3D							
7098-	49	35	FF	DB	3F	2E	2D	2D	7190-	3D	0D						

VÉRIFICATION : T = 0 : FOR I = 28584 TO 29073 : T = T + PEEK (I) : NEXT : PRINT T      RÉSULTAT : 37781

VÉRIFICATION GÉNÉRALE : T = 0 : FOR I = 24576 TO 29073 : T = T + PEEK (I) : NEXT : PRINT T  
RÉSULTAT : 354459

Sauvegarder par un **BSAVE FONTE.SPECIALE, A24576, L4498**

## ERREURS (à corriger S.V.P.)

**N°1** Page 17 (8<sup>e</sup> ligne avant la fin). Lire :  
• CALL 24793 (ici, adresse décimale de \$60D9)

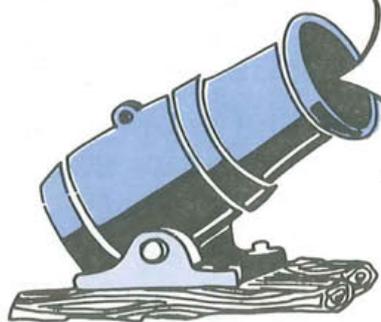
Page 46 (ligne 270). Supprimer ND = ND + 1

Page 64 (ligne 20) . 0306 : 85 06 (STA 006)

**N°2** Page 57 : BSAVE DIA.CARAC, A\$6100, L\$300 (au lieu de 400)

Page 60 : deuxième colonne, deuxième ligne, lire CATALOG, et non LIST.





## Des amis qui vous veulent du bien

• par GUY-HACHETTE

**S** i vous êtes riche ou puissant — voire les deux à la fois —, ne cherchez surtout pas à compter vos prétendus amis : ils sont forcément nombreux, et leur enthousiasme est tel qu'ils pratiqueraient volontiers, si vous leur en fournissiez l'occasion, quelque vulgaire forme de baise-main.

Par contre, si vous avez récemment connu quelque revers de fortune, c'est le moment ou jamais d'ouvrir un nouveau répertoire de vos vrais copains, ceux des bons et des mauvais jours. Comment ? Seulement quelques noms ? Vous dites qu'ils n'ont pas toujours été tendres avec vous ? Peut-être se sont-ils contentés de manifester une objectivité que votre aveuglement prit alors pour de l'hostilité...

Si j'en juge par tout ce que je lis sur Apple depuis quelque temps, ma vieille Pomme doit connaître des difficultés... car ses anciens "amis" semblent se réjouir des malheurs qui frappent le géant de l'informatique personnelle.

Pour ma part, je ne pratique pas le catastrophisme, et je me vois mal dans la peau du sadique qui appuie sur la tête d'un monsieur essayant d'échapper à la noyade.

J'ignore si Apple souffre (autant que les autres) de la crise actuelle, mais le contraire m'étonnerait. Par contre, j'estime que le brillant passé de cette marque plaide en sa faveur. On a montré, naguère, chez Apple, avec des hommes que l'on accuse aujourd'hui d'avoir été de mauvais gestionnaires, que les (bonnes) idées, associées au dynamisme de la jeunesse, pouvaient créer un grand courant commercial.

Bien sûr, il y avait (ils existent encore) les fans de l'Apple, prêts à acheter n'importe quoi, pourvu que cela soit marqué de la populaire pomme.

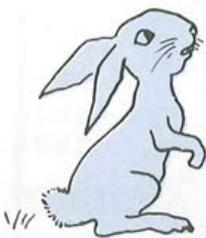
Il est non moins certain que l'évolution rapide de la micro-informatique et, surtout, le lancement de Mac, machine révolutionnaire, mais financièrement interdite à toute une catégorie d'utilisateurs, ont creusé un fossé que le manque de communication avait déjà ébauché entre Apple et les hobbistes.

Je sais, par les Lectrices et Lecteurs de *Tremplin Micro*, qu'un club parisien (trop, peut-être) ne saurait résoudre tous les problèmes, surtout lorsqu'il devient visiblement une affaire de gros sous : on ne paye pas une cotisation pour avoir le droit de lire, chaque mois, que Mac est le meilleur, etc. ou celui d'acheter un tee-shirt.

Il reste que vous et moi possédons un (ou des) Apple et que nous en sommes dans l'ensemble très satisfaits.

Que l'on ne compte pas sur moi, en tout cas, pour applaudir certaines initiatives, mais pas davantage pour hurler avec les loups.

Quand il y a un ver dans une pomme, c'est souvent parce qu'elle est délicieuse !



# QUESTIONS

Nous répondons ici aux questions de lecteurs néophytes.

• *Pouvez-vous m'expliquer clairement comment fonctionne une boucle de type FOR A = 1 TO 12 STEP 3 ?*

La meilleure explication vous sera fournie par votre Apple. Tapez donc, en mode direct :

**FOR A = 1 TO 12 STEP 3 : PRINT A : NEXT**

et vous verrez s'afficher les valeurs 1, 4, 7 et 10,

c'est-à-dire  $1 + 3 = 4 + 3 = 7 + 3 = 10...$

En l'absence de STEP, le compteur de la boucle est augmenté de 1 à chaque passage.

**Question à 2F50** : qu'est-ce qui se passerait avec  $STEP = 0$  ? Essayez !

• *Pourquoi utilisez-vous toujours le dollar après un INPUT, pour des valeurs numériques. Exemple : INPUT "NOMBRE" ; A\$. ... puis A = VAL(A\$) ? C'est compliqué et inutile.*

Si vous programmez ainsi votre INPUT :

— 10 INPUT "DONNEE " ; A

et tapez un RETURN sans avoir d'abord enfoncé une touche numérique, le message — ? REENTER

va apparaître... et démolir votre bel écran. En utilisant une valeur alphanumérique, vous restez maître (ou maîtresse) de la situation.

Autre méthode, avec ONERR GOTO :

5 ONERR GOTO 10

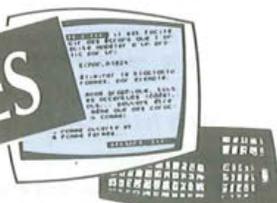
10 HTAB 1 : VTAB 5 : INPUT A

• *Je ne vois pas très bien à quoi peut servir l'instruction SPEED (que j'ai d'ailleurs remarquée dans certains programmes de Tremplin Micro) ?*

SPEED (vitesse) définit la vitesse d'affichage des caractères (ou d'émission vers un autre terminal). Elle doit être comprise entre 0 et 255. SPEED permet de fabriquer des boucles d'attente. Par exemple :

10 SPEED = 1 : PRINT " " : SPEED = 255

# REponses



# CALLS

Nous n'utilisons pas tous ces CALLS dans nos programmes, mais gageons que certains d'entre eux vous rendront service !

**CALL-1184** ..... \$FB60

Efface l'écran et affiche "APPLE II"

**CALL-1036** ..... \$FBF4

Avance le curseur d'une position

**CALL-1008** ..... \$FC10

Reculé le curseur d'une position

**CALL-998** ..... \$FC1A

Remonte le curseur d'une ligne

**CALL-958** ..... \$FC42

Efface depuis le curseur, jusqu'au bas de l'écran

**CALL-922** ..... \$FC66

Saut de ligne

**CALL-868** ..... \$FC9C

Efface la ligne à partir du curseur (celui-ci ne change pas de place)

**CALL-198** ..... \$FF3A

Emission d'un bip

**CALL-151** ..... \$FF69

Entrée dans le moniteur avec apparition de l'\*

## GRAPHISME (HI-RES)

**CALL-3100** ..... \$F3E4

Permet de visualiser la page 1 (POKE-16302,0 annule les lignes de texte, en bas de l'écran et TEXT — ou POKE-16303,0 — permet de retrouver l'écran TEXT)

**CALL-3082** ..... \$F3F6

Colorer la page graphique de cette manière :

HGR2 : HCOLOR = 5 : HPL0T 0,0 :

CALL-3082

# Votre bibliothèque INFORMATIQUE

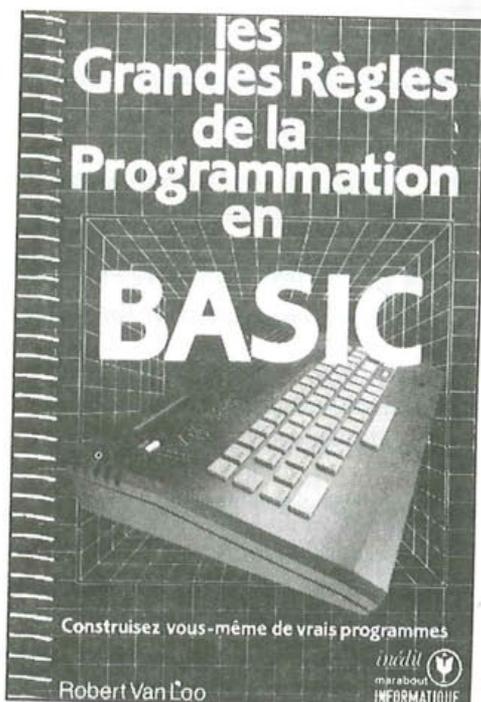
par Guy-HACHETTE

## ● Familiarisez-vous (aussi) avec le MSX

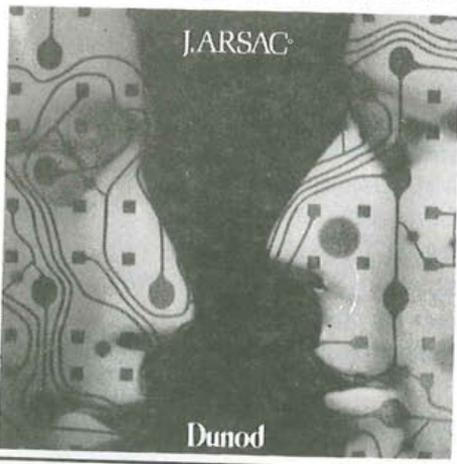
Vous possédez un Apple, mais vous avez entendu parler du M.S.X. (comme *Microsoft Extended*). Cette version du Basic deviendra-t-elle LA version de tous les producteurs de micros de la planète ? Il est permis d'en douter, mais on peut raisonnablement penser que cette norme sera à l'origine de multiples nouveautés, au cours des prochains mois.

Le livre de Robert Van Loo a le mérite de commencer par l'abc, et celui — à signaler — d'être à la portée de tous les débutants. Etant donné qu'il s'agit d'une édition de poche, relativement bon marché, je crois qu'il est digne de figurer dans votre bibliothèque Informatique.

Je ne l'ai pas étudié en détail, mais je l'ai parcouru. Il n'est jamais ennuyeux et pas davantage exagérément didactique. C'est une approche intelligente de la programmation, avec des chapitres utiles : glossaire du Basic MSX, problèmes de tri, codes des caractères, etc.



## JEUX et casse-tête A PROGRAMMER



## ● Des idées... et même de bonnes idées

Jacques Arsac a une mauvaise opinion des revues de micro-informatique, parce que «les jeux ainsi publiés sont d'une qualité médiocre... mal rédigés, horriblement mal commentés, comme si l'auteur voulait empêcher le lecteur de les comprendre...». N'en jetez plus, cher Maître ! Grâce à vous, tout cela va changer !

N'empêche qu'il a raison : «celui qui programme ses jeux s'amuse deux fois»... comme je l'ai écrit dans *Tremplin Micro n°1*, mais avec moins de talent, bien entendu.

Pour le reste, je ne suis pas d'accord avec son jugement, ne serait-ce qu'en raison de l'absence de la défense. A quoi serviraient les ouvrages de vulgarisation, comme celui de Jacques Arsac, si les périodiques se révélaient capables de les remplacer ?

Son livre — qui se veut structuré — n'est pas aussi clair qu'il le pense, mais il est possible que ma modeste cervelle de journaliste soit seule en cause dans cette affaire.

Par contre (dans *Tremplin Micro*, on donne la parole à la défense), il est bourré de raisonnements intéressants. C'est une invitation à la programmation, mais à étudier plutôt qu'à lire.

Pour terminer, que l'on me permette une constatation (en micro-minuscules) : il y manque tout de même, en *Basic*, quelques solutions géniales — et inédites — aux jeux qui y sont évoqués.

# I Est-ce une anagramme ?

**V**OUS serez probablement amené, dans un programme de jeu... ou d'enseignement, à traiter le problème des anagrammes. Est-ce une anagramme... ou non ? Divers procédés permettent de répondre à cette question. Le programme DEMO de Clément Renard vous fournit deux solutions : la première en Basic pur, la seconde avec un complément en langage machine.

## Première partie en Basic :

Les valeurs ASCII des N lettres du mot de base sont poquées à partir de l'adresse 8192 (choisie arbitrairement) ou \$2000.

A ce stade, on contrôle qu'il s'agit bien des lettres capitales (l'espace est accepté).

Ensuite, chaque fois qu'une lettre de la pseudo-anagramme correspond à l'une des valeurs poquées, on transforme cette dernière en un beau zéro.

Devinez la suite !

## Deuxième partie en langage machine :

Si on lit 0 à l'adresse \$06, c'est que le mot n'existe pas.

Si on lit 1 à l'adresse \$07, c'est une anagramme (0 = non).

```

10 TEXT : NORMAL : PRINT CHR$ (12); CHR$ (21): GO
   SUB 300
20 HOME : INVERSE : HTAB 6: PRINT " VERIFICATION D
   'UNE ANAGRAMME ": NORMAL
30 T$ = "": FOR I = 1 TO 10:T$ = T$ + "-----": NEXT
   : VTAB 2: PRINT T$
40 VTAB 6: CALL - 958: CALL - 198
50 INPUT "MOT/LETTRES -> ";M$: IF M$ = "" THEN 180
60 L = LEN (M$): FOR I = 1 TO L:A = ASC ( MID$ (M
   $,I,1)): IF (A < 65 OR A > 90) AND A < > 32 THEN 4
   0
70 POKE I + 8191,A: NEXT
80 VTAB 8: CALL - 958: CALL - 198
90 INPUT "ANAGRAMME -> ";A$: IF A$ = "" THEN 40
100 IF LEN (A$) < > L THEN 80
110 FOR I = 1 TO L:A = ASC ( MID$ (A$,I,1))
120 FOR J = 8192 TO 8191 + L: IF A = PEEK (J) THE
   N POKE J,0: GOTO 140
130 NEXT J: GOTO 150
140 NEXT I: GOTO 160
150 PRINT : PRINT "CE N'EST PAS UNE ANAGRAMME": GO
   TO 170
160 PRINT : PRINT "C'EST UNE ANAGRAMME"
170 VTAB 22: HTAB 1: PRINT "<1> BASIC <2> LM <3> F
   IN ";: GET R$
180 PRINT : IF R$ = "1" THEN 20
190 IF R$ = "2" THEN 220
200 IF R$ < > "3" THEN 170
210 TEXT : HOME : END
220 HOME : INVERSE : PRINT " ET AVEC UNE ROUTINE E
   N LANGAGE MACHINE ": NORMAL
230 VTAB 2: PRINT T$
240 PRINT : VTAB 8: CALL - 198: CALL - 958
250 INVERSE : PRINT "MOT-BASE";: NORMAL : PRINT "
   -> ";: CALL 768: IF PEEK (6) = 0 THEN 170
260 PRINT : VTAB 10: CALL - 198: CALL - 958
270 INVERSE : PRINT "ANAGRAMME";: NORMAL : PRINT "
   -> ";: CALL 787
280 IF PEEK (7) = 1 THEN 160
290 GOTO 150

```

(SUITE PAGE 34)

## ■ Est-ce une anagramme ? (suite)

```

300 RESTORE : FOR I = 768 TO 828: READ A: POKE I,A
   : NEXT : RETURN
310 DATA 32,44,213,162,255,232,189,0,2,240,5,157,0
   ,32,208,245,134,6,96,32,44,213,160,255,200,162,255,
   232,185,0,2,221,0,32,208,7,169,0,157,0,32,240,9,228
   ,6,208,236,169,0,133,7,96,196,6,208,224,169,1,133,7
   ,96
    
```

La ligne 310 correspond exactement au programme expliqué ci-après.

Lorsqu'une routine en langage machine est courte (ce qui est le cas) la mettre en DATA évite de la charger à partir de la disquette... et permet de la réinstaller facilement.

## ANAGRAMME.LM

0300-	20 2C D5	JSR	\$D52C	Entrée INLIN (voir fiche numéro 4)
0303-	A2 FF	LDX	£\$FF	A la ligne suivante, on aura donc \$FF + 1 = \$00
0305-	E8	INX		X = X + 1 (registre)
0306-	BD 00 02	LDA	\$0200,X	On lit la valeur écrite à l'adresse \$200 + X
0309-	F0 05	BEQ	\$0310	Si le résultat est 0, saut à \$310
030B-	9D 00 20	STA	\$2000,X	Sinon écriture à l'adresse \$2000 + X
030E-	D0 F5	BNE	\$0305	Si la dernière comparaison n'a pas donné l'égalité...
0310-	86 06	STX	\$06	La longueur de la ligne est dans X... écriture en \$06
0312-	60	RTS		Retour au Basic.
0313-	20 2C D5	JSR	\$D52C	Entrée de l'anagramme
0316-	A0 FF	LDY	£\$FF	] Initialisation de Y
0318-	C8	INY		
0319-	A2 FF	LDX	£\$FF	] Même combat avec le registre X
031B-	E8	INX		
031C-	B9 00 02	LDA	\$0200,Y	Lecture d'un caractère...
031F-	DD 00 20	CMP	\$2000,X	... et comparaison avec la liste de base
0322-	D0 07	BNE	\$032B	S'il n'y a pas égalité, saut en \$32B
0324-	A9 00	LDA	£\$00	Sinon A est mis à 00...
0326-	9D 00 20	STA	\$2000,X	... pour être transféré dans la case-mémoire ainsi annulée
0329-	F0 09	BEQ	\$0334	Saut en \$334
032B-	E4 06	CPX	\$06	A-t-on eu tous les caractères ? (nombre en \$6)
032D-	D0 EC	BNE	\$031B	Si la réponse est non, encore un tour !
032F-	A9 00	LDA	£\$00	Sinon on a terminé... mais l'anagramme...
0331-	85 07	STA	\$07	... n'en est pas une
0333-	60	RTS		Retour au Basic
0334-	C4 06	CPY	\$06	Finis ou non ?
0336-	D0 E0	BNE	\$0318	Non ! alors, ça repart !
0338-	A9 01	LDA	£\$01	Oui ! parfait... et le 1, écrit à l'adresse \$07 va renseigner le Basic....
033A-	85 07	STA	\$07	... enfin retrouvé !
033C-	60	RTS		

- On notera que cette routine ne matérialise pas les variables. Pour le faire, une petite transformation s'imposerait (des solutions existent dans d'autres programmes de la revue).
- Il est toujours possible d'améliorer un sous-programme en langage-machine, mais dans le cas de routines très courtes, le gain (en octets et en temps) se révèle négligeable.

# Echange (swap) de variables

Une petite routine qui peut très sensiblement accélérer vos programmes de tri. Personne ne vous oblige à recopier fidèlement la DEMO ci-dessous. Vous pouvez l'élaguer. Une simple lecture vous montrera comment...

```

100 TEXT : NORMAL : PRINT CHR$(12)
    ; CHR$(21): HOME
110 GOSUB 340: POKE 1014,0: POKE 101
    5,3
120 A$(1) = "BERNADETTE":A$(2) = "NAD
    EGE":A$(3) = "GUY"
130 PRINT "PROGRAMME EN LANGAGE MACH
    INE
    -----": PRINT : PRINT : LIST
    140,170: GOSUB 330
140 CALL - 198
150 & ,A$(1),A$(2)
160 & ,A$(3),A$(2)
170 X = X + 1: IF X < 1000 THEN 150
180 PRINT : PRINT "ENVIRON 20 SECOND
    ES": GOSUB 330
190 X = 0
200 HOME : PRINT : PRINT "PROGRAMME
    EN BASIC
    -----": PRINT : PRINT
210 LIST 220,240: GOSUB 330: CALL -
    198
220 T$ = A$(1):A$(1) = A$(3):A$(3) =
    T$
230 T$ = A$(3):A$(3) = A$(2):A$(2) =
    T$
240 X = X + 1: IF X < 1000 THEN 220
250 PRINT "ENVIRON 36 SECONDES": GOS
    UB 330
260 PRINT : PRINT "SYNTAXE : &,C$,A$
    (8)... par exemple"
270 GOSUB 330
280 PRINT : VTAB 22: PRINT "<1> MENU
    DISQUETTE <2> A REFAIRE <3> FIN"
    : GOSUB 330
290 R = PEEK ( - 16384) - 48: IF R =
    1 THEN VTAB 21: PRINT : PRINT
    CHR$(4)"RUN MENU"
300 PRINT : HOME : IF R = 2 THEN X =
    0: GOTO 130
310 IF R < > 3 THEN 280
320 END
330 CALL - 198: POKE - 16368,0: WA
    IT - 16384,128,127: POKE - 1636
    8,0: RETURN
340 FOR I = 768 TO 799: READ J: POKE
    I,J: NEXT : RETURN
350 DATA 32,190,222,32,227,223,133,1
    33,132,134,32,190,222,32,227,223,
    160,2,177,133,72,177,131,145,133,
    104,145,131,136,16,243,96
    
```

## LA LIGNE 350

0300-	20 BE DE	JSR	\$DEBE	Teste la virgule
0303-	20 E3 DF	JSR	\$DFE3	Recherche la variable
0306-	85 85	STA	\$85	] Stockage dans FORPNT des valeurs de Y et A (adresse de la variable)
0308-	84 86	STY	\$86	
030A-	20 BE DE	JSR	\$DEBE	Reteste la virgule
030D-	20 E3 DF	JSR	\$DFE3	Recherche de la seconde variable
0310-	A0 02	LDY	\$02	Y est mis à 2
0312-	B1 85	LDA	(\$85),Y	] Adresse lue dans FORPNT + Y et empilée
0314-	48	PHA		
0315-	B1 83	LDA	(\$83),Y	] Adresse lue dans VARPNT + Y et passée dans FORPNT + Y
0317-	91 85	STA	(\$85),Y	
0319-	68	PLA		] Récupération sur la pile, et échange
031A-	91 83	STA	(\$83),Y	
031C-	88	DEY		Moins 1 pour Y
031D-	10 F3	BPL	\$0312	Si pas terminé, on fait un autre tour
031F-	60	RTS		Retour au Basic

# ProDOS et VOUS

NESTOR

**S** I vous ne disposez pas encore du système d'exploitation ProDOS, hâtez-vous de réparer cette lacune et offrez-vous sans tarder la version 1.0.1. Elle présente l'avantage (non négligeable) d'être débarrassée de certains défauts de jeunesse. Nous continuerons, dans *Tremplin Micro*, à vous dévoiler les faces cachées de ProDOS, encore peu connu des bidouilleurs et autres chasseurs d'octets (mais ils mettent les bouchées doubles !).

**PREFIX** Le mot de passe pour accéder à une disquette ProDOS au début, c'est un peu agaçant, mais on s'y habitue très bien.

**Conseil :** Choisissez des préfixes courts, du genre /TM1, par exemple.

**Question :** Faut-il passer par un utilitaire pour changer le PREFIX d'une disquette ?

La réponse est non. Si le préfixe actuel est TOTO, et que vous désirez en faire LILI, il suffit de taper, dans l'ordre :

RENAME/TOTO,/LILI (pour changer), puis PREFIX/LILI (pour accéder à la disquette).

Clair ? Essayez toujours !

et le sous-sous-sous catalogue s'affiche sous vos yeux, avec son programme TEST dûment enregistré.

Parfait ! mais comme tout cela ne sert à rien, il s'agit maintenant d'en débarrasser votre accueillante disquette. Vous pensez benoîtement qu'un DELETE/TM4/ABASE va régler cette épineuse question. Erreur. L'Apple vous répond : FILE LOCKED. Clair et net, bien qu'en anglais.

Il faut absolument défaire, à l'envers, ce qui a patiemment — et avec quelle maîtrise ! — été édifié dans le bon sens, comme ceci :

- DELETETEST
- DELETE/TM4/ABASE/A1/A2
- DELETE/TM4/ABASE/A1
- DELETE/TM4/ABASE

Avouez que c'est plutôt sécurisant. Pas question d'effacer, par mégarde, comme mon arrière-arrière petit-fils, l'œuvre impérissable de mon grand-père !

Amusez-vous bien avec ProDOS, et, surtout, soyez au moins aussi sérieux que votre conseiller habituel, le sage

NESTOR

## CRÉER DES SOUS-VOLUMES

Imaginons que, un peu par vice, vous ressentiez l'impératif besoin de sauver, dans un sous-sous-sous volume intitulé A2, le programme TEST que vous a inspiré votre bon génie.

Vous avez devant vous le catalogue de la disquette /TM4 (c'est son PREFIX). Comme vous êtes une personne compliquée, vous allez successivement créer les sous-catalogues /ABASE, /A1 ... et /A2 (lequel accueillera votre programme TEST). Bien sûr, tous ces noms sont arbitraires. Voici la marche à suivre :

- CREATE/TM4/ABASE
- CREATE/TM4/ABASE/A1
- CREATE/TM4/ABASE/A1/A2
- SAVE/TM4/ABASE/A1/A2/TEST

Pour finir, vous tapez :

- PREFIX/TM4/ABASE/A1/A2
- CAT

## ADRESSE ET LONGUEUR D'UN PROGRAMME

Après avoir appelé un programme par BLOAD, vous pouvez connaître son adresse et sa longueur en tapant les lignes ci-après (en mode direct, il est plus simple de faire CATALOG... celui-ci fournissant l'adresse et la longueur des fichiers binaires... mais en HEXA) :

A = PEEK(48825) + PEEK(48826) \* 256 : L = PEEK(48840) + PEEK(48841) \* 256 : PRINT "A=" A ",L=" L

RÉPONSE : A=768, L=45, (par exemple)

# PETITS TRUCS D'ASSEMBLEUR : la numération hexadécimale

**D**ANS un programme en assembleur, les nombres sont précédés du signe \$. Cela signifie qu'ils sont exprimés en numération hexadécimale. Il est impossible de faire autrement. En effet, sur un demi-octet, on peut inscrire les valeurs 0 à 15 (vérifiez cette assertion en comptant en binaire sur 4 bits). Cela correspond à 16 valeurs différentes, zéro étant une valeur. La numération hexadécimale permet donc d'exprimer (1) :

- par un seul caractère, (0 à \$F), la valeur d'un demi-octet,
- par 2 caractères la valeur d'un octet. Toutes les valeurs qu'un octet peut prendre vont de 0 à \$FF.

En comptant ainsi, le programmeur en arrive très vite à se représenter les deux demi-octets accolés, et à "voir" la valeur de chaque bit (2). En numération décimale, ce serait impossible. C'est pour cela que l'on compte toujours en hexadécimal.

Prenons un exemple. Supposons, que dans un jeu, l'on donne alternativement la main aux blancs, puis aux noirs, chacun ayant 4 coups à jouer. Il est possible de régler cette boucle de la manière suivante :

COMPTEUR DS 1 : Je bloque un octet, que j'appelle COMPTEUR.

```
LDA £$AA
```

```
STA COMPTEUR : je lui donne la valeur $AA
```

Le dièse (ou la livre, avec le clavier AZERTY) qui précède \$AA signifie que je donne la valeur \$AA et non la valeur contenue dans l'octet d'adresse \$AA, auquel cas j'aurais écrit "LDA \$AA", sans le dièse (ou la livre). Comme \$A s'écrit 1010 (en numération binaire), l'octet COMPTEUR contient 4 bits à 1 qui alternent avec 4 bits à zéro : 10101010.

**BOUCLE LSR COMPTEUR**

```
BEQ TERMINUS
BCS BLANC
JSR JEUNOIR
JMP BOUCLE
```

**BLANC JSR JEUBLANC**

```
JMP BOUCLE
```

**TERMINUS...** suite du programme.

**JEUNOIR** = sous-programme qui fait jouer les noirs.

**JEUBLANC** = sous-programme qui fait jouer les blancs.

Chaque sous-programme se termine par RTS (=RETURN). RTS renvoie à l'instruction qui suit le JSR.

L'instruction "LSR COMPTEUR" fait glisser le contenu de l'octet COMPTEUR d'un bit vers la droite. Partant de la valeur \$AA, le bit qui sort est alternativement un 0 et un 1. Il tombe dans la retenue ou CARY. L'octet se vide petit à petit. Au bout de 8 tours, il est à zéro.

L'instruction "BEQ TERMINUS" signifie *si la valeur la plus récemment calculée, prise ou donnée, est zéro (3), se brancher à TERMINUS, sinon continuer en séquence*. Comme COMPTEUR passe à zéro après 8 tours, cette instruction limite la boucle à 8 itérations.

L'instruction "BCS BLANC" signifie *si la retenue est à 1, se brancher à BLANC, sinon continuer en séquence*. On a vu que la retenue est alternativement à zéro et à un. Cette instruction donne donc alternativement la main aux blancs et aux noirs.

Cet exemple montre comment la numération hexadécimale permet d'utiliser les bits d'un ou de plusieurs octets comme les engrenages d'un rouage. Les instructions AND ORA et EOR donnent un outillage de même nature. Elles seront illustrées par quelques exemples dans un prochain numéro de *Tremplin Micro*.

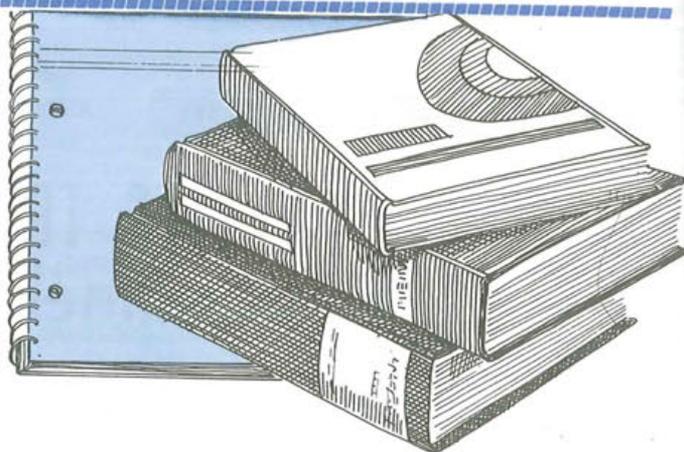
Madeleine HODÉ.

(1) Lire à ce sujet la page 44 de *Tremplin Micro* n°1.

(2) Un octet = 8 bits, bien sûr.

(3) Raccourci hardi. Il faut voir cas par cas dans la table des instructions si elles peuvent ou non modifier le bit z du registre d'état.

# Afficher un texte, empiler... dépiler



**A**FFICHER un texte à partir d'une routine en assembleur est très simple. Il suffit en effet de placer l'adresse du premier caractère de la chaîne dans les registres A et Y (à l'envers, comme le montre notre exemple, page suivante), puis d'appeler le sous-programme STROUT (\$DB3A). Et la longueur ? inutile de l'indiquer : STROUT, aussi intelligent que les auteurs de ses jours, s'arrêtera gentiment quand il rencontrera des guillemets... ou un octet mis à zéro. Ça vous explique un peu pourquoi on ne peut pas — facilement — afficher les guillemets !

## SOUS-PROGRAMMES (DE L'APPLE) UTILISÉS

Pour se brancher sur un sous-programme, on utilise la fonction JSR (code : 20)... qui correspond au GOSUB du Basic.

- \$FC58 : Home
- \$FE80 : SETINV.— Mode inverse
- \$DB3A : STROUT.— Affiche une chaîne dont l'adresse est dans A et Y, et se terminant par des guillemets ou par un 0
- \$FE84 : SETNORM.— Rétablit le mode normal
- \$F94A : PRBL2.— Sort X blancs (nombre dans registre X)
- \$FC22 : VTAB.— Va à la ligne spécifiée dans CV (position verticale du curseur)
- \$FD6F : Une entrée de GETLN sans affichage de prompt
- \$FDED : COUT.— Affiche le caractère dont le code ASCII est dans A.

### TAPEZ D'ABORD VOTRE PROGRAMME

R = RETURN

#### PARTIE PROGRAMME

```

$CALL-151 R
*300:20 58 FC 20 80 FE A9 3B A0 03 20 3A
DB 20 84 FE A2 22 20 4A F9 A9 06 85 25
20 22 FC 20 6F FD A2 00 8A 48 BD 00 02 4
8 E8 C9 8D D0 F7 86 06 A2 00 68 68 20 ED
FD E8 E4 06 D0 F7 60 R
    
```

#### PARTIE TEXTE

```

*33B:D2 C5 CE D4 D2 C5 DA A0 D6 CF D4 D2
C5 A0 D0 C8 D2 C1 D3 C5 A0 A8 CD CF C9
CE D3 A0 C4 C5 A0 B2 B5 B6 A0 C3 C1 D2 C
1 C3 D4 C5 D2 C5 D3 A9 00 R
    
```

#### CONTRÔLE

```

* CTRL-C R
$T=0:FORI=768 TO 873:T=T+PEEK(I):NEXT:PR
INT T
16210
    
```

← VOUS DEVEZ OBTENIR CE RÉSULTAT

SAUVEZ VOTRE PROGRAMME PAR BSAVE DÉPILER.LM, A\$300, L\$6A

## QU'EST-CE QUE LA PILE ?

Un espace mémoire logé en page \$01, de l'adresse \$100 à \$1FF. Si vous pratiquez (un peu) l'anglais, sachez aussi que la pile obéit à la règle *LIFO* : traduisez par **LAST IN-FIRST OUT**,

c'est-à-dire, en bon français, **DERNIER ENTRÉ-PREMIER SORTI**.

L'instruction *PHA* (48) pousse le contenu de *A* (Accumulateur) dans la pile. Le pointeur de pile est alors décrémenté de 1.

A l'inverse, à la suite d'une instruction *PLA* (68), le pointeur de pile est incrémenté de 1, et le contenu de la case mémoire ainsi pointée, passe dans *A*.

Le plus intéressant, c'est que l'on récupère les données de la pile dans l'ordre inverse de leur stockage... ce qui va nous permettre, ici, d'afficher instantanément le contenu du buffer à l'envers. Normalement, une instruction *JSR* ne modifie pas le contenu de la pile, sauf dépassement. Il ne faut jamais oublier, en effet, que la pile ne dispose que de 256 octets.

0300-	20 58 FC	JSR	\$FC58	Affichage du texte (mémorisé de \$33B à \$369), retour au mode normal, et fin de ligne avec 34 espaces (LDX = \$22, soit 34).
0303-	20 80 FE	JSR	\$FE80	
0306-	A9 3B	LDA	£\$3B	
0308-	A0 03	LDY	£\$03	
030A-	20 3A DB	JSR	\$DB3A	
030D-	20 84 FE	JSR	\$FE84	
0310-	A2 22	LDX	£\$22	
0312-	20 4A F9	JSR	\$F94A	
0315-	A9 06	LDA	£\$06	
0317-	85 25	STA	\$25	
0319-	20 22 FC	JSR	\$FC22	Curseur positionné au début de la sixième ligne d'écran.
031C-	20 6F FD	JSR	\$FD6F	
031F-	A2 00	LDX	£\$00	Initialisation de boucle de lecture Contenu de X (0) mis dans A A est empilé (au retour, ce sera le 0 final) Lecture \$200 + X Contenu empilé X est incrémenté de 1 Le dernier caractère est-il RETURN ? Si non, encore un tour X est sauvegardé dans \$6 (longueur) puis remis à zéro Ça, c'est le dernier rentré... RETURN... et voici l'avant-dernier ! Affichage Incrémentation Comparaison avec 06 (longueur) Encore un tour Vers le Basic.
0321-	8A	TXA		
0322-	48	PHA		
0323-	BD 00 02	LDA	\$0200,X	
0326-	48	PHA		
0327-	E8	INX		
0328-	C9 8D	CMP	£\$8D	
032A-	D0 F7	BNE	\$0323	
032C-	86 06	STX	\$06	
032E-	A2 00	LDX	£\$00	
0330-	68	PLA		
0331-	68	PLA		
0332-	20 ED FD	JSR	\$FDED	
0335-	E8	INX		
0336-	E4 06	CPX	\$06	
0338-	D0 F7	BNE	\$0331	
033A-	60	RTS		

Le pire, c'est que, dans le cas présent, l'utilisation de la pile est inutile... ce n'était qu'un prétexte. En effet, quand on rencontre le caractère RETURN, dans le buffer d'entrée, la longueur du texte saisi se trouve dans le registre X... et il suffit, sans aucunement faire appel à la pile, de relire le texte à l'envers, en l'affichant au fur et à mesure, comme ci-contre :

31F :	A2 00	LDX	£\$00	Lecture : quand A contient \$8D (RETURN), terminé !  X = X - 1 (élimination de RETURN) Lecture Affichage Si différent de 0 GOTO 329
321 :	BD 00 02	LDA	\$0200,X	
324 :	E8	INX		
325 :	C9 8D	CMP	£\$8D	
327 :	D0 F8	BNE	\$0321	
329 :	CA	DEX		
32A :	BD 00 02	LDA	\$0200,X	
32D :	20 ED FD	JSR	\$FDED	
330 :	D0 F7	BNE	\$0329	
332 :	60	RTS		

# LA PILE

(suite et fin)

Ce programme de démonstration vous montre comment transmettre une variable à une routine en assembleur... et comment tirer parti de la pile. D'autres formules existent. A vous de donner libre cours à votre imagination ! Notez que la ligne 120 n'a rien à voir avec la pile. Elle se contente d'afficher les 21 instructions de DEPILER2.LM, votre routine en langage machine.

```

10 TEXT : HOME : FOR I = 768 TO 808:T = T + PEEK (I): NEXT : IF T < > 5
481 THEN PRINT CHR$(4)"BLOAD DEPILER2.LM"
20 A$ = "-----zyxwvutsrqponmlkjihgfedcba0987654321ZYXWVUTSRQPONMLKJI
HGFEDCBA"
30 PRINT A$: PRINT
40 CALL 768,A$
50 GOSUB 100: CALL - 198: GOTO 40
60 B$ = "ZYXWVUTSRQPONMLKJIHGFEDCBA A l'envers, puis à l'endroit
0987654321"
70 R = 0: HOME : PRINT B$: PRINT : GOTO 90
80 CALL 768,B$
90 GOSUB 100: CALL - 198: GOTO 80
100 GET R$:R = R + 1: IF R < 8 THEN RETURN
110 X = X + 1: IF X < 2 THEN 60
120 POKE 809,169: POKE 810,0: POKE 811,133: POKE 812,58: POKE 813,169: POK
E 814,3: POKE 815,133: POKE 816,59: POKE 817,169: POKE 818,21: POKE 819
,32: POKE 820,99: POKE 821,254: POKE 822,96: HOME : CALL 809
130 PRINT : VTAB 24: PRINT "EXPLICATIONS DANS ";: INVERSE : PRINT "TREMPLI
N MICRO";: NORMAL
140 CALL - 198: POKE - 16368,0: WAIT - 16384,128: POKE - 16368,0
150 VTAB 22: PRINT : VTAB 24: PRINT "<1> MENU DE DISQUETTE <0> FINI DE JOU
ER";: GET R$: POKE - 16368,0: IF R$ = "1" THEN VTAB 22: PRINT : PRINT
CHR$(4)"RUN MENU"
160 HOME
  
```

Vous pourriez aussi écrire CALL 768, A\$(9) variable indiquée, par exemple.

## ROUTINE DEPILER2.LM

0300-	20 BE DE	JSR	\$DEBE	Pour tester la virgule après CALL 768
0303-	20 7B DD	JSR	\$DD7B	Evalue la formule et met le résultat en FAC
0306-	20 FD E5	JSR	\$E5FD	Est-ce que FAC adresse une chaîne ?
0309-	A0 00	LDY	£\$00	Initialisation registre Y
030B-	B1 5E	LDA	(\$5E), Y	En 5E et 5F, on lit l'adresse de la variable (pointeur provisoire)
030D-	48	PHA		
030E-	C9 22	CMP	£\$22	Si guillemets, fin de lecture,
0310-	F0 03	BEQ	\$0315	sinon on incrémente Y de 1 pour aller lire un autre caractère
0312-	C8	INY		
0313-	D0 F6	BNE	\$030B	
0315-	84 06	STY	\$06	Stockage de la longueur de la variable
0317-	68	PLA		Elimination des guillemets (on dépile)
0318-	A0 00	LDY	£\$00	Y remis à zéro
031A-	68	PLA		Récupération du dernier caractère...
031B-	91 5E	STA	(\$5E), Y	... qui devient le premier en écriture
031D-	C8	INY		Y est incrémente de 1
031E-	C4 06	CPY	\$06	Comparaison avec longueur connue
0320-	D0 F8	BNE	\$031A	Boucle... ou fin
0322-	20 FB DA	JSR	\$DAFB	Envoie un retour chariot
0325-	20 3D DB	JSR	\$DB3D	Affiche la chaîne
0328-	60	RTS		Retour au Basic

## ADRESSES-ÉCRAN (Valables en 40 ou 80 colonnes)

Votre écran compte 24 lignes de 40 caractères, mais elles sont numérotées de 0 à 23 (les colonnes sont elles-mêmes numérotées de 0 à 39).

• POKE 37,0 = VTAB 1 = LIGNE 0      • POKE 36,0 = HTAB 1 = COLONNE 0

Essayez ce petit programme de démonstration (revoyez aussi la partie "FENÊTRE" de notre fiche n°1).

```
10 TEXT : HOME : FOR I = 1 TO 22 : POKE 36, INT(RND(1) * 39) : POKE 37, INT(RND(1) * 23) :
   INVERSE : PRINT "" : NORMAL : NEXT
```

LIGNE	HEX	DEC	LIGNE	HEX	DEC
0	\$400 - 427	1024 - 1063	12	\$628 - 64F	1576 - 1615
1	\$480 - 4A7	1152 - 1191	13	\$6A8 - 6CF	1704 - 1743
2	\$500 - 527	1280 - 1319	14	\$728 - 74F	1832 - 1871
3	\$580 - 5A7	1408 - 1447	15	\$7A8 - 7CF	1960 - 1999
4	\$600 - 627	1536 - 1575	16	\$450 - 477	1104 - 1143
5	\$680 - 6A7	1664 - 1703	17	\$4D0 - 4F7	1232 - 1271
6	\$700 - 727	1792 - 1831	18	\$550 - 577	1360 - 1399
7	\$780 - 7A7	1920 - 1959	19	\$5D0 - 5F7	1488 - 1527
8	\$428 - 44F	1064 - 1103	20	\$650 - 677	1616 - 1655
9	\$4A8 - 4CF	1192 - 1231	21	\$6D0 - 6F7	1744 - 1783
10	\$528 - 54F	1320 - 1359	22	\$750 - 777	1872 - 1911
11	\$5A8 - 5CF	1448 - 1487	23	\$7D0 - 7F7	2000 - 2039

## AFFICHAGE PAR POKE Utilisez les valeurs suivantes :

C	INV	FLA	NOR	C	INV	FLA	NOR	C	INV	FLA	NOR
A	1	65	129	N	14	78	142	0	48	112	176
B	2	66	130	O	15	79	143	1	49	113	177
C	3	67	131	P	16	80	144	2	50	114	178
D	4	68	132	Q	17	81	145	3	51	115	179
E	5	69	133	R	18	82	146	4	52	116	180
F	6	70	134	S	19	83	147	5	53	117	181
G	7	71	135	T	20	84	148	6	54	118	182
H	8	72	136	U	21	85	149	7	55	119	183
I	9	73	137	V	22	86	150	8	56	120	184
J	10	74	138	W	23	87	151	9	57	121	185
K	11	75	139	X	24	88	152	ESP	32	96	160
L	12	76	140	Y	25	89	153				
M	13	77	141	Z	26	90	154				

LÉGENDE : C=CARACTÈRE — INV=INVERSE — FLA=FLASH — NOR=NORMAL

## INPUT PAR LA ROUTINE \$D52C (INLIN)

```

100 TEXT : PRINT CHR$ (12) ; CHR$ (21) : HOME
110 FOR I = 768 TO 813 : READ R : POKE I, R : NEXT : I = 0
120 I = I + 1 : HOME
130 CALL 768, A$(I)
140 IF A$(I) <> "" THEN 120
150 HOME : FOR J = 1 TO I : PRINT A$(J) : NEXT
160 DATA 32, 190, 222, 32, 227, 223, 133, 133, 132, 134, 165, 184, 164, 185, 133, 135, 132, 136, 32,
    44, 213, 133, 13, 133, 14, 200, 32, 237, 227, 32, 61, 231, 32, 123, 218, 165, 135, 164, 136, 133,
    184, 132, 185, 76, 183, 0
    
```

**La ligne 160 correspond à ces 46 OCTETS (\$2E)**

300 :	20	BE	DE	JSR	\$DEBE	CHKCOM : Cherche la virgule
303 :	20	E3	DF	JSR	\$DFE3	PTRGET : Lit le nom de la variable, la trouve ou la crée, en passant par CHRGET
306 :	85	85		STA	\$85	L'adresse de la variable (pointée par A et Y) est stockée dans VARPNT (\$85-\$86)
308 :	84	86		STY	\$86	
30A :	A5	B8		LDA	\$B8	On lit en \$B8-\$B9 (TXTPTR) l'adresse du dernier caractère de CHRGET. Elle est passée dans FORPNT, pointeur d'une chaîne à déplacer.
30C :	A4	B9		LDY	\$B9	
30E :	85	87		STA	\$87	INLIN : Caractères ASCII dans le buffer. Le dernier est un 0
310 :	84	88		STY	\$88	
312 :	20	2C	D5	JSR	\$D52C	A la sortie, A=0 et Y=1. A est passé dans \$0D-\$0E (caractères de fin de chaîne), puis Y est incrémenté pour valoir 02.
315 :	85	0D		STA	\$0D	
317 :	85	0E		STA	\$0E	STRLT2 : Construit un descripteur de chaîne
319 :	C8			INY		
31A :	20	ED	E3	JSR	\$E3ED	Laissons l'APPLESOFT travailler pour nous
31D :	20	3D	E7	JSR	\$E73D	
320 :	20	7B	DA	JSR	\$DA7B	Lire et écrire, à l'inverse de ce qui a été fait plus haut (lignes 30A-310)
323 :	A5	87		LDA	\$87	
325 :	A4	88		LDY	\$88	CHRGOT : C'est CHRGET, mais sans caractère. Retour au Basic.
327 :	85	B8		STA	\$B8	
329 :	84	B9		STY	\$B9	
32B :	4C	B7	00	JMP	\$00B7	

**CONTRÔLE :** T = 0 : FOR I = 768 TO 813 : T = T + PEEK (I) : NEXT : PRINT T  
**Résultat : 6160**

### EN BASIC

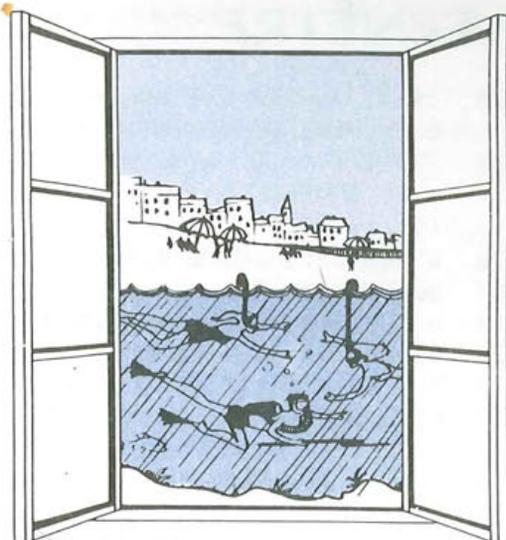
C'est simple,  
mais un peu longuet

```

100 HOME : A$ = "" : CALL — 10964
110 FOR I = 512 TO 767 : IF PEEK (I) <> 0 THEN A$ = A$ + CHR$
    (PEEK (I)) : NEXT
120 IF A$ = "" THEN 140
130 J = J + 1 : B$(J) = A$ : GOTO 100
140 HOME : FOR I = 1 TO J : PRINT B$(I) : NEXT
    
```

Cet INPUT accepte tous les caractères (y compris — attention ! — les caractères de contrôle), mais pas les guillemets. Une variable peut commencer par des espaces.

# Ouvrez donc une fenêtre !



**L** E langage machine, associé au Basic, vous autorise, avec un minimum d'octets, à ouvrir (et à refermer) une fenêtre... en un éclair !

```

10 TEXT : NORMAL : PRINT CHR$ (12); CHR$ (21): HOME
20 GOSUB 500
30 IF ( PEEK (806) + PEEK (828)) < > 192 THEN PRINT : PRINT CHR$ (4)"
   BLOAD FENETRE.LM"
40 VTAB 8: HTAB 1: CALL - 958: INPUT "VTAB DEB -> ";V
50 VTAB 10: HTAB 1: CALL - 958: INPUT "VTAB FIN -> ";VF
60 IF VF < V OR VF > 23 THEN CALL - 198: GOTO 50
70 VT = VF + 1 - V:HM = INT (255 / VT)
80 VTAB 12: HTAB 1: CALL - 958: INPUT "HTAB DEB -> ";HD
90 VTAB 14: HTAB 1: CALL - 958: INPUT "HTAB FIN -> ";HF
100 IF HF < HD OR VF > 40 THEN CALL - 198: GOTO 90
110 IF HF + 1 - HD > HM THEN VTAB 20: PRINT "MAXIMUM=255 OCTETS
   LIGNES DE "HM" CARACTERES": GOSUB 380: GOTO 80
120 GOSUB 400: HOME
130 FOR I = V TO VF: VTAB I: HTAB HD: PRINT MID$ ("12345678912345678912
   3456789123456789",HD,HF + 1 - HD): NEXT
140 CALL 768: GOSUB 380: HOME
150 VTAB 12: PRINT "RECUPERATION PAR ROUTINE EN BASIC": PRINT : GOSUB 370:
   HOME
160 P = PEEK (24):VV = V: HTAB HD: VTAB VV
170 FOR I = 8192 TO 8192 + P - 1:PS = PS + 1
180 PRINT CHR$ ( PEEK (I) - 128);: IF PS = HF + 1 - HD THEN PRINT "":PS
   = 0:VV = VV + 1: HTAB HD: VTAB VV
190 NEXT
200 GOSUB 380: HOME
210 VTAB 12: PRINT "RECUPERATION PAR ROUTINE ASSEMBLEUR": PRINT : GOSUB 37
   0: HOME
220 GOSUB 410: GOSUB 380: HOME : VTAB 12: INVERSE : PRINT "APPLICATION": N
   ORMAL : GOSUB 380
230 M$ = "<.....><.....><.....><.....><.....><.....
   ><.....><.....>"
240 V = 4:VF = 20:HD = 5:HF = 19: GOSUB 400
250 T$ = "": FOR I = 1 TO 10:T$ = T$ + "****": NEXT : FOR I = 1 TO 23: VTAB
   I: PRINT T$: NEXT
260 CALL 768

```



## I FENÊTRE (suite)

```
270 T1$ = "": FOR I = HD TO HF: T1$ = T1$ + " ": NEXT : FOR I = V TO VF: VTA
  B I: HTAB HD: PRINT T1$: NEXT
280 FOR I = 1 TO LEN (M$) STEP 11: N = N + 1: VTAB V - 1 + N * 2: HTAB HD
  + 1: INVERSE : PRINT CHR$ (N + 64);: NORMAL : PRINT SPC( 1) MID$ (M$,
  1,11): NEXT
290 GOSUB 380: GOSUB 470: REM MEMO MENU
300 GOSUB 420: REM RESTAURATION ECRAN
310 GOSUB 380: GOSUB 490: REM RESTAURATION MENU
320 GOSUB 380: ES = ES + 1: IF ES < 5 THEN 300
330 EX = EX + 1: IF EX < 2 THEN ES = 0: HOME : GOTO 300
340 HOME : VTAB 22: PRINT "<1> AUTRE FENETRE <2> FIN DE PROGRAMME": GOSUB
  380
350 HOME : IF PEEK ( - 16384) = 49 THEN RUN
360 END
370 PRINT "ENFONCEZ UNE TOUCHE POUR EXECUTION"
380 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: POKE - 16368,0:
  RETURN
390 REM LES PARAMETRES DE LA FENETRE SONT POQUES EN PAGE 0 ($6 $7 $8 $9)
400 POKE 6,V - 1: POKE 7,VF - 1: POKE 8,HD - 1: POKE 9,HF: RETURN
410 HOME
420 CALL 807: REM POUR ROUTINE ->ECRAN
430 CALL 768: REM ECRITURE ECRAN
440 CALL 807: REM POUR RESTAURATION ->MEM
450 RETURN
460 REM MEMORISATION DU MENU (APPLICATION) A PARTIR DE $2100
470 POKE 785,( PEEK (785)) + 1: POKE 831,( PEEK (831)) + 1:: CALL 768: POK
  E 785,( PEEK (785)) - 1: POKE 831,( PEEK (831)) - 1: RETURN
480 REM RECUPERATION DU MENU (APPLICATION)
490 POKE 785,( PEEK (785)) + 1: POKE 831,( PEEK (831)) + 1:: GOSUB 420: PO
  KE 785,( PEEK (785)) - 1: POKE 831,( PEEK (831)) - 1: RETURN
500 VTAB 12: PRINT "APRES AVOIR REGARDE CETTE DEMONSTRATION,lisez attentiv
  ement les instructions du programme dans TREMLIN MICRO, et ne NE-GLIGE
  Z PAS la courte routine en langage machine:"
510 PRINT : PRINT "il est probable que vous l'utiliserez unjour ou l'autre
  , dans l'un de vos petitsprogrammes, pour ouvrir et fermer... uneFENETR
  E..."
520 PRINT : PRINT " ";: GOSUB 370: RETURN
```

### TRANSMISSION DES PARAMÈTRES

On utilise les adresses \$6 à \$9 et \$18-19 de la page zéro.

\$06 : Première ligne de la fenêtre (V)

\$18 On y trouve le nombre de caractères enregistrés à partir de \$2000 (ce n'est utile que dans le programme DEMO).

\$07 : Dernière ligne de la fenêtre (VF)

\$08 : Première colonne ou HTAB (HD)

\$19 Pointeur utilisé pour le nombre de lignes (dans Fenêtre LM).

\$09 : Dernière colonne (HF)

**MÉMORISATION** Ce programme utilise l'adresse 2000, puis 2100, mais on pourrait stocker les précieux octets à la fin du programme en BASIC (en les protégeant par LOMEM).

**CONSEIL** : Tapez le programme DEMO, REMarques comprises, et efforcez-vous d'en bien comprendre le fonctionnement.

# FENÊTRE.LM

**BSAVE FENETRE.LM, A\$300, L\$42**

```

0300- A2 00 LDX £$00
0302- A5 06 LDA $06
0304- 85 19 STA $19
0306- A0 00 LDY £$00
0308- 20 C1 FB JSR $FBC1
030B- A4 08 LDY $08
030D- B1 28 LDA (<$28),Y
030F- 9D 00 20 STA $2000,X
0312- C8 INY
0313- E8 INX
0314- C4 09 CPY $09
0316- D0 F5 BNE $030D
0318- A5 19 LDA $19
031A- C5 07 CMP $07
031C- F0 06 BEQ $0324
031E- 18 CLC
031F- 69 01 ADC £$01
0321- 4C 04 03 JMP $0304
0324- 86 18 STX $18
0326- 60 RTS
0327- A2 00 LDX £$00
0329- BD 0D 03 LDA $030D,X
032C- 48 PHA
032D- BD 3D 03 LDA $033D,X
0330- 9D 0D 03 STA $030D,X
0333- 68 PLA
0334- 9D 3D 03 STA $033D,X
0337- E8 INX
0338- E0 05 CPX £$05
033A- D0 ED BNE $0329
033C- 60 RTS
033D- BD 00 20 LDA $2000,X
0340- 91 28 STA (<$28),Y
    
```

Mise à zéro du registre X (compteur)  
 On lit dans \$6 le numéro de la première ligne...  
 ... et on le charge dans la case-mémoire \$19  
 Initialisation de Y (boucle pour 1 ligne)  
 \$FBC1 va calculer l'adresse du début de la ligne  
 Y est chargé avec HTAB (début fenêtre)  
 Lecture de l'octet qui est affiché à l'adresse \$28-29 + Y  
 Ecriture à l'adresse 2000 + X  
 On incrémente Y...  
 ... ainsi que X  
 On compare Y au contenu de \$9 (fin de fenêtre)  
 S'il n'y a pas égalité, boucle  
 Accumulateur chargé avec \$19  
 Comparaison avec \$7 (dernière ligne fenêtre)  
 Si c'est terminé, on saute à \$324  
 Sinon, annulation de la retenue  
 ... pour incrémenter de 1  
 Vers une nouvelle ligne  
 X est écrit dans la case-mémoire \$18  
 Retour au Basic.

Il s'agit, ici, de remplacer les lignes 30D et 30F par les lignes 33D et 340, et vice-versa.  
 Rappelons que PHA (48) permet de mettre sur la pile et PLA (68) de récupérer l'octet qui a été empiler le dernier.

Ces instructions permettent de restaurer la fenêtre.

## POUR TAPER FENÊTRE.LM,

exécutez exactement ces opérations; quand le programme sera enregistré sur disquette, tapez CTRL-C et RETURN pour sortir du moniteur (ou CTRL-C + RESET).

**VÉRIFICATION** à effectuer en mode direct. Le résultat doit être 6500.

\$CALL-151

**RETURN**

```

*300:A2 00 A5 06 85 19 A0 00 20 C1 FB A4
08 B1 28 9D 00 20 C8 E8 C4 09 D0 F5 A5
19 C5 07 F0 06 18 69 01 4C 04 03 86 18 6
0 A2 00 BD 0D 03 48 BD 3D 03 9D 0D 03 68
9D 3D 03 E8 E0 05 D0 ED 60 BD 00 20 91
28
    
```

**RETURN**

\*BSAVE FENETRE,A\$300,L\$42

**RETURN**

```

$T=0:FORI=768 TO 833:T = T+PEEK(I):NEXT:
PRINTT
6500
    
```

**RETURN**

# & GOTO

## Goto acrobatique par le truchement de l'Ampersand

Lorsque le signe & est rencontré dans un programme en Basic, celui-ci effectue un saut inconditionnel à l'adresse \$3F5 (1013). On comprend immédiatement que ce saut ne servirait à rien si on ne plaçait pas, à cette adresse, une autre instruction de saut, mais cette fois vers une routine en langage machine.

```

1 TEXT : PRINT CHR$ (4)"PR€3": PRINT : HOME
2 PRINT "GOTO ACROBATIQUE PAR LE TRUCHEMENT DE L'AMPER
  SAND (&)": VTAB 2: GOSUB 410
3 PRINT " 10 A = 70:B = 100": PRINT " 20 T$ = " CHR$ (
  34)"TOTO" CHR$ (34): PRINT " 30 & A"
4 PRINT " 40 T$ = " CHR$ (34)"LILI" CHR$ (34): PRINT "
  50 PRINT T$"
5 PRINT " 60 IF X = 1 THEN X = X + 1: & B": PRINT " 7
  0 X = X + 1: IF X = 3 THEN END": PRINT " 80 PRINT T
  $"
6 PRINT " 90 & 40": PRINT "100 PRINT " CHR$ (34)"LIG
  NE 100" CHR$ (34): PRINT "110 & 8 * 10"
7 PRINT "$RUN": PRINT
8 GOSUB 360
10 A = 70:B = 100
20 T$ = "TOTO"
30 & A
40 T$ = "LILI"
50 PRINT T$
60 IF X = 1 THEN X = X + 1: & B
70 X = X + 1: IF X = 3 THEN 120
80 PRINT T$
90 & 40
100 PRINT "LIGNE 100"
110 & 8 * 10
120 POKE 34,2: GOSUB 410
130 POKE 32,36: POKE 33,44: VTAB 16: PRINT : INVERSE :P$
  = "ESSAYONS DE COMPRENDRE": GOSUB 420: NORMAL
140 VTAB 5: PRINT
150 P$ = "- " + CHR$ (34) + "& A" + CHR$ (34) + " = GOT
  O 70 puisque A = 70": GOSUB 420
160 VTAB 9: PRINT
170 P$ = "- ici, X vaut donc 1": GOSUB 420
180 VTAB 10: PRINT
190 P$ = "- on affiche T$ = " + CHR$ (34) + "TOTO" + CH
  R$ (34): GOSUB 420
200 VTAB 11: PRINT :P$ = "- GOTO 40, tout simplement": G
  OSUB 420

```

**GOTO** est remplacé par & et le numéro de ligne par des variables ou par des nombres.

Ce programme de démonstration vous montre, par exemple, comment modifier la fonction GOTO... et surtout, comment ça marche.

**Attention !** une carte 80 colonnes est indispensable pour réussir un affichage correct.

Il peut être intéressant, notamment dans un club, de mettre au point de petits programmes de DÉMO, parfois plus "parlants" que de longues explications.

Notez que l'on sauvegarde très simplement (en Basic) le contenu des adresses 1013 à 1015 (3F5 à 3F7). Cela se passe à la ligne 360. Il est aussi facile de restituer à l'Apple ses paramètres d'origine, dès qu'ils ne sont plus utilisés (ligne 400).

Ces pokes constituent à eux seuls le programme en langage machine, implanté à partir de l'adresse 768.

Bruit familier aux lecteurs de T.M. ... et raser quand on se repasse le programme !

```

210 VTAB 6: PRINT :P$ = "- T$ y devient " + CHR$(34) +
    "LILI" + CHR$(34): GOSUB 420
220 VTAB 7: PRINT :P$ = "- et on l'affiche": GOSUB 420
230 VTAB 8: PRINT :P$ = "- X vaut 2 ...& B = GOTO 100":
    GOSUB 420
240 VTAB 12: PRINT :P$ = "- " + CHR$(34) + "LIGNE 100"
    + CHR$(34) + " s'imprime": GOSUB 420
250 VTAB 13: PRINT :P$ = "- GOTO 80 (8 * 10)": GOSUB 420
260 VTAB 10: PRINT :P$ = "- T$ est devenu " + CHR$(34)
    + "LILI" + CHR$(34) + " depuis la ligne 40": GOSUB
    420
270 VTAB 11: PRINT :P$ = "- cette fois, c'est un GOTO 40
    ": GOSUB 420
280 VTAB 6: PRINT :P$ = "- T$ y est toujours " + CHR$(34) +
    "LILI" + CHR$(34): GOSUB 420
290 VTAB 7: PRINT :P$ = "- et on l'affiche encore une fo
    is": GOSUB 420
300 VTAB 8: PRINT :P$ = "- et X ne vaut pas 1, mais bien
    2...": GOSUB 420
310 VTAB 9: PRINT :P$ = "- et maintenant 3... c'est la f
    in (END)": GOSUB 420
320 GOSUB 400
330 VTAB 18: PRINT : PRINT "<1> TOUT RECOMMENCER <2> FIN
    (HOME) "; GET R$: IF R$ = "2" THEN TEXT : HOME :
    END
340 IF R$ = "1" THEN RUN
350 POKE - 16368,0: GOTO 330
360 FOR I = 1013 TO 1015:M(I - 1012) = PEEK (I): NEXT
370 POKE 1013,76: POKE 1014,0: POKE 1015,3
380 POKE 768,32: POKE 769,123: POKE 770,221: POKE 771,32
    : POKE 772,82: POKE 773,231: POKE 774,32: POKE 775,26
    : POKE 776,214: POKE 777,144: POKE 778,3
390 POKE 779,76: POKE 780,65: POKE 781,217: POKE 782,162
    : POKE 783,90: POKE 784,76: POKE 785,18: POKF 706,212
    : RETURN
400 FOR I = 1013 TO 1015: POKE I,M(I - 1012): NEXT : RET
    URN
410 FOR I = 1 TO 10: PRINT "_____"; NEXT : PRINT "":
    RETURN
420 FOR I = 1 TO LEN (P$): PRINT MID$(P$,I,1); FOR J
    = 1 TO 10:BUZZ = PEEK (49200): NEXT : NEXT : PRINT
    ""
430 RR = RR + 1: IF RR = 4 OR RR = 7 OR RR = 9 OR RR = 11
    OR RR = 14 THEN 450
440 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127:
    POKE - 16368,0: RETURN
450 AF = AF + 1: POKE 32,0: VTAB AF + 16: HTAB 1: INVERSE
    : ON AF GOTO 460,480,470,480,480
460 PRINT "TOTO": GOTO 490
470 PRINT "LIGNE 100": GOTO 490
480 PRINT "LILI"
490 NORMAL : POKE 32,36: PRINT : GOTO 440

```

# Utilisation de la page 2 (TEXT) pour stocker un écran

La PAGE 1 de votre Apple est utilisée pour l'affichage de la page TEXT, mais il existe une PAGE 2, capable de remplir le même rôle, mais habituellement occupée par le début du programme en Basic. En effet, la PAGE 1 occupe l'espace-mémoire \$400-7FF (1024-2047) et la PAGE 2 l'espace-mémoire \$800-BFF (2048-3071). Il s'agit donc, dans un premier temps, d'obliger votre cher APPLE à mémoriser le programme BASIC à partir de l'adresse 3073... ce que fait la ligne 10 de notre DEMO. Tromper l'ennemi (ou le meilleur ami) en somme ! Quel usage tirer de cette fameuse page 2 ? Cela dépendra de votre imagination.

```

10 TEXT : NORMAL : PRINT CHR$(12); CHR$(21): HOME
   : HGR2 : POKE 49234,0: REM ECRITURE MASQUEE DE LA
   PAGE INFOS
20 IF PEEK(103) + PEEK(104) * 256 < > 3073 THEN
   POKE 103,1: POKE 104,12: POKE 3072,0: PRINT CHR$(4)
   "RUN PAGE1.2"
30 C$ = "TAB": IF PEEK(-637) < 255 THEN C$ = "CTRL
   -I"
40 GOSUB 280: GOSUB 190
50 T$ = "": TEXT : VTAB 23: PRINT " PAGE D'INFORMATI
   ONS EN TAPANT "C$": VTAB 22: FOR I = 1 TO 10:T$ = T$
   + "----": NEXT I: PRINT T$: POKE 34,0: POKE 35,21
60 HOME
70 GET R$: CALL 774,R$:R = ASC(R$): IF R > 31 AND R
   < 127 THEN PRINT R$;
80 IF R = 8 AND PEEK(36) < > 0 THEN POKE 36, PEEK
   (36) - 1: GOTO 70
90 IF R = 21 AND PEEK(36) < 39 THEN POKE 36, PEEK
   (36) + 1: GOTO 70
100 IF R = 11 AND PEEK(37) < > 0 THEN POKE 37, PEE
   K(37) - 1: CALL -990: GOTO 70
110 IF R = 10 AND PEEK(37) < 20 THEN POKE 37, PEEK
   (37) + 1: CALL -990
120 IF R = 2 THEN INVERSE
130 IF R = 1 THEN NORMAL
140 IF R = 6 THEN END
150 IF R = 22 THEN POKE 35,24: VTAB 22: HTAB 1: INVER
   SE : PRINT "<0> POUR HOME ";: NORMAL : GET S$: PRIN
   T " ": IF S$ < > "0" THEN VTAB 22: PRINT T$: VTAB
   1: POKE 35,21:R = 3
160 IF R = 22 THEN HOME
170 GOTO 70
190 INVERSE : FOR I = 1 TO 22: VTAB I: HTAB 1: PRINT " ";

```

← Permet d'écrire sans être vu par les petits curieux !

← Attention ! quand vous taperez RUN, vous irez lire le programme PAGE1.2 sur la disquette. Sauvez votre œuvre au préalable !

← Quand on n'a pas la touche TAB, on tape un CTRL-I. C'est pareil !

Cette partie du programme n'est pas vraiment bidon... et il y manque peu de chose pour en faire une routine de saisie d'écrans, mais ce n'est qu'une DEMO, ne l'oubliez pas.

## Ecriture en PAGE2

Vous aurez certainement une idée pour une application plus personnelle...

Paramètres pour transfert PAGE1 vers PAGE2 →

POKES OR NOT POKES ? THAT IS THE (MAUVAISE) QUESTION...

```
200 IF I = 1 OR I = 22 THEN FOR J = 2 TO 39: HTAB J:
    PRINT " ";: NEXT
210 HTAB 40: PRINT " ": NEXT : NORMAL
220 POKE 32,2: POKE 33,36: POKE 34,2: POKE 35,22
230 VTAB 2: HTAB 1: PRINT
240 INVERSE : READ A$: PRINT A$: NORMAL : PRINT : PRIN
    T : READ A$: PRINT A$: PRINT : PRINT : PRINT : FOR
    I = 1 TO 4: READ A$: PRINT " "A$: PRINT : NEXT
250 INVERSE : PRINT : PRINT "ET PLACE A VOTRE GRANDE I
    MAGINATION!": NORMAL
260 POKE 60,0: POKE 61,4: POKE 62,255: POKE 63,7: POKE
    66,0: POKE 67,8: CALL 768: REM TRANSFERT PAGE 1 A
    PAGE 2
270 POKE 34,0: POKE 32,0: POKE 33,40: POKE 35,24: HOME
    : RETURN
280 POKE 768,216: POKE 769,160: POKE 770,0: POKE 771,7
    6: POKE 772,44: POKE 773,254: REM TRANSFERT PAGE 1
    A 2
290 POKE 774,32: POKE 775,190: POKE 776,222: POKE 777,
    32: POKE 778,123: POKE 779,221: POKE 780,32: POKE 7
    81,253: POKE 782,229: POKE 783,160: POKE 784,0: POK
    E 785,177: POKE 786,94: POKE 787,201
300 POKE 788,09: POKE 789,208: POKE 790,9: POKE 791,17
    3: POKE 792,85: POKE 793,192: POKE 794,32: POKE 795
    ,12: POKE 796,253: POKE 797,173: POKE 798,84: POKE
    799,192: POKE 800,96: RETURN : REM TEST CTRL-I ET A
    FFICHAGE PAGE2
310 DATA RESUME DES COMMANDES,Utilisez les FLECHES d
    e l'APPLE,- CTRL-A = Mode NORMAL,- CTRL-B = Mode IN
    VERSE,- CTRL-F = Fin de programme,- CTRL-V = Vider
    l'ECRAN
```

## • LES LIGNES DE POKES (280 à 300)

Ces trois instructions utilisent les paramètres transmis par les POKES de la ligne 260 (nous avons déjà utilisé cette méthode à plusieurs reprises).

0300-	D8	CLD	Annulation du mode décimal éventuel
0301-	A0 00	LDY ££00	Registre Y mis à 0
0303-	4C 2C FE	JMP \$FE2C	MOVE existe... pourquoi s'en priver ?

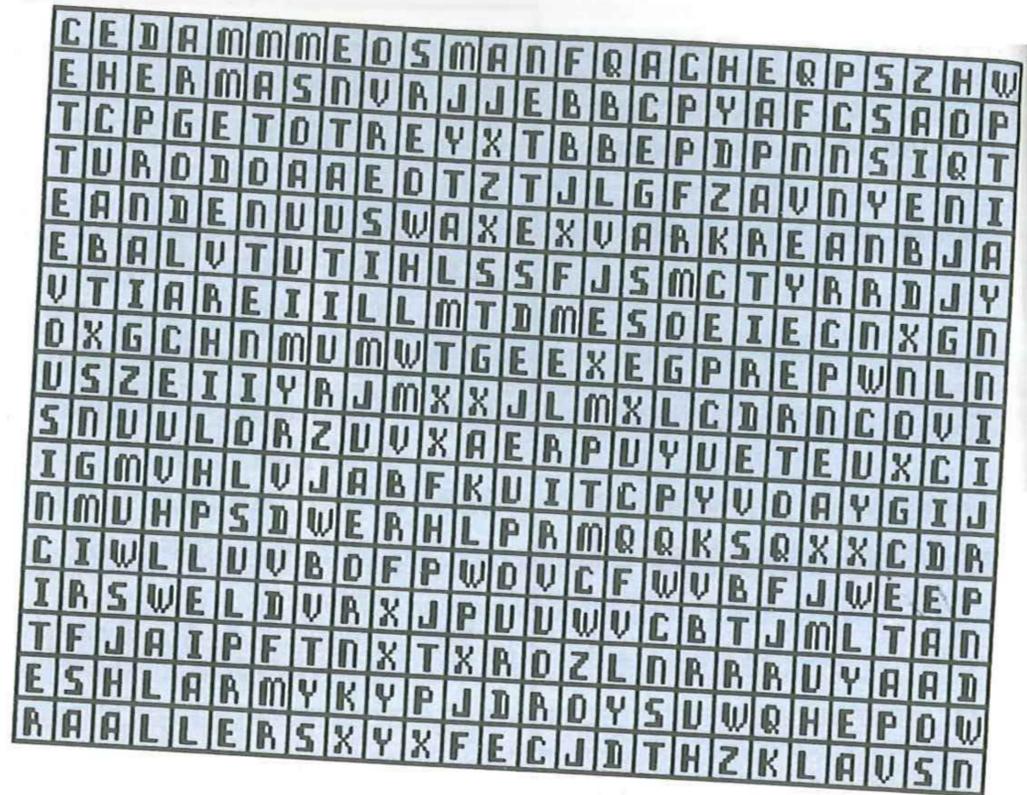
Interprétation du caractère saisi par GET, mais notre routine est prévue pour un INPUT normal (voir la pile, page 40).

0306-	20 BE DE	JSR \$DEBE	Y a-t-il une virgule après CALL 768 ?
0309-	20 7B DD	JSR \$DD7B	Evaluation de la variable ; Résultat en FAC
030C-	20 FD E5	JSR \$E5FD	Est-ce bien une chaîne ?
030F-	A0 00	LDY ££00	Y mis à zéro
0311-	B1 5E	LDA (£5E),Y	Adressage indirect (adresse lue en 5E-5F+Y)
0313-	C9 09	CMP ££09	Est-ce CTRL-I ?
0315-	D0 09	BNE \$0320	Non ? retour au Basic
0317-	AD 55 C0	LDA \$C055	Oui... alors POKE-16299,0 (PAGE2)
031A-	20 0C FD	JSR \$FD0C	Correspond au GET du Basic
031D-	AD 54 C0	LDA \$C054	Equivalent de POKE-16300,0 (PAGE1)
0320-	60	RTS	Vers programme Basic

# Logogrille\*

## LA PHRASE CACHÉE

**F**AITES une farce à vos amis en leur envoyant cette étrange grille, à charge, pour eux, d'y découvrir le message que votre Apple y aura dissimulé... à votre instigation bien sûr !



## LE PROGRAMME SE COMPOSE DE

LOGOGRILLE (en Basic) : pages 50 à 53

LOGO-CAR (Fonte élémentaire) : pages 53 et 54

LOGO.EGILOR (Embryon de fonte) : page 55

```

100 LOMEM: 16385
105 TEXT : NORMAL : PRINT CHR$ (12); CHR$ (21): HOME
    : HGR
110 G = 6500: GOTO 490
115 :
120 REM *****
125 REM *** AFFICHAGE AVEC BRUIT ***
130 REM *****
135 :
140 FOR I = 1 TO LEN (T$): PRINT MID$ (T$,I,1);: FOR
    J = 1 TO 10:X = PEEK (49200): NEXT : NEXT : RETUR
    N
145 :
150 REM *****
155 REM *** PAGE GRAPHIQUE ***
160 REM *****
165 :
170 MS = 0:MT = 0: HOME : TEXT : CALL 62450
175 FOR I = 0 TO 187 STEP 11: HPLOT 0,I TO 275,I: NEXT
180 FOR I = 0 TO 275 STEP 11: HPLOT I,0 TO I,187: NEXT
185 POKE - 16297,0: POKE - 16302,0: POKE - 16300,0:
    POKE - 16304,0
190 D = 3:IN = 11:T = D:S = - 8:SL = 0:TL = 0
195 LA = 25:HA = 17: FOR I = 6000 TO 5999 + (LA * HA):
    POKE I,0: NEXT
200 I = 1:S = S + IN: GOTO 285
205 I = I + 1: IF I > LEN (T$) THEN 330
210 N = N + 1: IF N > 1 AND AL < > D THEN N = 0: GOTO
    220

```

← LOMEM protège la totalité du programme, ainsi que la page graphique

← G = adresse de chargement de la fonte EGILOR.

← Pour abrégier le temps d'affichage (avec bruit), modifiez la seconde boucle : FOR J = 1 TO 5, par exemple.

← CALL 62450 efface la page dans laquelle on écrit (ici : HGR). Les lignes 175 et 180 tracent la grille.

← HGR sans effacement et sans les lignes de texte.

← La grille est simulée, en mémoire, par une série de POKES.

\* LOGOGRILLE est une marque déposée.

On n'explique pas cet imbroglia, mais il n'est pas interdit de chercher de meilleurs algorithmes, bien au contraire !

Ces deux lignes évitent les blocages ; on efface tout et on recommence !

Les 26 caractères de la fonte sont numérotés de 1 à 26.

Donc A=1 (ASC 65-64)

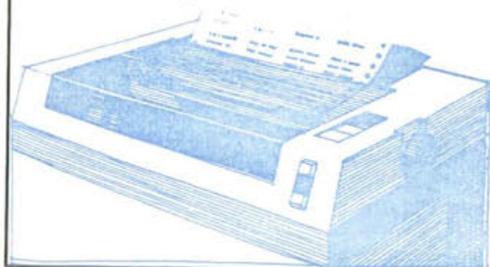
```

215 IF AL < > 0 THEN 225
220 AL = 1 + INT ( RND (1) * 7) : IF AL > 4 THEN AL = A
    L - 3
225 IF S > = SL AND ( INT ( RND (1) * 3) = 0) THEN AL
    = 4
230 MS = S : MT = T
235 ON AL GOTO 240,250,260,270
240 IF S > 256 THEN 280
245 S = S + IN : GOTO 285
250 IF S < 14 OR T > 168 OR T < TL THEN 280
255 S = S - IN : GOTO 285
260 IF T > 168 THEN 280
265 T = T + IN : GOTO 285
270 IF T < 14 OR SL > S THEN 280
275 T = T - IN : GOTO 285
280 AL = 0 : GOTO 220
285 H = (S + 8) / IN : V = (T + 8) / IN : AD = V * LA + H : A
    D = 5974 + AD
290 IF MS = 267 AND MT = 3 AND PEEK (AD + LA) < > 0
    THEN 170
295 IF MS = 267 AND MT > 168 AND PEEK (AD - LA) < >
    0 THEN 170
300 IF PEEK (AD) < > 0 THEN S = MS : T = MT : GOTO 220
305 A = ASC ( MID$ (T$,I,1))
310 POKE AD,A
315 IF S > SL THEN SL = S
320 IF T > TL THEN TL = T
325 DRAW A - 64 AT S,T : GOTO 205
330 S = - 8 : T = 3
335 FOR I = 6000 TO 6424 : S = S + IN : IF S = 278 THEN S
    = 3 : T = T + IN
340 A = PEEK (I) : IF A = 0 THEN A = 65 + INT ( RND (1)
    ) * 26)
345 DRAW A - 64 AT S,T : NEXT
350 :
355 REM *****
360 REM *** IMPRESSION ***
365 REM *****
370 :

```

(SUITE PAGE 52)

## LES PROBLÈMES d'imprimante



Il est impossible, dans le cadre d'une revue spécialisée, de publier des programmes et des conseils concernant toutes les imprimantes du marché.

Les compatibilités, quand elles existent, sont souvent sujettes à caution.

Exigez de votre fournisseur une documentation complète — et en français — et demandez à voir fonctionner la bête avant tout achat.

Nous nous intéresserons par contre à *Image Writer*, la plus vendue, actuellement, aux possesseurs d'Apple.

# Logogrille (suite)

```

375 : PR# 1: PRINT CHR# (9)"G": PR# 0
380 POKE - 16301,0
385 HOME : PRINT : VTAB 22: PRINT "<1> IMPRESSION DE L
A REPOSE <2> NON ";: GET R#
390 ON R# < > "1" AND R# < > "2" GOTO 385: PRINT R#
395 IF R# = "1" THEN POKE - 16302,0: GOTO 405
400 GOTO 430
405 POKE - 16302,0: PRINT : PR# 1: PRINT CHR# (27)"n
";: PRINT CHR# (27)"L020"
410 C = 0: FOR I = 6000 TO 6424:C = C + 1: IF C = 26 TH
EN PRINT :C = 1
415 A = PEEK (I): IF A = 0 THEN A = 32
420 PRINT CHR# (A);: NEXT : PRINT
425 PR# 0
430 HOME : TEXT : END
435 :
440 REM *****
445 REM *** INSTALLATION FONT ***
450 REM *****
455 :
460 POKE 232,G - INT (G / 256) * 256: POKE 233, INT (
G / 256): RETURN
465 :
470 REM *****
475 REM *** LOGOGRILLE ***
480 REM *****
485 :
490 HCOLOR= 3: ROT= 0: SCALE= 1
495 PRINT CHR# (4)"BLOAD LOGO.EGILOR,A"G
500 GOSUB 460:S = 40
505 T# = "4525263441": FOR I = 1 TO 10: DRAW VAL ( MID
# (T#,I,1)) AT S,50:S = S + 19: IF I = 6 OR I = 7 T
HEN S = S - 3
510 NEXT :S = S - 19:G = 7200
515 PRINT CHR# (4)"BLOAD LOGO.CAR,A"G: SPEED= 50
520 FOR I = 10 TO 1 STEP - 1: XDRAW VAL ( MID# (T#,I
,1)) AT S,50:S = S - 19: IF I = 8 OR I = 7 THEN S =
S + 3
525 NEXT :S = 0: SPEED= 255: GOSUB 460
530 :
535 REM *****
540 REM *** SAISIE DE LA PHRASE ***
545 REM *****
550 :
555 TEXT : POKE 32,11: POKE 33,20: PRINT : INVERSE : F
OR I = 1 TO 3: PRINT " ": NEXT : C
ALL - 998: CALL - 998: PRINT " LA PHRASE CACHEE "
: NORMAL : CALL - 926: TEXT
560 T# = "": FOR I = 1 TO 10:T# = T# + "____": NEXT : N
ORMAL : PRINT T#

```

Avec une interface améri-  
caine (vendue par BIP), il  
suffit de taper **CHR\$(9) "G"**  
pour obtenir la copie de la  
page HGR. Hélas ! pas de  
slot pour cette interface sur  
l'Apple IIc !

## Notez que :

- **POKE — 16301,0** permet de visualiser les 4 dernières lignes de texte de la page HGR (21 à 24).
- **POKE — 16302,0** restitue la page graphique entière (sans effacer le texte d'ailleurs).

← **232 et 233 (\$E8 et \$E9)** : l'Apple lit ici l'adresse du début de votre table de formes.

← Ligne d'initialisation obligatoire.

← Tracé du titre LOGOGRILLE en gros caractères (nullement obligatoire).

← Effacement de ce même titre. On se fait plaisir, tout simplement !

CALL—912 provoque le déroulement des lignes de l'écran texte vers le haut (SCROLL).

Permet la saisie de la ponctuation (inutile ici).

```

565 PRINT : PRINT :T$ = "TAPEZ CI-DESSOUS UNE PHRASE (
    UN MESSAGE)D'ENVIRON 150 LETTRES, UNIQUEMENT EN MA-
    JUSCULES, ET SANS PONCTUATION (ELLE SERA EN TOUT CAS
    ELIMINEE)."
```

```

570 GOSUB 140: FOR I = 1 TO 13: CALL - 912: FOR J = 1
    TO 10:X = PEEK (49200): NEXT : NEXT
```

```

575 VTAB 13: HTAB 1: CALL - 958: FOR I = 1 TO 40: PRI
    NT "....";: NEXT
```

```

580 VTAB 13: HTAB 1: PRINT "": CALL - 657:T$ = "": F
    OR X = 512 TO 767:A = PEEK (X): IF A = 141 THEN 59
    5
```

```

585 A = A - 128: IF A > 64 AND A < 91 THEN T$ = T$ + C
    HR$ (A)
```

```

590 NEXT X
```

```

595 VTAB 17: PRINT : GOSUB 140
```

```

600 VTAB 23: HTAB 1: PRINT "<1> D'ACCORD <2> TOUT A
    REFAIRE -> ";: GET R$: IF R$ = "2" THEN 575
```

```

605 IF R$ < > "1" THEN 600
```

```

610 GOTO 175
```

## LOGO.CAR

Cette fonte (qui n'est pas standard) est seulement composée des 26 lettres (capitales) de l'alphabet. Elle a été conçue pour ce programme.

**ENREGISTREMENT.**— Taper CALL—151, puis rentrer les valeurs, en les séparant par un espace, comme ceci :

\* 6000 : 26 00 38 00 4A 00 5B 00 et RETURN, etc.

6000-	26	00	38	00	4A	00	5B	00	60A8-	4E	00	00	6D	31	DF	37	6D
6008-	6B	00	7B	00	8A	00	99	00	60B0-	31	3F	3F	2E	4D	FE	3B	2E
6010-	AB	00	BE	00	CB	00	DB	00	60B8-	4D	4E	00	00	00	09	29	2D
6018-	EC	00	F8	00	0E	01	1F	01	60C0-	1E	37	35	37	35	BF	2D	75
6020-	30	01	40	01	52	01	63	01	60C8-	01	00	09	49	35	37	35	FF
6028-	74	01	81	01	92	01	A1	01	60D0-	13	4D	35	FF	33	2D	2D	4E
6030-	B6	01	C6	01	D3	01	00	00	60D8-	00	00	31	6D	31	DF	37	6D
6038-	29	2D	FE	3B	2E	4D	FE	3B	60E0-	1E	3F	2E	0D	15	DF	37	6D
6040-	2E	2D	35	DF	37	6D	71	01	60E8-	71	01	00	36	35	37	35	37
6048-	00	09	2D	15	1F	37	6D	3E	60F0-	35	37	2D	2D	4E	00	00	DF
6050-	3F	2E	4D	FE	3B	2E	2D	75	60F8-	29	0D	AD	DF	1F	37	6D	4D
6058-	01	00	4A	29	AD	DF	37	35	6100-	FE	FB	37	6D	4D	FE	FB	37
6060-	37	6D	11	DF	37	2D	2D	4E	6108-	6D	4D	4E	00	00	2D	2D	AD
6068-	00	00	2D	2D	AD	1F	37	6D	6110-	DF	37	6D	31	DF	37	6D	31
6070-	FE	37	6D	FE	BF	2D	2D	4E	6118-	DF	37	6D	71	01	00	00	29
6078-	00	00	2D	2D	F5	3B	2E	8D	6120-	AD	DF	37	6D	31	DF	37	6D
6080-	3F	37	35	37	2D	2D	4E	00	6128-	31	DF	37	2D	75	09	00	00
6088-	00	2D	2D	2D	DE	3B	2E	8D	6130-	2D	AD	DF	37	6D	31	3F	3F
6090-	3F	37	35	37	75	49	01	00	6138-	2E	3E	2E	4E	49	00	00	AD
6098-	29	29	2D	DE	3B	2E	4D	3A	6140-	29	AD	DF	37	6D	31	DF	37
60A0-	1F	37	6D	31	DF	37	2D	2D	6148-	2D	0D	1E	1F	77	6D	4E	00

6150-	00	00	2D	15	1F	37	6D	3E	6198-	DF	37	6D	F1	1F	0E	4E	09
6158-	3F	2E	4D	FE	3B	2E	4D	4E	61A0-	00	6D	49	FE	FB	37	6D	4D
6160-	00	00	29	29	2D	DE	3B	2E	61A8-	FE	FB	37	6D	4D	1E	1F	1F
6168-	8D	3F	4E	35	FF	13	2D	2D	61B0-	0E	0D	4E	09	00	00	6D	31
6170-	4E	00	00	00	2D	2D	F5	3B	61B8-	DF	77	0D	1E	17	0D	15	DF
6178-	2E	3E	2E	3E	2E	4E	09	00	61C0-	37	6D	71	01	00	4A	6D	31
6180-	DF	6D	31	DF	37	6D	31	DF	61C8-	DF	37	6D	F1	3F	0E	36	4E
6188-	37	6D	31	DF	37	2D	75	09	61D0-	09	00	00	2D	2D	3E	17	F5
6190-	00	00	6D	31	DF	37	6D	31	61D8-	BF	35	37	2D	2D	4E	00	

**ÉCRITURE SUR DISQUETTE** : BSAVE LOGO-CAR A\$6000, L479

**VÉRIFICATION** : Après avoir tapé la ligne suivante en mode direct, le résultat devra être **36353**

T = 0 : FOR I = 24576 TO 25054 : T = T + PEEK(I) : NEXT : PRINT T

## LOGO.EGILOR

Vous n'êtes pas obligé de titrer votre programme de cette manière... et dans ce cas, inutile de saisir cet embryon de fonte !

6000-	06	00	10	00	52	00	9F	00
6008-	C4	00	F8	00	4E	01	00	00
6010-	2D	2D	2D	2D	2D	35	3F	3F
6018-	3F	3F	3F	37	2D	0D	FE	3F
6020-	2E	6D	FE	3F	2E	6D	2D	35
6028-	3F	FF	3F	2E	6D	2D	F5	1B
6030-	1F	3F	2E	6D	FE	3F	2E	6D
6038-	4D	49	11	3F	3F	3F	3F	3F
6040-	37	2D	2D	2D	2D	2D	35	3F
6048-	3F	3F	3F	3F	6F	49	49	49
6050-	09	00	2D	2D	2D	2D	2D	35
6058-	3F	3F	3F	3F	3F	37	2D	0D
6060-	4D	49	35	FF	DB	FB	3F	2E
6068-	6D	FE	3F	2E	6D	FE	3F	2E
6070-	6D	4D	2D	2D	3E	3F	FF	FB
6078-	3F	2E	6D	4D	49	35	FF	DB
6080-	FB	3F	2E	6D	4D	49	35	3F
6088-	3F	3F	3F	3F	37	2D	2D	2D
6090-	2D	2D	35	3F	3F	3F	3F	3F
6098-	6F	49	49	49	09	00	00	49
60A0-	2D	2D	FE	3F	2E	6D	FE	3F
60A8-	2E	6D	FE	3F	2E	6D	FE	3F
60B0-	2E	6D	FE	3F	2E	6D	FE	3F
60B8-	2E	6D	FE	3F	2E	6D	3E	3F
60C0-	6F	49	01	00	2D	2D	FE	3F
60C8-	2E	6D	FE	3F	2E	6D	FE	3F
60D0-	2E	6D	FE	3F	2E	6D	FE	3F
60D8-	2E	6D	FE	3F	2E	6D	4D	49

## LOGOGRILLE

60E0-	11	3F	3F	3F	3F	3F	37	2D
60E8-	2D	2D	2D	2D	35	3F	3F	3F
60F0-	3F	3F	6F	49	49	49	09	00
60F8-	2D	2D	2D	2D	2D	35	3F	3F
6100-	3F	3F	3F	37	2D	0D	4D	49
6108-	35	FF	DB	FB	3F	2E	6D	4D
6110-	49	35	FF	DB	FB	3F	2E	6D
6118-	4D	49	35	FF	DB	FB	3F	2E
6120-	6D	4D	49	35	FF	DB	FB	3F
6128-	2E	6D	4D	49	35	FF	DB	FB
6130-	3F	2E	6D	4D	49	35	3F	3F
6138-	3F	3F	3F	37	2D	2D	2D	2D
6140-	2D	35	3F	3F	3F	3F	3F	6F
6148-	49	49	49	09	00	00	2D	2D
6150-	2D	2D	2D	35	3F	3F	3F	3F
6158-	3F	37	2D	0D	4D	49	35	FF
6160-	DB	FB	3F	2E	6D	4D	49	35
6168-	FF	DB	FB	3F	2E	6D	4D	49
6170-	35	3F	3F	3F	FF	3F	2E	6D
6178-	2D	2D	2D	35	3F	3F	3F	FF
6180-	3F	2E	6D	0D	2D	15	3F	DF
6188-	1F	3F	2E	6D	4D	29	AD	3F
6190-	DF	1B	1F	3F	2E	6D	4D	09
6198-	2D	3E	FF	DB	3F	3F	4D	49
61A0-	49	49	01	00				

**ÉCRITURE SUR DISQUETTE** : BSAVE LOGO.EGILOR, A\$6000, L420

**VÉRIFICATION** : Taper la ligne suivante, en mode direct. Résultat : 35419

T = 0:FOR I = 24576 TO 24575 + 420 : T = T + PEEK(I) : NEXT : PRINT T

# Cadres express

**N**OUS avons publié, dans le numéro 1 de *TM* (page 64), un exemple de cadre express. Les efforts conjugués d'Yvan KOENIG et du sage NESTOR me permettent de vous présenter la suite (et peut-être la fin ?) de ce chapitre. Notez que les cadres ainsi obtenus occupent tout l'écran (24 lignes).

Guy HACHETTE.

## CADRE.DEMO

```

10 TEXT : NORMAL : PRINT CHR$ (12); C
   HR$ (21); HOME : PRINT : PRINT : PRI
   NT " CADRE NUMERO 1
      -----"
20 PRINT CHR$ (4)"BLOAD CADRE"
30 CALL 768
40 VTAB 3: HTAB 4: CALL - 198: PRINT
   "UNE TOUCHE SVP "; GET R$: POKE -
   16368,0
50 HTAB 3: PRINT "CADRE BLOQUE (NUMERO
   2)": HTAB 3: PRINT "-----"
   -----"
60 PRINT ""
70 PRINT CHR$ (4)"BLOAD CADRE.BLOQUE"
80 CALL 768
90 PRINT "VOUS EVOLUEZ MAINTENANT": PR
   INT "A L'INTERIEUR DE VOTRE CADRE"
100 PRINT : PRINT "DANS UN PROGRAMME EN
   BASIC": PRINT "UTILISEZ PLUTOT CADR
   E.ROUTINE": PRINT : PRINT "POKE 6,32
   : CALL 768: CADRE INVERSE": PRINT "
   POKE 6,160 : CALL 768: EFFACEMENT":
   PRINT "POKE 6,170 : CALL 768: ASTERI
   SQUE"
110 PRINT : PRINT "%BLOAD CADRE.ROUTINE
   , POUR UN ESSAI": CALL - 998: CALL
   - 998
  
```

Dans un programme en Basic, utilisez Cadre.routine de la manière suivante :

Cadre en inverse POKE 6,32 : CALL 768

Cadre astérisques POKE 6,170 : CALL 768

Effacement (du cadre) POKE 6,160 : CALL 768

## CADRE

```

0300- A9 20 LDA £$20
0302- D0 03 BNE $0307
0304- 20 0C FD JSR $FD0C
0307- 85 06 STA $06
0309- A2 16 LDX £$16
030B- 8A TXA
030C- 20 C1 FB JSR $FBC1
030F- A5 06 LDA $06
0311- A0 27 LDY £$27
0313- 91 28 STA ($28),Y
0315- A0 00 LDY £$00
0317- 91 28 STA ($28),Y
0319- CA DEX
031A- D0 EF BNE $030B
031C- A2 27 LDX £$27
031E- 9D D0 07 STA $07D0,X
0321- 9D 00 04 STA $0400,X
0324- CA DEX
0325- 10 F7 BPL $031E
0327- 60 RTS
  
```

## CADRE BLOQUÉ

```

0300- 20 58 FC 20 0C FD 48 68
0308- 48 20 F6 FD A2 27 86 24
0310- 68 48 20 F6 FD A5 25 C9
0318- 17 D0 EC A2 28 68 9D CF
0320- 07 CA D0 FA A2 28 9D FF
0328- 03 CA D0 FA A9 02 85 22
0330- 85 20 A9 24 85 21 A9 17
0338- 85 23 20 58 FC 60
  
```

## CADRE.ROUTINE

```

0300- A2 16 LDX £$16
0302- 8A TXA
0303- 20 C1 FB JSR $FBC1
0306- A5 06 LDA $06
0308- A0 27 LDY £$27
030A- 91 28 STA ($28),Y
030C- A0 00 LDY £$00
030E- 91 28 STA ($28),Y
0310- CA DEX
0311- D0 EF BNE $0302
0313- A2 27 LDX £$27
0315- 9D D0 07 STA $07D0,X
0318- 9D 00 04 STA $0400,X
031B- CA DEX
031C- 10 F7 BPL $0315
031E- 60 RTS
  
```

# Cadre-line (en mode text)

```

100 TEXT : PRINT CHR$(12); CHR$(2
11): PRINT CHR$(4)"BLOAD CADRE.L
LINE"
110 HOME : INVERSE : PRINT " ECRAN H
LIN BASIC, PUIS LANGAGE MACHINE "
: NORMAL : GOSUB 260
120 HOME : LIST 130,220: GOSUB 260
130 COLOR= 170
140 HLIN 0,39 AT 0: HLIN 0,39 AT 46:
VLIN 0,46 AT 0: VLIN 0,46 AT 39
145 :
150 GOSUB 260
160 HOME : CALL 768
170 GOSUB 260
180 POKE 769,160: CALL 768
190 GOSUB 260
200 POKE 769,255: CALL 768
210 GOSUB 260
220 POKE 769,170: CALL 768
230 GOSUB 260
240 FOR I = 806 TO 818: READ J: POKE
I,J: NEXT : HOME : CALL 806
250 END
260 CALL - 198: POKE - 16368,0: WA
IT - 16384,128,127: POKE - 1636
8,0: RETURN
270 DATA 169,0,133,58,169,3,133,59,1
69,18,76,99,254
  
```

**L** ES lignes 130 et 140 du programme ci-contre vous montrent comment tracer — instantanément, et sans se soucier de la ligne 24 — un cadre de 24 × 40.

Les mêmes lignes ont ensuite été interprétées en langage machine.

Vous obtiendrez l'effacement avec COLOR = 160.

COLOR = 255 vous est soumis... pour vous montrer les limites de ce programme.

## En langage machine :

- \$F819 (HLINE) trace une ligne horizontale dans la page GR. Il faut que :

- A contienne le numéro de la ligne (0 à 47)
- Y le numéro de la colonne-départ (0 — 39)
- \$2C le numéro de la colonne-arrivée (0 — 39)

Vous pouvez en déduire que Y est forcément inférieur à \$2C.

- \$F828 (VLIN) trace une ligne verticale.

- Y doit contenir le numéro de la colonne (0 — 39)
- A le numéro de la ligne de départ (0 — 47)
- \$2D le numéro de la ligne d'arrivée (0 — 47). A doit donc être inférieur à \$2D.

## BSAVE CADRE-LINE A\$300, L\$26

```

0300- A9 AA LDA £$AA
0302- 85 30 STA $30
0304- A9 27 LDA £$27
0306- 85 2C STA $2C
0308- A9 00 LDA £$00
030A- A8 TAY
030B- 20 19 F8 JSR $F819
030E- A0 00 LDY £$00
0310- A9 2E LDA £$2E
0312- 20 19 F8 JSR $F819
0315- A9 2F LDA £$2F
0317- 85 2D STA $2D
0319- A9 00 LDA £$00
031B- A8 TAY
031C- 20 28 F8 JSR $F828
031F- A9 00 LDA £$00
0321- A0 27 LDY £$27
0323- 4C 28 F8 JMP $F828
  
```

Accumulateur chargé avec 170 (\$AA)

Le code de la couleur est stocké à l'adresse \$30

\$27 (39) = numéro de colonne-arrivée...

... stocké dans \$2C

A est mis à zéro...

... ainsi que Y

GOTO HLINE (A et Y sont détruits)

Y est remis à zéro...

A est chargé avec \$2E (46)

GOTO HLINE (A et Y sont détruits)

A chargé, avec \$2F (47) ou \$2E...

et stocké en \$2D

A mis à zéro...

... ainsi que Y

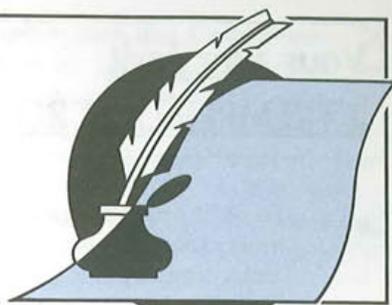
GOTO VLINE (A est détruit)

A est remis à zéro...

et Y à \$27 (39)

VLIN et retour au Basic

# Vous avez écrit à TREMPLIN MICRO



• Tout d'abord «Bonjour» et félicitations pour votre revue dont l'aspect pédagogique ne fait aucun doute. En ce sens, je me permets de vous demander conseil, car je me trouve en butte à un certain nombre de problèmes que je n'ai pu résoudre.

En voici la liste :

1 Il existe des codes écran en L.M. Mais comment afficher en Applesoft un caractère inverse **seulement** à partir du code ASCII (modifié ?), mais **sans** l'instruction INVERSE ?

2 En L.M., y a-t-il une instruction qui permette d'afficher un caractère **sans déplacer** par la suite le curseur (comme le fait, elle, l'instruction COUT1 en \$FDF0) ? Cela me permettrait de tracer des cadres de 24 x 40 caractères (cela se fait dans Disk Mucher).

Je remarque d'autre part que les deux programmes des pages 63-64 de *Tremplin Micro* n°1 ont la partie inférieure des cadres en 23<sup>e</sup> ligne et non en 24<sup>e</sup> (problème inéluctable du retour chariot et décalage vers le haut d'une ligne ?). Il faut noter page 64 une erreur dans le listing : mettre STA 06 et non LDA 06 en ligne 20, sinon la saisie ne peut être chargée en 06 ; le code correspondant étant par la suite : 306 : 85 06.

3 Comment passer du Basic au L.M. une valeur négative (par exemple poker la valeur décimale -67) ?

4 Je désire remplacer VTAB xx par POKE 37,xx-1 dans le but de traduire un programme Applesoft en L.M. car : VTAB 1 à 24, or Poke 37, 0 à 23 (comme la routine CV).

Malheureusement, il y a **disfonctionnement**, à l'image du programme qui suit :

```
10 HOME
20 FOR U = 1 TO 24
50 POKE 36,U
60 VTAB U
70 PRINT U;
80 NEXT
90 VTAB 1: GET R$
110 HOME
120 FOR U = 0 TO 23
```



```
130 POKE 36,U
140 POKE 37,U
150 PRINT U;
160 NEXT
170 VTAB 1: GET R$
```

DUFOREST Fabien (CLICHY/SOUS/BOIS)



TM

Quelle avalanche de questions !

1. Vous ne pouvez afficher en mode inverse (sans l'instruction INVERSE) que par des POKES judicieusement placés. Il est indispensable de connaître les adresses des lignes de l'écran (voir à ce sujet la fiche numéro 4 de TM).

2. Les cadres de la page 55 répondent à votre attente. On peut parfaitement tracer un cadre de 24 lignes... en BASIC, comme ceci, par exemple :

```
5 TEXT : HOME
10 FOR I = 1 TO 24: VTAB I: HTAB
  1: PRINT "*";: IF I = 1 OR I
  = 24 THEN FOR J = 2 TO 39: PR
  INT "*";: NEXT
20 IF I < > 24 THEN VTAB I: HT
  AB 40: PRINT "*";
30 NEXT : POKE 2039,170: VTAB 3:
  HTAB 2: PRINT ""
40 VTAB 3: HTAB 2: GET R$
```

3. Plus grande valeur positive, sur un octet : 127 (\$7F)

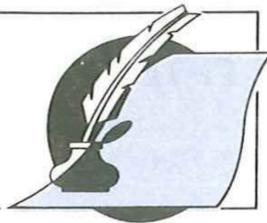
Plus grande valeur négative : -128 (\$80)

C'est le bit 7 (qui est en fait le 8<sup>e</sup>, puisque la série commence à 0), le plus à gauche, qui est le bit de signe. Il est à 0 pour un nombre positif et à 1 pour une valeur négative. Je vous conseille, sur ce point (et sur beaucoup d'autres) l'excellent livre de A.P. Stephenson : *La Programmation du 6502* (Dunod).

Pour poker -7, il suffit de faire : POKE x, 256-7 (soit 249, ou \$F9).

SUITE DU COURRIER (PAGE 58)

## Vous avez écrit à TREMPLIN MICRO



TM

Le petit programme démo ci-après devrait vous aider à mieux comprendre.

Vous transmettez un nombre positif (plus petit que 128) à une routine en L.M.

Celle-ci transforme ce nombre en sa valeur négative, puis additionne les deux... plus 1. Devinez quel est inmanquablement le résultat ?

```
10 HOME
20 INPUT C: IF C > 127 THEN 10
30 CALL - 998: PRINT C" = ";
40 CALL 768,C
50 PRINT " ADDITION DES DEUX +
  1 : ";: CALL 794: PRINT : GOT
  0 20
```

```
0300- 20 F5 E6 JSR $E6F5
0303- 8A TXA
0304- 85 06 STA $06
0306- 49 FF EOR £$FF
0308- 18 CLC
0309- 69 01 ADC £$01
030B- 85 07 STA $07
030D- 20 DA FD JSR $FDDA
0310- 18 CLC
0311- E6 06 INC $06
0313- A5 06 LDA $06
0315- 65 07 ADC $07
0317- 85 08 STA $08
0319- 60 RTS
031A- A5 08 LDA $08
031C- 20 DA FD JSR $FDDA
031F- 60 RTS
```

4. Remplacer les HTAB et VTAB par des POKES ne présente pas un grand intérêt (sauf en 80 colonnes, où il faut d'ailleurs utiliser POKE 1403,x au lieu de POKE 36,x). Votre POKE 37,U restera inopérant si vous ne placez pas, à la suite un CALL — 990 (\$FC22) qui va déplacer le curseur jusqu'à la ligne précitée.

Nestor.

• A quoi sert la touche →| ? Je vous avoue que je n'ai jamais pu l'utiliser.

• Peut-on employer, sur l'Apple IIc, le mini-assembleur de la disquette 3.3 MASTER-SYSTEM ? Si oui, comment l'extraire de la disquette ci-dessus (listez S.V.P. les instructions/commandes nécessaires) et comment le charger sur une autre dis-

quette ? Etant donné que vous avez affaire au néophyte des néophytes, soyez assez aimable pour lister en détail toutes les instructions, y compris la préparation de la disquette destination...

M.G.M. (31420 AURIGNAC)

TM

Alors là, pas d'accord ! Pourquoi ? simplement parce que vous ne ferez rien de votre mini-assembleur si vous ne consentez pas, au moins une fois, à lire les ouvrages (fort bien rédigés) qui vous ont été remis avec votre Apple IIc. Mais je vais tout de même répondre partiellement à vos deux questions :

1. La touche →| est appelée touche TAB. Vous ne pouvez la tester qu'à la suite du GET. Exemple :  
GET R\$ : IF CHR\$(9) = R\$ THEN... quelque chose  
CONTROLE I joue le même rôle que la touche TAB.
2. Pour utiliser le mini-assembleur de l'Apple, il convient de commencer par charger le BASIC INTEGER. Ensuite, procéder ainsi : INT (le prompt du BASIC INTEGER apparaît)  
CALL — 2458 (le point d'exclamation s'affiche... à vous de jouer !).

Pour le reste, consultez le livre de Nicole Bréaud-Pouliquen, *LANGAGE MACHINE ET ASSEMBLEUR DU 6502*, PSI EDITEUR.

- Dans la routine ci-après (N=850), tout se passe bien jusqu'à N=814, puis l'ordinateur ne me rend plus la main, et je suis obligé de taper un RESET

```
FOR I = 1 TO N
FOR J = 1 TO 7
K = ASC (MID$(TA$(I),J,1))
NB (K - 64) = NB (K - 64) + 1
NEXT J
NEXT I
```



TM

Votre Apple s'accorde tout simplement le temps de faire le ménage... en éliminant les valeurs successives de K... qui encombrant sa mémoire. C'est un FRE(0) automatique. Accordez-lui une bonne minute de pause (parfois davantage)... ou évitez de passer par le K intermédiaire, comme ceci :

```
FOR I = 1 TO N : FOR J = 1 TO 7 : C (ASC
(MID$ (M$(I), J, 1)) - 64) = C (ASC (MID$
(M$(I), J, 1)) - 64) + 1 : NEXT : VTAB 1 : HTAB
1 : PRINT I : NEXT
```

• Pas trop de tests de logiciels... qui foisonnent partout (quand vous les faites, donnez les prix).

• Je dispose d'un Apple IIe avec carte 80 colonnes étendue, de la souris et d'une carte super série. J'aimerais savoir comment utiliser les 64 K supplémentaires de ma carte, et j'ignore tout des possibilités de la carte super série. Je ne sais pas davantage tirer parti de mon imprimante Image Writer. Je ne crois pas être le seul à avoir ces curiosités.

• Vous avez parlé de ProCODE : je ne l'ai trouvée nulle part. Quant à ProDOS, il reste obscur pour bien des utilisateurs.

P. G. (13008 MARSEILLE)

**TM** Les deux derniers numéros de *Tremplin Micro* ont dû satisfaire certaines de vos curiosités, mais nous serions réellement des journalistes extraordinaires si, instantanément, nous nous révélions capables de vous éclairer sur tous les sujets évoqués dans votre lettre.

**Image Writer** pose en effet des problèmes à bon nombre d'utilisateurs. Sa documentation manque de clarté, bien que, comme toutes les documentations Apple, elle soit très bien réalisée.

Quand les auteurs de telles brochures comprendront que les usagers attendent plutôt des EXEMPLES, que des explications, la situation s'améliorera... et ce n'est pas vrai que pour Apple.

Il en va de même pour la carte super série (que nous ne connaissons d'ailleurs pas).

A propos des prix des logiciels, il est bon de savoir que, lorsque nous donnons le prix d'un produit et l'adresse de son fabricant (ou de son importateur), l'article est considéré comme une annonce publicitaire.

**ProCODE** est un produit de *Version-Soft* (19, rue Ganneron, 75018 Paris, Tel. 387.94.87). Une disquette Démo vous est encore proposée dans le présent numéro de *T.M.*

• Bravo ! **TREMPIN MICRO**, mais j'ai deux suggestions à vous faire :

— Pourriez-vous créer une rubrique sur le déplombage et la protection des programmes ?

— Et une autre apprenant à se servir des utilitaires connus ?

D. B. (Aix-les-Bains)

**TM** Si je comprends bien : nous expliquons comment copier un logiciel... puis comment utiliser la copie sans même photocopier la doc. Une ruse de Goupil (non, il n'y a pas d'allusion à ce-micro) ?

Mais dites-moi, ami Dany, pourquoi vouloir un

**TM** article sur la «protection des programmes» ? Vous n'allez tout de même pas protéger les vôtres... et copier ceux des autres ?

Bon, je sais : les logiciels sont chers et les étudiants (notamment) pauvres, mais il faut absolument comprendre (et même lutter) pour que d'autres jeunes — ceux qui vivent de LEURS LOGICIELS — puissent cueillir les fruits de leurs longs efforts.

D'accord ? la discussion est ouverte. Personnellement, je ne suis ni juge ni partie, mais plutôt témoin, un témoin qui, néanmoins, a sa petite idée sur la question... et vous l'expose franchement :

Tout ce temps inutilement passé à déplomber des logiciels ne serait-il pas mieux employé à tenter d'en créer d'autres, meilleurs ?

Et comme j'envie un peu les jeunes génies qui jonglent avec les précieux octets, je les crois capables, précisément dans le domaine de la création, de réaliser des choses assez extraordinaires.

Et excusez-moi pour cet accès de franchise (pas tellement commercial, mais qu'importe ?), c'est tout de même mieux que de dire du mal de son voisin, non ?

G.-H.



**microtel**

**NOUVEAU BUREAU**

Le Conseil d'Administration de la FÉDÉRATION NATIONALE des clubs MICROTEL, vient de désigner son nouveau bureau.

Auprès de M. Laurent VIROL, reconduit comme Président, on trouve Mme Marie-Josée VARLOOT (secrétaire), M. Claude PERDRILLAT (trésorier), M. Michel DAPPE (vice-président) et M. Claude PRUNET qui assure la direction technique, assisté de Jean SAQUET et d'Yves-Jacques VERNAY.

Rappelons que le Conseil d'Administration est, par ailleurs, composé de : MM. Alain BELLEVRE (ADEMIR), Bernard BLUMENTHAL (Délégué Général à la Fondation X2000), Antoine DUCRET (Fédération MICROTEL Lorraine), Thierry GAUDIN (responsable du Centre de Prospective et d'Evaluation du Ministère de l'Industrie et de la Recherche), Serge GEORGE (MICROTEL-Club Vanves), Marc LEVAILLANT (AMII), Roger MERLE (Amiposte), et Julien SPIESS (Fédération MICROTEL II<sup>e</sup> de France).

La Fédération Nationale compte, après 7 ans d'existence, plus de 15000 adhérents au travers de ses fédérations régionales et associations affiliées sectorielles.

## Vous avez écrit à TREMLIN MICRO



• J'apprécie votre revue, cependant je souhaiterais pouvoir y trouver des programmes d'enseignement, et de divertissement utile.

Lorsque je vois les jeunes ou moins jeunes utiliser le "micro" pour des jeux d'arcades et autres... qui sont des merveilles de graphisme et d'expressions musicales, je comprends qu'ils ne soient pas très intéressés par les pauvres (en graphisme, son...) programmes "EAO".

Alors, par l'intermédiaire de votre revue, ne serait-il pas possible d'avoir des programmes de production de sons, musique synthétisée, de graphisme, d'animation, etc.

D'autre part, pourriez-vous m'indiquer comment intégrer dans un programme Basic Applesoft des documents (pages d'écran...) constitués avec MOUSEPAINT et également PRINTSHOP ?

Pourriez-vous commenter plus en détail vos programmes ? Pour un débutant, comme moi, cela permettrait d'apprendre à mieux programmer.

C.S. (17250 St-PORCHAIRE)

TM

A vous (entre autres...) de montrer aux jeunes que la programmation constitue à elle seule un jeu ! Nous sommes persuadés que plusieurs Lectrices ou Lecteurs de T.M. ont déjà résolu des problèmes similaires aux vôtres, notamment en ce qui concerne MOUSEPAINT.

**Commentaires de nos programmes :** On ne conseillera jamais assez aux débutants de lire un bon abc du Basic et de l'Applesoft... et d'essayer tous les exemples. Ensuite, pour bien comprendre un programme copié dans une revue (ou dans un livre), il est nécessaire de le taper dans l'ordre où il sera exécuté, même si cette solution paraît irrationnelle. Ressasser des explications qui sont méthodiquement classées dans des recueils spécialisés ne rend pas vraiment service aux débutants. Tous les pédagogues savent qu'il est préférable d'inciter un élève à rechercher lui-même, dans le dictionnaire, l'orthographe des mots, plutôt que de la lui donner, aussi souvent qu'il la demande. Nous avons, ici, la faiblesse d'estimer nos Lecteurs et Lectrices et nous désirons les amener à aimer la programmation pour ce qu'elle est : un jeu, bien sûr, mais également un exercice intellectuel dont on reconnaîtra un jour les bienfaits.

D'aucuns vont encore m'accuser de me prendre (trop) au sérieux. Quelle erreur ! G.-H.

## GRAPH.MEN

(page 9)

Si vous manquez d'expérience, vous pouvez taper, sauvegarder et contrôler cette routine comme indiqué ci-contre.

Veillez à ne pas confondre les Ø et les O.

§CALL-151

```
*300:20 58 FC A2 2C A9 06 85 25 A9 0A 85
      24 20 22 FC BD 00 03 C9 00 F0 06 20 F0
      FD E8 D0 F3 E0 B2 F0 07 E6 25 E6 25 E8 D
      0 E1 20 E4 FB 60
```

```
*32C:A0 A0 A0 A0 A0 20 0D 20 05 20 0E 20
      15 20 00 31 A0 D3 C1 C9 D3 C9 C5 A0 C4
      C5 D3 A0 C4 CF CE CE C5 C5 D3 00 32 A0 C
      1 C6 C6 C9 C3 C8 C1 C7 C5 A0 C4 D5 A0 C7
      D2 C1 D0 C8 00 33 A0 CD C5 CD CF D2 C9
      D3 C1 D4 C9 CF CE A0 C7 D2 C1 D0 C8 00
```

```
*37A:34 A0 CD C5 CD CF A0 C4 C5 D3 A0 C4
      CF CE CE C5 C5 D3 00 35 A0 C5 C4 C9 D4
      C9 CF CE A0 C4 D5 A0 C7 D2 C1 D0 C8 00 3
      6 A0 C6 C9 CE A0 C4 C5 A0 D0 D2 CF C7 D2
      C1 CD CD C5 00 00 68 A8 68 A6 DF 9A 48
      98 48 60 00
```

BSAVE GRAPH.MEN,A\$300,L\$BE

```
§T=0:FORI=768T0768+189:T=T+PEEK(I):NEXT:
?:?"TOTAL-CONTROLE=" T
```

```
TOTAL-CONTROLE= 29649
```

CONTROLE

# INDEX des articles déjà parus

Additions binaires (p. 19).....	N° 2	Les variables dans la mémoire de l'Apple (p. 46).....	N° 3
<i>Jeu didactique pour apprendre à composer en binaire</i>		<i>Initiation à l'assembleur</i>	
Affichage accéléré en langage machine (p. 63).....	N° 1	Lignes DATA ou lignes de Pokes (p. 55).....	N° 3
Amortissement (p. 38).....	N° 1	List avec l'ampersand (p. 48).....	N° 1
Avec ProCODE (p. 52).....	N° 1	Listage et édition en langage machine (p. 14).....	N° 1
Boucles simples en langage machine (p. 50).....	N° 1	<i>Utilitaire fort pratique</i>	
Boutabout (p. 30).....	N° 3	Listage-contrôle d'un fichier «T» (p. 44).....	N° 2
<i>Routine LM permettant de mettre bout à bout deux programmes en Basic</i>		Logic-grilles (p. 57).....	N° 1
Carte langage (Fiche 3) (p. 41).....	N° 3	<i>Jeu visuel</i>	
Charabia à décrypter (p. 10).....	N° 2	Ma chère vieille pomme (p. 41).....	N° 1
<i>Jeu dont la solution a paru dans le numéro 3</i>		Majuscules en inverse (p. 32).....	N° 3
Copie d'écran avec carte 80 colonnes (p. 40).....	N° 1	<i>Une routine permettant d'afficher une ligne dont seules les majuscules sont en inverse, par Y. Koenig</i>	
Cryptotexte (p. 13).....	N° 3	Onerr Goto (p. 38).....	N° 2
<i>Pour coder ou décoder à volonté (solution du charabia du numéro 2)</i>		Ouest-Est (p. 38).....	N° 3
Créer des écrans sur 80 colonnes (p. 3).....	N° 3	<i>Petit jeu géographique en mode Text</i>	
<i>Utilitaire dont vous ferez un usage intensif</i>		Ouvrez l'œil ! (p. 52).....	N° 3
Créez vos écrans en vous amusant (p. 17).....	N° 1	<i>Jeu visuel ultra-court</i>	
<i>Raccourcissez vos programmes en mémorisant, sur la disquette, les écrans de présentation</i>		Pair ou impair avec quelques octets (p. 35).....	N° 2
'Datez vos programmes en ProDOS (p. 31).....	N° 3	<i>Une petite étude des nombres aléatoires... pour le plaisir</i>	
De bons livres, et en français (p. 61).....	N° 1	Partez en voyage avec Max (p. 12).....	N° 3
De nouvelles perspectives pour vos histogrammes (p. 30).....	N° 1	<i>Présentation du logiciel Max the Globe Trotter</i>	
<i>Un programme utilitaire indispensable</i>		Plus loin avec l'assembleur (p. 39).....	N° 2
Des octets en déplacement (p. 26).....	N° 1	Pokes utiles (p. 5).....	N° 1
<i>Utilisez mieux la mémoire auxiliaire de votre carte 80 colonnes</i>		Pour oublier le jeu du pendu (p. 32).....	N° 2
Dessine-moi un cercle (p. 47).....	N° 3	<i>Une solution : le jeu de l'homme libre !</i>	
Deux trucs pour votre Apple (p. 40).....	N° 2	Print using (p. 48).....	N° 1
Disquettes à votre nom (p. 6).....	N° 1	Programme en EXEC (Fiche 1) (p. 34).....	N° 1
<i>Personnalisez vos disquettes en DOS 3.3</i>		Programmer ? Bien sûr que cela en vaut la peine ! (p. 39).....	N° 1
Disséquer un nombre (p. 38).....	N° 2	ProDOS et la carte langage (p. 43).....	N° 3
Double listage en 80 colonnes (p. 11).....	N° 2	ProDOS et vous (Ram, Catalog, Run et Brun) (p. 44)....	N° 3
<i>Basic et langage machine</i>		Que va-t-il se passer ? (p. 54).....	N° 1
Découvrez l'assembleur avec Madeleine Hodé (p. 15).....	N° 3	Quelques notes pour votre Apple (p. 51).....	N° 3
Ecrans condensés (p. 36).....	N° 3	Saisie avec tri immédiat (p. 49).....	N° 1
<i>Mémoriser... l'essentiel et économiser les octets</i>		Saisie d'un texte, suivie de son affichage (p. 45).....	N° 3
Ecrire en page 2 (HGR2) (p. 25).....	N° 1	<i>Initiation à l'assembleur</i>	
Ecriture sur pages graphiques (p. 54).....	N° 3	Saisie LM assistée (p. 48).....	N° 3
<i>Programme de Robert Faverge, Professeur au Lycée Thiers de Marseille</i>		<i>Utilitaire destiné à vous simplifier la vie</i>	
Effacement d'écran par la gauche (p. 62).....	N° 1	Sostraction en langage machine (p. 43).....	N° 2
Evidemment qu'elle vous intéresse... la mémoire de votre Apple ! (p. 42).....	N° 1	<i>La suite de l'initiation à l'assembleur</i>	
Fichiers programmes (Fiche 1) (p. 34).....	N° 1	Spécial Bonjour (p. 3).....	N° 1
Fichiers EXEC (Fiche 1) (p. 34).....	N° 1	Tracer un cadre en langage machine (p. 64).....	N° 1
Gribouille (p. 3).....	N° 2	Transfert d'un écran de 80 colonnes en mémoire auxiliaire (Fiche 2) (p. 41).....	N° 2
<i>Présentation de l'original traitement de texte de Madeleine Hodé</i>		<i>Une fiche indispensable</i>	
Générateur de programmes (p. 33).....	N° 3	Transfert d'écrans texte (Fiche 3) (p. 42).....	N° 3
<i>Utilitaire créant automatiquement un programme de saisie de données</i>		Un accessoire utile : le magnétophone (p. 31).....	N° 2
Hexadécimal ou décimal, et vice versa (p. 44).....	N° 1	Un certain jour de la semaine (p. 12).....	N° 1
HTAB en 80 colonnes (p. 48).....	N° 1	<i>Quel jour êtes-vous né(e) ?</i>	
Imprimantes et graphisme (p. 56).....	N° 2	Un drive pour 1 500 F ? Banco ! (p. 46).....	N° 2
Imprimez des listages formatés de vos programmes (p. 27).....	N° 2	<i>Pourquoi payer votre matériel au prix fort ?</i>	
<i>Éditez correctement vos meilleurs programmes</i>		Un moyen de savoir si un fichier existe (p. 36).....	N° 1
Inversion de HGR1 ou HGR2 (p. 37).....	N° 1	Une petite révision d'anglais (p. 28).....	N° 1
Jeu (erreur) (p. 45).....	N° 1	<i>La première leçon (à vous d'écrire les suivantes !)</i>	
L'envers ne vaut pas l'endroit (p. 8).....	N° 1	Valeur ASCII après GET (p. 48).....	N° 1
<i>Jeu simple, et améliorable</i>		Visublocs (p. 21).....	N° 3
La fenêtre (Fiche 1) (p. 33).....	N° 1	<i>Un gros morceau... pour une nouvelle approche d'une représentation graphique des chiffres</i>	
La page zéro de l'Apple IIe (Fiche 1) (p. 33).....	N° 1	Vos chiffres valent bien un fromage ! (p. 49).....	N° 2
La personne X (p. 17).....	N° 3	<i>La suite logique des histogrammes, avec une certaine compatibilité</i>	
<i>Jeu de logique élémentaire</i>		Vos menus à la carte (p. 5).....	N° 2
Le curseur (Fiche 1) (p. 33).....	N° 1	<i>En Basic et langage machine... pour un affichage ultra-rapide</i>	
Les caractères «souris» de l'Apple IIc (p. 34).....	N° 2	Vous êtes au courant, mais cela vous coûte combien ? (p. 46).....	N° 1
Les fichiers aléatoires (p. 15).....	N° 2	<i>Petit utilitaire de gestion personnelle</i>	
<i>Enregistrements de taille fixe (initiation)</i>		Vous avez emprunté ? (p. 27).....	N° 1
Les nombres décimaux et votre micro (p. 16).....	N° 1	Vous dites addition ? (p. 51).....	N° 1
		<i>Initiation au langage machine</i>	



# MICROSONDAGES

Merci de bien vouloir répondre à ces questions :

1. Possédez-vous le numéro 1 de Tremplin Micro ?  OUI  NON
2. Possédez-vous le numéro 2 de Tremplin Micro ?  OUI  NON
3. Avez-vous déjà commandé une disquette ?  OUI  NON

## Bulletin de commande et d'abonnement

Nom ..... Prénom .....

Adresse complète .....

Code postal     Ville .....

LES DISQUETTES FONCTIONNENT SOUS DOS ET ProDOS  
(à condition de posséder une version Apple de ce SYSTEME D'EXPLOITATION)

1	Abonnement à 5 numéros (un an) : FRANCE	150 F
1	DISQUETTE de TREMLIN MICRO N°1 (Ile et Ilc)	105 F
1	DISQUETTE de TREMLIN MICRO N°2 (Ile et Ilc)	105 F
1	DISQUETTE de TREMLIN MICRO N°3 (Ile et Ilc)	105 F
1	DISQUETTE de TREMLIN MICRO N°4 (Ile et Ilc)	105 F
1	NUMÉRO 1 DE TREMLIN MICRO	30 F
1	NUMÉRO 2 DE TREMLIN MICRO	30 F
1	NUMÉRO 3 DE TREMLIN MICRO	30 F
1	NUMÉRO 4 DE TREMLIN MICRO	30 F
1	DISQUETTE DEMO DE ProCODE	25 F

Participation aux frais d'envoi (gratuit pour les abonnés) + 10 F

Numéro d'abonné ou de client : ..... Total à payer ..... F

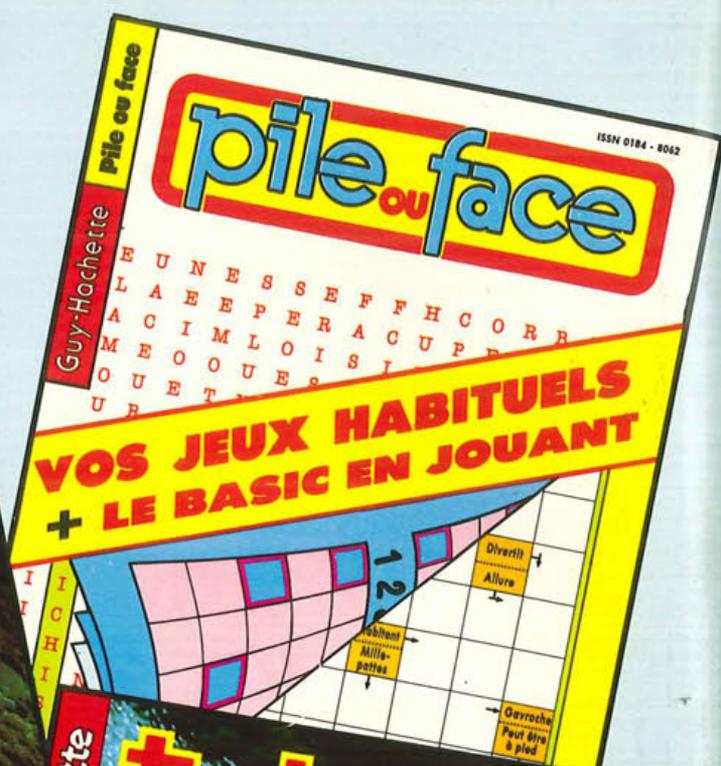
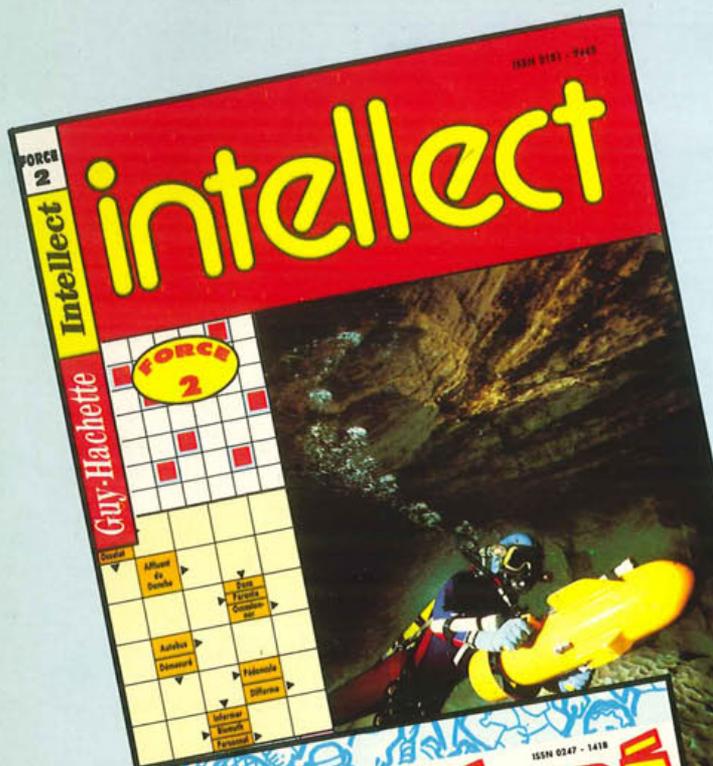
Merci de libeller votre règlement à l'ordre de TREMLIN MICRO / Editions JIBENA

Mode de règlement choisi :  Chèque  Mandat-lettre  Carte Bleue

	N° de votre Carte Bleue <input type="text"/>	Signature (obligatoire) _____
	Date d'expiration <input type="text"/>	
	Montant à débiter _____ F	

Bulletin à retourner à : TREMLIN MICRO — Senillé — 86100 CHATELLERAULT

**Guy-Hachette** pense à vos loisirs !



TREMLIN MICRO — Bimestriel — C'est une publication des Editions JIBENA, 4, rue de la Cour-des-Noues, 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.