tremplin micro

La boîte aux idées :
aide-mémoire infaillible

Mastermind :
version pour Apple

Initiation :
langage machine et basic

Graphisme en mode texte

Toujours plus avec votre Apple !

Qui Se Ressemble
S'Assemble

Chose Promise Chose Due

N°8 - Bimestriel - Deuxième année
3 Mai - 2 Juillet 1986
254 FB - 11 FS - 33 F

tremplin micro 8

BIMESTRIEL

Le numéro : 33 F
Abonnement d'un an 190 F
(6 numéros)

EDITIONS JIBENA

Direction-Rédaction :
Editions JIBENA
Guy-HACHETTE
La Petite Motte — Senillé
86100 CHÂTELLERAULT.

Téléphone :
49-93-66-66

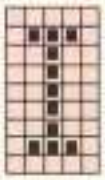
PUBLICITÉ :
Joelle (même numéro)

Commission paritaire :
Demande en cours.

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

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

Plus loin avec votre Apple



L paraît — ce sont les statistiques qui le prétendent — que des tas d'ordinateurs familiaux font une fin peu glorieuse dans les placards, où ils voisinent avec d'autres gadgets plus ou moins coûteux.

Cela ne me surprend pas outre mesure : comment employer utilement ces petites "merveilles" quand on a son content de jeux du style "envahisseurs"... et quand les possibilités de programmation sont limitées par une notoire insuffisance de la mémoire, ou encore par la lenteur de la lecture-écriture sur cassette (souvent aléatoire par-dessus le marché) ?

Votre Apple — dont les premières versions seraient assurément considérées comme des micros familiaux — figure heureusement dans une autre catégorie, celle de l'*Informatique Personnelle*. Ce qui ne l'empêche pas d'avaler, si vous le lui demandez, les logiciels de jeux. Qui peut le plus peut le moins.

Si son vieux 6502 fait aujourd'hui figure d'ancêtre, il reste — y compris dans sa nouvelle version de 65C02 — l'un des microprocesseurs les plus faciles à programmer. Les différences de rapidité que l'on note d'un Basic à un autre, se révèlent beaucoup plus difficiles à évaluer (même si elles existent) quand elles mettent en jeu des routines en langage machine.

On accomplit, notamment en matière d'affichage, de remarquables performances si l'on choisit de s'adresser directement au microprocesseur, dans le langage qu'il connaît le mieux.

Ce n'est peut-être pas une méthode très conviviale, mais c'est indiscutablement efficace... pour la grande satisfaction du programmeur.

Je n'en suis que plus à l'aise pour répondre par un oui sans aucune restriction à la question que nous posent des centaines de Lectrices et de Lecteurs : "Faut-il encore apprendre à programmer en assembleur alors que les têtes pensantes de chez Apple nous annoncent des machines capables de nous obéir au doigt, à l'œil... et à la parole ?".

Oui, le jeu — car c'est bien d'un jeu qu'il s'agit — en vaut la chandelle. Ne laissons pas l'avenir de l'informatique individuelle aux mains d'une poignée d'initiés. Nous les aimons bien, et nous apprécions comme il convient les excellents logiciels qu'ils nous mijotent, mais de grâce, qu'ils permettent à nos modestes cerveaux de fréquenter, au moins pour s'amuser, les sentiers non banalisés dans lesquels ils évoluent — souvent parce qu'ils disposent de documentations plus complètes que les nôtres — en terrains connus.

Oui, plus loin avec votre Apple, pour le plaisir ! **GUY-HACHETTE.**

Sommaire 8

LANGAGE MACHINE – INITIATION

LA BOÎTE AUX IDÉES

CATALOGUE par Guy DESENFANT ... 34

TRANSLIST par Dominique LUGAT

Ou comment créer un fichier source à partir d'une simple routine en langage machine ... 23

APPLICATION

DIMDEL par Yvan KOENIG

Ou comment changer la dimension d'un tableau en cours de programme 47

UTILITAIRE – 65C02 et IMAGEWRITER II

À CONNAÎTRE

MASTERMIND

par Michel DEVAUX

Une version pour Apple *II* 10

DES TRUCS

LISEZ AUSSI :

TREMPLIN MICRO Le numéro **9** paraîtra le **3 juillet.**

UTILITAIRE | DOS 3.3 | ProDOS

LA BOÎTE aux idées

- **Petite base de données**
- **Bloc-notes ou pense-bête**

Vous n'aurez que l'embarras du choix

POURQUOI vous priver plus longtemps du plaisir de mémoriser — en vrac — les plus farfelues de vos élucubrations ? **LA BOÎTE AUX IDÉES** est là pour vous servir, et d'une manière on ne peut plus simple, avec :
— BOÎTE-BAS, programme en Basic — BOÎTE, fichier binaire de données.

BOÎTE-BAS fonctionne sur tout Apple IIe ou IIc équipé d'une carte 80 colonnes. Une adaptation sur 40 colonnes est possible, mais l'affichage sera moins bon. La version ci-après est destinée aux Apple IIe normaux. Pour Apple IIc ou IIe équipés du 65C02, on modifiera seulement la ligne 295.
Pourquoi avoir préféré le fichier BIN au fichier TXT ? D'une part, en raison de la rapidité de son écriture sur disquette et de son chargement, d'autre part parce que le programme l'utilise sans faire appel à une variable.

LIMITE La longueur du fichier est limitée à la taille-mémoire de votre Apple. Ici, LOMEM a été fixé à 26000, ce qui autorise l'écriture de 255 données. Chaque enregistrement exigeant 80 caractères, 100 données supplémentaires vous obligeront à donner à LOMEM la valeur 34000. Veillez à ne pas dépasser la taille de la mémoire disponible... et n'oubliez pas de réserver quelques milliers d'octets pour le fonctionnement du programme (peu gourmand d'ailleurs).

PREMIÈRE UTILISATION RUN BOÎTE.BAS... puis option 1 du menu. Pour qu'une ligne de données (pas plus de 78 caractères) soit validée, il est indispensable de répondre par (1) D'ACCORD... sinon c'est à refaire.
Consultez le tableau de la page 4 avant toute utilisation. Il comporte des indications qui ne figurent (volontairement) pas dans le programme.

CRÉATION DU FICHIER

La création du fichier BOÎTE est automatique. Ne pas verrouiller ce fichier car il est entièrement réécrit lors de chaque mise à jour.
Pour le copier sur une autre disquette, taper successivement :

- RUN BOÎTE-BAS ...puis changer de disquette
- 1 (option saisie du menu)
- RETURN (immédiatement après le numéro de la donnée)
- 4 (vers menu général)... et le fichier BOÎTE est écrit sur votre seconde disquette (préalablement initialisée, bien sûr).

UTILISATION DE BOÎTE.BAS

SAISIE DES DONNÉES

Après avoir choisi l'option 1 (menu général), tapez normalement votre phrase (signes de ponctuation autorisés). A vous de savoir si les minuscules seront ou non utilisées, de même que l'accent circonflexe. La longueur de chaque donnée est limitée à 78 caractères. Plusieurs commodités vous sont offertes (en Basic) :

CTRL-D : Curseur au début de la ligne
CTRL-F : Curseur à la fin de la ligne
CTRL-E : Effacement de la ligne
CTRL-S : Récupération d'une ligne effacée par erreur
DEL : Effacement du caractère précédant le curseur

Pour insérer, se contenter de revenir en arrière, puis taper le caractère à insérer. Il prendra place derrière la lettre recouvrant le curseur.

RETURN : Pour terminer la saisie d'une ligne... (Pour terminer la saisie, RETURN sur une ligne vide).

FLÈCHE DROITE et FLÈCHE GAUCHE :
Autorisent les déplacements vers la droite et vers la gauche

LECTURE DES DONNÉES

(à partir du Menu général)
CTRL-L ... comme Liste vous permet d'accéder au fichier de données, ligne par ligne... à partir du numéro de votre choix. Une ligne de commandes vous guide (possibilité de lire la ligne suivante avec RETURN... ou de corriger après un CTRL-C).

RECHERCHE

C'est évidemment le point fort du programme. Vous pourrez rechercher un mot, un groupe de mots... voire une simple lettre, mais pas la ponctuation (le mot à chercher est en effet saisi par un INPUT normal... les programmeurs peuvent s'en donner à cœur joie et modifier cette partie de BOÎTE-BAS !).

Si vous cherchez "qui", toutes les phrases comportant ces trois lettres vont instantanément s'afficher, y compris quand "qui" sera inclus dans acquitter ou équilibre.

Par contre, si vous cherchez "qui " (qui suivi d'un espace), vous n'obtiendrez que les phrases comportant "qui ", suivi d'un espace. Les possibilités sont nombreuses.

(La disquette de *Tremplin Micro* comporte un fichier de proverbes qui en donne une bonne idée).

Et maintenant, à vous de jouer !

Clément Renard.

BOÎTE-BAS

```
100 LOMEM: 26000: REM POUR 255 DONNEES ENVIRON
105 L$ = "":R$ = "":R = 0:H = 0: REM VARIABLES LES PLUS
    UTILISEES
110 TEXT : NORMAL :D$ = CHR$ (4): PRINT D$"PR£3": PRINT
115 BA = 5500: REM ADRESSE DU FICHIER "BOITE"
120 GOSUB 680: GOSUB 465
125 INVERSE :T$ = ":                          :": VTAB 1: PRINT
    T$: PRINT " LA BOITE AUX IDEES ": PRINT T$: NORMAL
130 T$ = "": FOR I = 1 TO 20:T$ = T$ + "____": NEXT : GO
    TO 140
135 POKE 1403,40: VTAB V: INVERSE : PRINT V;: NORMAL : R
    ETURN
140 V = 1: GOSUB 135: PRINT " SAISIE D'UNE... IDEE (GENI
    ALE)"
145 V = 2: GOSUB 135: PRINT " RECHERCHE D'UNE IDEE NON M
    OINS GENIALE"
150 TEXT
155 V = 3: GOSUB 135: PRINT " TERMINE POUR AUJOURD'HUI -
    > ";: CALL - 198: GET R$: PRINT
160 R = VAL (R$)
165 PRINT T$: POKE 34,5: VTAB 22: PRINT T$: POKE 35,21
```

Ligne 115 : Si vous modifiez les commentaires en les allongeant, votre programme sera trop long et sera "écrasé" par le fichier de données (installé à partir de 5500).
Alors, soyez prudent !

POKE 1403, valeur remplace **HTAB** (qui, au-dessus de 40, n'est pas supporté par tous les Apple).

SAISIE

Elle est classique, avec *GET*, en Basic. Il serait possible de l'optimiser en la remplaçant par une routine en langage machine (une boîte de disquettes vierges à l'auteur de la meilleure !).

- Le fait d'avoir déclaré les variables L$, R$, R et H au début du programme en améliore sensiblement la rapidité.
- Sur les anciens Apple *IIe*, la touche ESCAPE ne doit pas être utilisée... car elle permet de se promener à travers l'écran (avec les flèches, bien sûr). Problème inexistant sur les *IIc* et *IIe* avec 65C02
- Avec l'Apple *IIc* et l'Apple *IIe* + 65C02, la ligne 295 sera remplacée par :

```
295 IF R = 21 THEN H =
      H + 1 : GOTO 225
```

MÉMORISATION

A partir du moment où vous avez accédé à la routine de saisie, le fichier BOÎTE sera automatiquement réécrit lorsque vous quitterez cette même routine de saisie pour l'option *MENU GÉNÉRAL*.

```
170 IF R$ = CHR$ (12) THEN 580
175 HOME
180 ON R GOTO 210,520,700: GOTO 150
185 :
190 REM ********************************
195 REM *      SAISIE DU TEXTE         *
200 REM ********************************
205 :
210 HOME :N = N + 1: INVERSE : PRINT N: NORMAL
215 L$ = " "
220 L = LEN (L$) - 1:H = L + 1: IF L > 78 THEN CALL - 19
    8:L$ = LEFT$ (L$,L - 1): GOTO 220
225 IF H - 1 > L THEN H = L - 1
230 PRINT : VTAB 9: HTAB 1: PRINT L$;: CALL - 958: POKE
    1403,H: GET R$
235 R = ASC (R$)
240 IF R > 19 THEN 275
245 IF R = 8 AND H > 1 THEN H = H - 1: GOTO 230
250 IF R = 4 THEN H = 1: GOTO 230
255 IF R = 6 THEN 220
260 IF R = 5 THEN S$ = L$: GOTO 215
265 IF R = 19 THEN L$ = S$:S$ = "": GOTO 220
270 IF R = 13 THEN 335
275 IF H - 1 = L THEN 315
280 IF R < > 127 THEN 295
285 IF H > 1 THEN L$ = LEFT$ (L$,H - 1) + RIGHT$ (L$,L +
     1 - H):H = H - 1:L = LEN (L$) - 1
290 GOTO 230
295 IF R = ASC ( MID$ (L$,H + 1,1)) THEN H = H + 1: GOTO
     230
300 IF R < 32 OR R > 126 THEN 230
305 IF L > 77 THEN 220
310 L$ = LEFT$ (L$,H) + R$ + RIGHT$ (L$,L + 1 - H):H = H
     + 1:L = LEN (L$) - 1: GOTO 230
315 IF R = 127 AND L > 0 THEN L$ = LEFT$ (L$,L): GOTO 22
    0
320 IF R < 32 OR R > 126 THEN 230
325 IF L > 78 THEN R$ = ""
330 L$ = L$ + R$: GOTO 220
335 PRINT : PRINT : PRINT L$
340 VTAB 17: PRINT "<1> D'ACCORD <2> CORRECTION <3> ANNU
    LATION <4> MENU GENERAL ";: CALL - 198: GET R$: PRIN
    T
345 R = VAL (R$): IF R < 1 OR R > 4 THEN 340
350 ON R GOTO 395,230,215,355
355 IF NM THEN N = NM:NM = 0: RETURN
360 N = N - 1
365 GOSUB 425: GOTO 150
370 :
375 REM ********************************
380 REM *     ECRITURE DU FICHIER      *
385 REM ********************************
390 :
```

→

BOÎTE-BAS (suite)

```
395 IF L$ = " " THEN N = N - 1: GOTO 210
400 AD = (BA - 80) + 80 * N:AL = AD + L:AF = AD + 79:M =
    0
405 FOR I = AD TO AL:M = M + 1: POKE I,128 + ASC ( MID$
    (L$,M,1)): NEXT
410 FOR I = AL + 1 TO AF: POKE I,0: NEXT
415 IF NM THEN N = NM:NM = 0: RETURN
420 GOTO 210
425 L = N * 80
430 PRINT CHR$ (4)"BSAVE BOITE,A"BA",L"L
435 RETURN
440 :
445 REM ********************************
450 REM *     LECTURE DU FICHIER       *
455 REM ********************************
460 :
465 ONERR GOTO 490
470 PRINT CHR$ (4)"BLOAD BOITE"
475 L = PEEK (43616) + PEEK (43617) * 256: REM 48840-41 .
    AVEC PRODOS
480 N = L / 80
485 RETURN
490 POKE 216,0: GOTO 125
495 :
500 REM ********************************
505 REM *          RECHERCHE           *
510 REM ********************************
515 :
520 AR = 848: REM STOCKAGE DU MOT CHERCHE
525 VTAB 6: INPUT "VOTRE MOT-CLE EST -> ";M$
530 IF M$ = "" THEN 150
535 L = LEN (M$): POKE 252,L: POKE 8,(N - 1) - INT (N /
    256) * 256: POKE 9, INT ((N - 1) / 256):M = 0
540 FOR I = AR TO AR + L - 1:M = M + 1: POKE I,128 + ASC
     ( MID$ (M$,M,1)): NEXT
545 PRINT : CALL 768: CALL - 198
550 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: PO
    KE - 16368,0: PRINT : HOME : GOTO 525
555 :
560 REM ********************************
565 REM *   LECTURE ET CORRECTION      *
570 REM ********************************
575 :
580 HOME : CALL - 198: PRINT : VTAB 7: INPUT "NUMERO DE
    DEPART DEMANDE -> ";R$:RE = VAL (R$)
585 PRINT : IF RE < 1 OR RE > N THEN 645
590 VTAB 7: CALL - 868: INVERSE : PRINT RE: NORMAL : VTA
    B 9: CALL - 868
```

LONGUEUR DU FICHIER

Calcul élémentaire, puisque chaque enregistrement est limité à 80 caractères. On a L = 80 * N

- Notez (ligne 410) que les lignes sont complétées par autant de 0 qu'il y a d'octets libres... jusqu'à 80.
- Le premier caractère de chaque donnée est un espace (ce n'était pas indispensable).

ONERR GOTO

Lors de la première création de fichier, il évite de se planter.

ProDOS

Sous ProDOS, ne pas oublier de modifier la ligne 475, comme indiqué par la REMarque...

RECHERCHE

On poque en 252 ($FC) la longueur du mot cherché (de 1 à FF... si vous voulez !), et en 8-9 le nombre actuel de lignes de données (qui peut donc dépasser $FF... ou 255, puisque deux octets lui sont réservés).

Quant aux caractères du mot, ils sont également poqués à partir de 848, c'est-à-dire à la suite de la routine en langage machine... qui sait les lire à cette adresse C.Q.F.D.

CORRECTION

Les copieurs de disquettes ne sauront jamais comment relire leur fichier... et le corriger, sauf s'ils ont le courage de lire le programme pour comprendre comment il fonctionne. C'est élémentaire, non ?

Et maintenant, pour en savoir plus là-dessus, lisez tout de même l'explication de cette toute petite routine de recherche.
(ci-dessous et au verso)

```
595 AD = (BA - 80) + (RE * 80): POKE 6,AD - INT (AD / 25
    6) * 256: POKE 7, INT (AD / 256)
600 POKE 8,0: POKE 9,0
605 CALL 833: REM SORTIE DE LA PHRASE
610 VTAB 18: PRINT "SUITE = <RETURN> -- CORRECTION = <CT
    RL-C> -- AUTRE OPTION = <UNE AUTRE TOUCHE> ";: CALL
    - 198: GET R$
615 PRINT :R = ASC (R$): IF R = 13 THEN RE = RE + 1: GOT
    O 585
620 IF R = 3 THEN NM = N:N = RE: GOTO 630
625 GOTO 580
630 L$ = "": FOR I = AD + 79 TO AD STEP - 1:R = PEEK (I)
    : IF R < > 0 THEN L$ = CHR$ (R - 128) + L$
635 NEXT : GOSUB 220
640 CO = 1: GOTO 580
645 IF CO = 1 THEN CO = 0: GOSUB 425
650 GOTO 150
655 :
660 REM ********************************
665 REM *    ROUTINE DE RECHERCHE      *
670 REM ********************************
675 :
680 FOR I = 768 TO 847: READ R: POKE I,R: NEXT : RETURN
685 DATA 169,044,133,6,169,21,133,7,198,8,165,8,201,255,
    208,9,198,9,165,9,201,255,208,1,96,24,165,6,105,80,1
    33,6,144,2,230,7,160,255,162,255,232,228,252
690 DATA 240,20,200,177,6,240,214,221,80,3,240,241,134,2
    53,152,56,229,253,168,76,38,3,165,6,164,7,32,58,219,
    169,13,32,237,253,76,8,3
695 :
700 TEXT : HOME
```

À SAVOIR

- $DB3A (STROUT) affiche la chaîne dont vous lui indiquez l'adresse (Y et A), mais attention ! elle doit obligatoirement se terminer par un 0 ou par des guillemets ("). La partie basse de l'adresse est mise dans l'accumulateur A et la partie haute dans le registre Y. Cela est obtenu par LDA et LDY (LD comme LOAD).
- $FDED (COUT) permet d'afficher le caractère qui est dans l'accumulateur A.

Si vous êtes néophyte, utilisez des trucs pour vous souvenir de la signification des **mnémoniques** (LDA, STA, etc.)

PAR EXEMPLE :

- LDA, LDX, LDY comme **LoaD** A, X, Y
- STA, STX, STY comme **STocke** A, X, Y
- CMP, CPX, CPY comme **ComParer** A, X, Y
- ADC comme **ADditionner**
- SBC comme **Soustraire**
- DEC comme **DECrémenter**
- DEX, DEY comme **DEcrémenter** X,Y
- INC comme **INCrémenter**
- INX et INY comme **INcrémenter** X,Y
- JSR... **Je Saute et Reviens**
- JMP... **Je Me Perds**... et je ne reviendrai pas... bref, du délire !

NESTOR

LA BOÎTE AUX IDÉES

(suite et fin)

Voici, pour terminer, l'explication de la routine de RECHERCHE (lignes 685 et 690), qui permet de trouver une lettre ou un groupe de lettres... et d'afficher la ou les lignes concernées.

DEUX POINTS D'ENTRÉE

- 768 ($300) pour la recherche proprement dite
- 833 ($341) pour le sous-programme "Routine et correction".

On utilise les adresses $6 - 7 - 8 - 9 - FC - FD de la page 0.

```
0300-  A9 2C      LDA   £$7C      ┐ $152C = 5500-80
0302-  85 06      STA   $06       │
0304-  A9 15      LDA   £$15      │ La partie basse de l'adresse est stockée dans
0306-  85 07      STA   $07       ┘ la mémoire $6, la partie haute dans la mémoire $7
0308-  C6 08      DEC   $08       ┐
030A-  A5 08      LDA   $08       │
030C-  C9 FF      CMP   £$FF      │ En $8-9, on a le nombre de lignes de données
030E-  D0 09      BNE   $0319     │ moins un (poqué depuis le Basic). On le
0310-  C6 09      DEC   $09       │ décrémente. Quand les deux valeurs passent
0312-  A5 09      LDA   $09       │ au-dessous de 0 (0 — 1 = $FF), on a terminé
0314-  C9 FF      CMP   £$FF      │ la lecture des n lignes
0316-  D0 01      BNE   $0319     ┘
0318-  60         RTS             RTS renvoie au Basic
0319-  18         CLC             ┐
031A-  A5 06      LDA   $06       │ Annulation de la retenue pour ajouter
031C-  69 50      ADC   £$50      │ $50 (80) à l'adresse contenue dans
031E-  85 06      STA   $06       │ $6-7. Quand il n'y a pas de retenue,
0320-  90 02      BCC   $0324     │ BCC évite d'incrémenter la mémoire $7
0322-  E6 07      INC   $07       ┘
0324-  A0 FF      LDY   £$FF      Registre Y à $FF
0326-  A2 FF      LDX   £$FF      Registre X à $FF
0328-  E8         INX             X = X + 1 (à noter que $FF + 1 = 0)
0329-  E4 FC      CPX   $FC       X est-il égal à $FC, (longueur du mot) ?
032B-  F0 14      BEQ   $0341     Si oui, on passe à l'affichage...
032D-  C8         INY             Non : alors Y = Y + 1
032E-  B1 06      LDA   ($06),Y   On lit l'octet contenu à l'adresse $6-7 + Y
0330-  F0 D6      BEQ   $0308     Si c'est un 0, la ligne a été lue... à une autre !
0332-  DD 50 03   CMP   $0350,X   Comparaison avec le caractère cherché
0335-  F0 F1      BEQ   $0328     S'il est identique, voyons le suivant éventuel
0337-  86 FD      STX   $FD       ┐ Valeur de X stockée dans FD
0339-  98         TYA             │ Y passe dans A
033A-  38         SEC             │
033B-  E5 FD      SBC   $FD       │ Retenue à 1 pour soustraire $FD (autrement
033D-  A8         TAY             │ dit X) de A (autrement dit Y).
033E-  4C 26 03   JMP   $0326     ┘ A repasse dans Y
                                  Et ça repart pour un tour !
```

```
0341-  A5 06      LDA   $06       ┐ Lisez l'explication page précédente
0343-  A4 07      LDY   $07       │
0345-  20 3A DB   JSR   $DB3A     │
0348-  A9 0D      LDA   £$0D      │
034A-  20 ED FD   JSR   $FDED     ┘
034D-  4C 00 03   JMP   $0300     Retour à la source !
```

■

VARIABLES dimensionnées

On saisit cinq données alphanumériques et on affiche la première

```
10   HOME
20   FOR I = 1 TO 5
30   INPUT A$(I)
40   NEXT I
50   PRINT A$(1)
```

A$(x) est une variable dimensionnée (tableau)

QUESTION

Pourquoi $843 - **$851** ? (ci-dessous)

On lira successivement la longueur puis l'adresse de chaque mot dans :

$843 - 44 - 45
$846 - 47 - 48
$849 - 4A - 4B
$84C - 4D - 4E
$84F - 50 - 51

```
]RUN
?TOTO
?NATHALIE
?LILI
?RENE
?ZOE
TOTO
```

Dans la mémoire de l'Apple, le début de la zone réservée aux variables dimensionnées est pointé par les octets 107 et 108 ($6B et $6C), mais les octets 131 et 132 fournissent le même renseignement... sur la dernière variable utilisée (ici, il s'agit de A$(1)).

On interroge $83 -$84, puis $843 -$851

```
]CALL-151

*83.84

0083- 43 08  = 0843 à l'endroit
*843.851

0843- 04 FC 95 08 F4
0848- 95 04 F0 95 04 EC 95 03
0850- E9 95
```

E9 95 = $95E9
95FF = 95FC+84 (longueur)

Notez que les mots sont stockés les uns à la suite des autres, à partir du sommet de la mémoire, en ASCII

```
*95E9.95FF

95E9- 5A 4F 45 52 45 4E 45
95F0- 4C 49 4C 49 4E 41 54 48
95F8- 41 4C 49 45 54 4F 54 4F
                  T  O  T  O
```

• par **NESTOR**

JEU | DOS 3.3 | ProDOS

Michel DEVAUX

MASTERMIND

Version pour Apple //

CETTE adaptation du fameux Mastermind ne décevra certainement pas les (nombreux) amateurs de jeux de logique. Elle a le mérite d'utiliser, sans aucune modification, la *FONTE.LM* parue dans le numéro 5 de *Tremplin Micro* (sous la signature du même auteur).

PROGRAMME BASIC

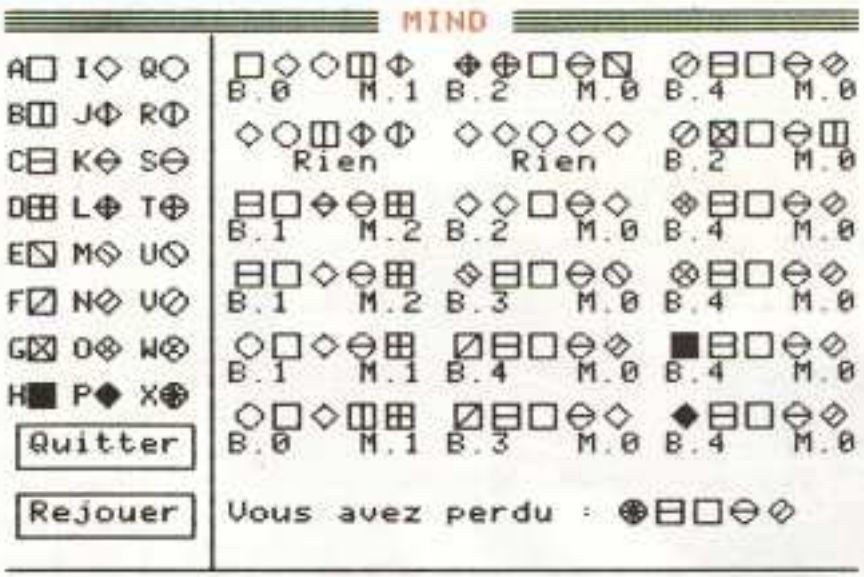
```
100 LOMEM: 18100: TEXT : NORMAL : PR
    INT CHR$ (21): HOME
105 PRINT : PRINT CHR$ (4)"BLOAD FON
    TE.LM"
110 PRINT : PRINT CHR$ (4)"BLOAD MIN
    DFORM"
115 POKE 232,0: POKE 233,3: SCALE= 1
    : ROT= 1
120 GOSUB 640
125 :
130 REM ********** ECRAN ***********
135 :
140 AD = 16384
145 HGR : POKE - 16302,0
150 HCOLOR= 7
155 FOR I = 0 TO 7 STEP 2
160 HPLOT 0,I TO 118,I
165 HPLOT 162,I TO 279,I
170 NEXT
175 HPLOT 0,191 TO 279,191
180 HOME
185 HTAB 19: PRINT "MIND"
190 L = 1:C = 1
195 FOR I = 1 TO 24
200 L = L + 2: IF L > 17 THEN L = 3:
    C = C + 3
205 C$ = CHR$ (I + 64)
210 VTAB L: HTAB C: PRINT C$
215 X = C * 7:Y = (L - 1) * 8 - 1:F
    = I
220 GOSUB 560
225 NEXT
230 CALL AD
235 FOR I = 1 TO 2
240 J = (I - 1) * 24 + 141
245 HPLOT 3,J TO 59,J TO 59,J + 15 T
    O 3,J + 15 TO 3,J
250 NEXT
255 HPLOT 65,8 TO 65,191
260 FOR I = 1 TO 3
265 FOR J = 1 TO 6
270 FOR K = 1 TO 5
275 HPLOT (I - 1) * 70 + 77 + (K - 1
    ) * 12,J * 24 - 2
280 NEXT K,J,I
285 :
290 REM ********** JEU ***********
295 :
300 FOR I = 1 TO 5
305 N(I) = INT ( RND (1) * 24 + 1)
310 NEXT
315 L = 1:C = 11:CO = 0
320 L = L + 3: IF L > 19 THEN L = 4:
    C = C + 10
325 B = 0:M = 0:CO = CO + 1
330 FOR I = 1 TO 5:AA(I) = 0:NN(I) =
     0: NEXT
335 FOR I = 1 TO 5
340 VTAB 1: HTAB 1: GET A$
345 A(I) = ASC (A$) - 64: IF A(I) <
```

```
    1 OR A(I) > 24 THEN PRINT CHR$ (
    7): GOTO 340
350 X = C * 7 + (I - 1) * 12 - 4
355 Y = (L - 2) * 8 - 2
360 F = A(I)
365 GOSUB 560
370 IF A(I) = N(I) THEN B = B + 1:AA
    (I) = 1:NN(I) = 1
375 NEXT
380 J = 0
385 J = J + 1: IF J = 6 THEN GOTO 41
    5
390 FOR I = 1 TO 5
395 IF AA(J) = 1 OR NN(I) = 1 THEN G
    OTO 405
400 IF A(J) = N(I) THEN M = M + 1:AA
    (J) = 1:NN(I) = 1
405 NEXT
410 GOTO 385
415 VTAB L: HTAB C
420 IF M = 0 AND B = 0 THEN PRINT "
     Rien": GOTO 430
425 PRINT "B.";B;"   M.";M
430 CALL AD
435 IF CO < 18 AND B < > 5 THEN GOTO
     320
440 VTAB 22: HTAB 11
445 IF B < > 5 THEN GOTO 470
450 PRINT "Vous avez GAGNE en ";CO;"
     coup";
455 IF CO > 1 THEN PRINT "s"
460 IF CO = 1 THEN PRINT
465 GOTO 475
470 PRINT "Vous avez perdu :"
475 CALL AD
480 IF B = 5 THEN GOTO 515
485 FOR I = 1 TO 5
490 X = 196 + (I - 1) * 12
495 Y = 167
500 F = N(I)
505 GOSUB 560
510 NEXT
515 VTAB 19: HTAB 2: PRINT "Quitter"
520 VTAB 22: HTAB 2: PRINT "Rejouer"
525 CALL 16384
530 VTAB 1: HTAB 1: GET A$
535 IF A$ < > "Q" AND A$ < > "R" THE
    N GOTO 530
540 IF A$ = "R" THEN RUN 130
545 TEXT : HOME : END
550 END
555 :
560 REM ********** FORMES ***********
565 :
570 IF F = 16 THEN DRAW 12 AT X + 4,
    Y: GOTO 630
575 FO = INT ((F - 1) / 8) + 1
580 DRAW FO AT X,Y
585 IF F = 2 OR F = 10 OR F = 18 OR
    F = 4 OR F = 12 OR F = 20 OR F =
     24 OR F = 16 THEN DRAW 4 AT X +
     4,Y + 1
590 IF F = 3 OR F = 11 OR F = 19 OR
    F = 4 OR F = 12 OR F = 20 OR F =
     24 THEN DRAW 5 AT X + 1,Y + 4
595 IF F = 5 OR F = 7 THEN DRAW 6 AT
     X + 1,Y + 1
600 IF F = 6 OR F = 7 THEN DRAW 9 AT
     X + 7,Y + 1
```

PRINCIPE DU JEU

L'ordinateur choisit une sélection de 5 formes graphiques (la même peut être sélectionnée plusieurs fois).

Il s'agit de découvrir cette sélection, par déductions successives, en un minimum de coups.



MASTERMIND (suite et fin)

```
605 IF F = 13 OR F = 15 THEN DRAW 7
    AT X + 3,Y + 3
610 IF F = 14 OR F = 15 THEN DRAW 10
     AT X + 5,Y + 3
615 IF F = 21 OR F = 23 OR F = 24 TH
    EN DRAW 8 AT X + 2,Y + 2
620 IF F = 22 OR F = 23 OR F = 24 TH
    EN DRAW 11 AT X + 6,Y + 2
625 IF F = 8 THEN FOR J = X TO X + 8
    : HPLOT J,Y TO J,Y + 8: NEXT J
630 RETURN
635 :
640 REM ******* PRESENTATION *******
645 :
650 TEXT : HOME
655 HTAB 13: INVERSE : PRINT " MASTE
    R.MIND ": NORMAL
660 PRINT : PRINT : PRINT : PRINT "
     L'ordinateur choisit une sélect
    ion de"
665 PRINT "5 formes graphiques parmi
     les 24 propo-"
670 PRINT "sées ."
675 PRINT : PRINT "  Votre but ? Déc
    ouvrir cette sélection"
680 PRINT "en un minimum de coups ."
685 PRINT : PRINT "  A chaque coup,
    l'ordinateur vous indi-";
690 PRINT "quera combien votre propo
    sition comporte";
695 PRINT "de formes BIEN placées et
     MAL placées ."
700 CALL - 198: VTAB 23: PRINT "APPU
    YEZ SUR UNE TOUCHE POUR COMMENCE
    R":: GET A$
705 RETURN
```

Pour fonctionner, le programme en Basic doit être accompagné, sur sa disquette par :

- FONTE.LM (Tremplin Micro n°5)
- MINDFORM (ci-dessous)

Adresses :

Fonte.LM : 16384 ($4000)
Mindform : 768 ($300)

Pour enregistrer les codes de MINDFORM, passez d'abord en mode moniteur (CALL — 151 ... RETURN), puis tapez chaque ligne en remplaçant le tiret par deux points (:).
Lorsque vous aurez terminé, tapez CTRL-e, puis **BSAVE MINDFORM, A$300, L$A8**

MINDFORM

```
0300- 10 00 28 00 39 00 4C 00
0308- 5E 00 63 00 68 00 70 00
0310- 74 00 7A 00 82 00 86 00
0318- 8C 00 00 00 00 00 00 00
0320- 00 00 00 00 00 00 00 00
0328- 2D 2D 2D 2D 36 36 36 36
0330- 3F 3F 3F 3F 24 24 24 24
0338- 00 09 09 15 15 15 15 17
0340- 17 17 17 1C 1C 1C 1C 0C
0348- 0C 0C 0C 00 49 2D 15 15
0350- 15 36 1E 1E 1E 3F 1C 1C
0358- 1C 24 0C 0C 05 00 36 36
0360- 36 06 00 2D 2D 2D 05 00
0368- 15 15 15 15 15 15 05 00
0370- 15 15 05 00 15 15 15 15
0378- 05 00 17 17 17 17 17 17
0380- 07 00 17 17 07 00 17 17
0388- 17 17 07 00 15 3F 17 2D
0390- 2D 15 3F 3F 3F 17 2D 2D
0398- 2D 2D 1E 3F 3F 3F 0E 2D
03A0- 2D 1E 3F 0E 05 00 00 00
```

Lorsque vous écrivez à Tremplin Micro...

Si vous attendez de nous un renseignement... un conseil, joignez toujours un timbre-poste pour la réponse.

Si vous nous adressez une disquette, conservez l'original de vos programmes et n'oubliez pas de joindre une enveloppe affranchie pour son retour.

Nous n'examinerons votre programme que s'il a été recopié sur une disquette.

ÉCRAN-DAMIER

CES démonstrations sont destinées à vous montrer les énormes possibilités du langage machine, utilisé pour de courtes routines. Vous constaterez notamment, en comparant *DAMIER.0* et *DAMIER.1* (page 14), qu'il est souvent facile, en traitant le problème d'une autre manière, d'accélérer considérablement la vitesse d'exécution.

DAMIER.BAS 40 COLONNES

```
100 TEXT : PRINT CHR$ (21): HOME : COLOR= 2
110 VTAB 12: HTAB 5: INVERSE : PRINT " DAMIER EN BASIC (
    BEAGLE BROS) ": NORMAL
120 GOSUB 290: HOME
130 FOR X = 0 TO 39 STEP 2: FOR Y = 1 TO 47 STEP 2:Z = Z
    = 0: PLOT X + Z,Y: NEXT Y,X: POKE 37,255
140 GOSUB 290: HOME
150 PRINT : PRINT : LIST 100: LIST 130: GOSUB 290: HOME
160 VTAB 12: HTAB 2: INVERSE : PRINT " DAMIER EN LANGAGE
    MACHINE, VERSION 0 "
170 NORMAL : GOSUB 290
180 PRINT : PRINT CHR$ (4)"BLOAD DAMIER.0"
190 FOR I = 1 TO 3: GOSUB 290: CALL 768: GOSUB 290: HOME
    : NEXT
200 VTAB 12: HTAB 2: INVERSE : PRINT " DAMIER EN LANGAGE
    MACHINE, VERSION 1 "
210 NORMAL : GOSUB 290
220 PRINT : PRINT CHR$ (4)"BLOAD DAMIER.1"
230 FOR I = 1 TO 3: GOSUB 290: CALL 768: GOSUB 290: HOME
    : NEXT
240 VTAB 22: PRINT "<1> ENCORE <2> MENU DISQUETTE <3> FI
    N ";: GET R$: PRINT
250 IF R$ = "1" THEN RUN
260 IF R$ = "2" THEN PRINT CHR$ (4)"RUN MENU"
270 IF R$ < > "3" THEN 240
280 HOME : END
290 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: PO
    KE - 16368,0: RETURN
```

Rien ne vous oblige à recopier la totalité de ce court programme de démonstration, encore que les premières lignes, empruntées à *Beagle Bros* soient fort originales (auriez-vous pensé à utiliser la fonction *PLOT...* en mode *TEXT...* pour transformer votre écran en damier ?)

Par contre, ne négligez surtout pas :

DAMIER.0 : affiche des caractères à l'écran. Il est lent.

DAMIER.1 : écrit directement dans la mémoire-écran. Il est très rapide.

DAMIER.0

```
0300-  20 58 FC     JSR   $FC58     On efface l'écran,
0303-  A9 18        LDA   #$18      puis on stocke $18 (24) dans la mémoire $6
0305-  85 06        STA   $06
0307-  A9 A0        LDA   #$A0      COUT1 ($FDF0) envoie un espace
0309-  20 F0 FD     JSR   $FDF0
030C-  A0 00        LDY   #$00      Y = 0
030E-  A2 01        LDX   #$01      X = 1
0310-  A5 06        LDA   $06       Lecture de la mémoire $6 (par A)
0312-  C9 01        CMP   #$01      On compare le contenu de A à 1
0314-  D0 04        BNE   $031A     Si ce n'est pas égal, saut à $31A
0316-  C0 13        CPY   #$13      Y est-il égal à $13 (19) ?
0318-  F0 01        BEQ   $031B     Oui : alors on saute à $31B
031A-  E8           INX             X = X + 1
031B-  A9 20        LDA   #$20      $20 (espace inverse) mis dans A
031D-  20 4C F9     JSR   $F94C     PRBL3 ($F94C) envoie A + X — 1 espace(s)
0320-  C8           INY             Y = Y + 1
0321-  C0 14        CPY   #$14      Y est-il à $14 (20) ?
0323-  90 E9        BCC   $030E     Si oui (C = 0), on revient à $30E
0325-  A5 24        LDA   $24       Lecture de CH (position horizontale du curseur)
0327-  48           PHA             A stocké sur la pile
0328-  38           SEC             Retenue à 1 pour soustraire
0329-  E5 24        SBC   $24       Position HOR. moins 1
032B-  85 24        STA   $24
032D-  C6 06        DEC   $06      Et moins 1 pour le nombre de lignes
032F-  F0 05        BEQ   $0336     Si on est à zéro, terminé !
0331-  68           PLA             On récupère A sur la pile
0332-  D0 D8        BNE   $030C     S'il est différent de 0, ligne impaire
0334-  F0 D1        BEQ   $0307     ou, au contraire, ligne paire
0336-  68           PLA             Ne pas oublier de dépiler A
0337-  A9 FF        LDA   #$FF      Retour du curseur en haut de l'écran... et
0339-  85 25        STA   $25       retour au Basic
033B-  60           RTS
```

BSAVE DAMIER.0, A$300, L$3C

DAMIER.1

```
0300-  20 58 FC     JSR   $FC58     Home
0303-  A0 01        LDY   #$01      Y = 1
0305-  20 C1 FB     JSR   $FBC1     BASCALC calcule l'adresse de base
0308-  A9 20        LDA   #$20      Espace inverse dans A
030A-  91 28        STA   ($28),Y   Stockage de A à l'adresse lue dans $28/29 + Y
030C-  C8           INY             Y incrémenté deux fois (damier !)
030D-  C8           INY
030E-  C0 28        CPY   #$28      Est-il à $28 ?
0310-  90 F8        BCC   $030A     S'il est au-dessous, un autre tour
0312-  E6 25        INC   $25       Sinon on incrémente CV (VTAB)
0314-  A5 25        LDA   $25       Puis on en lit la valeur
0316-  C5 23        CMP   $23       Que l'on compare à $23 (où il y a $18)
0318-  F0 08        BEQ   $0322     Si A = $18 (24), c'est fini
031A-  C0 28        CPY   #$28      Y est-il égal à $28 (40) ?
031C-  F0 E5        BEQ   $0303     Si oui, un espace pour commencer la ligne
031E-  A0 00        LDY   #$00      Non : alors Y = 1 (pas d'espace)
0320-  F0 E3        BEQ   $0305     GOTO $305
0322-  A9 FF        LDA   #$FF      Voir DAMIER.0
0324-  85 25        STA   $25
0326-  60           RTS
```

BSAVE DAMIER.1, A$300, L$27

Passons maintenant sur 80 colonnes

VERSION MIXTE Si vous êtes courageux (bien sûr que vous l'êtes !), recopiez les 47 octets de cette version mixte. Elle a le mérite (?) de donner le même résultat sur 40 ou 80 colonnes.

```
300 : 20  58  FC  A0  01  20  C1  FB  A9  20  91  28  8D  55  C0  91  28  C8
      C8  8D  54  C0  C0  28  90  F0  E6  25  A5  25  C5  23  F0  08  C0  28
      F0  DD  A0  00  F0  DB  A9  FF  85  25  60
```

Pour terminer : **BSAVE DAMIER.80cp, A$300, L$2F**

Utilisation CALL 768 (après un PR£3... si vous désirez le damier sur 80 colonnes).

La gestion de votre écran 80 colonnes est plus complexe que celle de l'écran 40 colonnes.

En effet, un caractère sur deux est en mémoire auxiliaire, à la même adresse qu'en mémoire normale (dite résidente).

Ainsi, pour lire ou écrire chaque caractère, il faut d'abord taper un :

POKE 49237,0 ($C055) qui donne accès à la MEM.AUX., puis un

POKE 49236,0 ($C054) qui ramène en mémoire vive.

En langage machine, on ne procède pas autrement (voir page suivante).

Ligne 330 : Adresse de base de chaque ligne-écran (valable en 40 ou en 80 colonnes).

La *DEMO* ci-contre est intéressante à analyser. N'hésitez pas à reprendre chaque instruction en mode direct... ■

DAMIER.BAS 80 COLONNES

```
100 PRINT CHR$ (4)"PR£3": PRINT : HOME
110 P = ABS (P - 1): POKE 49236 + P,0
120 READ A: IF NOT (A) THEN POKE 49236,0: GOTO 150
130 B = A + 39
140 FOR I = A TO B: POKE I,32: NEXT : GOTO 110
150 GOSUB 310: GOSUB 320: PRINT : PRINT : PRINT " ET MAI
    NTENANT PASSONS A LA ROUTINE EN LANGAGE MACHINE, EVI
    DEMMENT PLUS RAPIDE": GOSUB 310
160 PRINT CHR$ (4)"BLOAD DAMIER.80cp"
170 CALL 768: GOSUB 310
180 GOSUB 320: LIST 190,210: GOSUB 310: CALL 768: GOSUB
    310
190 PRINT CHR$ (17): GOSUB 310
200 PRINT CHR$ (18): GOSUB 310
210 PRINT CHR$ (17): GOSUB 310
220 CALL 768: GOSUB 310: GOSUB 320: LIST 230,250: GOSUB
    310: CALL 768: GOSUB 310
230 PRINT CHR$ (18): CALL 768: GOSUB 310
240 PRINT CHR$ (21): CALL 768: GOSUB 310
250 PRINT CHR$ (4)"PR£3": PRINT : CALL 768: GOSUB 310
260 GOSUB 320: PRINT : PRINT : PRINT "<1> A REFAIRE  <2>
     MENU GENERAL DE LA DISQUETTE  <3> FIN-PROGRAMME EN
    MEMOIRE ";: GET R$: PRINT
270 IF R$ = "1" THEN RUN
280 IF R$ = "2" THEN PRINT CHR$ (4)"RUN MENU"
290 IF R$ = "3" THEN HOME : END
300 GOTO 260
310 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: PO
    KE - 16368,0: PRINT : RETURN
320 POKE 37,18: CALL - 958: RETURN
330 DATA 1024,1152,1280,1408,1536,1664,1792,1920,1064,11
    92,1320,1448,1576,1704,1832,1960,1104,1232,1360,1488
    ,1616,1744,1872,2000,0
```

ÉCRAN-DAMIER (suite et fin)

DAMIER.80cp

```
0300-  20 58 FC   JSR  $FC58      Home
0303-  A0 00      LDY  £$00       Y = 0
0305-  20 C1 FB   JSR  $FBC1      BASCALC calcule l'adresse de base
0308-  A9 20      LDA  £$20       Espace inverse mis dans l'accumulateur
030A-  E6 25      INC  $25        CV (VTAB en Basic) = CV + 1
030C-  8D 55 C0   STA  $C055      Pour lire ou écrire en MEM.AUX. (page 2)
030F-  91 28      STA  ($28),Y    A copié à l'adresse lue en $28/29 + Y
0311-  C8         INY             Y = Y + 1
0312-  C0 28      CPY  £$28       Y est-il égal à $28 (40) ?
0314-  90 F9      BCC  $030F      S'il est plus petit, ligne inachevée
0316-  48         PHA             Contenu de A empilé
0317-  98         TYA             Y est mis dans A
0318-  18         CLC             Annulation de la retenue
0319-  69 57      ADC  £$57       A = A + $57... lire en bas de page
031B-  A8         TAY             Nouvelle valeur de A dans Y
031C-  8D 54 C0   STA  $C054      Pour lire ou écrire en MEM. normale
031F-  68         PLA             On récupère A sur la pile ($20)
0320-  91 28      STA  ($28),Y    On l'écrit à l'adresse lue en $28/29 + Y
0322-  C8         INY             Y = Y + 1
0323-  C0 A8      CPY  £$A8       Y est-il égal à $A8 ?... (lire en bas de page)
0325-  90 F9      BCC  $0320      Non : on n'a pas écrit toute la ligne
0327-  E6 25      INC  $25        CV = CV + 1 (VTAB)
0329-  A5 25      LDA  $25        Valeur de CV dans A
032B-  C5 23      CMP  $23        Est-elle égale à celle contenue dans $23 ?
032D-  D0 D4      BNE  $0303      Non : on a encore des lignes à écrire
032F-  A9 FF      LDA  £$FF       ┐ Curseur à la ligne 0... et
0331-  85 25      STA  $25        │ retour au Basic
0333-  60         RTS             ┘
```

BSAVE DAMIER.80cp, A$300, L$34

IL FAUT SAVOIR

BASCALC ($FBC1) calcule, on le sait, l'adresse de base de chaque ligne de l'écran, à partir de la valeur (0 à $17) contenue dans $25 (CV). On trouve cette adresse de base dans *BASL* et *BASH* ($28 et 29). A noter que la mémoire $23 contient le numéro (1 à 24) de la dernière ligne de l'écran. Si elle n'a pas été modifiée, on sait donc y trouver $18 (24).

128 octets ($80) séparent l'adresse de base de la ligne 0... de celle de la ligne 1... et ainsi de suite, de 2 en 2 (paire-impaire, paire-impaire, etc.)... d'où les étranges additions observées dans la routine ci-dessus ($28 + $57 = $7F). De même, plus loin, $A8 = $28 + $80.

Et voilà ! j'espère, avec ces quelques démonstrations, vous avoir donné envie de faire, vous aussi, quelques courtes incursions dans le "ténébreux" (en apparence) langage machine. C'est un jeu de logique qui vaut bien le Mastermind et ses nombreux dérivés !

NESTOR

AND.DÉMO

7	6	5	4	3	2	1	0
N	V	■	B	D	I	Z	C

HUIT bits constituent un octet... du moins avec le 6502 (ou 65C02) et quelques autres. Les signes cabalistiques **NVBDIZC** sont des indicateurs qui prennent la valeur 0 ou 1 suivant le mode ou l'état du processeur. Ainsi, C = 1 quand la retenue est à 1 ; Z = 1 si le dernier résultat est nul (s'il y a eu égalité dans une comparaison). Il est bon de savoir que le bit de signe est N (à 1 quand le dernier calcul a donné un résultat négatif).

Mais l'objet de cet article est autre : *comment forcer un ou plusieurs bits à 0... avec l'instruction AND ?*

Avant d'aller plus loin, signalons qu'il est impossible de comprendre les opérations sur les bits sans avoir sous les yeux un tableau des codes HEXA et (surtout) binaires des 255 premiers nombres. C'est l'objet de la **fiche** de ce numéro.

AND est une opération logique (comme ORA et EOR) et opère bit par bit, très simplement : 1 — 1 = 1, 1 — 0 = 0, 0 — 1 = 0. Mais plusieurs exemples valent mieux qu'un long discours (LDA signifie que l'on place dans l'Accumulateur la valeur indiquée) :

LDA	11111111 $FF	11111111 $FF	11111111 $FF
AND	00000000 $0	00000001 $01	10000001 $81
RÉSULTAT	00000000 $0	00000001 $01	10000001 $81

AVOUEZ QUE C'EST PRATIQUE !

QUESTION :

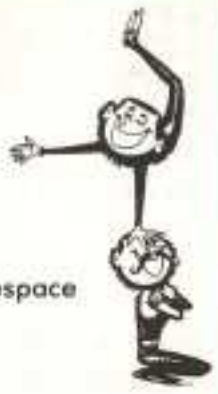
Comment forcer à 0 le seul bit 4 ? Comme ceci :

LDA	11111111 $FF	01010101 $55
AND	11101111 $EF	11101111 $EF
RÉSULTAT	11101111 $EF	01000101 $45

APPLICATION Et voici un petit programme d'application, entièrement en langage machine. Objet : afficher les caractères de l'alphabet en mode inverse... mais seulement si leur valeur est impaire. Consultez notre FICHE. Vous constaterez que le bit 0 de tous les caractères impairs... est à 1, d'où l'intérêt de AND. Passons donc à la routine (lancée par CALL 768).

AND.DÉMO

```
0300-  20 58 FC   JSR  $FC58    Home
0303-  A9 00      LDA  #$00     A = 0
0305-  85 06      STA  $06      Adresse 6 à 0
0307-  E6 06      INC  $06      Mémoire 6 + 1
0309-  A5 06      LDA  $06      Mémoire 6 dans Accumulateur
030B-  29 01      AND  #$01     ET VOICI AND...
030D-  F0 11      BEQ  $0320    Si on obtient 0, pas d'affichage
030F-  A2 02      LDX  #$02     Autrement X = 2 pour envoyer 1 espace
0311-  A5 06      LDA  $06      Relecture de la mémoire 6
0313-  20 4C F9   JSR  $F94C    PRBL3 affiche A + X — 1 espace
0316-  A6 06      LDX  $06      Mémoire 6 dans X
0318-  A9 00      LDA  #$00     A = 0
```



AND.DÉMO (suite)

```
031A-   20 24 ED   JSR   $ED24     On affiche les deux octets de X, A
031D-   20 62 FC   JSR   $FC62     Retour chariot
0320-   A5 06      LDA   $06       On récupère la valeur de A
0322-   C9 1A      CMP   #$1A      Est-elle égale à $1A (26) ?
0324-   D0 E1      BNE   $0307     Non : encore un tour
0326-   A9 06      LDA   #$06      A = $06
0328-   85 20      STA   $20       $06 stocké en $20          ] POKE 32,6
032A-   A9 21      LDA   #$21      A = $21
032C-   85 21      STA   $21       $21 stocké en $21          ] POKE 33,33 ($21)
032E-   20 58 FC   JSR   $FC58     HOME
0331-   A9 FF      LDA   #$FF      ]
0333-   85 25      STA   $25       ] Curseur en haut de l'écran
0335-   A9 00      LDA   #$00      ]
0337-   85 3A      STA   $3A       ]
0339-   A9 03      LDA   #$03      ] On place en $3A et $3B l'adresse de départ
033B-   85 3B      STA   $3B       ] de cette routine et dans A le nombre d'ins-
033D-   A9 10      LDA   #$10      ] tructions à désassembler, puis $FE63 fait le
033F-   20 63 FE   JSR   $FE63     ] reste
0342-   A0 16      LDY   #$16      Y = $16 (22)
0344-   B9 8D 06   LDA   $068D,Y   Lecture de la mémoire $68D + Y
0347-   C9 A0      CMP   #$A0      Est-ce un espace ?
0349-   F0 02      BEQ   $034D     Si oui on ne le transforme pas... on saute !
034B-   29 3F      AND   #$3F      Non : avec AND, caractère mode inverse
034D-   99 8D 06   STA   $068D,Y   Il est réécrit
0350-   88         DEY             Y = Y — 1
0351-   D0 F1      BNE   $0344     Si on n'est pas à 0, encore un tour
0353-   A9 00      LDA   #$00      ]
0355-   85 20      STA   $20       ] Y étant à 0, on pouvait économiser une ins-
                                   ] truction en utilisant STY $20
0357-   A9 28      LDA   #$28      ]
0359-   85 21      STA   $21       ] A = $28 (40) pour rétablir la fenêtre d'écran
                                   ] normale
035B-   A9 12      LDA   #$12      ]
035D-   85 25      STA   $25       ] CV $12 (VTAB 19)
035F-   20 62 FC   JSR   $FC62     Retour chariot
0362-   20 22 FC   JSR   $FC22     VTAB place le curseur en CV
0365-   A2 08      LDX   #$08      ]
0367-   20 4A F9   JSR   $F94A     ] X = 8 pour envoyer... 8 espaces
036A-   20 80 FE   JSR   $FE80     SETINV affichage en mode inverse
036D-   A0 03      LDY   #$03      ]
036F-   A9 81      LDA   #$81      ] STROUT affiche le message dont l'adresse
0371-   20 3A DB   JSR   $DB3A     ] est mise dans Y et A (il se termine par 0)
0374-   20 84 FE   JSR   $FE84     SETNORM... mode normal
0377-   20 E4 FB   JSR   $FBE4     BELL2 se fait entendre
037A-   20 88 FB   JSR   $FB88     KBDWAIT attend une touche
037D-   20 58 FC   JSR   $FC58     HOME
0380-   60         RTS             BASIC
```

De 381 à 397 : message **IMPAIR OU PAIR AVEC AND :**

381 : 49 4D 50 41 49 52 20 4F 55 20 4E 4F 4E 20 41 56 45 43 20 41 4E 44 00

Pour terminer : CTRL-C RETURN, puis **BSAVE AND.DÉMO, A$300, L$98** ■

MIROIR

Toutes les lignes de l'écran inversées en un clin d'œil !

```
100 TEXT : PRINT CHR$ (21): HOME
110 GOSUB 250
120 ONERR GOTO 190
130 FOR I = 1 TO 24: PRINT I;
140 IF I = 11 THEN PRINT " ":: INVERSE : PRINT " CONTROL
    E-C POUR INTERROMPRE ":: NORMAL
150 IF I < 24 THEN PRINT ""
160 NEXT
170 VTAB 1: CALL 768
180 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: PO
    KE - 16368,0: GOTO 170
190 HOME : POKE 58,0: POKE 59,3: CALL - 418: GET R$: PRI
    NT
200 PRINT : HOME : VTAB 22: PRINT "<1> ENCORE <2> MENU D
    ISQUETTE <3> FIN ":: CALL - 198: GET R$: PRINT
210 HOME : IF R$ = "1" THEN GOTO 120
220 IF R$ = "2" THEN PRINT CHR$ (4)"RUN MENU"
230 IF R$ < > "3" THEN 200
240 END
250 FOR I = 768 TO 768 + 33: READ R: POKE I,R: NEXT : RE
    TURN
260 DATA 162,23,138,32,193,251,160,0,177,40,72,200,192,4
    0,208,248,160,0,104,145,40,200,192,40,208,248,202,16
    ,229,169,255,133,37,96
```

EMPILER-DÉPILER

La dernière valeur *empilée* va devenir la première *dépilée*... ce qui permet de retourner facilement les 40 octets de chaque ligne.

Ne tapez pas la routine en langage machine... puisqu'elle est contenue dans la ligne 260.

```
0300-   A2 17      LDX   #$17      X = $17 (nombre de lignes d'écran - 0- 23)
0302-   8A         TXA               X passe dans A (sans être modifié)
0303-   20 C1 FB   JSR   $FBC1      BASCALC donne l'adresse du premier octet
0306-   A0 00      LDY   #$00      Y = 0 pour la boucle
0308-   B1 28      LDA   ($28),Y   Lecture de la mémoire dont l'adresse est en $28-29 + Y
030A-   48         PHA               Contenu de A empilé
030B-   C8         INY               Y = Y + 1
030C-   C0 28      CPY   #$28      Y est-il égal à $28 (40) ?
030E-   D0 F8      BNE   $0308     Non : encore un tour (on va à $308)
0310-   A0 00      LDY   #$00      Y = 0 pour seconde boucle
0312-   68         PLA               On récupère une valeur de la pile
0313-   91 28      STA   ($28),Y   On l'écrit à l'adresse lue en $28-29 + Y
0315-   C8         INY               Y = Y + 1
0316-   C0 28      CPY   #$28      Y est-il égal à $28 (40) ?
0318-   D0 F8      BNE   $0312     Non : on recommence
031A-   CA         DEX               X = X — 1 (nombre de lignes)
031B-   10 E5      BPL   $0302     Si on est positif, encore un tour !
031D-   A9 FF      LDA   #$FF
031F-   85 25      STA   $25       Curseur en haut de l'écran et retour au Basic.
0321-   60         RTS
```

TÊTE EN BAS

ET maintenant, renversons les lignes de l'écran... comme cela, simplement pour le plaisir, en attendant qu'une (bonne) occasion d'utiliser la routine **se** présente.

INITIATION

```
100 TEXT : PRINT CHR$ (21): HOME : GOSUB 175
105 ONERR GOTO 145
110 E = 35: FOR I = 1 TO 24
115 PRINT I; SPC( E)I;
120 IF I = 9 THEN HTAB 8: INVERSE : PRINT " CONTROLE-C P
    OUR STOPPER ";: NORMAL :E = E - 1
125 IF I < > 24 THEN PRINT ""
130 NEXT
135 VTAB 1: PRINT ""
140 GOSUB 170: CALL 768: GOTO 135
145 HOME : VTAB 22: CALL - 198: PRINT "<1> ENCORE <2> ME
    NU DISQUETTE <3> FIN ";: GET R$
150 PRINT : HOME : IF R$ = "1" THEN 110
155 IF R$ = "2" THEN PRINT CHR$ (4)"RUN MENU"
160 IF R$ < > "3" THEN 145
165 END
170 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: PO
    KE - 16368,0: RETURN
175 FOR I = 768 TO 852: READ R: POKE I,R: NEXT : RETURN
180 DATA 169,0,133,6,169,32,133,7,162,23,138,32,193,251,
    160,0,177,40,145,6,200,192,40,208,247,165,6,24,105,4
    0,133,6,144,2,230,7,202,16
185 DATA 227,169,32,133,7,169,0,133,6,162,0,138,32,193,2
    51,160,0,177,6,145,40,200,192,40,208,247,165,6,24,10
    5,40,133,6,144,2,230,7,232,224,24,208,225,169,255,13
    3,37,96
```

Dans la routine en langage machine (lignes 180-185) expliquée ci-contre, on recopie l'écran à partir de l'adresse $2000, puis on le réécrit en commençant par la ligne du bas.
On pourrait utiliser la pile (un peu comme on l'a fait avec MIROIR) de la manière suivante :

- Lire et empiler la ligne 0
- Lire et empiler la ligne 23
- Dépiler pour écrire la ligne 0
- Dépiler pour écrire la ligne 23

... et ainsi de suite, jusqu'à 11 et 12 (voir page 22).

BPL

Si, après le CALL — 151 bien connu, vous tapez ces 5 instructions (uniquement ce qui est en gras), puis * 300 G... RETURN, vous constaterez que la mémoire $6 contient $FF. En effet, BPL renvoie à l'adresse $304 tant que le résultat est égal ou supérieur à 0.

```
300 : A9 02   LDA   #$02
302 : 85 06   STA   $06
304 : C6 06   DEC   $06
306 : 10 FC   BPL   $0304
308 : 60      RTS
```

TÊTE EN BAS (LIGNES 180-185)

```
0300-  A9 00      LDA   £$00
0302-  85 06      STA   $06
0304-  A9 20      LDA   £$20
0306-  85 07      STA   $07
0308-  A2 17      LDX   £$17
```

Initialisation des pointeurs $6-7 avec $2000 (pour mémorisation de l'écran)

Initialisation nombre de lignes ($17 = 23)

```
030A-  8A         TXA
030B-  20 C1 FB   JSR   $FBC1
030E-  A0 00      LDY   £$00
0310-  B1 28      LDA   ($28),Y
0312-  91 06      STA   ($06),Y
0314-  C8         INY
0315-  C0 28      CPY   £$28
0317-  D0 F7      BNE   $0310
0319-  A5 06      LDA   $06
031B-  18         CLC
031C-  69 28      ADC   £$28
031E-  85 06      STA   $06
0320-  90 02      BCC   $0324
0322-  E6 07      INC   $07
0324-  CA         DEX
0325-  10 E3      BPL   $030A
```

X passe dans A (mais n'est pas modifié)
BASCALC calcule l'adresse de base
Y = 0 pour la boucle
Lecture écran et mémo. à partir de $2000
Y = Y + 1
A-t-on lu 40 caractères ?
Non : encore une lecture
Contenu de la mémoire $6 dans A
Annulation de la retenue
A = A + $28 (40)
A remis dans le pointeur $6
S'il n'y a pas eu de retenue, on saute
Sinon il faut incrémenter le pointeur $7
X = X — 1
Voir bas de page précédente

```
0327-  A9 20      LDA   £$20
0329-  85 07      STA   $07
032B-  A9 00      LDA   £$00
032D-  85 06      STA   $06
032F-  A2 00      LDX   £$00
```

Nouvelle initialisation des pointeurs $6 et $7

Puis de X (cette fois, on part de 0)

```
0331-  8A         TXA
0332-  20 C1 FB   JSR   $FBC1
0335-  A0 00      LDY   £$00
0337-  B1 06      LDA   ($06),Y
0339-  91 28      STA   ($28),Y
033B-  C8         INY
033C-  C0 28      CPY   £$28
033E-  D0 F7      BNE   $0337
0340-  A5 06      LDA   $06
0342-  18         CLC
0343-  69 28      ADC   £$28
0345-  85 06      STA   $06
0347-  90 02      BCC   $034B
0349-  E6 07      INC   $07
034B-  E8         INX
034C-  E0 18      CPX   £$18
034E-  D0 E1      BNE   $0331
```

Le déroulement est rigoureusement le même que dans la première partie, si ce n'est que l'on incrémente X au lieu de le décrémenter... et que l'on lit à l'adresse indiquée par les pointeurs... pour écrire à celle indiquée par BASCALC.

RAPPEL : L'adressage indirect permet de lire (ou d'écrire) à l'adresse indiquée + Y. Ainsi quand Y vaut 5 et que les pointeurs $6-7 contiennent 0020, on lit le contenu de la mémoire $2000 + Y, soit $2005

```
0350-  A9 FF      LDA   £$FF
0352-  85 25      STA   $25
0354-  60         RTS
```

CV = 0
Retour au Basic

TÊTE.EN.BAS (VERSION 2)

LA routine de retournement d'écran va de $300 à 347 (L$48) et correspond aux lignes 175-180 du programme DÉMO.

La partie $348-360 remplit l'écran de lignes de caractères.

```
100 TEXT : PRINT CHR$ (21): HOME
105 GOSUB 170
110 ONERR GOTO 140
115 CALL 840
120 VTAB 24: INVERSE : PRINT "PRESSE
    Z UNE TOUCHE (CTRL-C POUR FIN)":
    : NORMAL
125 VTAB 1: PRINT ""
130 CALL - 198: POKE - 16368,0: WAIT
     - 16384,128,127: POKE - 16368,0
    : CALL 768:X = ABS (X - 1): GOTO
     130
135 X = X + 1: INVERSE : PRINT X:: N
    ORMAL : RETURN
140 IF X THEN CALL 768
145 VTAB 24:X = 0: GOSUB 135: PRINT
    " ENCORE ";: GOSUB 135: PRINT "
    MENU DE DISQUETTE ";: GOSUB 135:
     PRINT " FIN > ";: CALL - 198: G
    ET R$
150 VTAB 1: PRINT "": IF R$ = "1" TH
    EN X = 0: GOTO 120
155 IF R$ = "2" THEN PRINT CHR$ (4)"
    RUN MENU
160 IF R$ < > "3" THEN 145
165 HOME : END
170 FOR I = 768 TO 864: READ R: POKE
     I,R: NEXT : RETURN : REM LA ROU
    TINE "TETE.EN.BAS" NE COMPREND Q
    UE LES 72 PREMIERS OCTETS (768 A
     839)
175 DATA 169,23,133,6,162,0,138,32,1
    93,251,160,39,177,40,72,136,16,2
    50,165,6,32,193,251,160,39,177,4
    0,72,136,16,250,138,32,193,251,1
    60,0,104,145,40
180 DATA 200,192,40,208,248,165,6,32
    ,193,251,160,0,104,145,40,200,19
    2,40,208,248,198,6,232,224,12,20
    8,195,169,255,133,37,96
185 DATA 32,88,252,162,0,138,32,193,
    251,138,105,193,160,39,145,40,13
    6,16,251,232,224,23,208,237,96
```

```
300-   A9 17      LDA   £$17
302-   85 06      STA   $06
304-   A2 00      LDX   £$00
306-   8A         TXA
307-   20 C1 FB   JSR   $FBC1
30A-   A0 27      LDY   £$27
30C-   B1 28      LDA   ($28),Y
30E-   48         PHA
30F-   88         DEY
310-   10 FA      BPL   $030C
312-   A5 06      LDA   $06
314-   20 C1 FB   JSR   $FBC1
317-   A0 27      LDY   £$27
319-   B1 28      LDA   ($28),Y
31B-   48         PHA
31C-   88         DEY
31D-   10 FA      BPL   $0319
31F-   8A         TXA
320-   20 C1 FB   JSR   $FBC1
323-   A0 00      LDY   £$00
325-   68         PLA
326-   91 28      STA   ($28),Y
328-   C8         INY
329-   C0 28      CPY   £$28
32B-   D0 F8      BNE   $0325
32D-   A5 06      LDA   $06
32F-   20 C1 FB   JSR   $FBC1
332-   A0 00      LDY   £$00
334-   68         PLA
335-   91 28      STA   ($28),Y
337-   C8         INY
338-   C0 28      CPY   £$28
33A-   D0 F8      BNE   $0334
33C-   C6 06      DEC   $06
33E-   E8         INX
33F-   E0 0C      CPX   £$0C
341-   D0 C3      BNE   $0306
343-   A9 FF      LDA   £$FF
345-   85 25      STA   $25
347-   60         RTS
348-   20 58 FC   JSR   $FC58
34B-   A2 00      LDX   £$00
34D-   8A         TXA
34E-   20 C1 FB   JSR   $FBC1
351-   8A         TXA
352-   69 C1      ADC   £$C1
354-   A0 27      LDY   £$27
356-   91 28      STA   ($28),Y
358-   88         DEY
359-   10 FB      BPL   $0356
35B-   E8         INX
35C-   E0 17      CPX   £$17
35E-   D0 ED      BNE   $034D
360-   60         RTS
```

UTILITAIRE | ProDOS

Dominique LUGAT

TRANSLIST

Ou comment créer un fichier source à partir d'une simple routine en langage machine ?

PRENONS un exemple précis : celui de TÊTE.EN.BAS (ci-contre). *Tremplin Micro* publie le désassemblage des N instructions de cette petite routine, mais sans passer par un assembleur. C'est donc la réplique exacte du listage que vous pouvez obtenir sur votre écran. La même routine, présentée par un Assembleur (type ProCODE) permet de situer les différents branchements (boucles), d'identifier les labels et, éventuellement, grâce à l'indication de l'adresse de départ (ORG), de modifier facilement toutes les adresses.

D'où l'intérêt présenté par TRANSLIST qui désassemble — tout seul et comme un grand — n'importe quelle routine binaire (installée à l'adresse $300 et de longueur inférieure à $D0), pour la mémoriser sous forme de fichier-texte... accepté par les assembleurs.

PROGRAMMES

Installez l'ensemble des programmes sur une disquette spéciale ProDOS. Ceux-ci sont en Basic, et au nombre de trois (plus un fichier) :

- TRANSLIST (le gros morceau), à partir de la page 25
- LABELS.CREER (court), page 24
- LABELS.LIRE (plus court encore !), page 28
- LABELS (le fichier créé par vos soins)

Création des labels

Pour fonctionner convenablement, TRANSLIST doit pouvoir consulter un fichier LABELS contenant au moins un label. Bien sûr, plus le nombre de labels sera grand et plus le fichier-source généré par TRANSLIST sera documenté (par contre, le temps de traitement augmentera d'autant).

Donc : RUN LABELS.CREER... et à vous de saisir VOS labels (nous vous suggérons, page 24, une première liste), dans la limite de... 300.

Si vous désirez relire votre fichier LABELS, utilisez LIT.LABELS.

POUR LES CONNAISSEURS

Charger le fichier LABELS à partir d'un Assembleur ne pose aucun problème si celui-ci n'utilise pas le type binaire pour ses fichiers-sources.

UTILISATION

IMPORTANT : n'oubliez pas de modifier le PRÉFIXE (ligne 110) pour qu'il corresponde à celui de votre disquette (variable PF$).

Dès que TRANSLIST est lancé, il vous demande de lui indiquer le...

...nom du programme binaire

Ou, ledit programme figure sur la même disquette que TRANSLIST... et il suffit d'en taper directement le titre... ou il se trouve sur une autre, et vous devrez alors indiquer le nouveau préfixe, suivi du titre (Exemple : TOTO ou /TRUC/TOTO). En cas de faute de frappe, utilisez la touche DEL pour effacer.

Pour le reste, observez ce qui se passe sur votre écran, et tricotez un cache-col pour votre grand-mère... si vous trouvez le temps long ! Quand le nombre de labels est élevé et que le

Quelques idées de LABELS...

ADRESSES $	LABELS	COMMENTAIRES
06	INDEXB	Adressage indexé
07	INDEXH	Partie haute
08	VAR1	Variable(s) page 0
09	VAR2	
19	VAR3	
1A	VAR4	
1B	VAR5	
1C	VAR6	
1D	VAR7	
1E	VAR8	
1F	VAR9	
20	WNDLFT	Marge gauche
21	WNDWDTH	Largeur fenêtre
22	WNDTOP	Haut de fenêtre
23	WNDBTM	Bas de fenêtre
24	CH	Position curseur horiz.
25	CV	Position curseur verti.
30	HMASK	Masque bit/point
03D0	VECDOS	Met le ProDOS en ligne
03EA	INTDOS	Reprise des E/Sorties
03F5	AMPER	& Ampersand ($4C)
03F6	AMPERB	& Partie basse
03F7	AMPERH	& Partie haute
C000	KBD	Saisie touche sans écho
C010	KBDSTRB	Nettoie le bit 7 $C000
C030	SPKR	Activation HP
DB3A	STROUT	Affiche une chaîne par Y A
DB5C	OUTDO	Affiche l'accumulateur
F3D8	HGR2	Effacement des pages
FBC1	BASCALC	Calcul adresse écran
FC22	VTAB	Déplace curseur en CV
FC42	CLREOP	Eff. jusque bas écran
FC58	HOME	Efface l'écran
FC62	CR	Envoie retour chariot
FCA8	WAIT	Boucle d'attente (Acc)
FD1B	KEYIN	Entrée carac. avec écho
FD0C	RDKEY	Fait clignoter curseur
FDDA	PRBYTE	Aff. acc. héxa sur 2 octets
FDED	COUT	Sortie accumulateur
FF3A	BELL	Biiip

programme atteint la limite maximum autorisée, on a le temps de faire la pause-café. C'est un peu comme avec un compilateur, mais ce n'est pas une raison pour se priver des (bons) services de TRANSLIST !

RESTRICTIONS

Suivant la version de votre Apple, certaines instructions (et notamment les nouvelles, propres au 65C02 ... du IIc) ne seront pas reconnues par TRANSLIST... et seront remplacées par DFB (code non interprété). Votre assembleur vous permettra de remettre de l'ordre dans tout cela.

Et maintenant, à vous de jouer ! Naturellement, si vous imaginez des améliorations capables de rendre cette version plus attrayante et plus performante, écrivez à son auteur...

LABELS.CREER

```
100 TEXT : PRINT CHR$ (21): HOME :D$
    = CHR$ (4):CM$ = "LABELS": PRIN
    T D$"OPEN"CM$: PRINT D$"READ"CM$
    : ONERR GOTO 110
105 FOR NB = 1 TO 300: INPUT L1$: IN
    PUT L2$: INPUT L3$: NEXT
110 PRINT D$"CLOSE"CM$:NB = NB - 1:
    REM NB=NBR.LABELS POUR AJOUTS
115 HOME : HTAB 5: VTAB 1: INVERSE :
     PRINT ": CREER OU COMPLETER 'LA
    BELS' :": HTAB 1: VTAB 8: PRINT
    "ADRESSE":: NORMAL : PRINT "  $"
    : INVERSE
120 HTAB 1: VTAB 11: PRINT "LABEL":
    HTAB 1: VTAB 14: PRINT "COMMENTA
    IRE": NORMAL : HTAB 1: VTAB 16:
    PRINT ":": HTAB 30: VTAB 6: PRIN
    T "->"NB
125 HTAB 11: VTAB 8: INPUT "":AD$: I
    F AD$ = "" THEN 125
130 IF AD$ = "*" THEN HOME : PRINT D
    $"CAT": END
135 HTAB 10: VTAB 11: INPUT "";LA$:
    IF LA$ = "" THEN 135
140 HTAB 3: VTAB 16: INPUT "";CO$: I
    F CO$ = "" THEN 140
145 PRINT D$"APPEND"CM$: PRINT AD$:
    PRINT LA$: PRINT CO$: PRINT D$"C
    LOSE":NB = NB + 1: HTAB 32: VTAB
     6: PRINT NB: GOTO 125
```

• TRANSLIST

```
100 ONERR GOTO 785: REM TRAITEMENT E
    RREUR
105 CLEAR : LOMEM: 12288: TEXT : PRI
    NT CHR$ (21): HOME :D$ = CHR$ (4
    ):R$ = CHR$ (13):G$ = CHR$ (7):T
    $ = "                    :": HTAB 13: VT
    AB 1: INVERSE : PRINT T$: HTAB 1
    3: VTAB 2: PRINT "  TRANSLIST  "
    : HTAB 13: VTAB 3: PRINT T$: NOR
    MAL
110 T$ = "": FOR I = 1 TO 39:T$ = T$
     + "_": NEXT :PF$ = "TM8": GOSUB
     820: REM CHARGE LM
115 HTAB 30: VTAB 1: PRINT "(C) D.LU
    GAT": HTAB 30: VTAB 3: PRINT "VE
    RSION 2.1"
120 HTAB 1: VTAB 10: PRINT "NOM OBJE
    T": HTAB 11: VTAB 13: PRINT "$30
    0 <--> $3CF": HTAB 3: VTAB 22: P
    RINT "?/VOLUME (/SOUS-VOL) POUR
    CATALOGUE"
125 HTAB 11: VTAB 10: PRINT "/VOLUME
    /FICHIER": CALL C:H = 11:V = 10:
    LO = 30: INVERSE : GOSUB 165: NO
    RMAL :N$ = PHR$
130 IF LEFT$ (N$,1) = "?" THEN HOME
    : PRINT D$"PREFIX": PRINT D$"CATAL
    OG" RIGHT$ (N$, LEN (N$) - 1):CT
    $ = "<TOUCHE RETOUR MENU>": GOSU
    B 790: RUN
135 OG$ = "300":CM$ = "LABELS":LO =
    300: GOTO 245: REM CONSTANTES
140 :
145 REM **********************
150 REM *    SP.SAISIE      *
155 REM **********************
160 :
165 PHR$ = "": FOR I = 0 TO LO - 1
170 HTAB H + I: VTAB V: POKE - 16368
    ,0: GET CAR$: GOSUB 215
175 IF I = 0 THEN HTAB H: VTAB V: PR
    INT SPC( LO)
180 IF CAR$ = CHR$ (27) THEN 165
185 IF CAR$ = R$ THEN 205
190 IF CAR$ = CHR$ (127) AND I > 0 T
    HEN PRINT CHR$ (8);: PRINT " ";:
     PRINT CHR$ (8);:I = I - 1:PHR$
    = MID$ (PHR$,1,I): GOTO 170
195 IF (CAR$ < CHR$ (33) OR CAR$ > C
    HR$ (63)) AND (CAR$ < CHR$ (65)
    OR CAR$ > CHR$ (90)) THEN 170
200 HTAB H + I: VTAB V: PRINT CAR$;:
    PHR$ = PHR$ + CAR$: NEXT
205 IF PHR$ = "" THEN 165
210 FOR P = 1 TO 200: NEXT : GOSUB 2
    15: RETURN
215 FOR ZZ = 1 TO 4:Z = PEEK ( - 163
    36): NEXT : RETURN : REM ZZZ!
220 :
225 REM **********************
230 REM *   LECTURE ECRAN    *
235 REM **********************
240 :
245 HOME : TEXT : POKE 34,1: POKE 35
    ,20:LI = 1: DIM S1$(300),S2$(300
    ),S3$(300),ES$(300)
250 FOR I = LEN (N$) TO 1 STEP - 1:C
    AR$ = MID$ (N$,I,1): IF CAR$ < >
     "/" THEN NO$ = NO$ + CAR$
255 IF CAR$ = "/" THEN FOR I = LEN (
    NO$) TO 1 STEP - 1:NN$ = NN$ + M
    ID$ (NO$,I,1): NEXT :NO$ = NN$:
    GOTO 265
260 NEXT :NO$ = N$: REM CONSERVE NOM
     SANS DIRS
265 IF RIGHT$ (NO$,2) = ".0" OR RIGH
    T$ (NO$,2) = ".C" THEN NO$ = LEF
    T$ (NO$, LEN (NO$) - 2)
270 IF RIGHT$ (NO$,5) = ".OBJ0" THEN
     NO$ = LEFT$ (NO$, LEN (NO$) - 5
    )
275 CALL C: HTAB 1: VTAB 12: PRINT "
    ********** CREATION SOURCE BASE"
280 HTAB 11: VTAB 14: PRINT "* "NO$:
     HTAB 11: PRINT "*": HTAB 30: VT
    AB 14: INVERSE : PRINT "CTL-C FI
    N  ";: HTAB 30: PRINT "CTL-S PAU
    SE": NORMAL : HTAB 17: VTAB 16:
    PRINT "ORG   $"OG$: HTAB 11: PRI
    NT "*";
285 HTAB 1: VTAB 21: PRINT T$;: GOSU
    B 755
290 :
295 REM **********************
300 REM * ANALYSE LIGNE 22 *
305 REM **********************
310 :
315 ONERR GOTO 455
320 FOR LI = 1 TO LO:ES$(LI) = " "
```

→

TRANSLIST (suite)

```
325 CALL B: REM DESASSEMBLE UNE INST
    RUCTION SUR 1,2 OU 3 OCTETS
330 FOR I = 0 TO 3:AD$ = AD$ + CHR$
    ( PEEK (1872 + I) - 128): NEXT :
     REM LIT ADR.
335 FOR I = 8 TO 9:C1$ = C1$ + CHR$
    ( PEEK (1872 + I) - 128): NEXT :
     FOR I = 11 TO 12:C2$ = C2$ + CH
    R$ ( PEEK (1872 + I) - 128): NEX
    T : FOR I = 14 TO 15:C3$ = C3$ +
     CHR$ ( PEEK (1872 + I) - 128):
    NEXT : REM LIT CODES
340 IF C2$ = "  " THEN CX = 1: GOTO
    355: REM NBR.CODES
345 IF C3$ = "  " THEN CX = 2: GOTO
    355
350 CX = 3
355 CPT = CPT + CX: REM CUMUL CODES
    PAR INSTRUCTION
360 FOR I = 20 TO 22:MN$ = MN$ + CHR
    $ ( PEEK (1872 + I) - 128): NEXT
     : REM LIT MNEMONIQUE
365 FOR I = 26 TO 34:OP$ = OP$ + CHR
    $ ( PEEK (1872 + I) - 128): NEXT
     : REM LIT OPERANDE
370 IF RIGHT$ (OP$,1) = " " THEN OP$
     = MID$ (OP$,1, LEN (OP$) - 1):
    GOTO 370: REM LES ESPACES SONT D
    E TROP
375 IF MN$ = "???" THEN GOSUB 410: G
    OTO 370: REM ASCIIS, OU 65C02 NO
    N DESASSEMBLE
380 INC = INC + 1: GOSUB 425
385 HTAB 17: PRINT MN$;: HTAB 23: PR
    INT OP$;: REM SORTIE VIDEO
390 S1$(LI) = AD$:S2$(LI) = MN$:S3$(
    LI) = OP$
395 IF MN$ = "RTS" THEN GOSUB 415: G
    OSUB 420: VTAB 20: PRINT : GOSUB
     425: PRINT " *";
400 IF CPT = CT THEN 455: REM SORTIE
     LECTURE
405 C1$ = "":C2$ = "":C3$ = "": GOSU
    B 415: NEXT LI: GOTO 455
410 GOSUB 415:MN$ = "DFB $" + C1$: R
    ETURN : REM 'HEX' EQUIVALANT À '
    DFB'(DIRECTIVE ASSEMBLEUR)
415 AD$ = "":MN$ = "":OP$ = "": RETU
    RN
420 INC = INC + 1:LI = LI + 1:ES$(LI
    ) = "*": RETURN
425 INC$ = STR$ (INC): VTAB 19: HTAB
     (12 - LEN (INC$)): PRINT INC;".
    ";: RETURN : REM ECRITURE A UNE
    TABULATION VARIABLE EN FONCTION
    DE LA LONGUEUR DU CHIFFRE.
430 :
435 REM ********************
440 REM * TRAITEMENT LABELS*
445 REM ********************
450 :
455 PRINT : PRINT : CALL C: PRINT "*
    ********* RECHERCHE LABELS SYSTE
    ME": ONERR GOTO 475
460 DIM L1$(300),L2$(300),TA$(200),C
    M$(200): REM TABLEAUX LABELS
465 PRINT D$"OPEN"CM$: PRINT D$"READ
    "CM$: REM LECTURE FICHIER PRE-EN
    REGISTRE
470 FOR I = 1 TO 300: INPUT L1$(I):L
    1$(I) = "$" + L1$(I): INPUT L2$(
    I): INPUT CM$(I): NEXT
475 PRINT D$"CLOSE"CM$: PRINT : CALL
     C: PRINT "<"I - 1"> LABELS EN S
    TOCK": PRINT : PRINT "PATIENCE..
    .";
480 :
485 REM ********************
490 REM * REMPLACEMENTS    *
495 REM ********************
500 :
505 FOR J = 1 TO I - 1: REM NBR.LABE
    LS
510 FOR K = 1 TO LI: IF S3$(K) = L1$
    (J) THEN S3$(K) = L2$(J):L = L +
     1:TA$(L) = S3$(K) + " EQU " + L
    1$(J) + " ;" + CM$(J): PRINT "."
    ;:Z = PEEK ( - 16336): REM CALCU
    L DU LABEL CORRESPONDANT
515 NEXT : NEXT
520 :
525 FOR I = 1 TO L: REM SUPPRIME LES
     DOUBLES
530 IF TA$(I) = TA$(I + 1) THEN 540
535 O = O + 1:TA$(O) = TA$(I)
540 NEXT :L = O
545 :
550 REM ********************
555 REM * LOCALISATION BRA.*
560 REM ********************
565 :
570 PRINT : PRINT : CALL C: PRINT "*
    ********* CALCULS BRANCHEMENTS":
```

```
    PRINT
575 DIM T1$(300): REM TABLEAU LABELS
580 FOR I = 1 TO LI: REM RECHERCHE B
    RANCHEMENTS INCONDITIONNELS ET A
    PPELS DE SOUS-PROGRAMMES
585 IF S2$(I) = "JSR" OR S2$(I) = "J
    MP" OR S2$(I) = "BRA" THEN GOSUB
     595
590 NEXT : GOTO 615
595 Z = PEEK ( - 16336):AD$ = S3$(I)
    : FOR J = 1 TO LI: IF "$" + S1$(
    J) = AD$ AND T1$(J) = "" THEN LA
     = LA + 1:T1$(J) = "SPG" + STR$
    (LA):S3$(I) = T1$(J): RETURN
600 IF "$" + S1$(J) = AD$ AND T1$(J)
     < > "" THEN S3$(I) = T1$(J): RE
    TURN
605 NEXT : RETURN
610 :
615 FOR I = 1 TO LI: REM RECHERCHE B
    RANCHEMENTS CONDITIONNELS
620 IF S2$(I) = "BEQ" OR S2$(I) = "B
    NE" OR S2$(I) = "BMI" OR S2$(I)
    = "BPL" OR S2$(I) = "BCS" OR S2$
    (I) = "BCC" OR S2$(I) = "BVS" OR
     S2$(I) = "BVC" THEN GOSUB 630
625 NEXT : GOTO 670
630 Z = PEEK ( - 16336):AD$ = S3$(I)
    : FOR J = 1 TO LI: IF "$" + S1$(
    J) = AD$ AND T1$(J) = "" THEN LA
     = LA + 1:T1$(J) = "BCL" + STR$
    (LA):S3$(I) = T1$(J): RETURN
635 IF "$" + S1$(J) = AD$ AND T1$(J)
     < > "" THEN S3$(I) = T1$(J): RE
    TURN
640 NEXT : RETURN
645 :
650 REM **********************
655 REM * ENR.FICH.SOURCE   *
660 REM **********************
665 :
670 PRINT : PRINT "********** ENREGI
    STREMENT SOURCE": PRINT : ONERR
    GOTO 785
675 CALL C: PRINT D$"OPEN SOURCE": P
    RINT D$"CLOSE": PRINT D$"DELETE
    SOURCE": PRINT D$"OPEN SOURCE":
    PRINT D$"WRITE SOURCE": PRINT "*
     "NO$: PRINT "*": IF T1$(1) = ""
     THEN T1$(1) = "PROG"
680 FOR I = 1 TO L: PRINT TA$(I): NE
    XT
685 PRINT "*": PRINT " ORG $"OG$: PR
    INT "*"
690 FOR LT = 1 TO LI
695 RE$ = T1$(LT) + ES$(LT) + S2$(LT
    )
700 IF S3$(LT) < > "" THEN RE$ = RE$
     + CHR$ (32) + S3$(LT)
705 IF E$(LI) = "*" AND LT = LI THEN
     720: REM SI RTS EN FIN DE PROG,
     PAS D'ETOILE EN SUIVANT
710 PRINT RE$:RE$ = ""
715 NEXT
720 PRINT D$"CLOSE": CALL D
725 GOTO 890: REM OU CT$ = "<TOUCHE>
    ": GOSUB 1500: CALL C: TEXT : HO
    ME : END...ET DEL 1620,1820 POUR
     SUPPRIMER LA PARTIE FACULTATIVE
     DU PROGRAMME
730 :
735 REM **********************
740 REM * CHARG.TYPE PROG.  *
745 REM **********************
750 :
755 PRINT D$"BLOAD"N$",A$"OG$: CALL
    A:CT = PEEK (48840) + PEEK (4884
    1) * 256: RETURN : REM CT=LONG.F
    ICHIER PRODOS
760 :
765 REM **********************
770 REM * ERREURS SYSTEME   *
775 REM **********************
780 :
785 PRINT D$"CLOSE": TEXT : HOME : P
    RINT "ERREUR PRODOS N°" PEEK (22
    2);: FOR I = 1 TO 1200: NEXT : R
    UN
790 PRINT : CALL C: HTAB 30: PRINT C
    T$: POKE - 16368,0: WAIT - 16384
    ,128,127: POKE - 16368,0: RETURN
     : REM SP.SAISIE TOUCHE
795 :
800 REM **********************
805 REM *    CHARGE L.M.     *
810 REM **********************
815 :
820 A = 12032: REM TRANSMISSION AUTO
    MATIQUE DE L'IMPLANTATION
825 B = 12043: REM DESASSEMBLAGE D'U
    NE LIGNE
830 C = 12096: REM BIP PARAMETRABLE
835 D = 12108: REM SIRENE FIN TRAITE
    MENT
840 FOR ADR = 12032 TO 12143: READ C
```

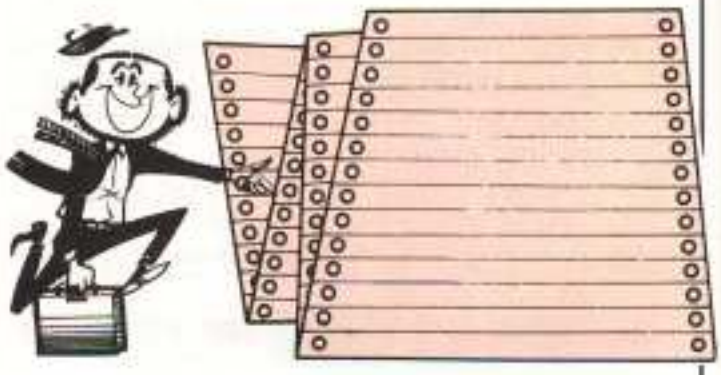
→

TRANSLIST (suite)

```
    ODE: POKE ADR,CODE: NEXT : RETUR
    N
845 DATA 173,185,190,133,8,173,186,1
    90,133,9,96,165,37,72,169,25,133
    ,36,169,22,133,37,32,34,252,32,1
    56,252,165,8,133,60,164,9,132,61
    ,162,1,32,120,254,169,1,32,99,25
    4,165,58,133,8,165,59,133,9,104,
    56,233,1,133,37,32,34,252,96
850 DATA 162,20,44,48,192,136,208,25
    3,202,208,247,96,169,0,141,115,4
    7,169,6,141,116,47,173,48,192,13
    6,240,9,202,208,250,174,115,47,7
    6,86,47,238,115,47,208,239,206,1
    16,47,208,234,96
855 :
860 REM ---- PARTIE FACULTATIVE
865 :
870 REM **********************
875 REM * EDITION ECRAN     *
880 REM **********************
885 :
890 PRINT G$: PRINT : PRINT "DEMONST
    RATION <E>CRAN OU <I>MPRIMANTE "
    ;: POKE - 16368,0: GET X$: IF X$
     = "I" OR X$ = "i" THEN IM = 1
895 IF IM = 0 THEN TEXT : HOME : PRI
    NT D$"PR£3": PRINT : INVERSE : P
    RINT "Ctrl-S pour stopper le lis
    ting... un peu de patience, merc
    i": NORMAL
900 IF IM = 1 THEN PRINT D$"PR£1": P
    RINT CHR$ (9)"80N"
905 PRINT : HTAB 10: PRINT "* FICHIE
    R SOURCE "NO$: HTAB 10: PRINT "*
    ": DIM M1$(100),M2$(100)
910 FOR I = 1 TO L: FOR J = 1 TO LEN
     (TA$(I)): REM RECONSTITUTION DE
    S VARIABLES POUR TABULER
915 PA$ = MID$ (TA$(I),J,1): IF PA$
    = " " THEN M1$(I) = LEFT$ (TA$(I
    ),J):J = J + 4:M2$(I) = MID$ (TA
    $(I),J): GOTO 925
920 NEXT J
925 NEXT I
930 FOR I = 1 TO L: FOR J = 1 TO LEN
     (M2$(I))
935 PA$ = MID$ (M2$(I),J,1): IF PA$
    = ";" THEN M3$(I) = MID$ (M2$(I)
    ,J - 1):M2$(I) = LEFT$ (M2$(I),J
     - 1): GOTO 945
940 NEXT J
945 NEXT I
950 FOR I = 1 TO L: HTAB 10: PRINT M
    1$(I);: HTAB 20: PRINT "EQU";: H
    TAB 29: PRINT M2$(I);: IF IM = 0
     THEN POKE 1403,40: REM ECRAN
955 IF IM = 1 THEN POKE 36,40: REM I
    MPRIMANTE
960 PRINT M3$(I): NEXT I: HTAB 10: P
    RINT "*": HTAB 20: PRINT "ORG
        $"OG$: HTAB 10: PRINT "*"
965 :
970 FOR P = 1 TO 3000: NEXT : FOR LT
     = 1 TO LI: HTAB 10: PRINT T1$(L
    T);: HTAB 20: PRINT S2$(LT);: HT
    AB 30: PRINT S3$(LT): FOR P = 1
    TO 500: NEXT : NEXT : HTAB 10: P
    RINT "*": HTAB 10: PRINT "*
        FIN"
975 PRINT G$: HTAB 30: PRINT "<M>enu
     <B>asic ":: POKE - 16368,0: GET
     X$: IF X$ = "M" OR X$ = "m" THE
    N PRINT D$"/TM8/MENU"
980 PRINT : END
```

LIT.LABELS

```
100 TEXT : HOME :D$ = CHR$ (4): PRIN
    T D$"PR£3": PRINT D$"OPEN LABELS
    ": PRINT D$"READ LABELS": ONERR
    GOTO 120
110 FOR I = 1 TO 300: INPUT A$: INPU
    T B$: INPUT C$: HTAB 1: PRINT I;
    : HTAB 12: PRINT B$;: HTAB 25: P
    RINT "EQU";: HTAB 30: PRINT "$"A
    $;: HTAB 40: PRINT "; "C$: FOR J
     = 1 TO 200: NEXT J: NEXT I
120 PRINT D$"CLOSE": END
```



Votre bibliothèque INFORMATIQUE

par Clément RENARD

• INITIATION À L'ALGORITHMIQUE (Belin)

Pour Chantal et Patrice Richard, tout enseignement sérieux de la programmation doit reposer sur une *initiation systématique à l'algorithmique*. C'est une idée qui a fait son chemin. De nombreux organismes (Ministère de l'Education Nationale, dans le cadre du plan "Informatique pour tous", Renault, Crédit Lyonnais, etc.) l'ont adoptée.

La première édition de cet ouvrage, parue en 1981, avait bénéficié d'un excellent accueil, mais il est probable que cette nouvelle édition — entièrement revue (exercices plus nombreux, mise en page originale...) — fera le bonheur d'un public varié : lycéens de première et de terminale, étudiants, enseignants, informaticiens professionnels... et autodidactes.

Pour concevoir un programme, il convient évidemment, dans un premier temps, d'analyser le problème, puis de concevoir l'algorithme permettant d'aboutir le plus simplement possible, à la solution.

Les auteurs proposent donc une méthode d'apprentissage accessible à tous les débutants, quelle que soit leur formation : ALADIN 135 exercices corrigés... mais aussi des explications claires, souvent illustrées par des tableaux significatifs. Rappelons que les auteurs ont obtenu, en 1984, pour leur ouvrage *Programmatique*, le Grand prix de la littérature micro-informatique.

(Editions Belin, 288 pages, moins de 100 F).

• APPLE 2 : TRUCS ET ASTUCES (MICRO APPLICATION)

Une traduction de l'édition allemande, publiée par DATA BECKER. On peut la recommander à tout utilisateur d'Apple... débutant.

On y apprendra en effet, en 300 pages, tout ce qu'il faut savoir pour tirer le meilleur parti possible d'un Apple *IIe* ou *IIc* : éléments fondamentaux du Basic, périphériques, accès mémoire et langage machine (rudiments), programmes graphiques, amélioration des structures d'un programme, etc.

Le plan de ce recueil est judicieux et le résultat intéressant. Un regret : les pages sortent directement d'un traitement de texte... sans aucune recherche typographique. C'est dommage, mais cette lacune n'entame en rien le sérieux de l'ouvrage.

(Micro Application, 300 pages, moins de 150 F).

LANGAGE MACHINE | DOS | ProDOS

INITIATION

VISUMEM

DEMO

Vous désirez visualiser, une à une, les n lignes d'une routine en langage machine ?

C'est tout simple : il suffit d'utiliser **VISUMEM**

Ce programme démo va non seulement installer VISUMEM à partir de l'adresse **$300**, mais vous montrer comment cela fonctionne.

Si vous désirez conserver VISUMEM sous forme de fichier B, lancez la DEMO par RUN, choisissez l'option **3** (FIN), puis tapez : **BSAVE VISUMEM, A768, L128**

```
100 TEXT : PRINT CHR$ (21): HOME
110 FOR I = 768 TO 895: READ R: POKE I,R: NEXT
120 DATA 32,88,252,160,23,185,104,3,153,212,7,136,208,24
    7,169,23,133,37,32,34,252,169,29,133,36,32,228,251,3
    2,111,253,224,0,240,69,32,199,255,32,167,255,166,62,
    165,63,134,58,133,59,165,59
130 DATA 72,165,58,72,160,0,177,58,72,169,1,32,99,254,10
    4,201,96,208,10,169,48,141,217,7,169,54,141,216,7,16
    9,34,133,36,104,170,104,32,36,237,172,0,192,16,251,1
    92,155,140,16,192,240,154,208
140 DATA 201,96,193,196,210,197,211,211,197,160,8,5,24,1
    ,160,196,197,160,196,197,208,193,210,212,186
150 INVERSE :T$ = ":                    :": PRINT T$: PRINT "
      V I S U M E M  ": PRINT T$: NORMAL
160 PRINT : PRINT "IL SUFFIT DE TAPER CALL 768  POUR ACCE
    DERA LA ROUTINE VISUMEM.": PRINT : PRINT "ENSUITE, A
    PPUYER SUR UNE TOUCHE POUR VI-SUALISER CHAQUE LIGNE.
    ": PRINT : PRINT "ESCAPE POUR CHANGER D'ADRESSE, ET
    RETURNPOUR TERMINER."
170 PRINT : PRINT "ACCES DIRECT PAR L'AMPERSAND (&-RETUR
    N) A CONDITION DE TAPER CES TROIS POKES:": PRINT : P
    RINT "  - POKE 1013,76:POKE 1014,0:POKE1015,3"
180 PRINT : PRINT "LE PROGRAMME VOUS DEMANDERA L'ADRESSE
     DEDEPART (HEXA), DESASSEMBLERA LA PREMIEREINSTRUCTI
    ON ET AFFICHERA SON ADRESSE DE-CIMALE."
190 PRINT : PRINT "ENFONCEZ UNE TOUCHE S.V.P. ":: CALL -
    198: GET R$: PRINT : CALL 768
200 HOME : VTAB 22: PRINT "<1> ENCORE <2> MENU DISQUETTE
    <3> FIN ":: CALL - 198: GET R$: PRINT
210 HOME : IF R$ = "1" THEN 150
220 IF R$ = "2" THEN PRINT CHR$ (4)"RUN MENU"
230 IF R$ < > "3" THEN 200
```

■

EXPLICATION DE LA ROUTINE (DATA)

0300-	20 58 FC	JSR	$FC58	C'est le HOME du Basic
0303-	A0 17	LDY	£$17	Y = $17 (23) : nbre d'octets à afficher
0305-	B9 68 03	LDA	$0368,Y	Adresse de lecture du texte + Y
0308-	99 D4 07	STA	$07D4,Y	Adresse-écran d'écriture + Y (VTAB24)
030B-	88	DEY		Y = Y — 1
030C-	D0 F7	BNE	$0305	Si Y <> 0 GOTO $305
030E-	A9 17	LDA	£$17	A = $17 (23... pour VTAB24)

→

```
0310-  85 25      STA  $25      CV (position verticale du curseur) = $17
0312-  20 22 FC   JSR  $FC22    VTAB va déplacer le curseur en CV
0315-  A9 1D      LDA  #$1D     A = $1D (29)
0317-  85 24      STA  $24      CH (position horizontale du curseur) = $1D
0319-  20 E4 FB   JSR  $FBE4    BELL2 pour 1 bip
031C-  20 6F FD   JSR  $FD6F    Une autre entrée de GETLIN (nbre de carac. dans X)
031F-  E0 00      CPX  #$00     X (longueur du nombre saisi) est-il égal à 0 ?
0321-  F0 45      BEQ  $0368    Si oui GOTO RTS (Basic)
0323-  20 C7 FF   JSR  $FFC7    LZMODE (0 mode)
0326-  20 A7 FF   JSR  $FFA7    GETNUM (récupère nombre hexa)
0329-  A6 3E      LDX  $3E      ┐
032B-  A5 3F      LDA  $3F      │ Récupération des valeurs en $3E-3F
032D-  86 3A      STX  $3A      │ et écriture en $3A-3B
032F-  85 3B      STA  $3B      ┘
0331-  A5 3B      LDA  $3B      Accumulateur chargé avec $3B
0333-  48         PHA           Valeur de A empilée
0334-  A5 3A      LDA  $3A      A chargé avec $3A
0336-  48         PHA           Valeur également empilée
0337-  A0 00      LDY  #$00     Y = 0 pour adressage indexé
0339-  B1 3A      LDA  ($3A),Y  Lecture à l'adresse $3A-3B + Y
033B   48         PHA           Valeur empilée
033C-  A9 01      LDA  #$01     A = 1
033E-  20 63 FE   JSR  $FE63    LIST2 désassemble A instruction (1)
0341-  68         PLA           A récupère le sommet de la pile
0342-  C9 60      CMP  #$60     Est-ce $60 (RTS) ?
0344-  D0 0A      BNE  $0350    Non : GOTO 350
0346-  A9 30      LDA  #$30     Oui : alors A = $30 (0 mode inverse)...
0348-  8D D9 07   STA  $07D9    affiché à la place du 0 normal
034B-  A9 36      LDA  #$36     A = $36 (6 mode inverse)...
034D-  8D D8 07   STA  $07D8    affiché à la place du 6 normal
0350-  A9 22      LDA  #$22     A = $22 (34)
0352-  85 24      STA  $24      CH = $22
0354-  68         PLA           Récupération du sommet de la pile
0355-  AA         TAX           A passe dans X
0356-  68         PLA           Dernière récupération
0357-  20 24 ED   JSR  $ED24    LINPTR affiche la valeur décimale de l'adresse
035A-  AC 00 C0   LDY  $C000    KBD (caractère frappé dans Y)
035D-  10 FB      BPL  $035A    Pas de caractère, on attend (GOTO 35A)
035F-  C0 9B      CPY  #$9B     Est-ce $9B (ESCAPE) ?
0361-  8C 10 C0   STY  $C010    Ecriture KBDSTRB (pour lire une autre touche)
0364-  F0 9A      BEQ  $0300    Si ESCAPE GOTO $300
0366-  D0 C9      BNE  $0331    Ou bien GOTO $331 pour autre saisie
0368-  60         RTS           FIN DE PROGRAMME

0369-  C1 C4      0374-  01 A0   ┐
036B-  D2         0376-  C4 C5   │
036C-  C5 D3      0378-  A0 C4   │ De $369 à 37F, texte suivant :
036E-  D3         037A-  C5 D0   │
036F-  C5 A0      037C-  C1 D2   │ ADRESSE HEXA DE DÉPART :
0371-  08         037E-  D4      │
0372-  05 18      037F-  BA      ┘
```

EN UTILISATION NORMALE, sans programme DEMO, taper BLOAD VISUMEM, puis CALL 768. Vous pouvez aussi utiliser l'AMPERSAND (voir DEMO).

MIXAGE

Comment mêler vos pages graphiques !

PAGE 1 PLUS PAGE 2

UNE technique simple permet de mixer les deux pages graphiques en un temps record... grâce à l'extraordinaire rapidité du langage machine et à l'opération logique ORA.

POURQUOI ORA PLUTÔT QUE AND ou EOR ?

Imaginons que le premier octet de la Page 1 soit (en binaire) 00000011 et que le premier octet de la Page 2 soit 00001001. Il s'agit d'obtenir, en mixant les deux données, 00001011. Laissons de côté les quatre premiers zéros (inutiles dans cet exemple), et regardons le résultat obtenu à partir des opérations logiques AND, EOR et ORA.

On constate immédiatement que seul ORA fournit le résultat — ou le mixage ! — escompté... d'où le court programme que voici...

	AND	EOR	ORA
	0 0 1 1	0 0 1 1	0 0 1 1
	1 0 0 1	1 0 0 1	1 0 0 1
	0 0 0 1	1 0 1 0	1 0 1 1

```
0300-  A9 20     LDA   £$20
0302-  85 07     STA   $07
0304-  A9 40     LDA   £$40
0306-  85 09     STA   $09
0308-  A9 00     LDA   £$00
030A-  85 06     STA   $06
030C-  85 08     STA   $08
030E-  A2 20     LDX   £$20
0310-  A0 00     LDY   £$00
0312-  B1 06     LDA   ($06),Y
0314-  11 08     ORA   ($08),Y
0316-  91 08     STA   ($08),Y
0318-  C8        INY
0319-  D0 F7     BNE   $0312
031B-  E6 07     INC   $07
031D-  E6 09     INC   $09
031F-  CA        DEX
0320-  D0 EE     BNE   $0310
0322-  60        RTS
```

$6 et $7 contiennent 00 20 pour lire la Page 1 (avec LDA ($06),Y).
$8 et $9 contiennent 00 40, adresse de la Page 2

Nombre de pages

Cette boucle permet de lire chaque octet de la Page 1, de le mixer avec l'octet correspondant de la Page 2, puis d'écrire le résultat en Page 2 (en écrasant l'ancien).

Incrémentation de la partie haute des adresses

X = X — 1

Si X différent 0 GOTO $310

FIN DE PROGRAMME

MIXAGE-DEMO

Cette courte démonstration vous permettra en outre d'installer **MIXAGE** à partir de l'adresse $300. Pour sauver cette routine LM, choisissez l'option **(B)ASIC** du menu final, puis tapez un simple **BSAVE MIXAGE, A$300, L$23**

```
100 D$ = CHR$ (13) + CHR$ (4):G$ = CHR$ (7): GOSUB 330:
    ONERR GOTO 110
110 HGR : HGR2 : HCOLOR= 3: TEXT : PRINT CHR$ (21): HOME
     : INVERSE : HTAB 10: VTAB 8: PRINT ": MIXAGE GRAPHI
    QUE :": NORMAL : HTAB 1: VTAB 12: PRINT "      Nous
    allons voir deux images,": PRINT
120 PRINT "   l'une en page 1, l'autre en page 2,": PRIN
    T : PRINT "    et enfin, leur mixage en page 2.": VT
    AB 20: PRINT "Appuyez sur <ESPACE> après chaque arre
    t": PRINT : HTAB 18: VTAB 4: PRINT "( )": GOSUB 310
130 :
140 REM **********************
150 REM * DESSIN PAGE 1     *
160 REM **********************
170 :
180 HGR :MA = 23:DI = 6:IN = 50:AA = 90:BB = 90:ST = .5:
     FOR I = 1 TO MA STEP ST:A = (( SIN (I) * 10) + I *
    I / DI) + IN + 40:B = (( COS (I) * 30) + I * I / DI)
     + IN: HPLOT AA,BB TO A,B:AA = A:BB = B: NEXT
190 :
200 REM **********************
210 REM * DESSIN PAGE 2     *
220 REM **********************
230 :
240 GOSUB 310: HGR2 :A = 10:B = 270:C = 155: HPLOT A,A T
    O B,A: HPLOT B,A TO B,C: HPLOT B,C TO A,C: HPLOT A,C
     TO A,A: GOSUB 310
250 :
260 CALL 768: REM VERS ROUTINE LM
270 :
280 GOSUB 310: TEXT : HOME : HTAB 3: VTAB 23: PRINT "<E>
    NCORE <B>ASIC <M>ENU DISQUETTE ":: POKE - 16368,0: G
    ET X$: IF X$ = "M" OR X$ = "m" THEN PRINT D$"RUN MEN
    U"
290 IF X$ = "E" OR X$ = "e" THEN 110
300 IF X$ = "B" OR X$ = "b" THEN HOME : END
310 POKE - 16368,0: VTAB 4: HTAB 19: GET X$: PRINT CHR$
    (7): RETURN
320 :
330 FOR ADR = 768 TO 802: READ CODE: POKE ADR,CODE: NEXT
     : RETURN
340 DATA 169,32,133,7,169,64,133,9,169,0,133,6,133,8,162
    ,32,160,0,177,6,17,8,145,8,200,208,247,230,7,230,9,2
    02,208,238,96
```

UTILISATION

MIXAGE (LM) peut être employé en mode direct... ou dans un programme en Basic.

Dans un cas comme dans l'autre, le processus sera le même :

- ***BLOAD MIXAGE***
- ***BLOAD IMAGE1, A$2000***
- ***BLOAD IMAGE2, A$4000***
- ***CALL 768***
- ***BSAVE IMAGE3, A$4000, L$2000***

TABLE DE VÉRITÉ

Voici la table de vérité de ORA (si au moins 1 des 2 bits est à 1 le résultat est 1).

OU	0	1
0	0	1
1	1	1

UTILITAIRE ProDOS

Guy DESENFANT

CATALOGUE

NE vous privez surtout pas du Super.catalogue que vous propose Guy DESENFANT, mais attention ! il ne fonctionne que sous ProDOS, avec le 65CØ2 et une carte 8Ø colonnes ! Pour le reste, nous vous en laissons la surprise... et sur deux colonnes !

```
100 TEXT : PRINT CHR$ (21): HOME : P
    RINT CHR$ (4)"PR£3": PRINT CHR$
    (1): GOSUB 925: ONERR GOTO 870
105 :
110 REM ******************************
115 REM * ACTIVER LES CARAC SOURIS *
120 REM ******************************
125 :
130 I$ = CHR$ (15):N$ = CHR$ (14):M$
     = CHR$ (27):XM$ = CHR$ (24)
135 PRINT I$: PRINT M$
140 :
145 REM ******************************
150 REM *    TRAITS HORIZONTAUX      *
155 REM ******************************
160 :
165 TR$ = "": FOR I = 1 TO 10:TR$ =
    TR$ + "LLLLLLLL": NEXT :TR$ = LE
    FT$ (TR$,78)
170 RT$ = "": FOR I = 1 TO 10:RT$ =
    RT$ + "SSSSSSSS": NEXT :RT$ = LE
    FT$ (RT$,78)
175 :
180 REM ******************************
185 REM * BOUCLE TRACE TRAITS VERT *
190 REM ******************************
195 :
200 FOR I = 1 TO 20: VTAB I: HTAB 1:
     PRINT "Z": VTAB I: HTAB 80: PRI
    NT "_": NEXT I
205 :
210 REM ******************************
215 REM * BCL.TRACE TRAITS HORIZON *
220 REM ******************************
225 :
230 VTAB 1: HTAB 2: PRINT TR$: VTAB
    3: HTAB 2: PRINT RT$: VTAB 5: HT
    AB 2: PRINT RT$: VTAB 7: HTAB 2:
     PRINT RT$
235 VTAB 6: HTAB 18: PRINT "Z": VTAB
     6: HTAB 26: PRINT "Z": VTAB 6:
    HTAB 40: PRINT "Z": VTAB 6: HTAB
     57: PRINT "Z": VTAB 6: HTAB 65:
     PRINT "Z"
240 :
245 REM ******************************
250 REM * BARRE VERTICALE CENTRALE *
255 REM ******************************
260 :
265 FOR J = 8 TO 20: VTAB J: HTAB 40
    : PRINT "Z": NEXT : VTAB 21: HTA
    B 2: PRINT TR$
270 :
275 REM ******************************
280 REM * DESACTIVE CARACT. SOURIS *
285 REM ******************************
290 :
295 PRINT N$: PRINT XM$
300 :
305 REM ******************************
310 REM *     CURSEUR INVISIBLE      *
315 REM ******************************
320 :
325 POKE 2043,160
330 A$ = "C A T A L O G U E":A = LEN
     (A$)
335 VTAB 2: POKE 1403, INT (80 - A)
    / 2: PRINT A$
340 VTAB 4: HTAB 3: PRINT "Nom du vo
    lume: ": VTAB 4: HTAB 60: PRINT
```

```
    "Blocs libres: "
345 B$ = "FICHIER":C$ = "TYPE":D$ =
    "DATE"
350 VTAB 6: HTAB 3: PRINT B$: VTAB 6
    : HTAB 20: PRINT C$: VTAB 6: HTA
    B 32: PRINT D$: VTAB 6: HTAB 42:
     PRINT B$: VTAB 6: HTAB 59: PRIN
    T C$: VTAB 6: HTAB 71: PRINT D$
355 VTAB 23
360 DIM FICHIER$(51)
365 :
370 REM ******************************
375 REM * ON PREND PREFIX COURANT  *
380 REM ******************************
385 :
390 PRINT CHR$ (4)"PREFIX": INPUT PF
    X$
395 :
400 REM ******************************
405 REM * OUVERTURE DU FIC CATALOG *
410 REM ******************************
415 :
420 PRINT CHR$ (4)"OPEN"PFX$",TDIR"
425 PRINT CHR$ (4)"READ"PFX$
430 INPUT VOLUME$
435 INPUT TITRE$
440 INPUT BLANC$
445 VTAB 23:I = 1
450 VTAB 23: INPUT FICHIER$(I): IF F
    ICHIER$(I) = "" THEN 465
455 FICHIER$(I) = LEFT$ (FICHIER$(I)
    ,20) + "       " + MID$ (FICHIER$
    (I),31,9)
460 VTAB 23: IF FICHIER$(I) < > "" T
    HEN I = I + 1: GOTO 450
465 VTAB 23: IF I > 27 THEN FLAG = 1
470 INPUT BLOC$
475 BLOC$ = MID$ (BLOC$,15,3)
480 PRINT CHR$ (4)"CLOSE"
485 :
490 REM ******************************
495 REM *    VERS SP. AFFICHAGE    *
500 REM ******************************
505 :
510 GOSUB 740
515 VTAB 23
520 :
525 REM ******************************
530 REM *           MENU           *
535 REM ******************************
540 :
545 IF FLAG = 1 THEN VTAB 22: HTAB 2
    : PRINT "<1> Page n° 1 <2> Nouve
    au PREFIX <3> Quitter": CALL - 1
    98: GET R$: IF ASC (R$) < 49 OR
    ASC (R$) > 51 THEN 545
550 IF (FLAG = 1 AND R$ = "1") THEN
    VTAB 22: HTAB 2: PRINT SPC( 70):
     GOSUB 685: GOSUB 740
555 IF (FLAG = 1 AND R$ = "3") THEN
    HOME : END
560 IF (FLAG = 1 AND R$ = "2") THEN
    VTAB 22: HTAB 2: PRINT SPC( 70):
     GOTO 610
565 IF FLAG < > 1 THEN VTAB 22: HTAB
     2: PRINT "<1> Nouveau PREFIX <2
    > Quitter": CALL - 198: GET R$:
    IF ASC (R$) < 49 OR ASC (R$) > 5
    0 THEN 565
570 IF (FLAG < > 1 AND R$ = "1") THE
    N VTAB 22: HTAB 2: PRINT SPC( 40
    ): GOTO 610
575 PRINT
580 IF (FLAG < > 1 AND R$ = "2") THE
    N HOME : END
585 :
590 REM ******************************
595 REM * CHOIX D'UN NOUV. PREFIX  *
600 REM ******************************
605 :
610 VTAB 22: HTAB 2: INPUT "Nouveau
    PREFIX: ";PFX$
615 :
620 REM ******************************
625 REM *    POUR LES ETOURDIS !    *
630 REM ******************************
635 :
640 IF LEFT$ (PFX$,1) < > "/" THEN P
    FX$ = "/" + PFX$
645 PRINT CHR$ (4)"PREFIX"PFX$
650 GOSUB 685
655 CLEAR : GOTO 360
660 :
665 REM ******************************
670 REM *    ON EFFACE LA PAGE     *
675 REM ******************************
680 :
685 FOR J = 8 TO 20
690 VTAB J: HTAB 2: PRINT SPC( 37):
    VTAB J: HTAB 41: PRINT SPC( 37)
695 NEXT
700 VTAB 4: HTAB 19: PRINT SPC( 37):
     VTAB 4: HTAB 75: PRINT "   ".
705 VTAB 22
710 RETURN
715 :
```

CATALOGUE (suite et fin)

```
720 REM ******************************
725 REM * AFFICHAGE PREMIERE PAGE  *
730 REM ******************************
735 :
740 J = 1
745 VTAB 4: HTAB 19: PRINT VOLUME$:
    VTAB 4: HTAB 75: PRINT BLOC$
750 IF J < 14 THEN VTAB J + 7: HTAB
    3: PRINT FICHIER$(J)
755 IF J > 13 AND J < 27 THEN VTAB J
     - 6: HTAB 42: PRINT FICHIER$(J)
760 IF J > 26 AND FLAG THEN GOSUB 77
    5: GOTO 515
765 IF J = I AND FLAG = 0 THEN RETUR
    N
770 J = J + 1: GOTO 750
775 CALL - 198: VTAB 23: HTAB 1: PRI
    NT "Pour la 2* page appuyez sur
    une touche.": POKE 2043,160: GET
     R$: IF R$ = "" THEN 775
780 VTAB 23: CALL - 864
785 GOSUB 685
790 VTAB 4:, HTAB 19: PRINT PFX$: VTA
    B 4: HTAB 75: PRINT BLOC$
795 :
800 REM ******************************
805 REM * AFFICHAGE DEUXIEME PAGE  *
810 REM ******************************
815 :
820 J = 27
825 IF J < 40 THEN VTAB J - 19: HTAB
     3: PRINT FICHIER$(J)
830 IF J > 39 THEN VTAB J - 32: HTAB
     42: PRINT FICHIER$(J)
835 IF J = I THEN RETURN
840 J = J + 1: GOTO 825
845 :
850 REM ******************************
855 REM *   TRAITEMENT DES ERREURS *
860 REM ******************************
865 :
870 CALL 768
875 IF PEEK (222) = 255 THEN RUN
880 IF PEEK (222) = 6 OR PEEK (222)
    = 7 THEN VTAB 22: CALL - 864: CA
    LL - 198: VTAB 22: HTAB 1: PRINT
     "Fichier non trouvé ": FOR A =
    1 TO 3000: NEXT : VTAB 22: CALL
    - 864: GOTO 610
885 IF PEEK (222) = 8 THEN VTAB 22:
    CALL - 864: CALL - 198: VTAB 23:
     HTAB 1: PRINT "Erreur d' Entrée
    /Sortie": FOR A = 1 TO 3000: NEX
    T : VTAB 22: CALL - 864: GOTO 61
    0
890 IF PEEK (222) = 16 THEN VTAB 22:
     CALL - 864: CALL - 198: VTAB 22
    : HTAB 1: PRINT "Erreur de synta
    xe": FOR A = 1 TO 3000: NEXT : V
    TAB 22: CALL - 864: GOTO 610
895 VTAB 22: CALL - 864: RESUME
900 :
905 REM ******************************
910 REM *   TRAITEMENT ERREURS LM  *
915 REM ******************************
920 :
925 POKE 216,0: POKE 768,104: POKE 7
    69,168: POKE 770,104: POKE 771,1
    66: POKE 772,223: POKE 773,154:
    POKE 774,72: POKE 775,152: POKE
    776,72: POKE 777,96: RETURN
```

ORGANISATION D'UN TOURNOI

PROBLÈME :

Tirage du calendrier des rencontres, jouées sur une ou plusieurs journées, suivant le nombre d'équipes engagées. Chaque équipe doit évidemment jouer une fois contre chacune des autres. Mais il y a une contrainte : aucune des équipes ne doit être amenée à jouer deux matchs consécutifs (pour lui permettre de récupérer).

MA SOLUTION :

L'aléatoire plus la logique ! Vous noterez au passage que le nombre de rencontres est fourni par la formule suivante (ou N = nombre de participants) : NR = (N — 1) * N/2. Pour le reste, suivez le déroulement du programme... et vous comprendrez. Il est sans doute possible de faire mieux, mais à quel prix ? Telle qu'elle se présente, cette routine pourra probablement aider certains organisateurs à mettre au point leur calendrier.

```
100 TEXT : PRINT CHR$ (21): HOME : DIM A%(190,1),B%(191,
    1):D$ = CHR$ (4):T$ = "": FOR I = 1 TO 20:T$ = T$ +
    "----":V$ = V$ + "    ": NEXT
110 :
120 REM ********************
130 REM * NOMBRE D'EQUIPES *
140 REM ********************
150 :
160 PRINT D$"PR£3": PRINT : HOME : PRINT "ORGANISATION D
    'UN TOURNOI  (5 A 20 EQUIPES)": PRINT T$
170 E = 0: VTAB 4: CALL - 958: CALL - 198: INPUT "NOMBRE
     D'EQUIPES ":N$: IF N$ = "" THEN 600
180 N = VAL (N$): IF N < 5 OR N > 20 THEN 170
190 B = N - 1
200 E = E + B:B = B - 1: IF B < > 0 THEN 200
210 VTAB 1: POKE 1403,53: INVERSE : PRINT E" RENCONTRES
    A ORGANISER:": NORMAL
220 :
```

100 : Je n'ai prévu que 20 équipes, mais il est possible d'aller plus loin en modifiant DIM (nombre de rencontres), puis la ligne 180 (nombre d'équipes).

190 : Si vous préférez la formule indiquée plus haut, allez-y gaiement : B = (N — 1) * N/2 et supprimez la ligne 200 !

ORGANISATION D'UN TOURNOI (suite)

```
230 REM ********************
240 REM * REPARTITION      *
250 REM ********************
260 :
270 POKE 34,3: PRINT : HOME
280 VTAB 4: INPUT "NOMBRE MAXIMUM DE RENCONTRES PAR JOUR
    : ":N$: IF N$ = "" THEN 280
290 NM = VAL (N$): IF NM > E THEN 280
300 :
310 REM ********************
320 REM *    TRAITEMENT    *
330 REM ********************
340 :
350 HOME :M = 0:Y = M:R = M:M1 = M:M2 = M
360 FOR I = 1 TO N - 1
370 IF I > 1 THEN PRINT LEFT$ (V$,(I - 1) * 4);
380 FOR J = I + 1 TO N
390 PRINT CHR$ (64 + I)"-" CHR$ (64 + J)" ";
400 M = M + 1:A%(M,0) = I:A%(M,1) = J
410 NEXT : PRINT : NEXT
420 V = PEEK (37) + 2
430 VTAB 20: HTAB 1: INVERSE : PRINT "MELANGE ALEATOIRE
    EN COURS": NORMAL : CALL - 198
440 FOR I = M TO 1 STEP - 1:A = INT ( RND (1) * I + 1):B
    %(I,0) = A%(A,0):B%(I,1) = A%(A,1):A%(A,0) = A%(I,0)
    :A%(A,1) = A%(I,1): NEXT
450 CALL - 198: VTAB 20: PRINT "UN PEU DE PATIENCE S'IL
    VOUS PLAIT!"
460 X = 0: IF F THEN GOSUB 700
470 F = 1
480 X = X + 1: IF X > M THEN 460
490 IF M1 = B%(X,0) OR M2 = B%(X,1) OR B%(X,0) = 0 OR B%
    (X,1) = 0 THEN 480
500 IF M1 = B%(X,1) OR M2 = B%(X,0) THEN 480
510 Y = Y + 1:F = 0
520 A%(Y,0) = B%(X,0):A%(Y,1) = B%(X,1):B%(X,0) = 0:B%(X
    ,1) = 0
530 M1 = A%(Y,0):M2 = A%(Y,1)
540 IF Y < M THEN 480
550 PRINT : IF V > 15 THEN HOME : GOTO 570
560 VTAB V: CALL - 958
570 FOR I = 1 TO Y: PRINT CHR$ (64 + A%(I,0))"-" CHR$ (6
    4 + A%(I,1))" ";
580 R = R + 1: IF R = NM THEN R = 0: PRINT
590 NEXT : PRINT
600 HTAB 44: VTAB 23: PRINT "<E>NCORE <B>ASIC <M>ENU DIS
    QUETTE ":: POKE - 16368,0: GET X$: IF X$ = "" THEN 4
    70
610 PRINT : IF X$ = "M" OR X$ = "m" THEN PRINT D$"RUN ME
    NU"
```

270 : POKE 34,3 protège les lignes de titre

290 : L'IF peut être supprimé si on le désire.

350 : Les variables sont mises (ou remises) à zéro.

420 : Sur quelle ligne est le curseur ? V est utilisé plus loin (ligne 560).

440 : Mélange aléatoire tout à fait classique.

460-480 : Essayez de comprendre comment se déroule le programme. Quand on a épuisé la liste des rencontres non programmées... sans en avoir trouvé une seule répondant aux conditions des lignes 490 et 500, F est égal à 1... et on saute au sous-programme "DECALAGE".

700 : Sans ce sous-programme, il y a "plantage" pur et simple.

710 : On autorise six essais... mais si la chose se produit, relancez le système !

```
620 IF X$ = "B" OR X$ = "b" THEN TEXT : HOME : END
630 IF X$ = "E" OR X$ = "e" THEN 170
640 GOTO 600
650 :
660 REM **********************
670 REM *   SP. DECALAGE     *
680 REM **********************
690 :
700 IF M1 < > A%(1,0) AND M2 < > A%(1,1) AND M1 < > A%(1
    ,1) AND M2 < > A%(1,0) THEN M1 = A%(1,0):M2 = A%(1,1
    ): FOR I = 1 TO Y - 1:A%(I,0) = A%(I + 1,0):A%(I,1)
    = A%(I + 1,1): NEXT :A%(Y,0) = M1:A%(Y,1) = M2: RETU
    RN
710 POP :P = P + P: IF P < 6 THEN 350
```

EXEMPLES

Chaque équipe bénéficie, entre chaque rencontre, d'un temps de repos qui est égal ou plus grand que la durée d'un match.

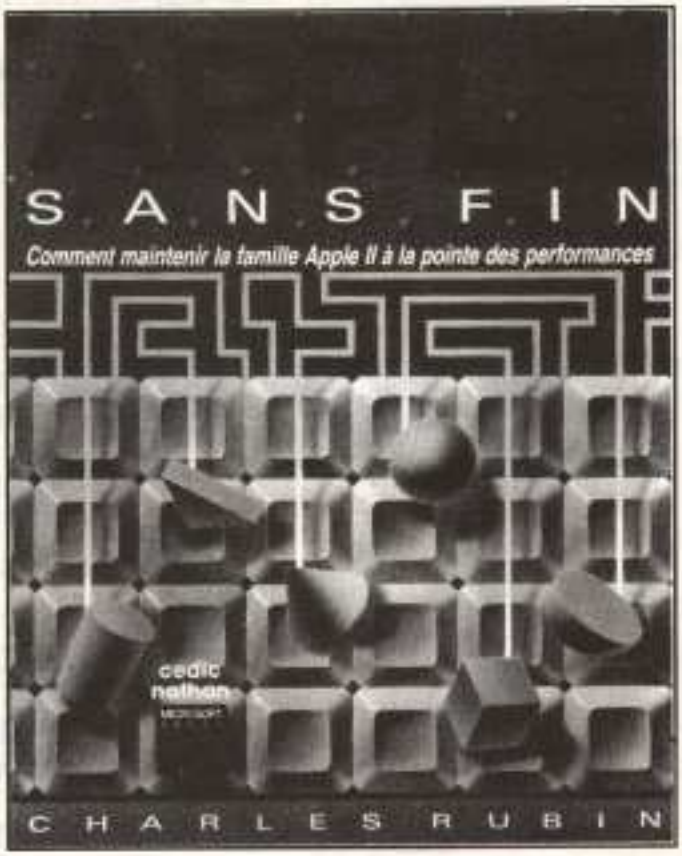
```
NOMBRE D'EQUIPES 7
21 RENCONTRES A ORGANISER:

NOMBRE MAXIMUM DE RENCONTRES PAR JOUR: 11
A-B A-C A-D A-E A-F A-G
    B-C B-D B-E B-F B-G
        C-D C-E C-F C-G
            D-E D-F D-G
                E-F E-G
                    F-G
MELANGE ALEATOIRE EN COURS
UN PEU DE PATIENCE S'IL VOUS PLAIT!

B-G A-C D-E A-F B-C A-E C-F D-G A-B E-F C-D
A-G B-D C-E B-F C-G D-F B-E F-G A-D E-G
```

```
NOMBRE D'EQUIPES 10
45 RENCONTRES A ORGANISER:

NOMBRE MAXIMUM DE RENCONTRES PAR JOUR: 15

A-B A-C A-D A-E A-F A-G A-H A-I A-J
    B-C B-D B-E B-F B-G B-H B-I B-J
        C-D C-E C-F C-G C-H C-I C-J
            D-E D-F D-G D-H D-I D-J
                E-F E-G E-H E-I E-J
                    F-G F-H F-I F-J
                        G-H G-I G-J
                            H-I H-J
                                I-J
MELANGE ALEATOIRE EN COURS
UN PEU DE PATIENCE S'IL VOUS PLAIT!

D-J A-F D-E B-H E-F B-J D-G E-J G-H C-F H-J B-D E-I D-H F-I
A-D F-H B-C A-J F-G E-H I-J C-H B-I C-G A-H G-I A-E H-I F-J
D-I C-J A-G C-E A-B E-G B-F C-D G-J A-C B-E D-F A-I B-G C-I
```

Votre bibliothèque INFORMATIQUE

par Clément RENARD

● L'APPLE SANS FIN (Cedic-Nathan)



Charles Rubin, l'auteur, éprouve visiblement une grande sympathie pour l'Apple *II*. Nous aussi. Soyons clairs : son livre s'adresse indiscutablement aux inconditionnels de l'Apple *II*, et plus particulièrement à ceux désirant améliorer les performances de leur machine.

L'Apple *II* fêtera-t-il ses dix années d'existence ? On peut raisonnablement le penser, tant la date est proche : avril 1987. Combien d'ordinateurs personnels sont ou seront un jour dans ce cas ? Répondre à la question, c'est forcément reconnaître à l'Apple *II* une longévité exceptionnelle... mais aussi comprendre la fidélité des "Apple-maniaques" pour leur micro.

Mais revenons à l'APPLE SANS FIN, utile synthèse des centaines d'études publiées sur les divers matériels fonctionnant sur Apple *II*.

Cet ouvrage est la traduction de *The Endless Apple*, paru en 1984, en langue anglaise, mais l'édition française fournit un répertoire actualisé des importateurs de produits disponibles sur le marché français.

Le désir d'améliorer les prestations d'un ordinateur personnel est généralement lié à la nécessité de mettre au point une application spécifique. De ce point de vue, le découpage du livre de C. Rubin est excellent, puisqu'il tient précisément compte des applications et non des types de produits.

La présence d'un *index* en rend la consultation encore plus agréable. Alors, je ferai mienne la conclusion du communiqué de presse : si vous voulez élargir votre horizon informatique, vous découvrirez dans ce livre jusqu'où l'Apple peut vous mener !

● LA FACE CACHÉE DU MO5 (Cedic-Nathan)

Je sais que de nombreux lecteurs (je n'oublie pas les lectrices... qu'elles se rassurent !) de *T.M.* sont amenés à utiliser le MO5, présent dans la plupart des écoles. Pour certains, il s'agit seulement de montrer à un enfant comment lancer et utiliser un jeu. D'autres tentent (souvent avec bonheur) d'adapter sur MO5 des programmes mis au point sur leur Apple... ou vice versa. D'où la présence, dans les colonnes de *T.M.*, de rubriques consacrées à d'autres machines.

Dans le cas présent, l'auteur, Jean-Baptiste Touchard, paraît au moins aussi curieux que les génies qui, depuis neuf ans, ont percé, un à un, les petits secrets du 65Ø2... sauf qu'il s'agit ici du 68Ø9.

Il affirme que son livre est sans danger... et c'est vrai, mais je vous assure que sa démarche est non seulement intéressante, mais conseillée à celles et à ceux — et, il en existe beaucoup — qui désirent faire de l'étude de l'informatique un passionnant jeu de découverte personnelle.

HISTO.TEXT

VOS GRAPHIQUES EN MODE "TEXT"

Il est facile, avec le nouveau 65C02 — qui permet d'utiliser les caractères *"SOURIS"* — de réaliser des graphiques en mode *TEXT*. Il est non moins facile d'éditer de tels graphiques sur la nouvelle imprimante IMAGEWRITER II. Si vous en doutez, essayez le programme de Clément Renard !

QUESTION 1 : Qu'obtiendra-t-on avec un Apple IIe non transformé ?

R : Ne cherchez pas : un écran assez curieux, sur lequel les caractères correspondant aux *Mouse characters* apparaîtront en mode inverse... et sous leur forme habituelle (les colonnes de grisés seront des "WW", les colonnes noires des "NN", le filet du bas une suite de "L", etc.).

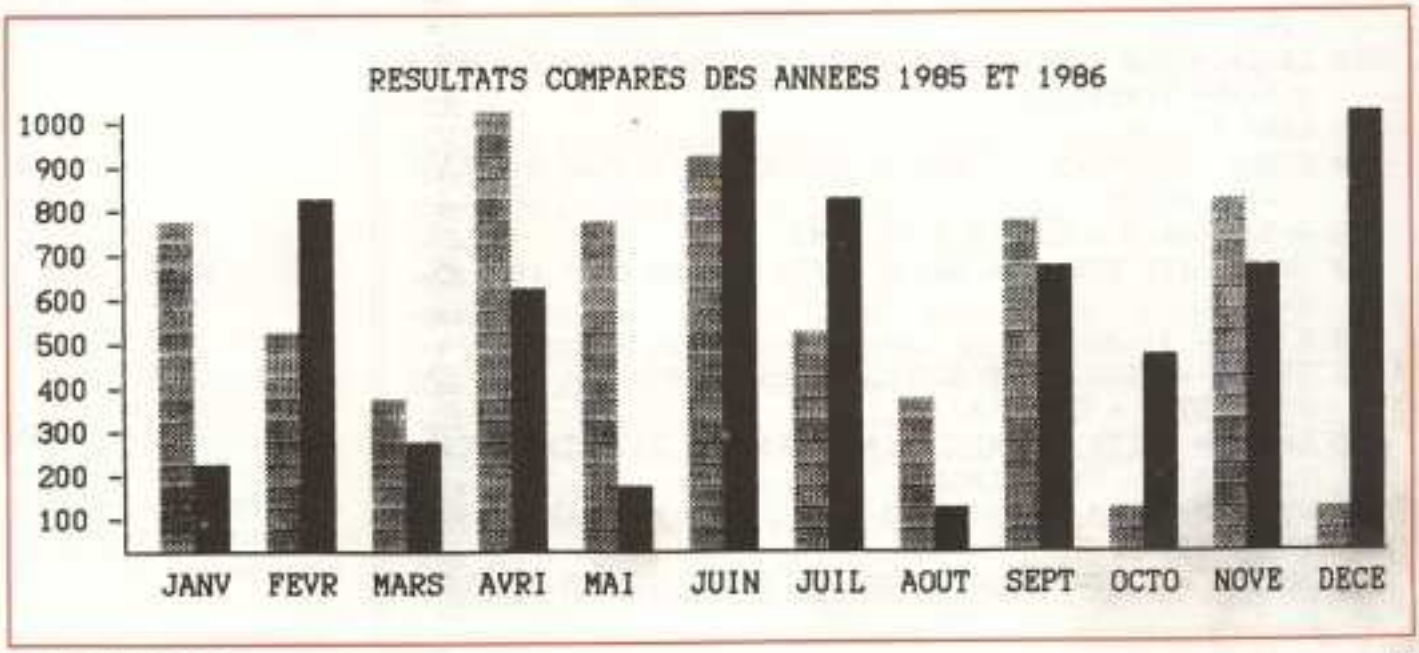
QUESTION 2 : Pourra-t-on néanmoins éditer ce graphique sur une IMAGEWRITER II ?

R : Bien sûr, puisque celle-ci possède les caractères *SOURIS*.

QUESTION 3 : Et avec une DMP... ou une IMAGEWRITER normale ?

R : Là, je vous attendais : c'est possible, mais en créant d'abord, à partir du programme paru dans le n°6 de *T.M.*, des caractères personnalisés. Ce n'est pas difficile... et ça marche. Parution de notre solution dans deux mois (si vous êtes pressé, adressez-moi une enveloppe timbrée à votre adresse et je vous ferai parvenir une photocopie du court programme à écrire.... avec les fameux caractères en DATA(s)).

NESTOR.



HISTO.TEXT (suite)

```
100 TEXT : PRINT CHR$ (4)"PR£3": PRINT : HOME
105 DIM M(12,1),L$(24),V(12,1): GOTO 150
110 :
115 REM ************************************
120 REM **   CARAC MOUSE OR NOT MOUSE     **
125 REM ************************************
130 :
135 PRINT CHR$ (27):: INVERSE : RETURN : REM CARACTERES
    SOURIS
140 PRINT CHR$ (24):: NORMAL : RETURN : REM CARACTERES N
    ORMAUX
145 :
150 GOSUB 455: HOME
155 :
160 REM ************************************
165 REM ** + GRANDE DONNEE/COEFFICIENT **
170 REM ************************************
175 :
180 X = 0: FOR I = 1 TO 12
185 IF V(I,0) > X THEN X = V(I,0)
190 IF V(I,1) > X THEN X = V(I,1)
195 NEXT
200 CO = X / 20:E = X:M = 9
205 IF X > 9999 THEN X = INT (X / 10): GOTO 205
210 E = X
215 FOR I = 1 TO 12:M(I,0) = V(I,0) / CO:M(I,1) = V(I,1)
     / CO: NEXT
220 :
225 REM ************************************
230 REM **         TRACE DU GRAPHE          **
235 REM ************************************
240 :
245 FOR I = 2 TO H STEP 2:VL$ = LEFT$ (V$,5 - LEN ( STR$
     (E)))
250 L$(I) = VL$ + STR$ (E) + " -_ ": VTAB I: POKE 1403,0
    : PRINT VL$E:L$ = ""
255 L$(I + 1) = "          _ "
260 GOSUB 135: HTAB 7: VTAB I: PRINT "L_": HTAB 8: VTAB
    I + 1: PRINT "_"
265 GOSUB 140:E = INT (X / 10 * M)
270 IF E < 100 THEN E = VAL ( LEFT$ ( STR$ (X / 10 * M),
    4))
275 M = M - 1: NEXT
280 TITRE$ = "RESULTATS COMPARES DES ANNEES " + STR$ (A1
    ) + " ET " + STR$ (A2)
285 L$(1) = LEFT$ (V$,(85 - LEN (TI$)) / 2) + TI$: VTAB
    1: HTAB 1:: PRINT L$(1)
290 VTAB 23: HTAB 1: PRINT L$(23);: VTAB 1: PRINT
295 GOSUB 135
300 L$(22) = LEFT$ (V$,7): HTAB 8: FOR I = 1 TO 72: VTAB
```

ATTENTION ! La carte 80 COLONNES est indispensable. Le programme ne fonctionne convenablement que sur Apple *IIe* + 65C02 ou Apple *IIc*.

Il est réellement très facile d'obtenir les caractères SOURIS, comme vous le montrent les lignes 135 et 140.

De la ligne 180 à la ligne 195, on recherche quelle est la plus grande valeur (X).
Si elle dépasse 9999 (4 chiffres), on la divise par 10... et encore par 10 si cela est nécessaire. Dans le cas de petites valeurs, on affichera 2 décimales.

Pendant le tracé de l'histogramme, le programme crée les 23 lignes (L$) qui permettront de l'imprimer... sans passer par l'intermédiaire d'une recopie d'écran 80 colonnes.

CARACTÈRES SOURIS

@= P=
A= Q=
B= R=
C= S=
D= T=
E= U=
F= V=
G= W=
H= X=
I= Y=
J= Z=
K= [=
L= \=
M=]=
N= ^=
O= _=

IMAGEWRITER 2

Notez que l'on utilise le mode unidirectionnel et que l'on modifie l'interlignage (T15). L'accès aux caractères *souris* est obtenu par :

PRINT CHR$(27) ; CHR$(38)

On retrouve les caractères normaux avec un :

PRINT CHR$(27) ; CHR$(36)

UTILISATION

Tel qu'il est, ce programme permet de comparer les résultats de deux années (complètes ou non), mais il est évidemment possible de l'utiliser à d'autres fins. C'est là qu'intervient l'imagination !

```
     22: PRINT "L";:L$(22) = L$(22) + "L": NEXT
305 FOR I = 1 TO 12: FOR K = 2 TO H
310 Z = H + .5 - M(I,0): IF K < Z THEN L$(K) = L$(K) + "
     ": GOTO 320
315 POKE 1403,3 + I * 6: VTAB K: PRINT "WW";:L$(K) = L$(
    K) + "WW"
320 Z = H + .5 - M(I,1): IF K < Z THEN L$(K) = L$(K) + "
     ": GOTO 330
325 POKE 1403,5 + I * 6: VTAB K: PRINT "NN";:L$(K) = L$(
    K) + "NN"
330 IF I < 12 THEN L$(K) = L$(K) + "  "
335 NEXT : NEXT
340 GOSUB 140: VTAB 22: PRINT
345 :
350 REM ************************************
355 REM **          IMPRESSION            **
360 REM ************************************
365 :
370 VTAB 24: HTAB 8: INVERSE : PRINT "PRESSER UNE TOUCHE
     POUR IMPRIMER CETTE PAGE (CTRL-C = A REFAIRE) ":: N
    ORMAL : CALL - 198: GET R$: VTAB 1: PRINT "": IF R$
    = CHR$ (3) THEN 150
375 IF R$ = CHR$ (27) THEN 780
380 HOME : PRINT CHR$ (21): PRINT
385 PRINT CHR$ (4)"PR£1": PRINT : PRINT CHR$ (9)"80N": P
    RINT CHR$ (27)">"; CHR$ (27)"T15"; CHR$ (27)"E":
390 FOR I = 1 TO 23: IF I = 1 OR I = 23 THEN PRINT L$(I)
    : GOTO 415
395 PRINT LEFT$ (L$(I),6);
400 FOR J = 7 TO LEN (L$(I)):L$ = MID$ (L$(I),J,1): IF L
    $ = "U" OR L$ = "_" OR L$ = "W" OR L$ = "L" OR L$ =
    "N" THEN PRINT CHR$ (27): CHR$ (38):L$; CHR$ (27): C
    HR$ (36);: GOTO 410
405 PRINT L$;
410 NEXT
415 PRINT : NEXT
420 PRINT CHR$ (4)"PR£0"
425 PRINT CHR$ (4)"PR£3": PRINT : HOME : GOTO 180
430 :
435 REM ************************************
440 REM **        INITIALISATION          **
445 REM ************************************
450 :
455 V$ = "": FOR I = 1 TO 20:V$ = V$ + "    ": NEXT
460 L$(23) = "          JANV  FEVR  MARS  AVRI  MAI   JUI
    N  JUIL  AOUT  SEPT  OCTO  NOVE  DECE"
465 H = 21: REM HAUTEUR MAXIMUM COLONNE
470 :
475 REM ************************************
480 REM **      SAISIE DES CHIFFRES       **
485 REM ************************************
490 :
495 HOME : PRINT "SAISIE DES DONNEES": GOSUB 135: PRINT
```

Si vous n'éteignez pas votre imprimante, ajoutez un : PRINT CHR$(27) "c" de mise à 0 du logiciel.

HISTO.TEXT (suite)

```
    "LLLLLLLLLLLLLLLLL": GOSUB 140
500 POKE 1403,16: PRINT "Taper un simple RETURN quand le
     chiffre est bon"
505 POKE 1403,16: GOSUB 135: PRINT "à";: GOSUB 140: PRIN
    T " en cas d'erreur -- ":: GOSUB 135: PRINT "A";: GO
    SUB 140: PRINT " pour affichage du graphe"
510 VTAB 22: PRINT : PRINT "DESIREZ-VOUS LIRE UN FICHIER
     (O/N) ? ";: CALL - 198: GET R$: PRINT "": IF R$ = "
    O" OR R$ = "o" THEN GOSUB 740

515 VTAB 6: CALL - 958
520 AB$ = "1985 1986": IF A1 AND A2 THEN AB$ = STR$ (A1)
     + CHR$ (32) + STR$ (A2)
525 VTAB 22: PRINT : PRINT "ANNEES CONCERNEES. SEPAREES
    PAR UN ESPACE: "AB$;: POKE 1403,43: CALL - 198: INPU
    T "";AN$
530 IF PEEK ( - 16286) > 127 THEN 510
535 IF PEEK ( - 16287) > 127 THEN 670
540 IF AN$ = "" THEN AN$ = AB$: CALL - 998: POKE 1403,43
    : PRINT AN$
545 IF MID$ (AN$,5,1) < > " " THEN 525
550 A1 = VAL ( LEFT$ (AN$,4)):A2 = VAL ( RIGHT$ (AN$,4))
    : IF A1 < 1900 OR A2 < 1900 THEN 525
555 VTAB 6: HTAB 6: PRINT A1;: HTAB 19: PRINT A2
560 FOR I = 1 TO 12
565 VI$ = " ": IF I > 9 THEN VI$ = ""
570 GOSUB 575: GOTO 580
575 VTAB 7 + I: PRINT VI$I"." LEFT$ (V$,25);: HTAB 12 -
    LEN ( STR$ (V(I,0))): PRINT V(I,0);: HTAB 25 - LEN (
     STR$ (V(I,1))): PRINT V(I,1);: RETURN
580 POKE 1403,30: PRINT A1;: GOSUB 135: PRINT "U";: GOSU
    B 140: INPUT " ";R$
585 IF PEEK ( - 16286) > 127 AND I > 1 THEN I = I - 1: G
    OTO 565
590 IF PEEK ( - 16287) > 127 THEN 670
595 IF R$ = "" THEN CALL - 998: GOTO 605
600 V(I,0) = VAL (R$): GOSUB 575
605 POKE 1403,45: PRINT A2:: GOSUB 135: PRINT "U":: GOSU
    B 140: INPUT " ";R$
610 IF PEEK ( - 16286) > 127 THEN CALL - 998: GOTO 580
615 IF PEEK ( - 16287) > 127 THEN 670
620 IF R$ = "" THEN 630
625 V(I,1) = VAL (R$): GOSUB 575
630 PRINT : IF I = 12 THEN VTAB 22: CALL - 958: PRINT "P
    RECISER SI CORRECTION OU AFFICHAGE DU GRAPHE  (CTRL-
    E) = ENREGISTREMENT ) ";: CALL - 198: GET R$: VTAB 1
    9: PRINT "": IF R$ = CHR$ (5) THEN 740
635 IF I = 12 THEN 610
640 NEXT
645 :
```

FICHIER

Vous pouvez mémoriser vos données (fichier on ne peut plus classique). Un conseil : donnez un titre très simple à vos fichiers. Exemple H85.86 (qui évoque les deux années concernées).

SAISIE

Pomme ouverte + return permet d'aller directement à l'affichage du graphique.

Pomme fermée + return renvoie à l'item précédent

N'oubliez pas de maintenir la touche pomme enfoncée quand vous appuyez sur RETURN.

La routine saisie est assez originale, même si elle présente quelques défauts. Elle permet notamment de savoir quelles valeurs ont été modifiées... mais seulement dans la mesure où l'on n'utilise pas "Pomme fermée + return"... car les corrections éventuelles modifient évidemment l'écran de base.
Vous comprendrez tout cela quand vous saisirez des données...

ÉVASION

Il n'est pas très facile (sauf par un RESET) de sortir du programme de Clément Renard... mais il existe pourtant une sortie naturelle, après l'affichage de l'histogramme. A ce moment, si vous répondez par ESCAPE... puis RETURN, vous vous retrouvez bel et bien à la ligne 790... mais un GOTO 10000 vous restituera — intactes — toutes vos données !

LIGNE 770

Classique routine de traitement des erreurs, évitant le plantage après un GOSUB.

SAUVÉ

GOTO 10000... et ça repart !

```
650 REM ************************************
655 REM **   PAS DE DONNEES: TERMINE!     **
660 REM ************************************
665 :
670 IF NOT V(1,0) AND NOT V(1,1) THEN HOME : INVERSE : C
    ALL - 198: VTAB 12: PRINT " DESOLE MAIS IL N'Y A PAS
     DE DONNEES POUR TRACER UN QUELCONQUE GRAPHE ": NORM
    AL : POP : GOTO 775
675 RETURN
680 :
685 REM ************************************
690 REM **     LECTURE D'UN FICHIER       **
695 REM ************************************
700 :
705 IF F$ = "" OR PEEK ( - 16286) > 127 THEN RETURN
710 VTAB 20: PRINT : PRINT CHR$ (4)"OPEN"F$: PRINT CHR$
    (4)"READ"F$: INPUT A1: INPUT A2: FOR I = 1 TO 12: IN
    PUT V(I,0): INPUT V(I,1): NEXT : PRINT CHR$ (4)"CLOS
    E"F$: RETURN
715 :
720 REM ************************************
725 REM **      MEMORISATION DONNEES      **
730 REM ************************************
735 :
740 VTAB 22: CALL - 958: INPUT "TITRE DE VOTRE FICHIER "
    ;F$
745 ONERR GOTO 770
750 IF R$ = "O" OR R$ = "o" THEN 705
755 IF F$ = "" OR PEEK ( - 16286) > 127 THEN 515
760 VTAB 20: PRINT : PRINT CHR$ (4)"OPEN"F$: PRINT CHR$
    (4)"WRITE"F$: PRINT A1: PRINT A2: FOR I = 1 TO 12: P
    RINT V(I,0): PRINT V(I,1): NEXT : PRINT CHR$ (4)"CLO
    SE"F$: GOTO 675
765 :
770 POKE 768,104: POKE 769,168: POKE 770,104: POKE 771,1
    66: POKE 772,223: POKE 773,154: POKE 774,72: POKE 77
    5,152: POKE 776,72: POKE 777,96: CALL 768: GOTO 515
775 :
780 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: PO
    KE - 16368,0: TEXT : HOME
785 PRINT "RIEN N'EST PERDU EN TAPANT:": PRINT : PRINT "
    $GOTO 10000": VTAB 2
790 END

            10000 POKE 51,0: GOTO 150
```

Dans le numéro 9 de *TREMPLIN MICRO* (3 juillet), nous publierons une courte routine qui vous permettra, avec une DMP ou une IMAGEWRITER normale, d'éditer un *HISTO.TEXT* (par l'intermédiaire des caractères personnalisés), mais l'offre de NESTOR (page 41) tient... et si vous êtes pressé !...

QUESTIONS

CONTRÔLE-X

Q : Peut-on, dans le cadre d'un INPUT traditionnel, annuler une ligne de saisie sans revenir en arrière ?

R : Oui. Il suffit de taper un CTRL-X et tout ce qui a été saisi est annulé.
CTRL-X (obtenu en pressant simultanément la touche CTRL et la touche X) affiche un \ (ç en AZERTY) et va au début de la ligne suivante... où l'on peut recommencer la saisie.

GUILLEMETS

Q : Est-il réellement impossible, sans passer par une routine en langage machine... ou par un GET, de saisir un espace au début d'une variable, ou encore les caractères de ponctuation ?

R : Même avec un INPUT classique, on peut commencer la saisie par un espace — qui sera pris en compte ! — et se servir des divers signes de ponctuation, mais il convient d'ouvrir d'abord les guillemets. Il ne sera pas nécessaire de les refermer.
Attention ! si le contenu de la variable doit aller dans un fichier, il sera indispensable de le faire précéder par un CHR$(34) lors de son écriture dans le fichier (exemple : PRINT CHR$(34) + A$).

LES 4 LIGNES DE TEXTE EN HGR2

Q : On peut obtenir 4 lignes de texte en bas de la page HGR2, mais je ne réussis pas à les écrire (l'espace est tantôt vide, tantôt rempli de caractères bizarres, en mode FLASH et INVERSE). Pourtant, cela fonctionne très bien avec la page HGR normale. Mon Apple comporte-t-il une anomalie ?

R : Comme vous le savez, il suffit d'exécuter une commande POKE — 16301,0 après un HGR2... pour avoir accès, si l'on peut dire, aux quatre dernières lignes d'un écran TEXT... qui est celui de la page 2 basse résolution. Or les adresses sont différentes de celles de la page 1. En fait, ce sont les mêmes + 1024. Pour écrire dans cette partie de l'écran, il convient donc d'y poquer une à une les valeurs ASCII + 128, comme ceci :

```
100 A$ = "VOICI UN SIMPLE ESSAI, UNE
    DEMONSTRATION" : REM (40 caractères)
110 FOR I = 1 TO 40 : POKE I + 2639,
    ASC(MID$(A$,I,1)) + 128 : NEXT
```

Je vous rappelle les adresses des 4 dernières lignes : 1616, 1744, 1872 et 2000 (+ 1024... lorsqu'il s'agit, comme ici, de la page 2).

VIRGULE ET DATA

Q : J'ai remarqué que, dans certains programmes on place plusieurs virgules, les unes à la suite des autres, dans une ligne DATA. Pourquoi ?

R : Chaque virgule correspond à un élément vide. C'est parfois pratique. Or, pour simuler un élément vide, il suffit, soit de créer une ligne vide comme celle-ci :

```
100 DATA (elle est bien vide, non ?)
```

soit d'ajouter une virgule quelque part :

```
100 DATA, TOTO,    = élément vide, puis
    TOTO, puis élément vide.
```

Il n'est pas interdit de taper X virgules pour figurer X éléments vides.

CALL — 151

Q : Si j'ai bien compris, pour recopier un programme en langage machine, il faut d'abord taper CALL — 151 puis la première adresse, suivi des codes, mais je n'obtiens que l'astérisque, et ceci malgré de multiples essais... si bien que j'ai abandonné !

R : Vous oubliez l'indispensable RETURN après votre CALL — 151. C'est RETURN qui valide toutes vos entrées, ne l'oubliez pas. Il est sous-entendu dans toutes les explications que nous donnons. D'autre part, d'après votre lettre, vous commettez une seconde erreur en faisant suivre la première adresse du programme par un tiret... que votre moniteur ne reconnaît pas. Il faut utiliser les deux points (:).
Pour le reste, vous avez compris, puisque vous respectez l'obligatoire espace entre chaque valeur. Sachez toutefois qu'il n'est pas nécessaire de taper 01 02 03 00.
Puisque le premier zéro ne compte pas, 1 2 3 0 font très bien l'affaire !

REPONSES

Yvan KOENIG

DIMDEL

Ou comment changer la dimension d'un tableau en cours de programme ?

Essayez en mode direct DIM A(12), par exemple, et A(13) = 99... Un BAD SUBSCRIPT ERROR vous signale que la tentative de donner une valeur à un élément de tableau d'indice supérieur à 12 est impossible. C'est ici qu'intervient DIMDEL.

UTILISATION

DIMDEL s'emploie au travers de l'ampersand (&)

& DIM A +	Pour ajouter un élément au tableau A().
& DIM A —	Pour supprimer un élément au tableau A().
& DIM A	Supprime le tableau A().
& DEL A	Permet de redimensionner, mais aussi donne la possibilité de mettre à zéro les éléments du tableau A(). Il suffit en effet de recréer le tableau effacé.

Revenons à l'exemple du tableau **A** dimensionné à **12** éléments. Pour le dimensionner à **13** éléments, afin que notre **A(13) = 99** soit accepté, il faut simplement **& DIM A +** avant l'affectation de la valeur **99** au tableau **A**. Le programme BASIC suivant en fait la démonstration (si vous avez tapé et enregistré, sous le nom de DIMDEL, la routine en langage machine désassemblée en page 49). Celle-ci est appelée à la ligne 100 du programme basic. A cette même ligne, la fonction & LOMEM : effectue un LOMEM spécial qui protège le langage machine. La routine est chargée juste après le basic. Gare aux ajouts de lignes qui l'écraseraient !

DIMDEL.DEMO

```
100 TEXT : PRINT CHR$ (21): HOME : PRINT CHR$ (4)"BRUN DIMDEL.A" PEEK (175) +
     256 * PEEK (176): & LOMEM: : HOME
110 HTAB 6: VTAB 2: PRINT "Prenons un tableau en exemple": HTAB 4: VTAB 10: P
    RINT "DIM A (12)":NMAX = 12: DIM A(NMAX)
120 FOR I = 0 TO NMAX:A(I) = I: NEXT : VTAB 4: FOR I = 0 TO NMAX: HTAB 18: PR
    INT A(I): NEXT : GOSUB 240: HTAB 1: VTAB 2: PRINT "Il faut employer...":
     PRINT SPC( 15)
130 GOSUB 250: HTAB 4: VTAB 10: INVERSE : PRINT "&  DIM A +": NORMAL : GOSUB
    250: HTAB 1: VTAB 2: PRINT SPC( 16): HTAB 20: VTAB 2: PRINT "et ajouter u
    ne valeur": GOSUB 250
140 & DIM A + :NMAX = NMAX + 1:A(NMAX) = 99: VTAB 4: FOR I = 0 TO NMAX: HTAB
    21: PRINT A(I): NEXT : GOSUB 250: HTAB 8: VTAB 21: PRINT "99 appartient a
    u tableau A()": GOSUB 250: GOSUB 240
150 HTAB 1: VTAB 2: PRINT " Réduire la dimension est aussi simple  ": GOSUB 2
    50: INVERSE : HTAB 13: VTAB 10: PRINT "-": HTAB 4: VTAB 12: PRINT "&  DIM
     A -": NORMAL : GOSUB 250
160 & DIM A - :NMAX = NMAX - 1: & DIM A - :NMAX = NMAX - 1: VTAB 4: FOR I = 0
     TO NMAX: HTAB 24: PRINT A(I): NEXT : GOSUB 250: HTAB 2: VTAB 21: PRINT "
```

→

DIMDEL.DEMO (suite)

Yvan KOENIG

```
    Le tableau A () est dimensionné à 11.": GOSUB 240: HTAB 4: VTAB 12: PRINT
     SPC( 10)
170 HTAB 4: VTAB 10: INVERSE : PRINT "&  DEL A";: NORMAL : PRINT "  ": GOSUB
    250: & DEL A: DIM A(20): HTAB 1: VTAB 2: PRINT "Initialiser / redimension
    ner un tableau"
180 VTAB 4: INVERSE : FOR I = 0 TO 12: HTAB 32: PRINT A(I): NEXT : NORMAL : G
    OSUB 250
190 PRINT CHR$ (7);: HTAB 2: VTAB 21: PRINT " <E>NCORE  <B>ASIC  <M>ENU DISQU
    ETTE  ";: GET X$
200 IF X$ = "E" OR X$ = "e" THEN CLEAR : GOTO 100
210 IF X$ = "B" OR X$ = "b" THEN TEXT : HOME : END
220 IF X$ = "M" OR X$ = "m" THEN PRINT : PRINT CHR$ (4)"RUN MENU"
230 GOTO 190
240 PRINT CHR$ (7);: HTAB 30: VTAB 24: PRINT "<TOUCHE> ";: POKE - 16368,0: GE
    T X$: HTAB 30: VTAB 24: PRINT SPC( 8);: RETURN
250 FOR J = 1 TO 900: NEXT : RETURN
```

RAPPELS SUR LES TABLEAUX EN APPLESOFT

(réservés aux durs de la programmation)

Il existe trois types de tableaux : entiers, flottants et chaînes. Ces tableaux sont traités différemment selon leur nature.

Un tableau comporte plusieurs parties, à savoir :

- Un descripteur du tableau et une zone de stockage d'éléments pour les tableaux de nombres.
- Un descripteur du tableau, une zone de stockage des descripteurs de chaînes et enfin le stockage des chaînes proprement dites (celles-ci peuvent être en partie haute de la mémoire ou dans le BASIC lui-même).

Les descripteurs de tableaux ont la structure suivante :

OCTET	CONTENU	INT	FLO	STR
0	1er caractère du nom	nég	nég	pos
1	2e caractère du nom	nég	pos	pos
2	partie basse nombre octets du tableau			
3	partie haute nombre octets du tableau			
4	nombre de dimensions			
5	partie haute dernière dimension			
6	partie basse dernière dimension			
.				
.				
x	partie haute première dimension			
x+1	partie basse première dimension			

Lorsque l'on dit "dimension", il faut comprendre "dimension + 1" puisque l'on doit ajouter le rang zéro (DIM A(10) donne 11 éléments).

En ce qui concerne les éléments, les entiers sont stockés sur deux octets, les flottants sont sur cinq octets et les descripteurs de chaînes comportent trois octets (un pour la longueur, deux pour l'adresse).

SI VOUS ÊTES NÉOPHYTE, LISEZ CECI :

Contentez-vous de taper les codes de la routine DIMDEL, en procédant de la manière habituelle : **CALL — 151** puis **2000 : 20 58 FF BA**, etc. sans oublier un espace entre chaque code. Validez chaque élément de saisie par un RETURN (ne faites pas des lignes trop longues), puis retapez une nouvelle ligne en commençant par l'adresse suivie de deux points.

Lorsque vous aurez terminé la saisie de tous les codes, un CTRL-C (+ RETURN) vous permettra de retrouver l'Applesoft. Enregistrez alors votre routine sur disquette par :

BSAVE DIMDEL, A$2000, L$145

DIMDEL

```
2000-  20 58 FF   JSR  $FF58     ┐ Un RTS garanti par Apple.
2003-  BA         TSX            ┘
2004-  BD FF 00   LDA  $00FF,X   ┐ Initialisation du vecteur & (Ampersand).
2007-  18         CLC            │
2008-  69 19      ADC  £$19      │
200A-  8D F6 03   STA  $03F6     │
200D-  BD 00 01   LDA  $0100,X   │
2010-  69 00      ADC  £$00      │
2012-  8D F7 03   STA  $03F7     │
2015-  A9 4C      LDA  £$4C      │ "JMP"
2017-  8D F5 03   STA  $03F5     │
201A-  60         RTS            ┘
201B-  48         PHA            ┐ Ici, on sauve le TOKEN et on le récupère
201C-  20 B1 00   JSR  $00B1     │ après un avancement.
201F-  68         PLA            ┘
2020-  C9 85      CMP  £$85      ┐ Quelle est la fonction appelée DEL, DIM ou
2022-  F0 19      BEQ  $203D     │ LOMEM ?
2024-  C9 86      CMP  £$86      │
2026-  F0 15      BEQ  $203D     │
2028-  C9 A4      CMP  £$A4      ┘
202A-  D0 41      BNE  $206D     ┐ C'est une erreur.
                                 ┘
```

FONCTION & LOMEM

```
202C-  AD F6 03   LDA  $03F6     ┐
202F-  69 29      ADC  £$29      │
2031-  85 50      STA  $50       │ Utile pour protéger tout le programme LM.
2033-  AD F7 03   LDA  $03F7     │
2036-  69 01      ADC  £$01      │
2038-  85 51      STA  $51       ┘
203A-  4C AC F2   JMP  $F2AC     ┐ L'Applesoft va terminer.
                                 ┘
```

FONCTION & DEL & DIM

```
203D-  85 40      STA  $40       ┐
203F-  A9 40      LDA  £$40      │
2041-  85 14      STA  $14       │ Il faut localiser le tableau, à l'aide de
2043-  20 E3 DF   JSR  $DFE3     │ PTRGET de l'Applesoft qui pointe sur une
2046-  A9 00      LDA  £$00      │ variable.
2048-  85 14      STA  $14       ┘
204A-  A0 02      LDY  £$02
204C-  18         CLC
204D-  A5 9B      LDA  $9B       ┐ Adresse du tableau LO.
204F-  71 9B      ADC  ($9B),Y   │ Ajout nombre octets LO.
2051-  85 3C      STA  $3C       ┘ Adresse LO tableau suivant.
2053-  C8         INY            ┐
2054-  A5 9C      LDA  $9C       │ Même chose partie haute
2056-  71 9B      ADC  ($9B),Y   │
2058-  85 3D      STA  $3D       │
205A-  A5 40      LDA  $40       ┘ et on récupère le TOKEN.
205C-  C9 85      CMP  £$85        DEL ?
205E-  F0 1D      BEQ  $207D
2060-  20 B7 00   JSR  $00B7     ┐ Quel mode ?
2063-  F0 18      BEQ  $207D     │ suppression du tableau.
2065-  C9 C8      CMP  £$C8      ┘ — pour supprimer un élément.
```

DIMDEL (suite)

Yvan KOENIG

```
2067-  F0 07       BEQ   $2070     ┐
2069-  C9 C9       CMP   £$C9      │ + pour ajouter un élément.
206B-  F0 03       BEQ   $2070     ┘
206D-  4C C9 DE    JMP   $DEC9     ┐ C'est une erreur.
2070-  6A          ROR             ┐ C'est bon. La commande est empilée, il faut
2071-  08          PHP             │ compter le nombre de dimensions.
2072-  C8          INY             │
2073-  B1 9B       LDA   ($9B),Y   │
2075-  C9 01       CMP   £$01      │ Plus d'une dimension ?
2077-  F0 1F       BEQ   $2098     │
2079-  28          PLP             │ Oui, sortie du programme. BAD SUBSCRIPT
207A-  4C 96 E1    JMP   $E196     ┘ ERROR
207D-  A5 9B       LDA   $9B       ┐ Initialisation des pointeurs de MOVE et fin
207F-  85 42       STA   $42       │ de zone des tableaux pour la suppression
2081-  A5 9C       LDA   $9C       │ du tableau.
2083-  85 43       STA   $43       │
2085-  38          SEC             │
2086-  A5 6D       LDA   $6D       │
2088-  85 3E       STA   $3E       │
208A-  88          DEY             │
208B-  F1 9B       SBC   ($9B),Y   │
208D-  85 6D       STA   $6D       │
208F-  A5 6E       LDA   $6E       │
2091-  85 3F       STA   $3F       │
2093-  C8          INY             │
2094-  F1 9B       SBC   ($9B),Y   │ Branche toujours
2096-  D0 5B       BNE   $20F3     ┘ partie haute adresse zone programme.
2098-  C8          INY             ┐
2099-  A5 81       LDA   $81       │ Premier caractère du nom.
209B-  10 01       BPL   $209E     │ Si pas entier.
209D-  88          DEY             │
209E-  A5 82       LDA   $82       │ Deuxième caractère du nom.
20A0-  10 02       BPL   $20A4     ┘ Si pas entier.
20A2-  88          DEY             ┐
20A3-  88          DEY             │
20A4-  84 40       STY   $40       │ RÉEL = 5  CHAÎNE = 3  ENTIER = 2
20A6-  A0 05       LDY   £$05      │ pour utilisation ultérieure, récupère la
20A8-  28          PLP             │ fonction.
20A9-  90 52       BCC   $20FD     │ C'est "+".
20AB-  B1 9B       LDA   ($9B),Y   │ Partie haute nombre d'éléments > 0, on
20AD-  D0 07       BNE   $20B6     ┘ peut continuer (Y = 5).
20AF-  C8          INY             ┐
20B0-  B1 9B       LDA   ($9B),Y   │ Partie basse nombre d'éléments.
20B2-  C9 02       CMP   £$02      │
20B4-  90 C4       BCC   $207A     ┘ 1 élément, ne rien supprimer.
20B6-  A0 06       LDY   £$06      ┐
20B8-  B1 9B       LDA   ($9B),Y   │ Partie basse nombre d'éléments.
20BA-  E9 01       SBC   £$01      │ Retranche 1 au nombre d'éléments.
20BC-  91 9B       STA   ($9B),Y   │
20BE-  88          DEY             ┘ Y = 5
```

```
20BF-  B1 9B      LDA  ($9B),Y
20C1-  E9 00      SBC  £$00
20C3-  91 9B      STA  ($9B),Y
20C5-  A0 02      LDY  £$02      ┐ Retranche longueur élément au nombre
20C7-  B1 9B      LDA  ($9B),Y   │ d'octets du tableau.
20C9-  38         SEC            │
20CA-  E5 40      SBC  $40       │
20CC-  91 9B      STA  ($9B),Y   │
20CE-  C8         INY            ┘ Y=3
20CF-  B1 9B      LDA  ($9B),Y
20D1-  E9 00      SBC  £$00
20D3-  91 9B      STA  ($9B),Y
20D5-  88         DEY            ┐ Y=2
20D6-  18         CLC            │
20D7-  A5 9B      LDA  $9B       ┐ Calcule l'adresse de début de zone à
20D9-  71 9B      ADC  ($9B),Y   │ descendre.
20DB-  85 42      STA  $42       │
20DD-  A5 9C      LDA  $9C       ┘
20DF-  C8         INY            ┐ Y=3
20E0-  71 9B      ADC  ($9B),Y   │
20E2-  85 43      STA  $43       │
20E4-  A5 6D      LDA  $6D       ┐ Pointeur de fin de zone des tableaux.
20E6-  85 3E      STA  $3E       ┘
20E8-  38         SEC            ┐
20E9-  E5 40      SBC  $40       │
20EB-  85 6D      STA  $6D       │ Initialise l'adresse de fin de zone à descen-
20ED-  A5 6E      LDA  $6E       │ dre et place le nouveau pointeur sur la fin
20EF-  85 3F      STA  $3F       │ de la zone des tableaux.
20F1-  E9 00      SBC  £$00      │
20F3-  85 6E      STA  $6E       ┘
20F5-  A0 00      LDY  £$00
20F7-  20 2C FE   JSR  $FE2C     MOVE du Moniteur.
20FA-  4C 95 D9   JMP  $D995     Fin d'instruction.
20FD-  C8         INY            ┐ Y=6
20FE-  B1 9B      LDA  ($9B),Y   │
2100-  69 01      ADC  £$01      │ Ajoute 1 au nombre d'éléments.
2102-  91 9B      STA  ($9B),Y   │
2104-  88         DEY            ┘ Y=5
2105-  B1 9B      LDA  ($9B),Y
2107-  69 00      ADC  £$00
2109-  91 9B      STA  ($9B),Y
210B-  A0 02      LDY  £$02
210D-  B1 9B      LDA  ($9B),Y   ┐ Ajoute au nombre d'octets du tableau la
210F-  65 40      ADC  $40       │ longueur d'un élément.
2111-  91 9B      STA  ($9B),Y   ┘
2113-  C8         INY            ┐ Y=3
2114-  B1 9B      LDA  ($9B),Y   │
2116-  69 00      ADC  £$00      │
2118-  91 9B      STA  ($9B),Y   │
211A-  A5 3C      LDA  $3C       │
211C-  A4 3D      LDY  $3D       │
211E-  85 9B      STA  $9B       ┘ Début de zone à remonter.
```

DIMDEL (suite)

Yvan KOENIG

```
2128-  84 9C     STY  $9C
2122-  A4 6E     LDY  $6E      ] Sommet de zone à remonter.
2124-  84 97     STY  $97
2126-  A5 6D     LDA  $6D
2128-  85 96     STA  $96
212A-  65 40     ADC  $40      ] + Longueur élément.
212C-  85 6D     STA  $6D      ] Nouvelle fin de zone des tableaux.
212E-  85 94     STA  $94      ] Sommet de la zone destination.
2130-  90 01     BCC  $2133
2132-  C8        INY
2133-  84 6E     STY  $6E
2135-  84 95     STY  $95
2137-  20 9A D3  JSR  $D39A    ] Exécute le déplacement avec l'Applesoft.
213A-  A9 00     LDA  #$00
213C-  A4 40     LDY  $40
213E-  88        DEY
213F-  91 9B     STA  ($9B),Y  ] Mise à zéro de l'élément.
2141-  D0 FB     BNE  $213E
2143-  F0 B5     BEQ  $20FA    FIN ROUTINE LM
```

(NOTA : DIMDEL s'applique aussi aux chaînes et aux entiers)

Offrez-vous 31 routines

mises au point par un spécialiste : Yvan KOENIG

Vous avez souvent besoin de routines en langage machine pour compléter l'APPLESOFT, mais votre page 3 déborde ? Ne vous énervez pas, AMPERELO est là avec ses 31 routines RELOGEABLES (plus de problème d'implantation !) :

PRINT USING aux normes françaises, **tri rapide**, **Input** contrôlé, **recherche** dans un tableau, **position** d'une sous-chaîne dans une chaîne, **remplacement** d'une sous-chaîne par une autre dans une chaîne, **SWAP** de variables ou de tableaux, **garnissage** de tableaux, **DIMDEL**, **IF THEN ELSE**, **GET** numérique contrôlé, **PEEK** et **POKE** sur deux octets, ils sont tous là. Et celà sans vous obliger à étudier le langage machine.

MOSAIQUE vous permettra en outre de couper-coller les modules dont vous aurez besoin.

En prime, LECTURE.SOURCES vous servira à lire sans assembleur, toutes les sources MERLIN/BIG/MAC. Il vous permettra également de transférer en mode TEXT les sources binaires de MERLIN pour en autoriser le traitement sous ProCODE.

Présentées en format DOS 3.3. les routines peuvent être transférées sous ProDOS.

Votre bibliothèque INFORMATIQUE

• PROGRAMMATION DU 6502 (Modulo/Belin)

Ce nouveau volume de la *Collection Modulo* est une traduction du *6502 Machine Code for Humans* (Alan Tootill et David Barrow), traduction de Robert Morin. Signalons au passage que MODULO EDITEUR est Canadien et Québécois, et qu'il est représenté en France par les Editions Belin.

J'ignore pourquoi l'on nous a privés aussi longtemps de cette excellente "Programmation du 6502". C'est à mon avis l'un des meilleurs bouquins sur la question. Que l'on ne s'y trompe pas : pour en tirer pleinement profit, il est indispensable d'y consacrer plusieurs heures, et de préférence devant un Apple.

Les auteurs ont préféré soumettre à leurs lecteurs des dizaines d'exemples plutôt que de décrire une à une les diverses instructions du 6502 (ce qui a d'ailleurs été fait, et avec succès, dans d'autres livres). Vous disposerez donc, avec *L'Assembleur du 6502*, d'un recueil de routines d'une grande richesse. En effet, ce sont pour la plupart des exemples pratiques (permuter deux mots de la page zéro, temporisation d'une seconde, translation d'un point, conversion du code ASCII en code DCB condensé, et une quarantaine d'autres sujets !). Si vous êtes néophyte, commencez par une bonne initiation (avec François Monteil — Eyrolles — ou Nicole Bréaud-Pouliaquen — PSI), mais ne tardez pas trop à vous offrir cette Programmation du 6502 : elle vous rendra assurément de grands services.

• L'ASSEMBLEUR FACILE DU 65C02 (Eyrolles)

Une version enrichie de l'*Assembleur facile du 65C02* (même auteur), tenant compte des nouvelles possibilités offertes par le descendant de l'un des plus populaires microprocesseurs : le 6502... cher à nos cœurs !

Il faut savoir que le 65C02 équipe non seulement nos Apple, mais aussi (entre autres machines) le Commodore LCD portable. Avec François Monteil, on entre directement dans le vif du sujet, sans passer par la longue et parfois fastidieuse description technique de la bête (je ne prétends pas qu'une telle description soit inutile).

Les systèmes numériques sont rapidement passés en revue, puis on arrive à la syntaxe assembleur du 65C02, très voisine, on le sait, de celle de son aîné.

La partie la plus importante de l'ouvrage est évidemment consacrée aux différents modes d'adressage et au jeu d'instructions du microprocesseur. Et c'est là que le bât blesse : bien des lecteurs regretteront, comme moi, de ne point disposer, sur leur machine, d'instructions telles que BBR et BBS (branchement relatif sur un bit). N'oublions pas que François Monteil s'adresse ici à TOUS les utilisateurs de 65C02 et non aux seuls possesseurs d'un Apple *IIe* ou d'un Apple *IIc*...

• PASCAL ISO/AFNOR (Dunod Informatique)

Support de cours, ce livre contient une initiation à la programmation par la méthode déductive. Une soixantaine d'exemples de programmes, tous testés sur ordinateur, et un aide-mémoire très complet facilitent la compréhension de tous les éléments du langage. La syntaxe de ces éléments est décrite par des diagrammes clairs et précis.

Manuel de référence, l'ouvrage d'Alain Tisserant décrit tous les aspects de la norme internationale ISO/AFNOR, mais en présente aussi les prochains développements. Les programmeurs expérimentés pourront y découvrir des particularités du langage qu'ils ne soupçonnaient pas. Quant aux étudiants et aux amateurs, ils y apprendront progressivement — mais en travaillant ! — à construire des programmes exécutables sur n'importe quel ordinateur.

• EXERCICES COMMENTÉS D'ANALYSE ET DE PROGRAMMATION (Dunod Informatique)

Peut-être connaissez-vous déjà, de Jean-Pierre Laurent, l'*Initiation à l'analyse et à la programmation*, publiée chez Dunod en 1982 (une deuxième édition vient de paraître) ? Cette fois, il récidive, mais en collaboration avec Jacqueline Ayel. Ce nouveau livre fait suite au premier et a pour principal objet la construction progressive et commentée d'algorithmes.

Chaque exercice — il y en a 66 — est considéré comme le support d'un enseignement dans l'art de conduire l'analyse des problèmes. On appréciera particulièrement les algorithmes suivants (liste non limitative) : dépouillement d'enquêtes, jeu du pendu, tri par la méthode du minimum, tri par insertion, jeu de mastermind, chemins dans un graphe, tri fusion...)

Mais attention ! ce n'est pas tout simple et les auteurs nous proposent réellement des exercices capables de nous aider à approfondir nos connaissances en analyse : tout un programme... en ce qui me concerne du moins ! **C. Renard.**

CHASSEURS de bogues

Miss Mouse et la Scribe

• *Je pense que le programme "What time is it Miss Mouse" fonctionne mal sur un Apple 6502. Cela semble provenir de la sauvegarde du contexte en début d'interruption. Ainsi, en ligne 121, la valeur de l'accumulateur sauvée sur la pile n'est pas la bonne car elle a été modifiée par la routine $FA40 appelée lors de l'interruption. Par contre, on retrouve la bonne valeur en $45. Il suffit donc de faire un LDA $45 avant PHA pour que tout rentre dans l'ordre.*
Sans cette modification, on aboutit après quelques manipulations en mode direct ou en Basic à un fatidique "SYNTAX ERROR".
Par contre, avec le 65C02 tout fonctionne normalement, car la gestion des interruptions est différente quant à la sauvegarde de l'accumulateur.
Un petit conseil à Alexandra de LYON. En rembobinant le ruban celui-ci est plus fragile, alors comme la protection de rupture n'est plus opérante, attention à la tête d'écriture en cas de déchirement du ruban ! Par contre, je n'ai pas trouvé d'utilisation véritable du fameux ressort, et sa suppression permet d'utiliser entièrement le ruban. Au contraire, lorsque le ressort est présent, j'ai constaté des défauts d'impression en fin de bande. Le logiciel Dazzle Draw me semble intéressant pour utiliser la couleur de la Scribe.

Jean-Paul A. (77640 JOUARRE).

Calculs imprécis

• *Voici deux courts programmes qui normalement devraient donner les mêmes résultats (sous DOS 3.3.).*

1°
```
10 INPUT H
20 H = INT(H * 100) / 100
30 PRINT H
```

2°
```
10 INPUT H : H = H * 100
20 H = INT(H) / 100
30 PRINT H
```

Or, si l'on entre en 10, la valeur 12,2, on obtient 12,19 à la fin du premier programme et 12,20 à la fin du second. Pourriez-vous m'indiquer d'où cela provient ?

J. P. (30000 NÎMES).

TM Vous avez trouvé vous-même la réponse : la deuxième programmation est la bonne. Le calcul intermédiaire (H = H * 100) place dans H une valeur exacte, ce qui n'est pas le cas dans la version 1 du mini-programme.

Catalogue thématique

• *D'abord merci de vos réponses rapides et précises à mes précédentes questions. Je vous écris aujourd'hui pour participer au jeu de la chasse aux bogues ; en effet, j'en ai trouvé deux, il me semble, dans le programme "CATALOGUE THEMATIQUE" du T.M. n°4.*

1° *Le remplissage du tableau C$(I) se fait à partir de l'élément 0 (ligne 40), mais l'affichage se fait à partir de l'élément 1 (ligne 120), résultat un fichier n'est pas affiché.*

2° *La boucle de classement (lignes 60-90) ne fonctionne pas ; en effet, si I=N, ce qui arrive avec mes disquettes, lors de la première exécution de la ligne 80, K=I—1 et l'instruction C$(K+1)=C$(K) écrase C$(I) par C$(K) et la variable à classer est perdue.*

Voici un extrait de la version corrigée de ce programme :

```
40 & C$(1): N = 0: I = 0
50 I = I + 1: IF C$(I) = " " THEN 100
60 IF MID$(C$(I),2,1) = CHR$ (8) THEN 50
70 N = N + 1: C$ = C$(I)
80 FOR K = N TO 1 STEP — 1: IF C$ > C$(K —
   1) THEN 90
85 C$(K) = C$(K — 1): NEXT K
90 C$(K) = C$: GOTO 50
```

QUESTION AUX CONNAISSEURS :

Grâce à T.M. je suis devenu un inconditionnel de ProDOS, surtout avec le volume fictif /RAM, mais comment libérer les pages mémoires bloquées par le système, l'équivalent du POKE 49984,0 (mais pour toutes les pages ou, au moins, pour les pages de la mémoire graphique) ? Je rêve de jeux du style "Mask of the sun" sans les interminables attentes de lecture disque !...

Jean-Claude H. (92160 ANTONY).

GRAPH.AIDE... erreur

• *Dans GRAPH.AIDE (numéro 7, page 28), il y a une erreur... sans grande importance, mais qu'il est préférable de corriger pour ne pas perdre la main.*
En effet, à la ligne 195, le GOTO 310 nous envoie... dans la nature. Je suggère donc de le remplacer par un GOTO 210.

Claude F. (37000 TOURS).

OUT OF MEMORY ERROR

• *Je vous écris pour vous poser un problème :*
Lorsque l'on fait défiler un texte de haut en bas et de bas en haut dans un programme basic (pour faire un petit traitement de texte par exemple), il se produit au bout d'un certain nombre de fois (entre 10 et 20) un "OUT OF MEMORY ERROR" dans des lignes qui n'ont aucune erreur. D'ailleurs, en compilant ce même programme tout marche très bien. Auriez-vous un remède à cette erreur du Basic Applesoft.

N. G. (51200 EPERNAY).

TM

Ce n'est pas une "erreur" du Basic Applesoft, mais une erreur de programmation. Vous devez abuser des GOSUB. Or, quand vous utilisez cette instruction, l'Applesoft stocke son numéro de ligne sur une pile (chaque GOSUB utilise 6 octets de mémoire !). Vous ne pouvez pas imbriquer plus de 25 sous-programmes les uns dans les autres. Si d'aventure le programme rencontre GOSUB 24 fois... avant de trouver le premier RETURN, il vous affiche un fatidique "OUT OF MEMORY ERROR".
Il convient d'y penser quand on programme en Basic, notamment s'il s'agit de faire défiler des lignes sur un écran (24 lignes...).

CALENDRIER PERPÉTUEL

• *A l'attention de Michel Edelin, auteur du CALENDRIER PERPÉTUEL paru dans T.M. n°5.*
Le calendrier semble limité à l'époque 1901 — 2000. 1901 car 1900 n'est pas bissextile ; 2000 par la volonté de l'auteur ? On peut élargir son application au calendrier Grégorien, à partir du 15 octobre 1582 jusqu'à... Il suffit d'éliminer les années centenaires non bissextiles, c'est-à-dire non divisibles par 400.

On pourrait alors écrire la ligne 85 comme ceci :

```
Y = M — (7 * INT(M/7)) + 1 :
B = (INT(U/4) = U/4) — (INT(U/400) <> U/400) :
M(²) = M(²) + B * (B > 0)
```

On vérifie alors que le calendrier de 1900 est correct. Le 15 octobre 82 était un vendredi. 1er janvier 3000 sera un mercredi si le calendrier Grégorien a toujours cours. A vous de juger si l'extension en vaut la peine (avec un coup d'œil sur la ligne 385).

Jean G. (42000 SAINT ÉTIENNE).

OPINION

APPLE IIc pas mort !

• *Je tiens tout d'abord à vous féliciter pour votre excellente revue où l'on trouve des programmes variés. J'ai été surpris en lisant en page 1 du numéro 6 que l'Apple IIc était une machine fermée et qu'elle avait selon vous peu d'avenir ("peut-on lui promettre une longue vie ? J'en doute"). Je tiens à vous signaler qu'il existe de nombreuses extensions, disponibles aux U.S.A. Je vous cite en vrac : extension 640 K, Z80 (disponible en France), horloge, carte Mockinboard, processeur 16 bits. Vous reconnaîtrez que l'on est loin de "la machine fermée" bien que ce ne soit pas encore la souplesse du IIe. Je pense que de telles affirmations font regretter aux possesseurs de IIc d'avoir acheté un IIc plutôt qu'un IIe.*
J'ai également quelques trucs à vous communiquer :
— Si vous utilisez un Apple IIc avec un moniteur couleur, vous serez certainement gêné par les lettres en mode graphique qui sont violettes, vertes, mais rarement blanches. Un poke — 16290,0 passe l'écran en mode monochrome. Par exemple, pour lire plus facilement les menus déroulants de MousePaint, vous faites poke — 16290,0 suivi de PR£6 pour booter la disquette MousePaint.
— Si vous disposez de Basic System 1.1, vous avez à votre disposition une commande ProDOS supplémentaire : BYE. En tapant BYE, le système vous demande d'entrer un préfixe, puis le nom du fichier SYS à exécuter.
Je tiens également à vous faire part d'une bogue que j'ai constatée dans le programme ECRANS.BAS du numéro 3 : arrivé à la dernière ligne, si l'on tape un caractère, l'écran remonte d'une ligne, détruisant la page que l'on avait faite. Peut-être avez-vous trouvé ce qui cause ce désagrément ?

Frédéric B. (19150 LAGUENNE).

TM

L'Apple *IIc* est une excellente machine, et je le reconnais volontiers, mais elle est loin d'être aussi ouverte que son aîné, l'Apple *IIe*.
De plus, la plupart des extensions exigent l'intervention d'une personne qualifiée. Quant à sa durée de vie (toujours par rapport à l'Apple *IIe*, champion dans sa catégorie, nous en reparlerons...)

Guy-HACHETTE.

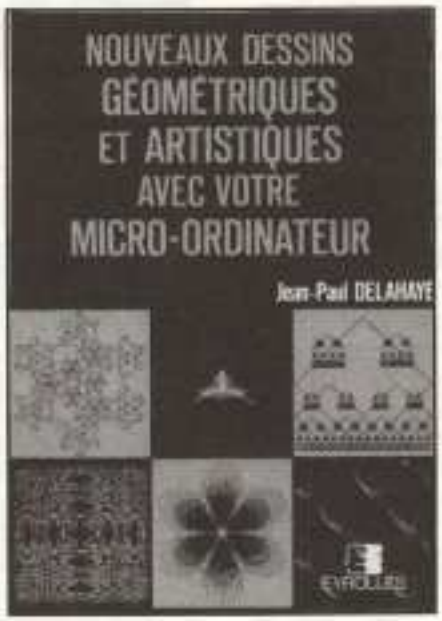
Votre bibliothèque INFORMATIQUE

• Lexique d'informatique et de micro-informatique (Eyrolles)

Un de plus ? Détrompez-vous : Jeanne Milsant a fait de la bonne besogne. Il est vrai qu'elle est maître ès Sciences et ingénieur en Informatique : cela lui donne tout de même des compétences, non ? Bref : on peut consulter son *Lexique* au fur et à mesure des nécessités, mais il n'est pas interdit de le lire séquentiellement ("séquentiel" figure, bien sûr, au nombre des termes expliqués).

Jeanne Milsant a tenu compte de la liste des nouveaux termes obligatoirement en usage dans les administrations, services et établissements publics de l'Etat, à partir de 1985. Notons que, dans ce *Lexique*, le Français est prioritaire. Je m'explique : ce sont les termes français qui sont définis, et non les mots anglais que l'on retrouve, accompagnés de leur traduction française, dans un index concis et pratique.

(170 pages sous couverture cartonnée, environ 140 F).



• Nouveaux dessins géométriques et artistiques avec votre micro-ordinateur (Eyrolles)

Jean-Paul Delahaye a voulu donner une suite à un ouvrage récent (même titre, si l'on excepte le qualificatif "Nouveaux") et il peut s'en féliciter. On peut évidemment utiliser ce second volume sans nécessairement acheter le premier.

Les programmes (300 dessins et autant de routines organisées autour de 20 blocs principaux) sont tous écrits dans un langage Basic standard (Microsoft) et sont donc adaptables sur la plupart des machines. Une annexe donne quelques exemples d'adaptations sur Apple *II*.

Un grand nombre de références concernant les arts géométriques, le dessin assisté par ordinateur, etc. figurent également à la fin du livre de Jean-Paul Delahaye. Elles seront certainement utiles à ceux qui désireront aller plus loin...

(312 pages sous couverture cartonnée, environ 160 F).

• Les fichiers en Basic sur micro-ordinateur (Eyrolles)

Si vous possédez déjà des notions de Basic et si vous ignorez comment archiver, mettre à jour et consulter les inestimables informations en votre possession, offrez-vous l'initiation de Claude Delannoy. Elle constitue un Abc sérieux sur la question, mais elle ne concerne pas UN MICRO en particulier. Toutefois, ayant choisi Apple, vous lui préférerez sans doute *"Les Fichiers en Basic sur Apple"*, du même auteur et dans la même collection, avec des programmes construits pas à pas et largement commentés... encore que je vous conseille de feuilleter les deux ouvrages chez votre libraire habituel avant de prendre une décision. D'après les lettres que reçoit la rédaction de *Tremplin Micro*, il semble que la gestion des fichiers constitue, pour bon nombre de nos lecteurs, sinon un obstacle, du moins un réel sujet de préoccupation.

(158 ou 152 pages sous couverture cartonnée, environ 120 F).

Clément RENARD.

Vous avez écrit à TREMPLIN MICRO

COMMUNICATIONS

• *Dans le N°6 — Dos de dernière page —, j'ai trouvé intéressant votre article* ***COMMUNICATIONS*** *sur VERSION-TEL. Vous y donnez des précisions fort utiles aux éventuels futurs acquéreurs. Cependant, je me permets de vous faire part de la manière dont j'ai résolu mes besoins pour émuler mon APPLE IIc en Minitel intelligent :*

La Société MARVIE — de PARIS — commercialise un Programme + un boîtier d'Interface propre à cette émulation. Il s'agit du M232.

Il faut avoir un Minitel — Gratuit de plus en plus souvent comme vous le savez —. L'interface M232 utilise donc le MODEM PTT, ce qui fait réaliser une économie qu'il est inutile de vous préciser ! Les fonctionnalités de cet ensemble — qui s'exécutent sans broncher — sont les suivantes :

- *Création de TEXTES paramétrés en fonctions et* ***temps*** *pour accéder* ***automatiquement*** *à un SERVEUR quelconque. A part la composition du N° de Tél. du serveur, les procédures se font sans intervention, y compris le stockage des pages sur la disquette.*
- *VISUALISATION et correction de ces textes.*
- *ACCÈS MANUEL au SERVEUR VIA l'APPLE qui affiche TOUTE PAGE vidéotex en MODE GRAPHIQUE complet.*
- *Modification du temps d'affichage.*
- *Passage en mode TEXTE à l'écran (en 40 ou 80 colonnes)*
- *SIMULATION des couleurs...*
- *ENVOI de toute page vidéotex directement sur TOUTE imprimante — ceci en mode TEXTE — afin d'augmenter la compatibilité. (Cet envoi peut se faire PENDANT l'accès au serveur — ce qui est peu intéressant — ou pendant une CONSULTATION de pages).*
- *Consultation des PAGES STOCKÉES et leur envoi sur le Minitel — ou sur l'imprimante en mode texte 40 / 80 colonnes.*

Voici les principales fonctions de ce programme. Je vous passe les petits détails présents dans le mode d'emploi fort bien réalisé. J'utilise ce matériel régulièrement depuis plus de deux mois maintenant, avec la plus grande satisfaction... dont celle d'avoir payé l'ensemble 750 F !

Pour terminer, je tiens à préciser que je n'ai AUCUN LIEN avec la Société MARVIE, et que ma lettre n'a qu'un but d'information. En effet, c'est après une assez longue recherche et plusieurs comparaisons de produits que je me suis orienté vers ce type de logiciel de communication.

Jean-Pierre G. (38120 ST-EGREVE)

TM Rien à ajouter à cette longue, mais très intéressante lettre.

COMPATIBILITÉ *II+* et *IIe*

• *Je souhaiterais faire bénéficier un ami d'une application personnelle (en cours de réalisation) sur un APPLE IIe, en Basic sous DOS 3.3. Malheureusement, il possède un II+ ; aussi, quelles sont ses chances de pouvoir faire tourner* ***telle quelle*** *mon application sur son II+ ?*

Le cas échéant, existe-t-il des équivalences pour certaines instructions BASIC permettant ce transfert ? Quels sont les problèmes de DOS qu'il risque de rencontrer ? Existe-t-il un artifice logiciel ?

Est-ce que la carte langage 16 K II+ pourra être utile ? Dernier point : pour corser le tout, mon application utilise 80 colonnes d'où problème de compatibilité avec celle du II+ ?

Philippe D.-N. (74290 VEYRIER-DU-LAC)

TM Les difficultés proviendront surtout de la carte 80 colonnes et auront donc trait à l'affichage. La carte langage pourra être utilisée dans la majorité des cas et dans les mêmes conditions que sur l'Apple *IIe*, mais des essais se révéleront indispensables.

Un conseil : demandez à votre ami de vous confier son unité centrale pendant plusieurs jours. Prenez aussi le temps de lire la documentation concernant sa carte langage et, surtout, sa carte 80 colonnes.

(Suite page 58)

Vous avez écrit à TREMPLIN MICRO

ET LA SCRIBE ?

• Je viens de lire dans T.M. n°6 les remarques désabusées de Michel C. (HAGUENEAU) qui a acheté une imprimante SCRIBE et qui semble vouloir la vendre parce qu'il ne sait pas s'en servir !

Ceci m'amène à faire plusieurs remarques :

1. *J'ai aussi une SCRIBE achetée à NANTES en avril 85 (2750 F) avec une doc. en français relativement bien faite. Après avoir tâtonné quelque temps, je me débrouille maintenant pour l'essentiel :*
   — graphiques par l'intermédiaire de la disquette "recopie graphique" de l'IMAGEWRITER dont le revendeur m'a donné une copie.
   — caractères spéciaux obtenus par programme en basic (ci-joint un petit programme-démo que je me suis fait et que vous pourriez peut-être faire suivre à Michel C. pour le dissuader de vendre sa machine).

2. *Votre réponse à Michel C. semble dire "vous auriez mieux fait d'acheter une IMAGEWRITER II" ; c'est pourtant vous-même qui dites, page 60 du même numéro, en réponse à Pierre G. "tous les possesseurs d'APPLE n'ont pas forcément les moyens de s'offrir une IMAGEWRITER" !*

3. *Je déplore un peu que la SCRIBE, matériel APPLE, n'ait pas (sauf erreur) eu droit à une seule ligne dans T.M. depuis sa sortie (pas plus d'ailleurs que dans l'autre revue trimestrielle consacrée à la Pomme). Son rapport qualité/prix m'a semblé intéressant et sa fringale de rubans n'est pas une tare pour une utilisation "hobbyque". Pourquoi, par exemple, dans la présentation des deux articles des pages 12 et 17 de T.M. n°6 ne pas mentionner la SCRIBE au même titre que l'IMAGEWRITER et la DMP (si le programme fonctionne avec, bien sûr) ?*

4. *L'utilisation de la couleur en impression graphique n'est pas abordée par la doc. de la SCRIBE ; alors, pourquoi pas un programme dans T.M., surtout s'il est utilisable également par l'IMAGEWRITER II ?*

5. *J'ai récemment appris par mon revendeur que la fabrication de la SCRIBE était arrêtée : serait-ce la raison du silence pesant qui règne dans les deux revues trimestrielles spécialisées ? Les possesseurs de SCRIBE seront sans doute moins nombreux que les heureux possesseurs d'IMAGEWRITER, mais ce n'est pas une raison pour les ignorer, non ?*

Bernard L. (44800 SAINT-HERBLAIN)

IMPRIMANTE SCRIBE

```
100 PRINT "METTRE L'IMPRIMANTE SOUS TE
    NSION"
105 PRINT "EN MODE BROUILLON."
110 PRINT "PUIS TAPER UNE TOUCHE."
115 WAIT - 16384,128
120 PRINT CHR$ (4)"PR£1"
125 PRINT CHR$ (27)"q" + CHR$ (14)
130 PRINT CHR$ (27)"L014" + CHR$ (27)"
    X"
135 PRINT "APERCU DES POSSIBILITES DE
    LA SCRIBE."
140 PRINT CHR$ (27)"c"
145 PRINT CHR$ (10) + CHR$ (10)
150 PRINT "Caractères NORMAUX BROUILLO
    N, tels qu'ils sortent sans comman
    de particulière"
155 PRINT CHR$ (27)"m"
160 PRINT "Caractères NORMAUX LETTRE ,
     tels qu'ils sortent avec la comma
    nde manuelle LETTER"
165 PRINT "ou en programmant CHR$ (27)
     et m entre guillemets"
170 PRINT CHR$ (27)"X"
175 PRINT "Caractères NORMAUX LETTRE s
    oulignés avec CHR$ (27) et X entre
     guillemets."
180 PRINT CHR$ (27)"Y"
185 PRINT "Fin du soulignement avec CH
    R$ (27) et Y entre guillemets."
190 PRINT CHR$ (14)
195 PRINT "EN-TETE avec CHR$ (14)."
200 PRINT CHR$ (15)
205 PRINT "Fin EN-TETE avec CHR$ (15).
    "
210 PRINT CHR$ (10)
215 PRINT "Interligne avec CHR$ (10)."
220 PRINT CHR$ (10)
225 PRINT "Voilà l'interligne."
230 PRINT "Voilà sans interligne."
235 PRINT CHR$ (27)"q"
240 PRINT "Petits caractères GRAS avec
     CHR$ (27) et q entre guillemets."
245 PRINT CHR$ (14)
250 PRINT "CARACTERES avec les deux co
    mmandes CHR$ (27) et q + CHR$ (14)
    ."
255 PRINT CHR$ (10)
260 PRINT "Ne pas depasser 68 caractèr
    es/ligne dans ce mode."
265 PRINT CHR$ (10)
270 PRINT "NOTE: ne pas oublier que l'
    imprimante continue avec le mode d
    emandé "
```

```
275 PRINT "tant qu'elle n'a pas été ét
    einte ou que le mode normal n'a pa
    s été"
280 PRINT "commandé, soit: CHR$ (27) e
    t c entre guillemets, ce qui remet
     tout à"
285 PRINT "zéro."
290 PRINT CHR$ (10)
295 PRINT CHR$ (27)"L009"
300 PRINT "Pour obtenir une marge de n
    caractères, faire CHR$ (27) et"
305 PRINT "Lnnn entre guillemets. La m
    arge est ici de 9 caractères, on"
310 PRINT "a donc fait: CHR$ (27) et L
    009 entre guillemets."
315 PRINT CHR$ (4)"PR£0"
```

TM Merci pour votre aimable collaboration. Vous avez ici la preuve que les colonnes de *T.M.* sont aussi ouvertes aux utilisateurs de la SCRIBE qui, effectivement, n'est plus commercialisée.

A noter que, d'après un autre correspondant (J.-Marc C. à ANZIN) :

① La doc. française de la SCRIBE est disponible, depuis le printemps 85, chez les concessionnaires Apple (en échange de la doc. d'origine).

② Sur Appleworks, l'accent circonflexe doit tout bêtement être tapé dans le texte, avant la lettre à accentuer.

TABULATIONS IW

• J'aimerais trouver, dans T.M., un petit programme permettant, de façon rapide (à partir d'un programme Basic ou d'un fichier de mise en page), la pose et l'utilisation des tabulations horizontales et verticales sur IMAGEWRITER.

M. G. (44130 BLAIN)

TM Dure... dure, la pose des tabulations sur IMAGEWRITER. Nos lecteurs y perdent le sommeil. Sur DMP, la recette donnée par la doc. fournissait de bons résultats... mais l'IW réagit étrangement avec les mêmes instructions... quand elle ne se contente pas de les ignorer. C'est sûrement très simple, trop pour les esprits compliqués qui se cherchent dans nos boîtes crâniennes !

CAO EN 3D

• Passionné de dessin en trois dimensions, mais guère doué pour les mathématiques, et un peu perdu dans les programmes, trop sophistiqués des livres consacrés à cette question, je suis impatient de voir paraître, dans T.M., un programme de CAO 3D, simple, structuré, performant, facile à adapter à mes propres dessins et matériels (Apple IIe, souris, joystick)...

B. B. (95160 TAVERNY)

TM Rien à voir avec B.B. (qu'est-ce que vous croyez ? je pensais à Bertrand Blier). Ce fameux logiciel existe, mais vous coûtera plus cher qu'un abonnement à *T.M.* Néanmoins, foi de Nestor, je vous promets de donner la priorité, au cours des mois à venir, à un programme inédit répondant à vos désirs.

ET SEIKOSHA ?

• Réponse, pour Monsieur S. (de Pierrevert) qui a des problèmes d'impression en 80 colonnes avec une Seikosha GP100A. J'ai un Apple IIe équipé d'une carte 80 colonnes "Chat Mauve". Au début j'avais une Seikosha GP100A, branchée par l'intermédiaire d'une interface IA83HC placée dans le Slot 1. La maigre documentation livrée avec cette carte indiquait que les listings en 80 colonnes étaient possibles. Après de longues recherches, j'ai pu trouver un "truc", pas formidable c'est vrai, mais efficace quand même. Je vous le livre, ainsi qu'à l'intention de M. Jean S. La solution consiste effectivement à taper, dans l'ordre, les instructions ci-dessous que M. S. reconnaîtra :

PR£1 : CTRL-I 80N : LIST : PR£3

Mais... car il y a un MAIS... ces ordres doivent être inclus dans le programme à lister... Il faut donc les taper sous la forme suivante et vérifier que la première ligne du programme porte un numéro commençant à 6 minimum :

1 PR£1	*4 PR£3*
2 CTRL-I 80N	*5 STOP*
3 LIST	*10 Programme à lister*

Une fois cela tapé, il suffit de faire RUN. Evidemment, il ne faut pas oublier d'enlever ces 5 lignes, le listing terminé, si on veut sauvegarder le programme. Je souhaite vivement que M. S. puisse obtenir ses listings en 80 colonnes comme j'avais réussi à en obtenir de cette manière, très contraignante, il faut le reconnaître, mais je n'ai jamais pu trouver d'autre solution.

Je dis bien "j'avais" car maintenant, je dispose d'une "IMAGEWRITER" interfacée par une "Super-Serial Card" et ma bonne Seikosha est retournée à son branchement d'origine, sur un "Dragon 32" dont le Basic Microsoft et son instruction LLIST sont infiniment plus efficaces !

R. L. (72000 LE MANS)

TM C.Q.F.D. Merci ! pour M. S. et pour d'autres lecteurs, placés devant les mêmes problèmes.

(Suite page 60)

Vous avez écrit à TREMPLIN MICRO

TOUJOURS IMAGEWRITER...

• *Comme suite à notre entretien par téléphone le 17 courant, et un entretien sur le même sujet avec quelqu'un du club APPLE. Je suis en mesure de dire :*

— Que mon interlocuteur du club APPLE ne connaît pas la fonction Gnnnn sur Imagewriter (mais que son accueil fut très sympa).

— Que le programme sur le sujet paru dans la fiche n°5, présente une anomalie lorsqu'il tourne avec IIc et IW1.

Voilà le tuyau pour obtenir les belles bordures qui ont orné les accessits et les billets d'honneur des copains de mon enfance.

```
120 PRINT CHR$(4)"PR£1" : PRINT : PRINT
    CHR$(27)"n" ; : PRINT CHR$(9)"80N" : PRINT
    CHR$(9)"Z"
```

Cela semble dû à une particularité de la carte super série (accessible uniquement par les Utilitaires). Les réglages DIP du IIc envoient un retour chariot et un Saut de Ligne après 80 caractères.

Le problème est que ce n'est pas dans le manuel de l'IW1 que se trouve la solution, ni dans celui des utilitaires, mais dans les 2 volumes (en Anglais, mais oui !) — APPLE IIc Reference Manual — non remboursés par la Sécurité Sociale, car produits de luxe. Sans vouloir être trop grossier, APPLE nous prend pour des... pigeons. Il est temps de trouver ces manuels en Français et de rêver que ces ouvrages seront offerts aux acheteurs de la susdite bécane. De plus, on se rend compte qu'Apple ne s'adresse plus aux bidouilleurs, qui ont fait sa renommée, mais se démocratise ; alors, il est souhaitable que les manuels soient plus explicites, voire assortis d'exemples. Pour ma part, c'est grâce à votre revue que j'apprends les possibilités du IIc (et ses impossibilités...). Je me rends compte que je fais là un réquisitoire contre APPLE, alors que je vous écris à vous qui palliez les carences. Continuez, mais je me plais à imaginer ce que serait votre revue si les II étaient réellement compatibles et si les manuels étaient complets. Que ne nous concocteriez-vous pas sur les pages ainsi rendues libres ?

Philippe C. (77820 LE CHATELET EN BRIE)

TM L'Imagewriter est une excellente imprimante, mais il faudrait qu'elle soit accompagnée d'un recueil de courtes routines explicatives. Il est sûr que, sur *IIc*, bon nombre d'utilisateurs rencontrent des problèmes dus à une documentation trop succincte (du moins en français). Le *IIc* et l'*IW* méritent un meilleur traitement.

MÉSAVENTURE (lettre d'un lecteur mécontent)

• *J'ai souhaité vous écrire pour vous signaler une mésaventure qui peut servir à d'éventuels acquéreurs du logiciel INSTANT PASCAL développé par THINK TCHNOLOGIE. La brochure "an introduction to programming" est dotée d'une admirable préface de John Sculley Président d'Apple. Très intéressé par le langage Pascal, j'ai désiré acquérir un logiciel susceptible de servir mes besoins de développement.*
Je suis allé chez mon vendeur habituel qui a pignon sur rue à Paris pour demander le classique PASCAL USED version 1.2.
Il ne possédait pas cette version, ni la nouvelle version 1.3 dont j'ignorais l'existence. Le vendeur m'a proposé INSTANT PASCAL.
Après avoir consulté la brochure précitée (en anglais), j'ai décidé de l'acquérir.
Le logiciel m'a coûté 1000 F...
Et pour cette somme, voilà ce que j'ai appris :

1ère SURPRISE : *La disquette MASTER est protégée. Pas de sauvegarde possible. On vous donne un double. En cas de problème, on vous invite à renvoyer la disquette défectueuse aux USA. (Les autres versions Pascal (les vraies) autorisent la copie de sauvegarde).*

2e SURPRISE : *La disquette de démonstration est dotée d'un formidable BUG qui vous empêche de poursuivre au-delà de la leçon 5. Heureusement il y a une version "clavier (KEYBOARD dans le texte)".*

3e SURPRISE : *Le livret fourni dans le kit vous propose, si vous passez aux choses sérieuses, d'acquérir "INSTANT PASCAL REFERENCE MANUAL". Ils le disent être disponible chez votre libraire habituel. Essayez, même chez A. Seedrin, on ne connaît pas !*

4e SURPRISE : *Dans la même brochure d'introduction à la programmation on vous signale que INSTANT PASCAL n'est pas compilé, mais interprété comme n'importe quel BASIC. Traduisez : je possède le PASCAL le plus lent de la planète.*

5e SURPRISE : *Vous ne pouvez pas développer de programme indépendant. De toute façon chez A. Seedrin on m'a répondu que le programme développé par l'utilisateur ne peut dépasser 20 K.*

Conclusion : question surprises, j'en ai pour mon argent !

En vous remerciant de votre attention et espérant que vous aurez assez de place pour insérer mon article dans le numéro de Mai.

Je vous prie de croire en l'expression de mes sentiments les meilleurs.

M. LAMBERT (9400 CRÉTEIL).

CODES ASCII, HEXA et BINAIRES

QUE l'on programme ou non en langage machine, il est souvent nécessaire de connaître la représentation binaire ou hexadécimale d'un caractère. La partie gauche de nos tableaux concerne les 128 codes ASCII standard (bit 7 à 0). La partie droite concerne les codes supérieurs de ces mêmes caractères (bit 7 à 1).

© CTRL

CODES ASCII INFÉRIEURS				CODES ASCII SUPÉRIEURS		
DÉC.	HEX.	BINAIRE 7 6 5 4 3 2 1 0		DÉC.	HEX.	BINAIRE 7 6 5 4 3 2 1 0
0	$00	0 0 0 0 0 0 0 0	©- à	128	$80	1 0 0 0 0 0 0 0
1	$01	0 0 0 0 0 0 0 1	©- A	129	$81	1 0 0 0 0 0 0 1
2	$02	0 0 0 0 0 0 1 0	©- B	130	$82	1 0 0 0 0 0 1 0
3	$03	0 0 0 0 0 0 1 1	©- C	131	$83	1 0 0 0 0 0 1 1
4	$04	0 0 0 0 0 1 0 0	©- D	132	$84	1 0 0 0 0 1 0 0
5	$05	0 0 0 0 0 1 0 1	©- E	133	$85	1 0 0 0 0 1 0 1
6	$06	0 0 0 0 0 1 1 0	©- F	134	$86	1 0 0 0 0 1 1 0
7	$07	0 0 0 0 0 1 1 1	©- G	135	$87	1 0 0 0 0 1 1 1
8	$08	0 0 0 0 1 0 0 0	©- H	136	$88	1 0 0 0 1 0 0 0
9	$09	0 0 0 0 1 0 0 1	©- I	137	$89	1 0 0 0 1 0 0 1
10	$0A	0 0 0 0 1 0 1 0	©- J	138	$8A	1 0 0 0 1 0 1 0
11	$0B	0 0 0 0 1 0 1 1	©- K	139	$8B	1 0 0 0 1 0 1 1
12	$0C	0 0 0 0 1 1 0 0	©- L	140	$8C	1 0 0 0 1 1 0 0
13	$0D	0 0 0 0 1 1 0 1	©- M	141	$8D	1 0 0 0 1 1 0 1
14	$0E	0 0 0 0 1 1 1 0	©- N	142	$8E	1 0 0 0 1 1 1 0
15	$0F	0 0 0 0 1 1 1 1	©- O	143	$8F	1 0 0 0 1 1 1 1
16	$10	0 0 0 1 0 0 0 0	©- P	144	$90	1 0 0 1 0 0 0 0
17	$11	0 0 0 1 0 0 0 1	©- Q	145	$91	1 0 0 1 0 0 0 1
18	$12	0 0 0 1 0 0 1 0	©- R	146	$92	1 0 0 1 0 0 1 0
19	$13	0 0 0 1 0 0 1 1	©- S	147	$93	1 0 0 1 0 0 1 1
20	$14	0 0 0 1 0 1 0 0	©- T	148	$94	1 0 0 1 0 1 0 0
21	$15	0 0 0 1 0 1 0 1	©- U	149	$95	1 0 0 1 0 1 0 1
22	$16	0 0 0 1 0 1 1 0	©- V	150	$96	1 0 0 1 0 1 1 0
23	$17	0 0 0 1 0 1 1 1	©- W	151	$97	1 0 0 1 0 1 1 1
24	$18	0 0 0 1 1 0 0 0	©- X	152	$98	1 0 0 1 1 0 0 0
25	$19	0 0 0 1 1 0 0 1	©- Y	153	$99	1 0 0 1 1 0 0 1
26	$1A	0 0 0 1 1 0 1 0	©- Z	154	$9A	1 0 0 1 1 0 1 0
27	$1B	0 0 0 1 1 0 1 1	©- °	155	$9B	1 0 0 1 1 0 1 1
28	$1C	0 0 0 1 1 1 0 0	©- ç	156	$9C	1 0 0 1 1 1 0 0
29	$1D	0 0 0 1 1 1 0 1	©- §	157	$9D	1 0 0 1 1 1 0 1
30	$1E	0 0 0 1 1 1 1 0	©- ^	158	$9E	1 0 0 1 1 1 1 0
31	$1F	0 0 0 1 1 1 1 1	©- _	159	$9F	1 0 0 1 1 1 1 1
32	$20	0 0 1 0 0 0 0 0	sp	160	$A0	1 0 1 0 0 0 0 0
33	$21	0 0 1 0 0 0 0 1	!	161	$A1	1 0 1 0 0 0 0 1
34	$22	0 0 1 0 0 0 1 0	"	162	$A2	1 0 1 0 0 0 1 0
35	$23	0 0 1 0 0 0 1 1	£	163	$A3	1 0 1 0 0 0 1 1
36	$24	0 0 1 0 0 1 0 0	$	164	$A4	1 0 1 0 0 1 0 0
37	$25	0 0 1 0 0 1 0 1	%	165	$A5	1 0 1 0 0 1 0 1
38	$26	0 0 1 0 0 1 1 0	&	166	$A6	1 0 1 0 0 1 1 0
39	$27	0 0 1 0 0 1 1 1	'	167	$A7	1 0 1 0 0 1 1 1
40	$28	0 0 1 0 1 0 0 0	(	168	$A8	1 0 1 0 1 0 0 0
41	$29	0 0 1 0 1 0 0 1	)	169	$A9	1 0 1 0 1 0 0 1
42	$2A	0 0 1 0 1 0 1 0	*	170	$AA	1 0 1 0 1 0 1 0
43	$2B	0 0 1 0 1 0 1 1	+	171	$AB	1 0 1 0 1 0 1 1
44	$2C	0 0 1 0 1 1 0 0	,	172	$AC	1 0 1 0 1 1 0 0
45	$2D	0 0 1 0 1 1 0 1	-	173	$AD	1 0 1 0 1 1 0 1
46	$2E	0 0 1 0 1 1 1 0	.	174	$AE	1 0 1 0 1 1 1 0
47	$2F	0 0 1 0 1 1 1 1	/	175	$AF	1 0 1 0 1 1 1 1
48	$30	0 0 1 1 0 0 0 0	0	176	$B0	1 0 1 1 0 0 0 0
49	$31	0 0 1 1 0 0 0 1	1	177	$B1	1 0 1 1 0 0 0 1
50	$32	0 0 1 1 0 0 1 0	2	178	$B2	1 0 1 1 0 0 1 0
51	$33	0 0 1 1 0 0 1 1	3	179	$B3	1 0 1 1 0 0 1 1
52	$34	0 0 1 1 0 1 0 0	4	180	$B4	1 0 1 1 0 1 0 0
53	$35	0 0 1 1 0 1 0 1	5	181	$B5	1 0 1 1 0 1 0 1
54	$36	0 0 1 1 0 1 1 0	6	182	$B6	1 0 1 1 0 1 1 0
55	$37	0 0 1 1 0 1 1 1	7	183	$B7	1 0 1 1 0 1 1 1
56	$38	0 0 1 1 1 0 0 0	8	184	$B8	1 0 1 1 1 0 0 0

(suite au verso)

TREMPLIN MICRO FICHE 8

CODES ASCII INFÉRIEURS				CODES ASCII SUPÉRIEURS		
DÉC.	HEX.	BINAIRE 7 6 5 4 3 2 1 0		DÉC.	HEX.	BINAIRE 7 6 5 4 3 2 1 0
57	$39	0 0 1 1 1 0 0 1	9	185	$B9	1 0 1 1 1 0 0 1
58	$3A	0 0 1 1 1 0 1 0	:	186	$BA	1 0 1 1 1 0 1 0
59	$3B	0 0 1 1 1 0 1 1	;	187	$BB	1 0 1 1 1 0 1 1
60	$3C	0 0 1 1 1 1 0 0	<	188	$BC	1 0 1 1 1 1 0 0
61	$3D	0 0 1 1 1 1 0 1	=	189	$BD	1 0 1 1 1 1 0 1
62	$3E	0 0 1 1 1 1 1 0	>	190	$BE	1 0 1 1 1 1 1 0
63	$3F	0 0 1 1 1 1 1 1	?	191	$BF	1 0 1 1 1 1 1 1
64	$40	0 1 0 0 0 0 0 0	à	192	$C0	1 1 0 0 0 0 0 0
65	$41	0 1 0 0 0 0 0 1	A	193	$C1	1 1 0 0 0 0 0 1
66	$42	0 1 0 0 0 0 1 0	B	194	$C2	1 1 0 0 0 0 1 0
67	$43	0 1 0 0 0 0 1 1	C	195	$C3	1 1 0 0 0 0 1 1
68	$44	0 1 0 0 0 1 0 0	D	196	$C4	1 1 0 0 0 1 0 0
69	$45	0 1 0 0 0 1 0 1	E	197	$C5	1 1 0 0 0 1 0 1
70	$46	0 1 0 0 0 1 1 0	F	198	$C6	1 1 0 0 0 1 1 0
71	$47	0 1 0 0 0 1 1 1	G	199	$C7	1 1 0 0 0 1 1 1
72	$48	0 1 0 0 1 0 0 0	H	200	$C8	1 1 0 0 1 0 0 0
73	$49	0 1 0 0 1 0 0 1	I	201	$C9	1 1 0 0 1 0 0 1
74	$4A	0 1 0 0 1 0 1 0	J	202	$CA	1 1 0 0 1 0 1 0
75	$4B	0 1 0 0 1 0 1 1	K	203	$CB	1 1 0 0 1 0 1 1
76	$4C	0 1 0 0 1 1 0 0	L	204	$CC	1 1 0 0 1 1 0 0
77	$4D	0 1 0 0 1 1 0 1	M	205	$CD	1 1 0 0 1 1 0 1
78	$4E	0 1 0 0 1 1 1 0	N	206	$CE	1 1 0 0 1 1 1 0
79	$4F	0 1 0 0 1 1 1 1	O	207	$CF	1 1 0 0 1 1 1 1
80	$50	0 1 0 1 0 0 0 0	P	208	$D0	1 1 0 1 0 0 0 0
81	$51	0 1 0 1 0 0 0 1	Q	209	$D1	1 1 0 1 0 0 0 1
82	$52	0 1 0 1 0 0 1 0	R	210	$D2	1 1 0 1 0 0 1 0
83	$53	0 1 0 1 0 0 1 1	S	211	$D3	1 1 0 1 0 0 1 1
84	$54	0 1 0 1 0 1 0 0	T	212	$D4	1 1 0 1 0 1 0 0
85	$55	0 1 0 1 0 1 0 1	U	213	$D5	1 1 0 1 0 1 0 1
86	$56	0 1 0 1 0 1 1 0	V	214	$D6	1 1 0 1 0 1 1 0
87	$57	0 1 0 1 0 1 1 1	W	215	$D7	1 1 0 1 0 1 1 1
88	$58	0 1 0 1 1 0 0 0	X	216	$D8	1 1 0 1 1 0 0 0
89	$59	0 1 0 1 1 0 0 1	Y	217	$D9	1 1 0 1 1 0 0 1
90	$5A	0 1 0 1 1 0 1 0	Z	218	$DA	1 1 0 1 1 0 1 0
91	$5B	0 1 0 1 1 0 1 1	°	219	$DB	1 1 0 1 1 0 1 1
92	$5C	0 1 0 1 1 1 0 0	ç	220	$DC	1 1 0 1 1 1 0 0
93	$5D	0 1 0 1 1 1 0 1	§	221	$DD	1 1 0 1 1 1 0 1
94	$5E	0 1 0 1 1 1 1 0	^	222	$DE	1 1 0 1 1 1 1 0
95	$5F	0 1 0 1 1 1 1 1	_	223	$DF	1 1 0 1 1 1 1 1
96	$60	0 1 1 0 0 0 0 0	\	224	$E0	1 1 1 0 0 0 0 0
97	$61	0 1 1 0 0 0 0 1	a	225	$E1	1 1 1 0 0 0 0 1
98	$62	0 1 1 0 0 0 1 0	b	226	$E2	1 1 1 0 0 0 1 0
99	$63	0 1 1 0 0 0 1 1	c	227	$E3	1 1 1 0 0 0 1 1
100	$64	0 1 1 0 0 1 0 0	d	228	$E4	1 1 1 0 0 1 0 0
101	$65	0 1 1 0 0 1 0 1	e	229	$E5	1 1 1 0 0 1 0 1
102	$66	0 1 1 0 0 1 1 0	f	230	$E6	1 1 1 0 0 1 1 0
103	$67	0 1 1 0 0 1 1 1	g	231	$E7	1 1 1 0 0 1 1 1
104	$68	0 1 1 0 1 0 0 0	h	232	$E8	1 1 1 0 1 0 0 0
105	$69	0 1 1 0 1 0 0 1	i	233	$E9	1 1 1 0 1 0 0 1
106	$6A	0 1 1 0 1 0 1 0	j	234	$EA	1 1 1 0 1 0 1 0
107	$6B	0 1 1 0 1 0 1 1	k	235	$EB	1 1 1 0 1 0 1 1
108	$6C	0 1 1 0 1 1 0 0	l	236	$EC	1 1 1 0 1 1 0 0
109	$6D	0 1 1 0 1 1 0 1	m	237	$ED	1 1 1 0 1 1 0 1
110	$6E	0 1 1 0 1 1 1 0	n	238	$EE	1 1 1 0 1 1 1 0
111	$6F	0 1 1 0 1 1 1 1	o	239	$EF	1 1 1 0 1 1 1 1
112	$70	0 1 1 1 0 0 0 0	p	240	$F0	1 1 1 1 0 0 0 0
113	$71	0 1 1 1 0 0 0 1	q	241	$F1	1 1 1 1 0 0 0 1
114	$72	0 1 1 1 0 0 1 0	r	242	$F2	1 1 1 1 0 0 1 0
115	$73	0 1 1 1 0 0 1 1	s	243	$F3	1 1 1 1 0 0 1 1
116	$74	0 1 1 1 0 1 0 0	t	244	$F4	1 1 1 1 0 1 0 0
117	$75	0 1 1 1 0 1 0 1	u	245	$F5	1 1 1 1 0 1 0 1
118	$76	0 1 1 1 0 1 1 0	v	246	$F6	1 1 1 1 0 1 1 0
119	$77	0 1 1 1 0 1 1 1	w	247	$F7	1 1 1 1 0 1 1 1
120	$78	0 1 1 1 1 0 0 0	x	248	$F8	1 1 1 1 1 0 0 0
121	$79	0 1 1 1 1 0 0 1	y	249	$F9	1 1 1 1 1 0 0 1
122	$7A	0 1 1 1 1 0 1 0	z	250	$FA	1 1 1 1 1 0 1 0
123	$7B	0 1 1 1 1 0 1 1	é	251	$FB	1 1 1 1 1 0 1 1
124	$7C	0 1 1 1 1 1 0 0	ù	252	$FC	1 1 1 1 1 1 0 0
125	$7D	0 1 1 1 1 1 0 1	è	253	$FD	1 1 1 1 1 1 0 1
126	$7E	0 1 1 1 1 1 1 0	..	254	$FE	1 1 1 1 1 1 1 0
127	$7F	0 1 1 1 1 1 1 1	□	255	$FF	1 1 1 1 1 1 1 1

tremplin micro 8

EN PROMOTION

AMPERELO d'Yvan KOENIG

Bulletin de commande et d'abonnement

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

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)

QUANTITÉ	DÉSIGNATION	PRIX UNITAIRE	TOTAL
	Abonnement à 6 numéros (un an) : FRANCE	190 F	
	Abonnement à 6 disquettes (un an) : FRANCE	600 F	
	MINIE, LOGICIEL DE CLAUDE AUBRY	180 F	
	AMPÉRELO d'Yvan KOENIG	170 F	
	RELIURE-ECRIN	40 F	
	DISQUETTE-INDEX (N°1 à 6)	30 F	
	DISQUETTE DE TREMPLIN MICRO :* *(IIe et IIc)* N°1 ☐ N°4 ☐ N°7 ☐ N°2 ☐ N°5 ☐ N°8 ☐ N°3 ☐ N°6 ☐	105 F	
	NUMÉRO DE TREMPLIN MICRO :* N°1 ☐ N°2 ☐ N°3 ☐ N°4 ☐ N°5 ☐ N°6 ☐	30 F	
	NUMÉRO DE TREMPLIN MICRO :* N°7 ☐ N°8 ☐	33 F	

* *Cochez la (ou les) case(s) de votre choix.*

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

Envoi en paquet-poste recommandé à partir de 400 F

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

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

Mode de règlement choisi : ☐ Chèque ☐ Mandat-lettre ☐ Carte Bleue

VISA

N° de votre Carte Bleue |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|

Date d'expiration |_|_|_|_| **Signature** (obligatoire)

Montant à débiter _____ F

tremplin micro

SONDAGE du numéro 8

Les renseignements concernant votre identité et votre adresse ne seront en aucun cas communiqués à une société étrangère aux EDITIONS JIBENA.

TREMPLIN MICRO — La Petite Motte — Senillé — 86100 CHÂTELLERAULT

QUI ÊTES-VOUS ?

** Facultatif*

Nom* ..... Prénom * .....

Année de naissance ..... Sexe ..... Profession .....

Adresse* .....

Code postal |_|_|_|_|_| Ville .....

VOTRE MATÉRIEL

Marquer d'une X les cases qui vous concernent.

☐ Apple II ☐ Apple II + ☐ Apple IIe ☐ Apple IIc avec ☐ 65Ø2 ☐ 65CØ2

Autres : .....

VOS OUTILS

Citez les quatre logiciels que vous utilisez le plus souvent **par ordre de préférence.**

1. ..... **3.** .....

2. ..... **4.** .....

ANCIENNETÉ

Cocher les cases de votre choix.

Depuis combien de temps possédez-vous un ordinateur personnel ? .....

Concevez-vous des programmes ? OUI ☐ NON ☐

Langage BASIC ☐ PASCAL ☐ LOGO ☐ AUTRE ☐ Précisez : .....

Programmez-vous en **LANGAGE MACHINE** ? OUI ☐ NON ☐

Utilisez-vous un assembleur ? OUI ☐ NON ☐ si OUI lequel ? .....

INTENTIONS D'ACHAT

Que souhaitez-vous acquérir, au cours des 6 prochains mois, en matière d'informatique personnelle ?

.....

RUBRIQUES PRÉFÉRÉES

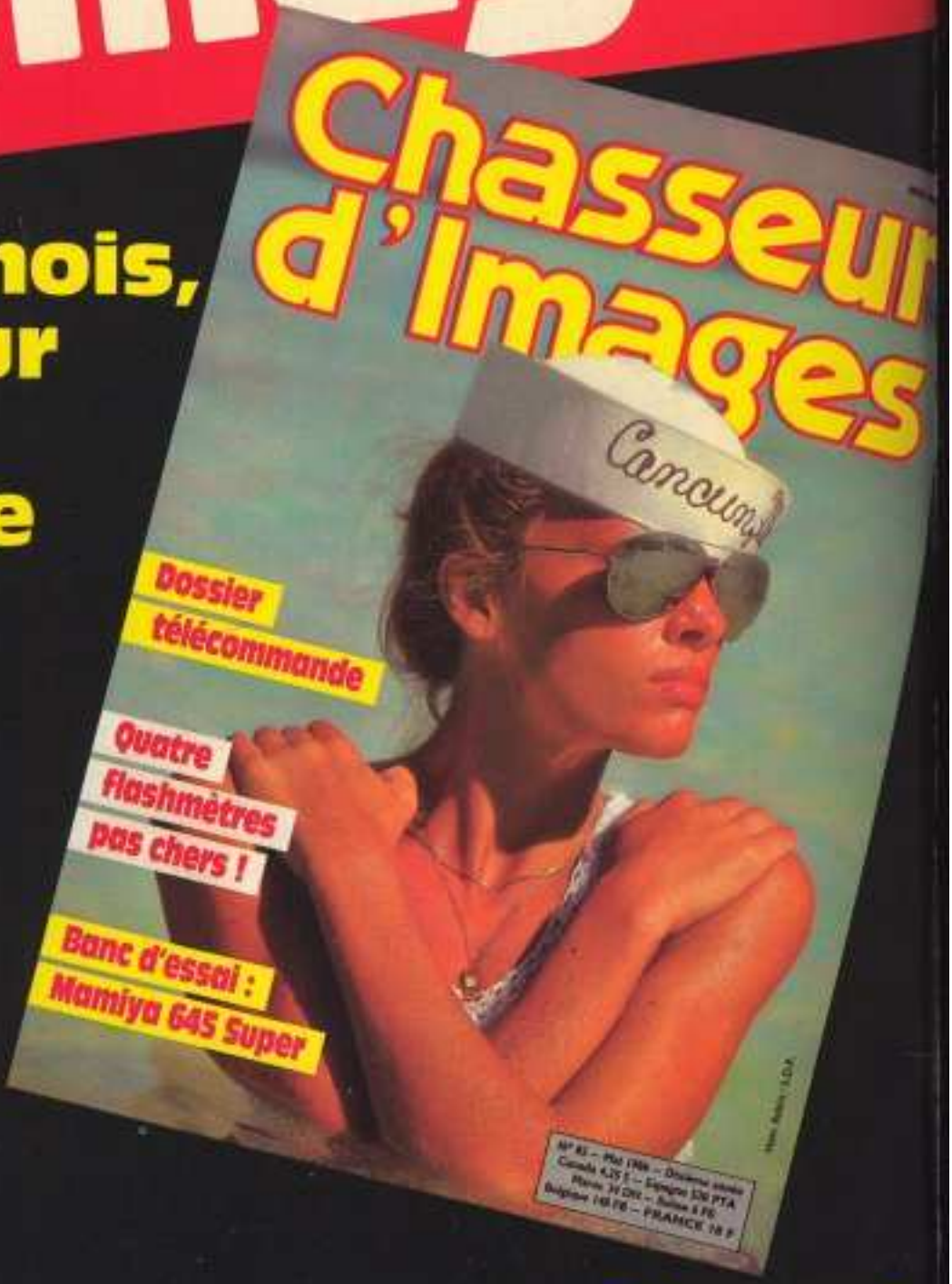
Indiquez-moi le numéro de la première page des rubriques que vous avez préférées :

.....

.....

Chasseur d'images

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



Chez votre marchand de journaux !