

tremplin micro

**Mettez votre Apple
à l'heure !**

**Ouvrez
une fenêtre
sur 80 colonnes !**

**Une nouvelle
commande
pour ProDOS**

Ecrans condensés

**Caractères souris
sur DMP ou ImageWriter**

**Copie d'écran
en 80 colonnes
sur ImageWriter**



N°9 - Bimestriel - Deuxième année
3 Juillet - 2 Septembre 1986
254 FB - 11 FS - 33 F

QUATRE OUVRAGES MAJEURS DU PSI

Clément RENARD

● ASSEMBLEUR DE L'APPLE II

Nicole BRÉAUD-POULIQUEN et son complice Daniel-Jean DAVID récidivent. Il s'agit en fait d'une réédition, mais tenant compte des nouvelles instructions du 65C02 et de l'existence de ProDOS.

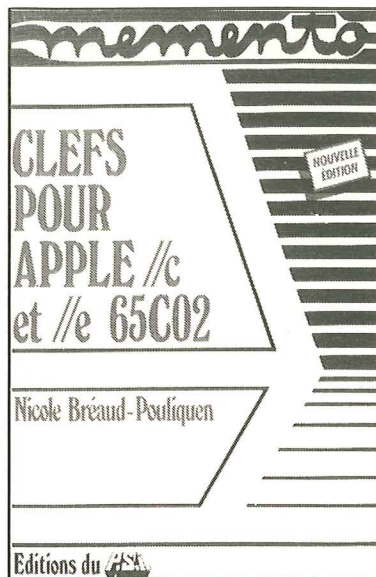
La présentation est améliorée : format 170 × 250, photocomposition, mise en page... De plus, les listes proviennent de l'Assembleur français ProCODE, ce qui nous comble de joie. L'auteur de ProCODE méritait bien ce coup de chapeau. Pour le reste, ce que nous avons déjà écrit sur cet ouvrage reste valable : c'est une sorte de Bible pour celles et ceux qui désirent s'initier à la programmation. Personnellement, j'aurais souhaité, dans cette édition remaniée, de plus nombreux exemples... mais que deviendraient des revues comme Tremplin Micro si l'on trouvait tout dans les livres ?



● CLEFS POUR APPLE IIc et IIe 65C02

Toujours avec Nicole Bréaud-Pouliquen, et encore avec une nouvelle édition revue et augmentée !

Les Applemaniques connaissent tous ce bouquin à spirale pour l'avoir d'abord exploité sur leur IIe, puis sur leur IIc. On avait pu, autrefois, déplorer la mauvaise lisibilité de la première version, mais ce défaut de jeunesse à disparu. Je vous recommande — et sans aucune restriction — ces *CLEFS POUR APPLE IIc et IIe 65C02*. En cherchant bien, vous y trouverez certainement toutes les adresses capables de faire votre bonheur.



● SYSTEME PRODOS DE L'APPLE II

Marcel COTTINI vous propose là un véritable manuel de référence. Qu'est-ce que le DOS 3.3... mais, surtout, quelle est exactement la structure de ProDOS ? Vous allez le découvrir dans cet ouvrage copieux (275 pages dans le format 170 × 250), mais sûrement pas sans effort.

C'est un bouquin destiné aux hobbyistes et c'est une étude de haut niveau. Si vous êtes suffisamment persévérant pour l'étudier de A à Z, devant votre Apple, il est certain que ProDOS n'aura plus guère de secrets pour vous : gestion des fichiers, organisation de la disquette ProDOS, gestion des

interruptions du 65C02, récupération avec NMI, IRQ et RESET... tout y passe. Notez que des annexes vous fourniront toutes les tables dont vous pouvez rêver.



● PROGRAMMATION SYSTEME DE L'APPLE II

Le même Marcel COTTINI espère vous aider à découvrir dans les 256 pages de ce gros recueil, les aspects techniques du BASIC APPLESOFT et les routines principales des microprocesseurs 6502 et 65C02.

Question immédiate : fait-il double emploi avec les deux livres de Nicole Bréaud-Pouliquen ? La réponse est non. Certes — et c'est bien normal — on y trouve des redites, mais la plupart des informations données par Marcel COTTINI me paraissent bien commentées et capables d'aider bon nombre de programmeurs débutants. Des POKE, PEEK et CALL autant qu'il est possible d'en espérer... mais aussi de petites routines qui, même quand elles sont déjà connues, rendront probablement service : Musique, Arrêt du Catalogue par ESC, Disquettes gonflées, etc. A lire... pardon : à décortiquer !

tremplin micro

9

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

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 :

Joëlle (même numéro)

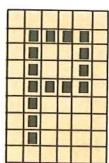
Commission paritaire :

Demande en cours.

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

TREPLIN 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.

Un jeu sans cesse renouvelé



ROGRAMMER est encore un jeu. C'est même l'une des plus captivantes récréations intellectuelles de notre époque. Lectrices et Lecteurs de *Tremplin Micro* en savent quelque chose. En effet, pour la plupart, ils ne se contentent plus de pratiquer le Basic, mais louchent vers le langage machine.

Jamais les ouvrages expliquant les mystères du 6502 (et du 65C02) ne se sont aussi bien vendus qu'actuellement. Il faut dire que les auteurs s'en donnent à cœur joie : Nicole Bréaud-Pouliquen, Marcel Cottini, etc. Toutefois, n'est-il pas trop tard pour se pencher sur la programmation d'un microprocesseur qui a son avenir... derrière lui ? Evidemment pas. Quand on se joue des instructions d'un 6502, on n'en est que mieux armé pour utiliser celles, plus nombreuses, d'un 68000, par exemple.

Pourquoi programmer en langage machine ? D'abord pour le plaisir, et peut-être parce que c'est moins évident que de consommer les instructions prédigérées de l'Applesoft. Ensuite parce que réaliser certains sous-programmes en langage machine permet en général de diviser les temps de traitement par quatre, voire par dix. Personnellement, je me souviens avoir remplacé une boucle destinée à inclure un CHR\$(8) dans une variable, à la suite des accents circonflexes et des trémas : ce n'était qu'un travail d'amateur, mais dont les résultats me parurent alors excellents.

C'est dire que *Tremplin Micro* ne modifiera pas sa politique actuelle. Evidemment, et vous le comprenez, c'est une tâche difficile, exigeant des efforts constants de la part de l'Editeur et de la Rédaction, ainsi qu'un soutien amical de la part de tous les Lecteurs*. Vous pouvez compter sur nous. Prouvez-nous, en faisant connaître *Tremplin Micro* à vos amis que vous appréciez Votre revue et la démarche générale de son responsable.

Merci de votre attention et **BONNE PROGRAMMATION !**

GUY-HACHETTE.

* Notre O.J.D. prouve en tout cas que plus de 11000 Lecteurs font confiance à *Tremplin Micro*.

Sommaire 9

Un numéro bourré de programmes et de renseignements utiles. Accordez-vous le temps de le disséquer. Vous constaterez que, **Tremplin Micro** en main, votre Apple vous réserve encore bon nombre de surprises agréables !

DES UTILITAIRES

- Une nouvelle incursion dans la mémoire de l'Apple 11
- Copie d'écran en 80 colonnes sur ImageWriter 19
- Caractères souris sur DMP ou ImageWriter 21

ÉCRANS CONDENSÉS Maurice CHAVELLI

Nouvelle version avec fenêtres 24

SPÉCIAL PRODOS

Super catalogue ProDOS 28

TYPIC : une nouvelle commande pour ProDOS

Une importante contribution de notre ami Yvan KOENIG 31

Mettez votre Apple à l'heure

Un programme de Marcel COTTINI 40

GRAPHISME

- ROTATION... ou comment faire tourner une partie d'image en Haute résolution ? 13
- Faux graphisme... ou comment éditer HISTO.TEXT (n°8) sur DMP ou IMAGEWRITER 1, sans les caractères souris 23
- Transformation d'une fonte imprimante en fonte graphique 51
- Copie HGR, en HGR2
Inversion d'image HGR 60

RUBRIQUES

- A propos de CLEAR et DEL 6
- Bibliothèque informatique 10, 65
- CLAVIER SECRET 17
- Que deviendra votre Apple ? 18

CHASSEURS DE BOGUES

- CARAC.CHARG, AUTO.COÛT, COMBIEN DE FEMMES ? 44
- VILLES DE FRANCE, DATA.GENERATOR, TRAJET MINUTÉ, ECRANS 80 COLONNES ... 45
- Questions/réponses 46
 - POKE 243,32
 - MID\$ abrégé
 - Touche... à mon clavier !
 - Reset.Boot
 - Où est le curseur ?
- Menu-type 53
- Conseils aux utilisateurs d'un Apple IIc avec imprimante 53
- Langage-machine (initiation) 55

OUVREZ UNE FENÊTRE SUR 80 COLONNES

- Un programme à étudier 3

UNE IDÉE DE JEU

- Mots en spirale 7

LISEZ AUSSI

- Des pommes par milliers 48
- Cercle économique 50

LE COURRIER DES LECTEURS COMMENCE PAGE 56

CURIOSITÉ

- Un INPUT en mode direct 47

TREMPIN MICRO N°10

paraîtra le 3 septembre 1986
(disquette disponible à partir du 20 août)



Ouvrez une fenêtre sur 80 colonnes



NE voyez, dans ce court programme, qu'un exemple destiné à montrer comment ouvrir une fenêtre dans un écran de 80 colonnes (nous avons déjà abordé le problème — mais sur 40 colonnes — dans le numéro 4 de *Tremplin Micro*).

Dans cette routine, la fenêtre est toujours ouverte à partir de HTAB1 et de VTAB1, mais ces paramètres sont évidemment modifiables, à condition d'allonger sensiblement le programme.

La sauvegarde de l'écran se fait à partir de la routine publiée sur la Fiche n°2 de *TM*.

Selon le microprocesseur utilisé, il faudra modifier le contenu de certaines adresses ainsi, avec les Apple IIe non transformés (6502), l'affichage sera obtenu par \$FDED (au lieu de \$FDF6). Il sera moins rapide. Si vous suivez pas à pas les diverses explications ci-après, la création d'une fenêtre ne devrait plus vous poser de problème.

FENÊTRE 80 COLONNES

100 : Attention ! *FEN* devra être modifié si vous disposez d'un Apple IIe avec 6502.

105 : Reportez-vous à la fiche du *Tremplin Micro* n°2 si vous désirez obtenir des éclaircissements sur *FENMEMO*. Cette routine est installée à partir de l'adresse \$2000... d'où le CALL 8192 de la ligne 175.

170 : POKE 6, LARGEUR et POKE 7, HAUTEUR — 1.

```

100 PRINT CHR$(4)"BLOAD FEN": REM ROUTINE POUR APPLE 65
    C02
105 PRINT CHR$(4)"BLOAD FENMEMO": REM TRANSFERT 80 COL
    (FICHE TM2)

110 TEXT : PRINT CHR$(4)"PR£3": PRINT : GOSUB 240
115 VTAB 1
120 :
125 REM DIMENSIONS DE FENETRE **
130 :
135 PRINT : VTAB 21: POKE 1403,42: INPUT "LARGEUR DE LA
    FENETRE (10-40) ";L$: IF L$ = "" THEN 220
140 L = VAL (L$): IF L < 10 OR L > 40 THEN 135
145 PRINT : VTAB 22: POKE 1403,42: INPUT "HAUTEUR DE LA
    FENETRE (10-23) ";H$: IF H$ = "" THEN 220
150 H = VAL (H$): IF H < 10 OR H > 23 THEN 145
155 :
160 REM TRANSMISSION PARAM. LM **
165 :
170 POKE 6,L: POKE 7,H - 1
175 CALL 8192: REM EXEMPLE DE MEMORISATION (FICHE TM2)
180 CALL 768
185 LIST
  
```

FENETRE

```

190 :
195 REM AUTRE FENETRE OU FIN **
200 :
205 GOSUB 215: CALL 852: CALL 8239: REM RECUPERATION DE
L'ECRAN
210 GOTO 135
215 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: PO
KE - 16368,0: RETURN
220 NORMAL : HOME : END
225 :
230 REM REMPLISSAGE DE L'ECRAN **
235 :
240 POKE 33,40: FOR I = 867 TO 892: READ R: POKE I,R: NE
XT : CALL 867: RETURN
245 DATA 32,88,252,169,212,32,246,253,174,226,7,224,160,
240,246,162,80,134,33,32,246,253,32,246,253,96
    
```

205 : CALL 8239 renvoie à FENMEMO, mais cette fois, il s'agit de réafficher la page-écran stockée en mémoire auxiliaire.

245 : Le remplissage de l'écran est intéressant à étudier. Avec Apple IIe-6502, remplacez les 7^e, 21^e et 24^e données (246) par 237 (voir ci-après, page 6, à partir de \$363).

FEN

Avec un Apple IIe-6502, placez ED dans les mémoires 319, 320, 328, 32F et 03 dans la mémoire 349.

VIDER LA FENÊTRE

0300-	A6 06	LDX	\$06	Registre X chargé avec le contenu de la mémoire \$6
0302-	86 21	STX	\$21	Stockage en \$21 (le POKE 33,X du Basic)
0304-	A6 07	LDX	\$07	Registre X chargé avec le contenu de la mémoire \$7
0306-	86 23	STX	\$23	Stockage en \$23 (le POKE 35,X du Basic)
0308-	CA	DEX		X est décrémenté (X = X - 1)
0309-	86 08	STX	\$08	Ecriture dans la mémoire \$08 (page zéro)
030B-	20 58 FC	JSR	\$FC58	Home à l'intérieur de la fenêtre définie

CADRE (partie supérieure)

030E-	A9 18	LDA	£\$18	Accumulateur chargé avec \$18 (24 en décimal)
0310-	85 23	STA	\$23	Ecriture dans \$23 (Bas de fenêtre standard)
0312-	A9 3F	LDA	£\$3F	Accumulateur chargé avec \$3F (63)...
0314-	85 32	STA	\$32	... pour obtenir le mode inverse (POKE 50,63)
0316-	A9 A0	LDA	£\$A0	Accumulateur chargé avec \$A0 (espace)
0318-	20 F6 FD	JSR	\$FDF6	\$FDF6 affiche un espace en mode inverse
031B-	A6 25	LDX	\$25	Autrement dit : X = PEEK(37), position vert. du curseur
031D-	F0 F9	BEQ	\$0318	Si ce n'est pas 0, encore un tour !

CADRE (les côtés)

031F-	20 F6 FD	JSR	\$FDF6	Affichage d'un espace en mode inverse (côté gauche)
0322-	A6 06	LDX	\$06	X chargé avec la largeur de la fenêtre
0324-	CA	DEX		X = X - 1
0325-	86 24	STX	\$24	POKE 36,X... si l'on préfère
0327-	20 F6 FD	JSR	\$FDF6	Affichage d'un espace inverse (côté droit)
032A-	C6 08	DEC	\$08	Mémoire 8 décrémentée de 1
032C-	D0 F1	BNE	\$031F	Si X différent de 0, GOTO \$31F

CADRE DE LA FENÊTRE (BAS)

032E-	20 F6 FD	JSR	\$FDF6	Affichage espace mode inverse
0331-	A6 25	LDX	\$25	Lecture position verticale du curseur (X = PEEK(37))
0333-	E4 07	CPX	\$07	Comparaison avec contenu mémoire \$7 (hauteur fenêtre)
0335-	F0 F7	BEQ	\$032E	Si identique GOTO \$32E, sinon terminé !

BLOCAGE DE LA FENÊTRE

0337-	A9 FF	LDA	£\$FF] Retour mode normal (POKE 50,255 en Basic)
0339-	85 32	STA	\$32	
033B-	A9 02	LDA	£\$02	Accumulateur chargé avec 02
033D-	85 25	STA	\$25	POKE 37,2 (position verticale du curseur)
033F-	85 24	STA	\$24	POKE 36,2 (position horizontale du curseur)
0341-	85 22	STA	\$22	POKE 34,2 : protection du cadre supérieur
0343-	85 20	STA	\$20	POKE 32,2 : protection du cadre gauche
0345-	A5 06	LDA	\$06	On réprend la largeur de la fenêtre...
0347-	38	SEC		Retenue à 1 pour soustraire
0348-	E9 04	SBC	£\$04	Moins 04 dans l'accumulateur
034A-	85 21	STA	\$21	POKE 33,A (largeur utile de la fenêtre)
034C-	A5 07	LDA	\$07	Hauteur dans l'Accumulateur
034E-	85 23	STA	\$23	POKE 35,A (hauteur utile de la fenêtre)
0350-	20 22 FC	JSR	\$FC22	VTAB(\$FC22) déplace le curseur enCV(\$25)
0353-	60	RTS		Retour au Basic.

ÉCRAN STANDARD

0354-	A9 00	LDA	£\$00	0 dans l'Accumulateur pour :
0356-	85 20	STA	\$20	POKE 32,0 (marge gauche standard)
0358-	85 22	STA	\$22	POKE 34,0 (haut de la fenêtre standard)
035A-	A9 50	LDA	£\$50	\$50 (80) dans l'Accumulateur :
035C-	85 21	STA	\$21	POKE 33,80 (fenêtre standard 80 colonnes)
035E-	A9 18	LDA	£\$18	\$18 (24) dans l'Accumulateur :
0360-	85 23	STA	\$23	POKE 35,24 (bas de fenêtre standard)
0362-	60	RTS		RETOUR AU BASIC

Pour terminer : **BSAVE FEN, A\$300, L\$63**

DÉBUTANTS : Rappelons que seuls les codes (en couleur) doivent être tapés. Commencez par un CALL-151 (suivi de RETURN), puis tapez la première adresse, suivie des codes. Après chaque RETURN (ne faites pas de saisies trop longues), tapez la nouvelle adresse... puis les codes, et ainsi de suite.

EXEMPLE : 300 : A6 06 86 21 A6 07 86 23 CA RETURN
 309 : 86 08 20 58 FC A9 18 85 23 RETURN

PLUS LOIN AVEC

Ce sont des sous-programmes qui reviennent...

\$FC22 (VTAB) Cette routine déplace le curseur en CV (contenu de \$25).

\$FC58 (HOME) Vide l'écran, mais seulement à l'intérieur de la fenêtre délimitée par le contenu de \$20 - 21 - 22 - 23.

\$FDED (COUT) Affiche le contenu de A en passant par JMP CSWL (\$36-37), adresse de la routine de sortie de caractère

\$FDF6 (COUTZ) Affiche aussi le contenu de A (plus rapide que COUT).

```

363- 20 58 FC JSR $FC58
366- A9 D4 LDA £$D4
368- 20 F6 FD JSR $FDF6
36B- AE E2 07 LDX $07E2
36E- E0 A0 CPX £$A0
370- F0 F6 BEQ $0368
372- A2 50 LDX £$50
374- 86 21 STX $21
376- 20 F6 FD JSR $FDF6
379- 20 F6 FD JSR $FDF6
37C- 60 RTS
    
```

Et d'abord le HOME classique
\$D4, c'est un T (celui de Tremplin Micro !)
COUTZ se fait un plaisir d'afficher ce T
Le contenu de 2018 est-il un espace ?
Si oui... ce n'est évidemment pas un T...
... et on continue



POKE 33,80... si vous préférez

Affichage des deux derniers T

RETOUR AU BASIC

Sur APPLE IIe-6502,
remplacer F6 par ED
dans les mémoires
\$369, \$377 et \$37A.

FENMEMO

(CALL 8192)

La fiche du numéro 2 de Tremplin Micro vous fournira tous éclaircissements sur cette routine.

Contrôle (en mode direct)

```
T = 0 : FOR I = 8192 TO 8290 : T = T + PEEK(I)
: NEXT : PRINT T
```

Résultat : 9522

Sauvegarde : BSAVE FENMEMO, A\$2000, L\$63

```

2000: 38 8D 55 C0 20 1C 20 A2
2008: 44 86 43 20 11 C3 8D 54
2010: C0 20 1C 20 A2 40 86 43
2018: 20 11 C3 60 A9 00 85 3C
2020: A2 04 86 3D A2 D0 86 3E
2028: A2 07 86 3F 85 42 60 18
2030: 8D 55 C0 20 54 20 A2 44
2038: 86 3D A2 47 86 3F 20 11
2040: C3 18 8D 54 C0 20 54 20
2048: A2 40 86 3D A2 43 86 3F
2050: 20 11 C3 60 A9 00 85 42
2058: 85 3C A2 04 86 43 A2 D0
2060: 86 3E 60
    
```

A propos de ...

CLEAR CLEAR affecte la valeur 0 à toutes les variables numériques du programme et une valeur nulle aux chaînes de caractères. Par contre, l'exécution du programme continue. HIMEM et LOMEM ne sont pas modifiés.

Le micro-programme ci-contre, montre bien ce qui se passe :

DEL Quand on utilise DEL dans le corps d'un programme, les lignes indiquées sont effectivement effacées, mais le programme s'arrête. Il est inutile d'essayer de le poursuivre en employant CONT.

Moralité : DEL doit être utilisé avec prudence... surtout pendant la mise au point d'un programme non mémorisé. NESTOR.

Nos lecteurs débutants se posent de nombreuses questions... élémentaires pour celles et ceux qui utilisent l'Applesoft depuis longtemps.

```

10 TEXT: HOME: LOMEM: 10000: HIMEM:
   30000
20 A = 10: PRINT A: B$ = "EXISTE": PRINT B$
30 GOSUB 80
40 CLEAR
50 PRINT A: PRINT B$
60 GOSUB 80
70 PRINT "TOTO": END
80 PRINT PEEK(105) + PEEK(106) * 256: REM
   LOMEM
90 PRINT PEEK(115) + PEEK(116) * 256:
   RETURN: REM HIMEM
    
```


Mots en spirale

une idée à développer



OBJET : Saisie d'un texte (81 caractères au maximum), puis affichage et impression éventuelle de ce texte... mais sous forme de spirale. Ce programme constitue une base intéressante, à exploiter

dans le cadre d'un jeu plus élaboré, faisant par exemple appel au graphisme... à la souris, etc. Gageons qu'il se trouvera, parmi les lectrices et lecteurs de *Tremplin Micro*, quelque esprit imaginatif, capable d'aller plus loin avec les mots en spirale ! A vos claviers !

100 : Affichage du titre. On emploie POKE 32,0 — POKE 33,40 et POKE 34,3 pour rétablir la fenêtre d'écran normal (TEXT aurait le même effet).

105 : Notez que le CLEAR mettra toutes les variables numériques à 0 et les chaînes à "".

Pendant la saisie du texte, le nombre de caractères utiles (ponctuation et espaces exclus) est affiché en permanence. La touche DELETE autorise l'effacement. On ne doit utiliser que des lettres CAPITALES.

```

100 TEXT : PRINT CHR$(21): HOME :V$ = "          ": P
    OKE 32,14: POKE 33,26: PRINT : VTAB 1: INVERSE : PRI
    NT V$: PRINT "  SPIRALES  ": PRINT V$: NORMAL : POKE
      32,0: POKE 33,40: POKE 34,3
105 CLEAR : FOR I = 1 TO 10:V$ = V$ + "    ": NEXT :D$ =
    CHR$(4)
110 HTAB 1: VTAB 5: CALL - 958: PRINT "Tapez une phrase
    comptant 81 caractères maximum. Il ne sera pas tenu
    compte des espaces et de la ponctuation éventuelle":
    POKE 34,8: PRINT
115 :
120 REM *****
125 REM *      SAISIE LETTRES      *
130 REM *****
135 :
140 T$ = "":V% = 11:L% = 0:H% = 0: HOME :B$ = "": VTAB V
    %: HTAB 1: GOTO 180
145 GOTO 165
150 R% = LEN (T$) - 1: IF NOT R% THEN CALL - 958: GOTO 1
    75
155 C$ = RIGHT$ (T$,1): IF C$ > CHR$(64) AND C$ < CHR$(
    91) THEN L% = L% - 1
160 T$ = LEFT$ (T$,R%):C$ = RIGHT$ (T$,1)
165 VTAB 9: HTAB 1: IF L% > 1 THEN PRINT L%" CARACTERES
    UTILES "
170 IF L% = 81 THEN 215
175 VTAB V%: HTAB H%: CALL - 958: PRINT C$:
180 GET C$:C% = ASC (C$): PRINT
185 IF C% > 64 AND C% < 91 THEN L% = L% + 1: GOTO 210
190 IF C% < > 127 THEN 205
195 IF H% > 1 THEN H% = H% - 1: GOTO 150
200 GOTO 140
205 IF C% = 13 THEN 220
210 T$ = T$ + C$:H% = H% + 1: GOTO 165
215 VTAB V%: HTAB H%: PRINT C$
  
```

Mots en spirale (suite)

```

220 IF NOT L% THEN 465
225 FOR I = 1 TO LEN (T$):C$ = MID$ (T$,I,1): IF C$ > CH
R$ (64) AND C$ < CHR$ (91) THEN B$ = B$ + C$
230 NEXT
235 PRINT : PRINT B$
240 PRINT : HTAB 8: VTAB 23: PRINT "<1> D'ACCORD <2> A R
EFAIRE ";; CALL - 198: GET R$
245 IF R$ = "1" THEN 260
250 IF R$ = "2" THEN 140
255 GOTO 240
260 PRINT : HTAB 8: VTAB 23: PRINT "EDITION SUR IMPRIMAN
TE O/N ";; CALL - 198: POKE - 16368,0: GET R$
265 IF R$ = "0" OR R$ = "o" THEN IM = 1
270 :
275 REM *****
280 REM *      CALCULS SPIRALES      *
285 REM *****
290 :
295 CI = SQR (L%):C% = INT (CI): IF C% * C% < > L% THEN
C% = C% + 1
300 POKE 34,4: HOME : VTAB 24: HTAB 3: INVERSE : PRINT L
%" LETTRES - DANS UN CARRE DE "C%" X "C%: NORMAL : P
OKE 35,22: VTAB 6
305 DIM M(C%,C%):Y% = INT (C% / 2 + .5):X% = Y%:A% = 1:B
% = 0:D% = C% * C%:R% = 0
310 FOR I = 1 TO C%:R% = ABS (R% - 1): IF NOT R% THEN 32
0
315 V% = 2
320 FOR J = 1 TO 2:Z% = B%:V% = V% - 1:B% = V%: IF B% =
- 2 THEN B% = 0
325 FOR K = 1 TO I:M(Y%,X%) = A%
330 IF A% = D% THEN 340
335 A% = A% + 1:Y% = Y% + B%:X% = X% + Z%: NEXT : NEXT :
NEXT
340 IF A% = L% THEN 395
345 VTAB 18: PRINT " <1> BLANC AU DEBUT <2> BLANC A LA F
IN ";; CALL - 198: GET R$: PRINT "":R% = VAL (R$): H
OME
350 R$ = LEFT$ (V$,D% - L%): ON R% GOTO 360,365
355 GOTO 345
360 B$ = R$ + B$: GOTO 395
365 B$ = B$ + R$
370 :
375 REM *****
380 REM *      EDITION CHIFFRES      *
385 REM *****
390 :
395 POKE 35,24: HOME :R% = 18 - (C% * 2): POKE 32,R%: PO
KE 33,40 - R%: PRINT : HOME
400 IF IM THEN PRINT : PRINT D$"PR#1"

```

235 : La variable B\$ contient le texte saisi, mais débarrassé des caractères parasites.

240 : Il est possible de taper un autre texte.

260 : La réponse "Oui" est testée en capitale et en minuscule.

295 : On calcule le côté du carré. Un rectangle poserait quelques problèmes supplémentaires !

305 : Ne pas oublier de dimensionner le tableau.

345 : Quand le nombre de lettres est plus petit que le nombre de cases disponibles, il faut savoir où mettre les cases vides.

395 : On affiche d'abord la grille sous forme de nombres.

EXEMPLE OBTENU SUR IMPRIMANTE IMAGE-WRITER, (le trait a été tracé à la main).

S O V T N E I C E
 Q I S S U O V E R
 U V S I M A S D P
 A O C N A M O T P
 L U E D N E V N A
 I S Q E E D A E S
 T S U Z P A S S L
 E U I L S P E N I
 S P P O S E Z O U

73	72	71	70	69	68	67	66	65
74	43	42	41	40	39	38	37	64
75	44	21	20	19	18	17	36	63
76	45	22	7	6	5	16	35	62
77	46	23	8	1	4	15	34	61
78	47	24	9	2	3	14	33	60
79	48	25	10	11	12	13	32	59
80	49	26	27	28	29	30	31	58
81	50	51	52	53	54	55	56	57

```

405 FOR I = 1 TO C%: FOR J = 1 TO C%: POKE 36,J * 4: PR
INT M(I,J);: NEXT J: PRINT : PRINT : NEXT I
410 POKE 32,0: POKE 33,40: IF IM = 1 THEN PRINT : GOTO 4
45
415 POKE 35,24: VTAB 23: CALL - 958: HTAB 11: PRINT "PRE
SSEZ UNE TOUCHE ";: CALL - 198: GET R$: PRINT
420 :
425 REM *****
430 REM *      EDITION LETTRES      *
435 REM *****
440 :
445 R% = 18 - (C% * 1.5): POKE 32,R%: POKE 33,40 - R%: P
RINT : HOME
450 FOR I = 1 TO C%: FOR J = 1 TO C%: POKE 36,(J * 3): P
RINT MID$(B$,M(I,J),1);: NEXT J: PRINT : PRINT : NE
XT I
455 IF IM = 1 THEN VTAB 5: CALL - 958: PRINT : PRINT D$"
PR&0"
460 POKE 32,0: POKE 33,40
465 HTAB 4: VTAB 23: PRINT " <E>NCORE <B>ASIC <R>ETOUR
MENU ";: CALL - 198: POKE - 16368,0: GET R$
470 IF R$ = "E" OR R$ = "e" THEN 105
475 IF R$ = "B" OR R$ = "b" THEN TEXT : HOME : END
480 IF R$ = "R" OR R$ = "r" THEN PRINT D$"RUN MENU"
485 GOTO 465
  
```

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 cela 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.

UTILISEZ LE BON DE COMMANDE, À LA FIN DU JOURNAL.

Votre bibliothèque

INFORMATIQUE

Rubrique indépendante
de Clément RENARD

- **Comment tenir sa comptabilité avec un micro-ordinateur**
(Editions de L'USINE NOUVELLE)

Cet ouvrage de Ronald HEUDE (ingénieur civil des Mines et conseil en organisation informatique) s'adresse aux responsables des petites et moyennes entreprises, candidats à une informatisation de leur comptabilité... ou victimes d'une première expérience malheureuse.

Les micro-ordinateurs sont indiscutablement capables, aujourd'hui, de tenir intégralement la comptabilité d'une petite ou d'une moyenne entreprise, mais il faut savoir choisir le bon modèle et ne se laisser influencer ni par la publicité des fabricants ni par celle des fournisseurs de logiciels.

L'objet de l'ouvrage est donc d'indiquer la méthode à suivre pour que l'implantation du micro et celle du logiciel soient parfaitement réussies. Ce n'est pas un hasard si le livre de Ronald HEUDE commence par un glossaire expliquant simplement les mots spécialisés qui sont utilisés plus loin.

On va successivement s'intéresser, et dans un langage d'une grande clarté, à l'Organisation des tâches et des traitements, à la Gestion des comptes clients, à la Comptabilité fournisseurs, à la Trésorerie Immobilisation, au Bilan et au Compte de résultat, à la Comptabilité des stocks, à la

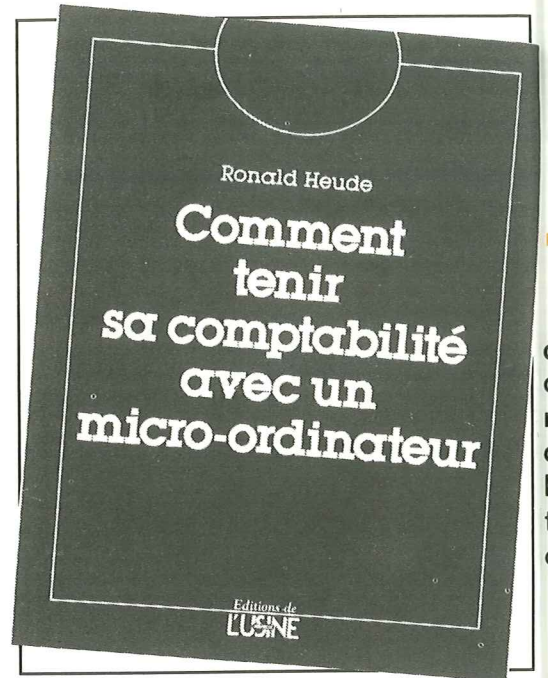
Comptabilité analytique, au Matériel et au logiciel, et enfin aux Ecritures courantes.

On aura ainsi fait le tour complet de la question et on sera en mesure de choisir machine et logiciel avec de bonnes chances de succès. Toutefois, il paraît évident que, malgré toutes les précautions prises, il faudra que les dirigeants acceptent aussi de changer — et sans regret — l'organisation administrative de l'entreprise...

Un livre utile, à lire pendant un long week-end, puis à discuter entre personnes responsables.

- **Les Fichiers en Basic sur Apple**
(EYROLLES)

Je vous avais invité, dans le dernier numéro de *Tremplin Micro* (page 56), à consulter le nouveau livre de Claude Delannoy, un spécialiste de la gestion des fichiers (et pas seulement sur Apple !), mais j'avais simplement parcouru "Les Fichiers en Basic sur Apple". Depuis, j'ai pris le temps de le lire et je n'en suis que plus à l'aise pour conseiller une consultation approfondie de cet ouvrage à tous les néophytes (nombreux, surtout dans ce domaine, si j'en juge par le nombre de lettres que reçoit la revue... sur cette question). Tout y est parfaitement expliqué et des exemples simples devraient permettre à tous les débutants de comprendre rapidement comment écrire et lire les données dans un fichier. Ils trouveront par ailleurs, dans la dernière partie du livre, de bonnes notions sur le tri en Basic.



Une nouvelle incursion dans la mémoire de l'APPLE

TREMPLEIN MICRO a déjà publié, plusieurs utilitaires permettant d'afficher le contenu des mémoires de l'Apple. La présentation de celui-ci est un peu différente, mais la philosophie reste la même. Il permet de modifier le contenu d'une case-mémoire, mais cela ne présente qu'un intérêt relatif. Par contre, l'utilisation de la fonction *USR* est à étudier.

L'affichage des valeurs décimales et hexadécimales est correct, mais (si l'on excepte le filtrage du CTRL-D... qui provoquerait un message d'erreur), celui des caractères ASCII est sujet à caution.

205 : On notera que l'utilisation de la touche ESCAPE permet de sauter au mini-menu final.

215 : Pour obtenir la suite, on peut indifféremment taper un S... ou un simple RETURN.

```

100 TEXT : NORMAL : PRINT CHR$ (21): HOME : GOSUB 415
105 POKE 50,63: PRINT "ADR +0 +1 +2 +3 +4 +5 +6
      +7 ADR": POKE 50,255: POKE 34,2: GOTO 295
110 :
115 REM *****
120 REM ** AFFICHAGE DES MEMOIRES **
125 REM *****
130 :
135 HOME
140 PRINT : INVERSE : PRINT " ";: NORMAL : PRINT " ADRES
      SE = ";D;" = $";: POKE 11,0:E = D: IF D > 32767 THEN
      E = - (65536 - D)
145 IF E = - 32768 THEN PRINT "8000": GOTO 155
150 X = USR (E): PRINT : PRINT
155 PRINT "DEC:":; POKE 11,11
160 FOR I = 0 TO 7:V%(I) = PEEK (D + I): PRINT TAB( 6 +
      I * 4);V%(I);: NEXT
165 PRINT TAB( 38)"DECHEX: ";
170 FOR I = 0 TO 7:X = USR (V%(I)): PRINT " ";: NEXT
175 PRINT TAB( 38)"HEXASC: ";
180 FOR I = 0 TO 7: IF V%(I) = 4 OR V%(I) = 134 THEN 190
185 PRINT TAB( 6 + I * 4): CHR$ (V%(I));
190 NEXT I: PRINT TAB( 38);"ASC"
195 N = N + 1: IF N < 3 THEN D = D + 8: GOTO 140
200 PRINT
205 VTAB 22: PRINT : CALL - 958: PRINT "AUTRE ADRESSE (A
      ) SUITE (S) MODIF (M) ";: GET R$: PRINT : IF R$ = CH
      R$ (27) THEN 370
210 IF R$ = "A" THEN 295
215 IF R$ = "S" OR R$ = CHR$ (13) THEN D = D + 8:N = 0:
      PRINT : GOTO 140
220 IF R$ < > "M" THEN 205
225 :
230 REM *****
235 REM **      MODIFICATION      **
240 REM *****

```



```

245 :
250 VTAB 22: PRINT : CALL - 958: INPUT "ADRESSE DECIMALE
    ? ";R$:AD = VAL (R$): IF AD < (D - 16) OR AD > D +
    7 THEN 205
255 VTAB 22: PRINT : CALL - 958: PRINT "NOUVELLE VALEUR
    ? "; PEEK (AD):: HTAB 19: INPUT " ";V$: IF V$ = "" TH
    EN 205
260 V = VAL (V$): IF V < 0 OR V > 255 THEN 255
265 POKE AD,V:D = D - 16:N = 0: GOTO 140
270 :
275 REM *****
280 REM ** ADRESSE DE REFERENCE **
285 REM *****
290 :
295 VTAB 20: CALL - 958: PRINT : PRINT "ADRESSE ";; INVE
    RSE : PRINT "D";: NORMAL : PRINT "ECIMALE OU ";; INV
    ERSE : PRINT "H";: NORMAL : PRINT "EXADECIMALE ? ";;
    GET R$:N = 0: PRINT : VTAB 12: IF R$ = "H" THEN 305
300 VTAB 23: INPUT "ADRESSE DECIMALE DE DEPART ? ";R$:D
    = VAL (R$): GOTO 135
305 VTAB 23: INPUT "ADRESSE HEXADECIMALE DE DEPART ? $";
    A$: IF A$ = "" THEN A$ = "0"
310 X = 0:D = 0:L = LEN (A$)
315 IF L > 4 THEN PRINT : PRINT A$;" C'EST TROP LONG!":
    GOTO 310
320 FOR I = L TO 1 STEP - 1:H$ = MID$ (A$,L - I + 1,1)
325 IF ASC (H$) > 64 AND ASC (H$) < 71 THEN X = ASC (H$)
    - 55: GOTO 335
330 X = VAL (H$)
335 D = D + X * (16 ^ (I - 1)): NEXT
340 GOTO 135
345 :
350 REM *****
355 REM ** VERS MENU DE DISQUETTE **
360 REM *****
365 :
370 PRINT : VTAB 21: PRINT : CALL - 958: PRINT "<1> ENCO
    RE <2> MENU DISQUETTE <3> FIN ";; CALL - 198: GET R$
    : PRINT : IF R$ < "1" OR R$ > "3" THEN 370
375 IF R$ = "1" THEN RUN
380 TEXT : HOME : IF R$ = "2" THEN PRINT CHR$ (4)"RUN ME
    NU"
385 END
390 :
395 REM *****
400 REM ** ROUTINE USR **
405 REM *****
410 :
415 FOR I = 768 TO 796: READ R: POKE I,R: NEXT
420 POKE 10,76: POKE 12,3: RETURN
425 DATA 32,12,225,166,161,165,160,32,65,249,96,32,12,22
    5,165,161,32,218,253,96,200,196,96,144,244,32,251,21
    8,96

```

250 : Si l'adresse décimale demandée ne figure pas dans la page affichée, elle est refusée.

255 : Si V\$ = "" (cas de RETURN), retour à la case-départ... ligne 105 !

265 : La page est réaffichée, pour montrer que la modification a été effectuée.

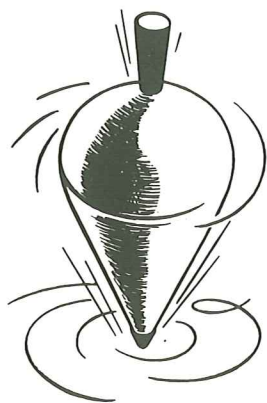
295 : La réponse est "DÉCIMALE" par défaut.

CONVERSION HEXA-DEC

Etudiez le contenu des lignes 320 à 335. Nous avons déjà publié plusieurs routines de conversion.

370 : On suppose que la disquette, comporte un programme intitulé MENU... capable d'afficher le Catalogue de la disquette et permettant de lancer telle ou telle routine.

425 : Quand le programme aura tourné au moins une fois, tapez un CALL-151 (RETURN), puis 300L... pour consulter le désassemblage de cette courte routine. Pour en savoir plus sur la fonction USR, se reporter au numéro 5 de T.M.



ROTATION

Comment faire tourner une partie d'image en HAUTE RÉOLUTION ?

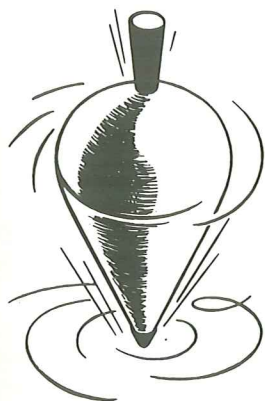
A routine de Bernard GOURC travaille sur la Page 1 (HGR), mais il faut savoir qu'elle détruit la Page 2 (HGR2). Les dimensions de la portion à faire tourner sont poquées aux adresses 24589 (largeur), 24577 (colonne gauche), 24583 (ligne haute). Un CALL 24576 provoque la rotation... qui peut se révéler plus ou moins longue, suivant le format de l'image à traiter.

ROTATION-DEMO

Le programme de démonstration utilise un tracé élémentaire, mais vous pourrez le remplacer par une image de votre choix.

En ce cas, remplacez seulement le contenu de la ligne 170 par :

```
PRINT D$"BLOOD VOTRE
IMAGE, A$2000": RETURN
```



```
100 TEXT : PRINT CHR$ (21): HOME : D$ = CHR$ (4):G$ = CHR
    $ (7): GOSUB 360: REM DESSIN
110 PRINT D$"BLOOD ROTATION": REM ROUTINE L.M.
130 :
140 REM *****
150 REM *   ROTATIONS PARAMETREES   *
160 REM *****
170 :
175 C = 0: POKE - 16302,0
180 FOR F = 14 TO 10 STEP - 2:C = C + 8:L = 50: PRINT G$
    ;
190 FOR T = 1 TO 4: REM 4 ROTATIONS
200 POKE 24589,F: REM LARGEUR
210 REM C=COLONE GAUCHE (0 A 39) L=PREMIERE LIGNE (0 A 1
    90)
220 POKE 24577,C: POKE 24583,L
230 CALL 24576: REM UNE ROTATION
240 NEXT T,F
250 POKE - 16301,0
260 PRINT G$:: HTAB 1: VTAB 22: PRINT "<E>ENCORE <B>ASIC
    <M>ENU DISQUETTE ";; GET X$
270 IF X$ = "E" OR X$ = "e" THEN 120
280 IF X$ = "B" OR X$ = "b" THEN TEXT : HOME : END
290 IF X$ = "M" OR X$ = "m" THEN PRINT : PRINT D$"RUNMEN
    U"
300 GOTO 260
310 :
```


6058:	D0 EE	76	BNE	AUTRE	60BD:	05 1F	130	ORA	COEF
605A:	A5 D6	77	LDA	LM	60BF:	91 26	131	STA	(\$26),Y
605C:	85 CF	78	STA	LC	60C1:	C6 19	132	DEC	MEML
605E:	E6 06	79	INC	C	60C3:	A5 07	133	LDA	L
6060:	C6 CE	80	DEC	E	60C5:	AA	134	TAX	
6062:	D0 E4	81	BNE	AUTRE	60C6:	CA	135	DEX	
6064:	18	82	CLC		60C7:	8A	136	TXA	
6065:	90 73	83	BCC	NNN	60C8:	C5 19	137	CMP	MEML
		84			60CA:	D0 DB	138	BNE	SUIV
		85	INVERSE CARAC C, L		60CC:	E6 09	139	INC	CPT
		86			60CE:	A5 1E	140	LDA	R
6067:	A9 00	87	INVERSE	LDA £\$00	60D0:	18	141	CLC	
6069:	85 09	88		STA CPT	60D1:	65 1E	142	ADC	R
606B:	A5 07	89		LDA L	60D3:	85 1E	143	STA	R
606D:	85 08	90		STA LL	60D5:	C9 80	144	CMP	£\$80
606F:	A5 07	91	ENCORE	LDA L	60D7:	D0 C0	145	BNE	REVIENT
6071:	20 11 F4	92		JSR \$F411	60D9:	60	146	RTS	
6074:	A4 06	93		LDY C			147		
6076:	B1 26	94		LDA (\$26),Y			148		
6078:	48	95		PHA			149		
6079:	A9 00	96		LDA £\$00	60DA:	A5 FE	150	NNN	LDA CC
607B:	91 26	97		STA (\$26),Y	60DC:	85 06	151		STA C
607D:	68	98		PLA	60DE:	A5 FC	152		LDA EE
607E:	A4 09	99		LDY CPT	60E0:	85 CE	153		STA E
6080:	99 00 70	100		STA \$7000,Y	60E2:	A6 CE	154		LDX E
6083:	E6 07	101		INC L	60E4:	CA	155		DEX
6085:	E6 09	102		INC CPT	60E5:	86 1E	156		STX R
6087:	A5 09	103		LDA CPT	60E7:	A9 00	157		LDA £\$00
6089:	C9 07	104		CMP £\$07	60E9:	85 18	158		STA MEMZ
608B:	D0 E2	105		BNE ENCORE	60EB:	A9 00	159	AAA	LDA £\$00
608D:	A5 08	106		LDA LL	60ED:	85 07	160		STA L
608F:	85 07	107		STA L	60EF:	A5 1E	161		LDA R
6091:	A9 00	108		LDA £\$00	60F1:	38	162		SEC
6093:	85 09	109		STA CPT	60F2:	E5 18	163		SBC MEMZ
6095:	A9 01	110		LDA £\$01	60F4:	85 19	164		STA MEML
6097:	85 1E	111		STA R	60F6:	A5 07	165	BBB	LDA L
6099:	A4 09	112	REVIENT	LDY CPT	60F8:	18	166		CLC
609B:	B9 00 70	113		LDA \$7000,Y	60F9:	65 06	167		ADC C
609E:	85 18	114		STA MEMZ	60FB:	85 D7	168		STA F
60A0:	A5 08	115		LDA LL	60FD:	A9 07	169		LDA £\$07
60A2:	18	116		CLC	60FF:	85 09	170		STA CPT
60A3:	69 06	117		ADC £\$06	6101:	A9 00	171		LDA £\$00
60A5:	85 19	118		STA MEML	6103:	18	172	CCC	CLC
60A7:	46 18	119	SUIV	LSR MEMZ	6104:	65 19	173		ADC MEML
60A9:	90 05	120		BCC A1	6106:	C6 09	174		DEC CPT
60AB:	A5 1E	121		LDA R	6108:	D0 F9	175		BNE CCC
60AD:	18	122		CLC	610A:	18	176		CLC
60AE:	90 02	123		BCC SUITE	610B:	65 D6	177		ADC LM
60B0:	A9 00	124	A1	LDA £\$00	610D:	85 08	178		STA LL
60B2:	85 1F	125	SUITE	STA COEF	610F:	A9 07	179		LDA £\$07
60B4:	A5 19	126		LDA MEML	6111:	85 09	180		STA CPT
60B6:	20 11 F4	127		JSR \$F411	6113:	A9 00	181		LDA £\$00
60B9:	A4 06	128		LDY C	6115:	18	182	DDD	CLC
60BB:	B1 26	129		LDA (\$26),Y	6116:	65 07	183		ADC L

ROTATION (suite)

6000: A9 09 85 06 85 FE A9 73
 6008: 85 CF 85 D6 A9 0F 85 CE
 6010: 85 FC A9 00 85 18 85 1E
 6018: A9 20 85 19 A9 40 85 1F
 6020: A2 20 A0 00 B1 18 91 1E
 6028: 88 D0 F9 E6 19 E6 1F CA
 6030: D0 F0 A9 40 85 E6 A9 06
 6038: 85 09 A5 CE 18 65 CE C6
 6040: 09 D0 F9 18 65 CF 85 D7
 6048: A5 CF 85 07 20 67 60 A5
 6050: CF 18 69 07 85 CF C5 D7
 6058: D0 EE A5 D6 85 CF E6 06
 6060: C6 CE D0 E4 18 90 73 A9
 6068: 00 85 09 A5 07 85 08 A5
 6070: 07 20 11 F4 A4 06 B1 26
 6078: 48 A9 00 91 26 68 A4 09
 6080: 99 00 70 E6 07 E6 09 A5
 6088: 09 C9 07 D0 E2 A5 08 85
 6090: 07 A9 00 85 09 A9 01 85
 6098: 1E A4 09 B9 00 70 85 18
 60A0: A5 08 18 69 06 85 19 46
 60A8: 18 90 05 A5 1E 18 90 02
 60B0: A9 00 85 1F A5 19 20 11
 60B8: F4 A4 06 B1 26 05 1F 91
 60C0: 26 C6 19 A5 07 AA CA 8A
 60C8: C5 19 D0 DB E6 09 A5 1E
 60D0: 18 65 1E 85 1E C9 80 D0
 60D8: C0 60 A5 FE 85 06 A5 FC
 60E0: 85 CE A6 CE CA 86 1E A9
 60E8: 00 85 18 A9 00 85 07 A5
 60F0: 1E 38 E5 18 85 19 A5 07
 60F8: 18 65 06 85 D7 A9 07 85
 6100: 09 A9 00 18 65 19 C6 09
 6108: D0 F9 18 65 D6 85 08 A9
 6110: 07 85 09 A9 00 18 65 07
 6118: C6 09 D0 F9 18 65 D6 85
 6120: FA A5 06 18 65 18 85 FB
 6128: A9 00 85 09 A9 40 85 E6
 6130: A5 FA 20 11 F4 A4 FB B1
 6138: 26 85 1F A9 20 85 E6 A5
 6140: 08 20 11 F4 A4 D7 A5 1F
 6148: 91 26 E6 FA E6 08 E6 09
 6150: A5 09 C9 07 D0 D6 E6 07
 6158: A5 CE C5 07 D0 98 E6 18
 6160: A5 CE C5 18 D0 85 60

Si vous ne possédez pas d'assembleur, contentez-vous de rentrer les 359 octets ci-dessus... après l'indispensable CALL-151.

VERIFICATION : T = 0 : FOR I = 24576 TO 24934 : T = T + PEEK(I) : NEXT : PRINT T ... RETURN

Le résultat doit être : 41647

6118: C6 09 184
 611A: D0 F9 185
 611C: 18 186
 611D: 65 D6 187
 611F: 85 FA 188
 6121: A5 06 189
 6123: 18 190
 6124: 65 18 191
 6126: 85 FB 192
 6128: A9 00 193
 612A: 85 09 194
 612C: A9 40 195
 612E: 85 E6 196
 6130: A5 FA 197
 6132: 20 11 F4 198
 6135: A4 FB 199
 6137: B1 26 200
 6139: 85 1F 201
 613B: A9 20 202
 613D: 85 E6 203
 613F: A5 08 204
 6141: 20 11 F4 205
 6144: A4 D7 206
 6146: A5 1F 207
 6148: 91 26 208
 614A: E6 FA 209
 614C: E6 08 210
 614E: E6 09 211
 6150: A5 09 212
 6152: C9 07 213
 6154: D0 D6 214
 6156: E6 07 215
 6158: A5 CE 216
 615A: C5 07 217
 615C: D0 98 218
 615E: E6 18 219
 6160: A5 CE 220
 6162: C5 18 221
 6164: D0 85 222
 6166: 60 223

RRR

DEC CPT
 BNE DDD
 CLC
 ADC LM
 STA LH
 LDA C
 CLC
 ADC MEMZ
 STA CH
 LDA £\$00
 STA CPT
 LDA £\$40
 STA \$E6
 LDA LH
 JSR \$F411
 LDY CH
 LDA (\$26),Y
 STA COEF
 LDA £\$20
 STA \$E6
 LDA LL
 JSR \$F411
 LDY F
 LDA COEF
 STA (\$26),Y
 INC LH
 INC LL
 INC CPT
 LDA CPT
 CMP £\$07
 BNE RRR
 INC L
 LDA E
 CMP L
 BNE BBB
 INC MEMZ
 LDA E
 CMP MEMZ
 BNE AAA
 RTS

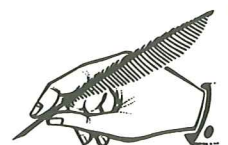
MÉMORISATION :

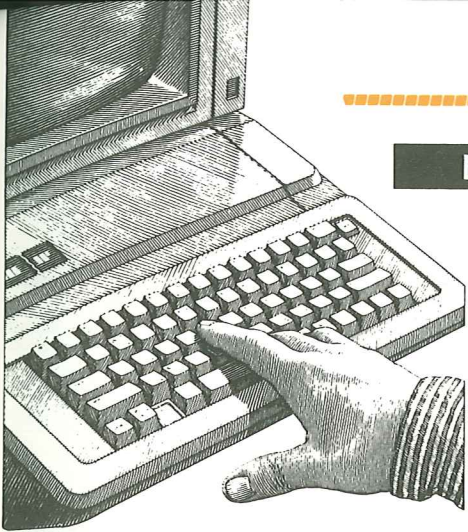
BSAVE ROTATION, A\$6000, L\$167

REMARQUE :

Pour travailler sur une image de la Page 2 (HGR2)... tout en détruisant la Page 1 (HGR), il suffit de modifier ainsi les lignes suivantes :

38 LDA £\$40
 40 LDA £\$20
 52 LDA £\$20
 197 LDA £\$20
 204 LDA £\$40





CLAVIER SECRET

Votre APPLE attend un code

UTILISATION :

Ce programme est à la fois une démonstration... et un petit utilitaire.

En effet, quand vous aurez établi votre *code secret* (qui sera obtenu par l'association de deux touches), il vous suffira de mémoriser la routine LM par un :

BSAVE CODE, A\$300, L\$1B

puis de l'utiliser, telle quelle, à l'entrée de n'importe quel programme codé (elle est relogeable et vous pourrez donc l'installer à n'importe quelle adresse).

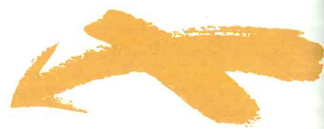
N'oubliez pas qu'il faut l'appeler par *CALL 770...* ou, dans le cas où elle est logée à une autre adresse que \$300 (768), par *CALL ADRESSE+2*

Amusez-vous bien !

```
100 TEXT : PRINT CHR$ (21): HOME : GOSUB 195
105 PRINT "CLAVIER SECRET": PRINT "-----"
110 POKE 34,3
115 PRINT "TAPEZ LES DEUX CARACTERES SECRETS      (CTRL
      SI VOUS LE DESIREZ)"
120 N = N + 1
125 VTAB 8 + N * 2: CALL - 198: GET A$: IF A$ < " " THEN
      GOSUB 185: GOTO 140
130 IF A$ = " " THEN GOSUB 190: GOTO 140
135 PRINT A$
140 V(N) = ASC (A$) + 128
145 IF N < > 2 THEN 120
150 POKE 768,V(2): POKE 769,V(1)
155 FOR I = 1 TO 2000: NEXT
160 HOME : VTAB 7
165 POKE 58,02: POKE 59,03: POKE 795,169: POKE 796,11: P
      OKE 797,32: POKE 798,99: POKE 799,254: POKE 800,96:
      CALL 795: PRINT
170 CALL 770
175 VTAB 6: PRINT "0300-   " ;: POKE 801,174: POKE 802,0:
      POKE 803,03: POKE 804,32: POKE 805,68: POKE 806,249
      : POKE 807,96: CALL 801: PRINT "   " ;: POKE 802,01: CA
      LL 801
180 GOTO 210
185 INVERSE : PRINT "CTRL-" ASC (A$): NORMAL : RETURN
190 PRINT "ESPACE": RETURN
195 POKE 768,0: POKE 769,0: POKE 770,162: POKE 771,2: PO
      KE 772,44: POKE 773,16: POKE 774,192: POKE 775,44: P
      OKE 776,0: POKE 777,192: POKE 778,16: POKE 779,251:
      POKE 780,173: POKE 781,0
200 POKE 782,192: POKE 783,221: POKE 784,255: POKE 785,2
      : POKE 786,208: POKE 787,240: POKE 788,202: POKE 789
      ,208: POKE 790,237: POKE 791,173: POKE 792,16: POKE
      793,192: POKE 794,96
205 RETURN
210 VTAB 21: PRINT : PRINT "<1> ENCORE <2> MENU DISQUETT
      E <3> FIN " ;: CALL - 198: GET R$: PRINT
215 IF R$ = "1" THEN RUN
220 TEXT : IF R$ = "2" THEN PRINT CHR$ (4)"RUN MENU"
225 IF R$ < > "3" THEN 210
230 HOME
```

La grande question

Que deviendra votre Apple ?



QUE deviendra votre machine actuelle quand les Apple IIe et IIc auront vécu leurs derniers jours (il semble, au moment où je rédige ces lignes, que l'événement soit imminent) ?

Tout dépend des raisons qui motivent une question dont en découle une autre, non moins importante : votre ordinateur personnel vous donne-t-il présentement satisfaction ? Autrement dit : qu'en attendez-vous de plus ?

Il est probable que vous n'avez pas encore épuisé toutes les ressources de votre Apple, tel qu'il est. Il est presque sûr que sa rapidité (inférieure à celle de bien des machines plus récentes) ne le pénalise pas, du moins à vos yeux. Alors, franchement, envisageriez-vous de l'abandonner au profit de son aîné ? Pour disposer du dernier modèle en date ? Pourquoi pas ? Pour vous familiariser le plus rapidement possible avec le 65C816 ? Bonne idée, mais rien ne presse.

Comptez donc les micros qui additionnent autant d'années de bons et loyaux services que le fameux Apple. Comptez-les sur les doigts d'une seule main... et ne vous trompez pas ! Quant à la compatibilité, si elle pose parfois quelques petits problèmes d'un modèle à l'autre — surtout depuis la naissance du IIc et du 65C02 —, elle reste acceptable sur Apple et tout porte à penser qu'on ne l'a pas oubliée quand on a conçu le Successeur.

Comme vous, j'attends ce fameux Successeur et je me vois déjà devant mon nouvel écran (au fait, disposerons-nous d'un écran couleur d'origine contrôlée ?), effleurant les touches du clavier numérique ou celles, évidemment programmables, qui transformeront enfin les logiciels en modèles de convivialité. Je m'imagine en train de chercher des câbles invisibles. Et surtout — tant pis si je fais sourire —, je programme déjà, en utilisant bien évidemment un Basic Plus plein de ressources, capable de reconnaître, sans les maltraiter, les vieilles

routines en Applesoft... et rendant inutiles la plupart des bidouilles qui permettent à des revues comme *Tremplin Micro...* de remplir leurs divers numéros.

Dans le monde entier, des millions de micros "domestiques" dorment dans des placards où ils ont rejoint d'autres gadgets plus ou moins coûteux. Combien d'Apple parmi eux ? Les statistiques ne le disent pas, mais nous savons tous que l'Ancêtre ne connaît que rarement ce misérable sort. Il suffit de consulter les prix pratiqués sur le marché de l'occasion pour en être totalement convaincu. Pourquoi ? Précisément parce que, dans un foyer (ou dans un bureau), un ordinateur personnel comme l'Apple parvient toujours à intéresser quelqu'un. On n'a certainement pas tout fait avec un Apple, mais on ne compte plus les applications tournant sur cette machine. Quant aux logiciels de jeux, n'en parlons pas : on ne cherche même plus à les recenser !

Mais parlons de vous... qui avez la chance de vous intéresser à la programmation et par conséquent d'appartenir à une catégorie particulière d'utilisateurs. D'autres jouent au Scrabble ou résolvent des mots croisés (vous savez que l'auteur de ces lignes édite une soixantaine de revues de jeux d'esprit). Vous préférez — et cela vous prend beaucoup de temps ! — vous adonner à la programmation. Une autre forme de récréation. Une gymnastique intellectuelle extraordinaire. La meilleure école pour habituer un cerveau à raisonner avec une logique... implacable ! Oui, parlons de vous... mais pour dire que, vraiment, dans l'état actuel des choses, la naissance d'un nouvel Apple ne vous pose certainement pas de problème particulier.

En fait, n'est-il pas vrai ? elle ne soulève qu'une seule question : le Successeur vous permettra-t-il oui ou non de donner libre cours à votre imagination et d'interroger, une à une, les précieuses adresses de son Cerveau ?

G.-H.

Copie d'écran

80 colonnes (IW)

CET utilitaire fonctionne sous ProDOS (et sous DOS 3.3 avec quelques modifications). Il permet de recopier, sur imprimante DMP ou IMAGEWRITER (nouveau modèle compris), les 24 lignes d'un écran de 80 colonnes (mode TEXT). Il fait appel à une courte routine en langage machine. On notera qu'il autorise une copie partielle de l'écran.

LD = Première ligne à éditer.

LF = Dernière ligne à éditer.

Naturellement, dans vos programmes, cette routine de remplissage disparaîtra... au profit de la vôtre !

210 : Le VTAB I est très important.

Sous DOS 3.3

Remplacez les lignes 240 et 250 par :

```
240 : PRINT D$"PR&0" :
      PRINT :
```

```
250 : PRINT D$"PR&3" :
      PRINT : HOME
```

Si vous rencontrez un problème de saut de ligne, modifiez la longueur de la ligne (voir la REMarque de la ligne 190) ou procédez éventuellement à un réglage des interrupteurs DMP.

```
100 TEXT : NORMAL : D$ = CHR$(4) : PRINT D$"PR&3" : PRINT
      : HOME
```

```
110 GOSUB 290 : REM ROUTINE LM 80 COLONNES
```

```
120 LD = 1 : LF = 24 : REM LIGNE DE DEBUT ET LIGNE FINALE
```

ÉCRITURE (TEST) SUR LES 24 LIGNES

```
130 A$ = "RAPIDE COPIE D'ECRAN (80 COLONNES)"
```

```
140 FOR I = 24 TO 1 STEP - 1 : VTAB I : POKE 1403, ABS ((I
      * 2) - 35) : PRINT A$ :
```

```
150 IF PEEK (50) = 255 THEN POKE 50,63 : GOTO 170
```

```
160 POKE 50,255
```

```
170 NEXT : PRINT
```

```
180 POKE 2039,198
```

COPIE D'ÉCRAN

```
190 PRINT D$"PR&1" : PRINT CHR$(9)"81N" : REM DMP/IW=81N :
      IW2=80N
```

```
200 FOR I = LD TO LF
```

```
210 VTAB I : CALL 768
```

```
220 PRINT : REM SUIVANT VOTRE IMPRIMANTE
```

```
230 NEXT : PRINT
```

```
240 PRINT D$"PR&A&C307" : REM RETOUR EN 80 COLONNES
```

```
250 CALL - 198 : POKE - 16368,0 : WAIT - 16384,128,127 : PO
      KE - 16368,0
```

```
260 VTAB 21 : PRINT : CALL - 958 : HTAB 14 : PRINT "<1> MEN
      U DE DISQUETTE <2> TERMINE " ; : GET A$
```

```
270 PRINT : HOME : IF A$ = "1" THEN PRINT CHR$(4)"RUN M
      ENU"
```

```
280 END
```

```
290 FOR I = 768 TO 810 : READ R : POKE I,R : NEXT : RETURN
```

```
300 DATA 160,0,141,85,192,177,40,141,84,192,32,30,3,32,2
      37,253,177,40,32,30,3,32,237,253,200,192,40,208,229,
      96,201,128,176,2,9,128,201,160,176,2,9,64,96
```

80 COL. IW/DMP

(LIGNE 300 PAGE PRÉCÉDENTE)

0300-	A0 00	LDY	£\$00	Ø dans le Registre Y
0302-	8D 55 C0	STA	\$C055	Page 2
0305-	B1 28	LDA	(\$28),Y	Lecture du caractère (Page 2)
0307-	8D 54 C0	STA	\$C054	On repasse en mémoire principale
030A-	20 1E 03	JSR	\$031E	Vers le traitement du caractère
030D-	20 ED FD	JSR	\$FDED	Ecriture
0310-	B1 28	LDA	(\$28),Y	Lecture du caractère (Page 1)
0312-	20 1E 03	JSR	\$031E	Vers le traitement
0315-	20 ED FD	JSR	\$FDED	Ecriture
0318-	C8	INY		Y = Y + 1
0319-	C0 28	CPY	£\$28	Y est-il égal à \$28 (40) ?
031B-	D0 E5	BNE	\$0302	S'il est différent, la boucle continue
031D-	60	RTS		Retour au Basic
031E-	C9 80	CMP	£\$80] Si le caractère est plus grand ou égal à \$80, saut
0320-	B0 02	BCS	\$0324	
0322-	09 80	ORA	£\$80] Si le caractère est plus grand ou égal à \$A0, saut (traitement terminé)
0324-	C9 A0	CMP	£\$A0	
0326-	B0 02	BCS	\$032A	Retour
0328-	09 40	ORA	£\$40	
032A-	60	RTS		

IMAGEWRITER 2 SOUS ProDOS

```

100 TEXT : NORMAL : D$ = CHR$ (4) : PRINT D$"PR£3" : PRINT
    : HOME
110 GOSUB 260 : REM ROUTINE LM 80 COLONNES
120 LD = 5 : LF = 20 : REM PREMIERE ET DERNIERE LIGNES A ED
    ITER
130 A$ = "RAPIDE COPIE D'ECRAN (80 COLONNES)"
140 FOR I = 24 TO 1 STEP - 1 : VTAB I : POKE 1403, ABS ((I
    * 2) - 35) : PRINT A$ :
150 IF PEEK (50) = 255 THEN POKE 50,63 : GOTO 170
160 POKE 50,255
170 NEXT : PRINT : POKE 2039,198
180 POKE 800,LF - 1 : REM DERNIERE LIGNE A EDITER
190 PRINT D$"PR£1" : PRINT CHR$ (9)"80N"
200 VTAB LD : CALL 768 : PRINT
210 PRINT D$"PR£A$C307" : REM RETOUR EN 80 COLONNES
220 CALL - 198 : POKE - 16368,0 : WAIT - 16384,128,127 : PO
    KE - 16368,0
230 VTAB 21 : PRINT : CALL - 958 : HTAB 14 : PRINT "<1> MEN
    U DE DISQUETTE <2> TERMINE " ; : GET A$
240 PRINT : HOME : IF A$ = "1" THEN PRINT CHR$ (4)"RUN M
    ENU"
250 END
260 FOR I = 768 TO 828 : READ R : POKE I,R : NEXT : RETURN
270 DATA 160,0,141,85,192,177,40,141,84,192,32,48,3,32,2
    37,253,177,40,32,48,3,32,237,253,200,192,40,208,229,
    165,37,201,23,176,12,169,141,32,237,253,230,37,32,34
    ,252,208,209,96,201,128,176,2,9,128,201,160,176,2,9,
    64,96
    
```

Programme
réservé aux
possesseurs
de la toute
nouvelle
IMAGEWRITER 2.

Il est réellement
très rapide
et fonctionne
sous ProDOS.

Si vous travaillez
avec le DOS 3.3,
modifiez les
lignes 210-220
comme indiqué
dans la routine
de la page
précédente.

Caractères souris

Sur DMP ou ImageWriter

CHOSE promise, chose due : voici les caractères souris (qui vous permettront notamment d'éditer, sur une imprimante DMP ou ImageWriter ancien modèle, les HISTO.TEXT de *Tremplin Micro* n°8). Pour être agréable à ceux qui n'aiment pas taper des codes en HEXA, j'ai tout transformé en DATAs, mais il n'est pas interdit de mémoriser autrement la *FONTE.SOURIS*, comme on le verra par la suite. Naturellement, si vous possédez une imprimante IMAGEWRITER 2 (veinard !) il est inutile de vous casser le poignet sur ce programme. L'IW 2 dispose, de naissance, des caractères souris. Les pinailleurs (que j'aime et respecte) me diront qu'il est inutile de passer par les adresses du DOS (ou de ProDOS) pour lire la longueur du fichier...

puisque, dans le cas présent, elle est connue. Je me suis contenté de reprendre ici, la routine CARAC.CHARG du numéro 6. Elle a le mérite de pouvoir servir à autre chose. Il suffit de créer d'autres lignes de DATAs.

Qu'on se le dise !

115 : Attention ! si vous utilisez ProDOS. Lisez la **REMARQUE** !

Si la *FONTE.SOURIS* était chargée par un **BLOAD SOURIS** (par exemple), au lieu d'être contenue dans des lignes de DATAs, la ligne 115 devrait être supprimée. En effet, **CARAC.CHARG** (adresse \$300) lit lui-même la longueur du fichier chargé.

```

100 TEXT : NORMAL : PRINT CHR$ (21): HOME
105 D$ = CHR$ (4):ES$ = CHR$ (27)
110 GOSUB 235
115 POKE 43616,37: POKE 43617,1: REM PRODOS48840 ET 4884
    1 (LONGUEUR FONTE)
120 :
125 REM *****
130 REM *      CHARGER ET IMPRIMER      *
135 REM *****
140 :
145 VTAB 12: PRINT "ALLUMEZ L'IMPRIMANTE IMAGE WRITER ";
    : GET R$: PRINT
150 PRINT D$"PR&1": PRINT ES$"N": PRINT CHR$ (9)"80N": P
    RINT CHR$ (15); CHR$ (9)"Z"
155 CALL 768: PRINT ES$"D 0";
160 PRINT D$: PRINT ES$"/";
165 FOR I = 64 TO 95: PRINT CHR$ (I);: IF I < > 70 THEN
    PRINT " ";
170 NEXT : PRINT
175 X = X + 1: IF X < 2 THEN PRINT CHR$ (14): GOTO 165
180 PRINT CHR$ (15);ES$" $": PRINT : PRINT D$"PR&0"
185 HOME : VTAB 12: PRINT "VOTRE FONTE ";: INVERSE : PRI
    NT F$;: NORMAL : PRINT " EST INSTALLEE"
190 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: PO
    KE - 16368,0: PRINT : HOME : GOTO 195
195 VTAB 12: PRINT "<1> MENU DISK <2> PROGRAMME <3> FIN
    ";: GET R$: PRINT R$
200 IF R$ = "1" THEN PRINT D$"RUN MENU"
205 IF R$ < > "2" THEN HOME : END
210 :
  
```

FONTE SOURIS

STRUCTURE

La ligne 245 contient les codes de commande de la fonte (43 pourrait être remplacé par 45 car aucun caractère ne dépasse 8 points).

Ensuite, chaque ligne contient 54 codes (9 par caractère), sauf la dernière (18 codes seulement).

Le premier code de chaque caractère indique sa valeur décimale (cela va de 64 à 95, c'est-à-dire du à au souligné).

VOIR À CE SUJET LE TABLEAU DE LA PAGE 43 DE T.M. n°8

Si vous désirez mémoriser la FONTE SOURIS sous forme de fichier binaire, lancez le programme, puis après impression, tapez un simple et efficace :

BSAVE SOURIS, **A**\$4000, **L**\$125
ou,
BSAVE SOURIS, **A**16384, **L**293

SOUS ProDOS

Gare à la REMarque de la ligne 325 !

```

215 REM *****
220 REM *   CARACTERES SOURIS   *
225 REM *****
230 :
235 FOR I = 16384 TO 16676: READ R: POKE I,R:T = T + R:
NEXT
240 IF T < > 26935 THEN CALL - 198: PRINT "ERREUR DANS L
A FONTE SOURIS": END
245 DATA 27,43,27,73
250 DATA 64,71,56,252,252,122,253,252,72,65,71,56,196,13
2,2,133,180,72,66,71,0,252,120,48,96,64,128,67,71,12
9,195,165,217,165,195,129,68,71,8,16,96,16,8,4,2,69,
71,247,231,159,239,247,251,253
255 DATA 70,71,88,196,68,84,125,127,23,71,103,20,20,52,9
8,130,128,128,72,71,8,28,42,73,8,8,8,73,103,0,128,0,
128,0,128,0,74,71,16,32,64,255,64,32,16,75,71,8,4,2,
255,2,4,8
260 DATA 76,71,1,1,1,1,1,1,1,77,71,32,112,248,32,32,32,6
3,78,71,255,255,255,255,255,255,255,79,71,129,137,15
6,190,127,136,136,80,71,136,136,127,190,156,137,129,
81,71,8,24,56,126,56,24,203
265 DATA 82,71,16,24,28,126,28,24,211,83,71,8,8,8,8,8,8,
8,84,71,255,128,128,128,128,128,128,85,71,8,8,8,73,4
2,28,8,86,71,170,85,170,85,170,85,170,87,71,85,170,8
5,170,85,170,85
270 DATA 88,71,252,130,130,130,130,130,132,89,71,132,132
,132,132,132,132,248,90,71,0,0,0,0,0,0,255,91,71,8,2
8,62,127,62,28,8,92,71,129,129,129,129,129,129,129,9
3,71,20,20,119,0,119,20,20
275 DATA 94,71,129,129,153,153,129,129,255,95,71,255,0,0
,0,0,0,0,4
280 :
285 REM *****
290 REM * CARAC.CHARG (TM NUMERO 6 *
295 REM *****
300 :
305 T = 0: FOR I = 768 TO 818: READ R: POKE I,R:T = T +
R: NEXT
310 IF T < > 5691 THEN CALL - 198: PRINT "ERREUR DANS RO
UTINE CARAC.CHARG": END
315 T = 0: RETURN
320 DATA 169,255,133,6,169,63,133,7,173,96,170,133,8,173
,97,170,133,9,230,6,165,6,208,2,230,7,198,8,165,8,20
1,255,208,6,165,9,240,12,198,9,160,0,177,6,32,237,25
3,76,18,3,96
325 REM SOUS PRODOS, 5691 devient 5939 (ligne 310): D'au
tre part (ligne 320) les 10e et 11e données devienne
nt: 200,190 & les 15e et 16e données deviennent 201,
190 au lieu de 96,170 & 97,170

```



2000
1800
1600
1400
1200
1000
800
600
400
200

Po
Su
Voici
tront

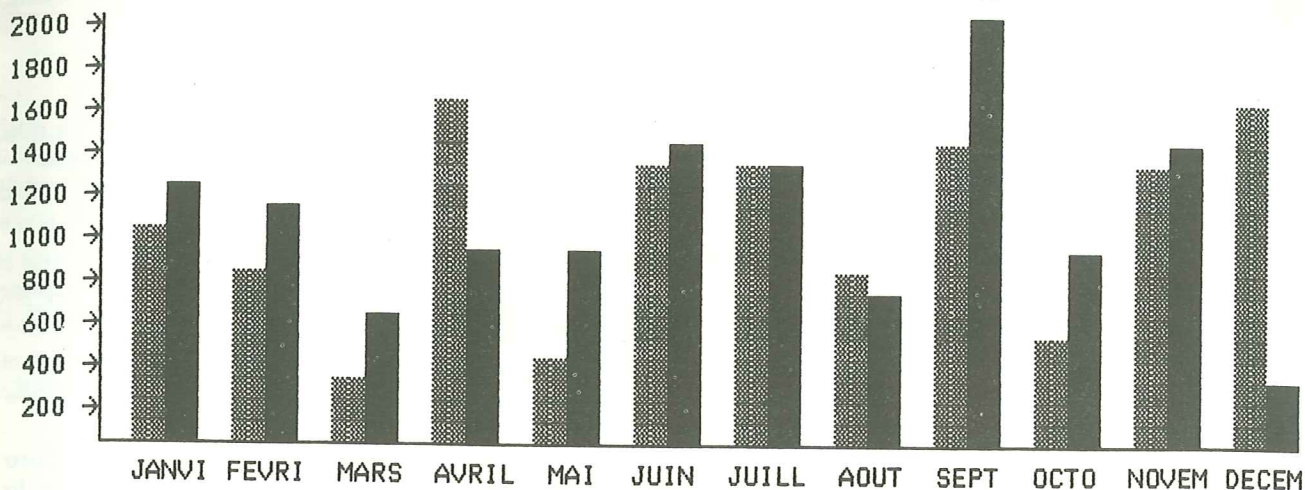
La liq
rema
longu
L'hist
page
l'imp
bonn

Vous
fallu
(103
aura
tats.

La fl
n'est
HIST
rem
c'est
(lign

TREA

RESULTATS COMPARES DES ANNEES 1985 ET 1986



Pour éditer HISTO.TEXT

Sur DMP ou IMAGEWRITER 1

UTILISATION :

- Créer un fichier EXEC avec la MODIF.
- Charger HISTO.TEXT (n°8)
- DEL 390,415
- EXEC le fichier MODIF
- SAVE HISTO.SSOURIS

Voici les quelques lignes de programme qui vous permettront de modifier HISTO.TEXT de TEMPLIN MICRO n°8.

HISTO.MODIF

```

386 IF NOT FL THEN GOSUB 1020: PRINT
390 FOR I = 1 TO 23: IF I = 1 THEN PRINT L$(I): PRINT :
      GOTO 415
391 IF I = 23 THEN PRINT "          JANVI FEVRI MARS AV
      RIL MAI JUIN JUILL AOUT SEPT OCTO NOVEM DECEM
      ": GOTO 415
395 PRINT LEFT$(L$(I),6);
400 PRINT CHR$(27)"N"; CHR$(27)"/"; MID$(L$(I),7, LEN
      (L$(I)) - 6); CHR$(27)"$"; CHR$(27)"E"
415 NEXT
1000 REM *****
1005 REM *      CARACTERES SOURIS      *
1010 REM *****
1015 :
1020 POKE 768,165: POKE 769,6: POKE 770,32: POKE 771,237:
      POKE 772,253: POKE 773,96
1030 FOR I = 1 TO 59: READ R: POKE 6,R: CALL 768: NEXT :F
      L = 1: RETURN
1035 DATA 27,43,27,73
1036 DATA 32,71,0,0,0,0,0,0,0
1040 DATA 76,71,1,1,1,1,1,1,1
1045 DATA 78,71,255,255,255,255,255,255,255
1050 DATA 45,71,8,8,8,73,42,28,8
1055 DATA 87,71,85,170,85,170,85,170,85
1060 DATA 95,71,255,0,0,0,0,0,4
    
```

La ligne des mois a été remaniée (problème de longueur).

L'histogramme de cette page vous montre que l'impression est plutôt bonne.

Vous noterez qu'il a fallu créer un ESPACE (1036) sans lequel on aurait obtenu des résultats... surprenants.

La flèche souris (85), qui n'est pas employée dans HISTO.TEXT normal, remplace CHR\$(45), c'est-à-dire le tiret (ligne 1050).

Ecrans condensés

Version avec fenêtres

LA série "Ecrans condensés" continue... vous pourrez maintenant, en plus des écrans complets, manipuler des fenêtres partielles. D'autre part, l'utilisation de l'Ampersand (&) facilite l'exploitation pour vous donner un véritable utilitaire. La version proposée utilise la carte langage et ne fonctionne donc que sous DOS. Si vous êtes un acharné de ProDOS je vous explique plus loin comment vous en sortir. Le mode de codage est toujours le même : tout caractère qui se répète est comptabilisé et on ne retient que le nombre à partir du troisième. **Seule limitation** : ne pas dépasser 256 caractères identiques (attention, pour les fenêtres surtout !).

CRÉATION

Le programme Basic ECR.CREER n'est qu'une optimisation du précédent*, la commande "F" a été ajoutée pour les fenêtres. La fenêtre peut se situer n'importe où sur l'écran ; on ne mémorise que ses dimensions. Pour le reste vous n'avez qu'à vous laisser guider... Le codage est assuré par une routine du programme assembleur ECRAN.LM. Vous n'avez que 12K pour la table, alors surveillez bien votre compteur !

LECTURE

Lisez et exécutez les programmes *ECR.DEMOC* (CRÉATION) et *ECR.DEMOL* (LECTURE) et vous comprendrez facilement comment les choses se passent. Naturellement, avant de lancer *ECR.DEMOL*, créez votre table avec *ECR.CREER*. Le bloc 2 de la carte langage constitue un tampon pour les fenêtres.

VOS ÉCRANS PERSONNELS

Pour créer vos propres écrans (première et indis-

pensable étape !), utilisez *FAIRE.ECRAN* paru dans le numéro 1 de *TREPLIN MICRO* (de la page 17 à la page 23). Bien entendu, il s'agira d'écrans en mode TEXT et non d'écrans graphiques. Choisissez (comme dans notre démonstration — *ECR.DEMOC*) des titres courts : E1, E2, et ainsi de suite.

POUR CHANGER L'ALLOCATION MÉMOIRE

Vous pouvez changer les adresses, en particulier si vous voulez adapter le programme à ProDOS (ce dernier occupant en effet la carte langage). Prenons un exemple : vous décidez de caser la table de \$6000 à \$7FFF, et le tampon pour les fenêtres de \$8000 à \$8FFF. Apportez les modifications suivantes à *ECRAN.LM* :

\$9207	:	\$80	au lieu de	\$D0
\$922D	:	\$8F	au lieu de	\$DF
\$9234	:	\$8F	au lieu de	\$DF
\$9278	:	\$80	au lieu de	\$D0
\$928E	:	\$8F	au lieu de	\$DF
\$9293	:	\$8F	au lieu de	\$DF
\$9314	:	\$60	au lieu de	\$D0
\$9319	:	\$60	au lieu de	\$D0
\$93E8	:	\$8F	au lieu de	\$DF



Dans le programme Basic ECR.CREER donnez à **HIMEM** (ligne 4) la valeur **23552** et la valeur **24576** (\$6000) à la variable **AA** de la ligne 5. Si vous avez 128K rien ne vous empêche d'utiliser la mémoire auxiliaire qui vous ouvre de vastes étendues...

Dans le programme *ECR.DEMOL*, fixez aussi **HIMEM** à **23552**.

* *CF TREPLIN MICRO N°5*

Commencez par une démonstration

Cette petite routine crée 5 écrans totalement "bidon" (le dernier figure une fenêtre).

Ils sont mémorisés automatiquement (E1, E2, E3, E4 et E5).

On termine par un **RUN ECRAN.CREER**

ECR.DEMOC

```

100 TEXT : PRINT CHR$ (21): HOME
105 N = N + 1:L$ = "": FOR I = 1 TO 20:L$ = L$ + CHR$ (6
    4 + N): NEXT
110 FOR I = 1 TO 20:L$ = L$ + STR$ (N): NEXT
115 FOR I = 1 TO 23: VTAB I: PRINT L$:: NEXT
120 GOSUB 140: IF N < 4 THEN 105
125 HOME : INVERSE : VTAB 8: PRINT ".....": V
    TAB 16: PRINT ".....": FOR I = 8 TO 16: V
    TAB I: PRINT ".": HTAB 16: PRINT ".": NEXT : NORMAL

130 N = N + 1: GOSUB 140
135 HOME : VTAB 12: PRINT "J'APPELLE ECRAN.CREER": PRINT
    : PRINT CHR$ (4)"RUN ECRAN.CREER"
140 VTAB 1:F$ = "E" + STR$ (N): PRINT "": PRINT CHR$ (4)
    "BSAVE"F$,A1024,L1024": RETURN
  
```

Le mode d'emploi du programme de Maurice CHAVELLI ne saurait être plus simple ni plus clair : c'est pratiquement automatique.

- 1 Tapez le titre du masque (E1... E2)
- 2 Tapez M pour mémoriser un écran normal ou F s'il s'agit d'une fenêtre. ESCAPE permet de ne pas mémoriser le masque affiché.
- 3 Quand vous aurez terminé, suivez les instructions et donnez ECR.TABLE comme titre... de TABLE. Vous constaterez avec plaisir que 5 écrans (de 5 ou 6 secteurs) n'en occupent plus que 4... en tout (cas de notre démonstration).

ECR.DEMOL

```

100 TEXT : PRINT CHR$ (21): HOME
105 HIMEM: 37376 :REM SOUS PRODOS:23552
110 PRINT CHR$ (4)"BRUNECRAN.LM"
115 POKE - 16247,0: POKE - 16247,0
120 PRINT CHR$ (4)"BLOADECR.TABLE,A#D000"
125 HOME
130 FOR C = 1 TO 4: & C: GOSUB 170: NEXT
135 & 5,8,13: GOSUB 170
140 FOR C = 3 TO 1 STEP - 1: & C: GOSUB 170: NEXT
145 Y = 10
150 FOR X = 2 TO 20 STEP 2:Y = Y + 1: & 5,X,Y: GOSUB 170
    : NEXT
155 FOR C = 1 TO 10: & : GOSUB 170: NEXT
160 IF PEEK ( - 16384) < 128 THEN 160
165 POKE - 16368,0: HOME : END
170 IF PEEK ( - 16384) < 128 THEN 170
175 POKE - 16368,0: RETURN
  
```

LIGNE 120 (SOUS PRODOS)
A#6000 AU LIEU DE A#D000

Apprenez à bien utiliser ECRAN.CREER

130 : Affichage successifs des écrans E1-E2-E3-E4.

135 : Affichage de la fenêtre E5 à la position 8-13.

140 : Affichage E3-E2-E1.

150 : Fenêtre E5 avec des paramètres différents.

155 : Les fenêtres disparaissent une à une.

A VOUS DE JOUER !

• ECRAN.CREER

```

1 REM -----
2 REM  INITIALISATIONS
3 REM -----
4 HIMEM: 24576: TEXT :D$ = CHR$ (4
  ): PRINT D$"BLOADECRAN.LM": ONER
  R GOTO 11
5 AA = 53248: REM ADRESSE DEFINITI
  VE
6 AD = 24576: REM ADRESSE DE STOCK
  AGE
7 POKE 249,0: POKE 250, INT (AD /
  256) + 1:N = 0: POKE AD,0: POKE
  AD + 1, PEEK (250): GOTO 12
8 REM -----
9 REM  MENU
10 REM -----
11 CALL 37999: REM PETITE MUSIQUE P
  OUR ERREUR
12 GOSUB 50: VTAB 20: INVERSE : PRI
  NT "ENTRER 'C' POUR LE CATALOGUE
  ": HTAB 6: PRINT "ENTRER 'F' P
  OUR TERMINER ": NORMAL : VTAB 1
  6: HTAB 6: INPUT "NOM DU MASQUE
  ? ";A$
13 IF A$ = "C" THEN HOME : PRINT D$
  "CATALOG": PRINT : PRINT "TOUCHE
  QUELCONQUE POUR LA SUITE": GOSU
  B 65: GOTO 12
14 IF A$ = "F" THEN 30
15 REM -----
16 REM  CHARGEMENT MASQUE
17 REM -----
18 PRINT D$"BLOAD";A$: POKE 216,0
19 GOSUB 65: IF A$ = CHR$ (27) THEN
  12
20 IF A$ = "M" THEN CALL 37661: GOT
  O 26
21 IF A$ = "F" THEN CALL 37682: GOT
  O 26
22 CALL 37999: GOTO 19
23 REM -----
24 REM  MISE A JOUR ADRESSE
25 REM -----
26 N = N + 1: POKE AD + 2 * N, PEEK
  (249): POKE AD + 1 + 2 * N, PEE
  K (250): GOTO 12
27 REM -----
28 REM  MENU DE SORTIE
29 REM -----
30 GOSUB 70: HTAB 12: PRINT "(S)AUV
  EGARDE": PRINT : HTAB 12: PRINT
  "(A)NNULLATION"
31 GOSUB 65: IF A$ = "A" THEN 7
32 IF A$ = "S" THEN 34
33 CALL 37999: GOTO 31
34 GOSUB 70: HTAB 6: INPUT "SOUS QU
  EL NOM? ";A$
35 REM -----
36 REM  COMPACTAGE TABLE
37 REM -----
38 POKE 60,0: POKE 61, PEEK (AD + 1
  ): POKE 62, PEEK (249): POKE 63,
  PEEK (250): POKE 66,N * 2: POKE
  67, PEEK (AD + 1) - 1: CALL 368
  64 + 1027
39 FOR E = 0 TO 2 * N - 2 STEP 2:V
  = PEEK (AD + E) + PEEK (AD + 1 +
  E) * 256 - 256 + 2 * N + AA - A
  D:H = INT (V / 256): POKE AD + E
  ,V - H * 256: POKE AD + 1 + E,H:
  NEXT
40 REM -----
41 REM  SAUVEGARDE TABLE
42 REM -----
43 PRINT D$"BSAVE";A$;" ,A";AD;" ,L";
  D
44 GOSUB 70: HTAB 6: PRINT "UNE AUT
  RE TABLE ?"
45 GOSUB 65: IF A$ = "O" THEN 7
46 HOME : END
47 REM -----
48 REM  S/P AFFICHAGE ECRAN
49 REM -----
50 HOME : VTAB 1: HTAB 8
51 PRINT "CREATION TABLE DE MASQUES
  "
52 HTAB 8: FOR E = 1 TO 25: PRINT "
  -";: NEXT
53 VTAB 5: HTAB 1: GOSUB 61
54 D = PEEK (249) + PEEK (250) * 25
  6 - AD - 256 + 2 * N
55 VTAB 8: HTAB 5: PRINT "LONGUEUR
  DE LA TABLE : ";D
56 VTAB 10: HTAB 5: PRINT "NOMBRE D
  E MASQUES : ";N
57 VTAB 12: GOSUB 61: VTAB 22: GOSU
  B 61
58 PRINT : PRINT "      <ESC>
  <M>      <F>";
59 VTAB 16: HTAB 6
60 RETURN
61 FOR E = 1 TO 40: PRINT "-";: NEX
  T : RETURN

```

```

62 REM -----
63 REM S/P ATTENTE TOUCHE
64 REM -----
65 IF PEEK ( - 16384) < 128 THEN 65
66 GET A$: RETURN
67 REM -----
68 REM S/P EFFACE LIGN. 16 A 21
69 REM -----
70 POKE 34,15: POKE 35,21: CALL - 9
36: TEXT : VTAB 16: RETURN

```

• **ECRAN.LM** BSAVE ECRAN.LM, A\$9200, L669

La version source commentée figure sur la disquette *TM9* (ProCODE).

```

9200: A9 00 85 F9 85 FB A9 D0
9208: 85 FA A9 4C 8D F5 03 A9
9210: 1A 8D F6 03 A9 92 8D F7
9218: 03 60 D0 3E A6 FB D0 04
9220: 20 6F 94 60 AD 80 C0 CA
9228: 8A 0A AA BD 00 DF 85 06
9230: 85 F9 BD 01 DF 85 07 85
9238: FA 20 64 94 8D 2A 94 20
9240: 64 94 8D 60 94 20 64 94
9248: 8D 23 94 20 64 94 8D 58
9250: 94 20 1B 94 C6 FB AD 82
9258: C0 60 20 F8 E6 86 FC 20
9260: B7 00 D0 1B AD 88 C0 20
9268: 0C 93 20 09 94 A5 FB F0
9270: 0A A9 00 85 FB 85 F9 A9
9278: D0 85 FA AD 8A C0 60 AD
9280: 81 C0 AD 81 C0 A6 FB 8A
9288: 0A AA A5 F9 9D 00 DF A5
9290: FA 9D 01 DF 20 F5 E6 8E
9298: A1 93 8E 2A 94 8A 20 E5
92A0: 93 20 F5 E6 E8 E0 19 90
92A8: 03 4C FC 93 8E D8 93 8E
92B0: 60 94 8A 20 E5 93 AD 88
92B8: C0 20 0C 93 20 64 94 85
92C0: FD AD 81 C0 AD 81 C0 AD
92C8: D8 93 38 E5 FD B0 03 4C
92D0: FC 93 8D 9A 93 8D 23 94
92D8: 20 E5 93 AD 88 C0 20 64
92E0: 94 18 6D A1 93 C9 29 90
92E8: 03 4C FC 93 8D D0 93 8D
92F0: 58 94 48 AD 81 C0 AD 81
92F8: C0 68 20 E5 93 20 8F 93
9300: E6 FB AD 88 C0 20 1B 94
9308: AD 8A C0 60 C6 FC 06 FC

```

```

9310: A6 FC BD 00 D0 85 06 BD
9318: 01 D0 85 07 60 A9 00 8D
9320: A1 93 8D 9A 93 A9 28 8D
9328: D0 93 A9 18 8D D8 93 4C
9330: 8F 93 A9 00 85 EB 20 85
9338: 94 A0 00 B1 28 C9 A0 D0
9340: 0B C8 C0 28 D0 F5 E6 EB
9348: A5 EB D0 EA 8C A1 93 A5
9350: EB 8D 9A 93 B1 28 C9 A0
9358: F0 05 C8 C0 28 D0 F5 8C
9360: D0 93 AC A1 93 B1 28 C9
9368: A0 F0 0E E6 EB A5 EB C9
9370: 18 F0 08 20 85 94 4C 62
9378: 93 A5 EB 8D D8 93 38 ED
9380: 9A 93 20 E5 93 AD D0 93
9388: 38 ED A1 93 20 E5 93 A9
9390: 01 85 FE A9 00 85 FF 85
9398: FD A9 00 85 EB 20 85 94
93A0: A0 00 B1 28 A6 FE D0 1B
93A8: C5 FD D0 05 E6 FF 4C CE
93B0: 93 48 A9 01 85 FE A5 FF
93B8: 20 E5 93 A9 00 85 FF 68
93C0: 4C CB 93 C5 FD D0 04 A2
93C8: 00 86 FE 20 E5 93 C8 C0
93D0: 00 D0 CF E6 EB A5 EB C9
93D8: 00 D0 C2 A5 FE D0 05 A5
93E0: FF 20 E5 93 60 A6 FA E0
93E8: DF F0 0D A2 00 81 F9 85
93F0: FD E6 F9 D0 02 E6 FA 60
93F8: 68 68 68 68 20 6F 94 AD
9400: 82 C0 60 A0 00 20 2C FE
9408: 60 A9 00 8D 23 94 8D 2A
9410: 94 A9 28 8D 58 94 A9 18
9418: 8D 60 94 A2 00 86 FD E8
9420: 86 FE A9 00 85 EB 20 85
9428: 94 A0 00 A6 FE D0 0D C6
9430: FF D0 04 A2 01 86 FE A5
9438: FD 4C 52 94 20 64 94 C5
9440: FD D0 0F 20 64 94 C9 00
9448: F0 06 85 FF A9 00 85 FE
9450: A5 FD 91 28 85 FD C8 C0
9458: 28 D0 D0 E6 EB A5 EB C9
9460: 18 D0 C3 60 A2 00 A1 06
9468: E6 06 D0 02 E6 07 60 A0
9470: C0 A9 12 38 48 E9 01 D0
9478: FC 68 E9 01 D0 F6 AD 30
9480: C0 88 D0 ED 60 48 4A 29
9488: 03 09 04 85 29 68 29 18
9490: 90 02 69 7F 85 28 0A 0A
9498: 05 28 85 28 60

```

CONTRÔLE : T = 0 : FOR I = 37376 TO 38044 : T = T + PEEK(I) : NEXT : PRINT RÉSULTAT : 93007

Super catalogue

ProDOS : CATPRO

Avec carte 80 colonnes

S vos disquettes ProDOS sont bourrées de fichiers, vous apprécierez certainement l'utilitaire de Benjamin Ané. Il présente un maximum de 92 fichiers sur deux masques (écrans... pour les néophytes), et lit aussi bien sous-catalogues que sous-sous-catalogues (DRIVE 1, DRIVE 2 et RAM). Bien entendu — on s'en doutait un peu ! — il permet de lancer les programmes en Basic

```
10 REM      UTILITAIRE PRODOS
           — par Ané Benjamin —
```

```
20 F$ = " U T I L I T A I R E - C A
      P T U R E D E C A T A L O G U
      E "
```

```
30 D$ = CHR$(4): PRINT D$"PR&3": P
      RINT : DIM FP$(93): ONERR GOTO 9
      00
```

```
40 REM      PRESENTATION
```

```
50 FOR I = 1 TO 92:FP$(I) = "": NEX
      T :X = 1:RE$ = ""
```

```
60 TEXT : HOME : INVERSE : PRINT F$
      ;: HTAB 64: PRINT " * M E N U
      * "
```

```
70 VTAB 5: PRINT " INFORMATIONS..."
      ;: NORMAL
```

```
80 PRINT " Avec ce programme vous l
      isterez catalogues & s/catalogue
      s," SPC( 22)"et vous pourrez lan
      cer le programme BASiC de votre
      choix.": PRINT
```

```
90 PRINT "Le signe * devant le nom
      du fichier vous indique que le f
      ichier est verrouillé. Le signe
      / devant le type précise que le
      fichier est du dernier s/cat DIR
      cité."
```

```
95 PRINT "Le signe s devant le type
      indique un sous fichier du dern
      ier s/cat DIR cité...."
```

```
100 PRINT : PRINT "Vous pouvez visua
      liser jusqu'à 92 fichiers, ce qu
      i est déjà fort honorable!. Le m
      aximun est... autant de K que vo
      tre disquette peut en contenir;
      vous imaginez !"
```

```
110 PRINT SPC( 200)"ESCAPE pour reve
      nir à ce MENU"
```

```
120 VTAB 16: FOR I = 1 TO 4: INVERSE
      : PRINT " ";I;" ": NORMAL : PRI
      NT : NEXT I: VTAB 16
```

```
130 HTAB 5: PRINT "volume du drive n
      °1": PRINT : HTAB 5: PRINT "volu
      me du drive n°2": PRINT : HTAB 5
      : PRINT "volume virtuel /RAM": P
      RINT : HTAB 5: PRINT "quitter l'
      utilitaire"
```

```
140 VTAB 20: HTAB 45: PRINT "VOTRE C
      HOIX...1 TO 4 ";
```

```
150 GET Z$: IF Z$ < "1" OR Z$ > "4"
      THEN 150
```

```
160 Z = VAL (Z$): ON Z GOTO 180,190,
      200,170
```

```
170 TEXT : HOME : END
```

```
180 PRINT D$;"PREFIX,S6,D1": GOTO 21
      0
```

```
190 PRINT D$;"PREFIX,S6,D2": GOTO 21
      0
```

```
200 PRINT D$;"PREFIX,S3,D2"
```

```
210 PRINT D$"PREFIX": INPUT PX$
```

```
220 VTAB 1: HTAB 64: INVERSE : PRINT
      " ";: HTAB 64: PR
      INT PX$
```

```
230 VTAB 20: HTAB 45: PRINT " travail  
  | en cours... ": NORMAL
```

```
240 REM          LECTURES
```

```
250 PRINT D$"OPEN";PX$,T DIR": PRIN  
T D$"READ";PX$: INPUT LP$,ET$,LV  
$
```

```
255 INPUT Y$:FP$(X) = LEFT$(Y$,28):  
VTAB 20: HTAB 67: PRINT X - 1;
```

```
260 IF MID$(FP$(X),18,3) = "DIR" TH  
EN FP$(X) = FP$(X) + PX$: GOTO 2  
90
```

```
265 IF Y$ < > "" THEN FP$(X) = FP$(X  
) + PX$
```

```
270 IF Y$ < > "" THEN X = X + 1: GOT  
O 255
```

```
275 INPUT RE$
```

```
280 PRINT D$"CLOSE";PX$: PRINT D$"PR  
EFIX,S6,D1": GOTO 400
```

```
285 :
```

```
290 A$ = MID$(FP$(X),2,15):PR$ = PX  
$ + A$:E$ = "/"
```

```
300 PRINT D$;"PREFIX";PR$: PRINT D$;  
"PREFIX": INPUT PP$
```

```
305 PRINT D$"OPEN";PP$,T DIR": PRIN  
T D$"READ";PP$: INPUT LP$,ET$,LV  
$
```

```
310 X = X + 1: INPUT Y$:FP$(X) = LEF  
T$(Y$,16): VTAB 20: HTAB 67: PR  
INT X - 1;
```

```
315 IF MID$(Y$,18,3) = "DIR" THEN 3  
45
```

```
320 IF FP$(X) < > "" THEN FP$(X) = (  
FP$(X) + E$ + MID$(Y$,18,11) +  
PP$): GOTO 310
```

```
325 PRINT D$"CLOSE";PP$: PRINT D$;"P  
REFIX/"
```

```
330 PRINT D$;"PREFIX": INPUT PX$: PR  
INT D$"READ";PX$: INPUT LV$
```

```
335 GOTO 255
```

```
340 :
```

```
345 B$ = MID$(FP$(X),2,15):PE$ = PP  
$ + B$:C$ = "s"
```

```
350 FP$(X) = (FP$(X) + C$ + MID$(Y$  
,18,11) + PP$)
```

```
355 PRINT D$;"PREFIX";PE$: PRINT D$;  
"PREFIX": INPUT P3$
```

```
360 PRINT D$"OPEN";P3$,T DIR": PRIN  
T D$"READ";P3$: INPUT LP$,ET$,LV  
$
```

```
365 X = X + 1: INPUT Y$:FP$(X) = LEF  
T$(Y$,16): VTAB 20: HTAB 67: PR  
INT X - 1;
```

```
370 IF FP$(X) < > "" THEN FP$(X) = (  
FP$(X) + C$ + MID$(Y$,18,11) +  
P3$): GOTO 365
```

```
375 PRINT D$"CLOSE";P3$: PRINT D$;"P  
REFIX";PP$
```

```
380 PRINT D$;"READ";PP$:X = X - 1: G  
OTO 310
```

```
390 REM          AFFICHAGE
```

```
400 TEXT : HOME : INVERSE : HTAB 40  
- LEN(PX$) / 2: PRINT PX$: NORM  
AL : PRINT
```

```
410 FOR I = 1 TO 3: PRINT "      "; L  
EFT$(ET$,6); MID$(ET$,16,15);:  
NEXT
```

```
420 PRINT : POKE 34,3: PRINT : GOTO  
470
```

```
430 IF FP$(I) = "" THEN FP$(I) = " .  
....."
```

```
440 INVERSE : PRINT (I);: NORMAL
```

```
450 PRINT LEFT$(FP$(I),20); MID$(F  
P$(I),26,3): RETURN
```

```
460 REM          TABLEAU 1
```

```
470 VTAB 5: FOR I = 1 TO 9: INVERSE  
: PRINT "0";: GOSUB 430: NEXT
```

```
480 VTAB 14: FOR I = 10 TO 16: GOSUB  
430: NEXT
```

```
490 VTAB 5: FOR I = 17 TO 32: HTAB 2  
8: GOSUB 430: NEXT
```

```
500 VTAB 5: FOR I = 33 TO 46: HTAB 5  
5: GOSUB 430: NEXT
```

```
510 VTAB 20: HTAB 55: INVERSE : PRIN  
T " flèche droite > SUITE..": NO  
RMAL
```

```
520 POKE 34,21: PRINT
```

```
530 IF Z = 3 THEN PRINT D$;"CATALOG/  
RAM"
```

```
540 POKE 34,22: PRINT
```

```
550 VTAB 22: PRINT RE$;: HTAB 45: PR  
INT "n°XX du programme BASic cho  
isi ? ";
```

```
560 GET Y$: IF ASC(Y$) = 21 THEN HO  
ME : GOTO 660
```

```
570 IF ASC(Y$) = 27 THEN 50
```

```
580 IF ASC(Y$) < 48 OR ASC(Y$) > 5  
2 THEN 560
```

```
590 PRINT Y$;
```

```
600 GET Z$: IF ASC(Z$) < 48 OR ASC  
(Z$) > 57 THEN 600
```

```
610 PRINT Z$:CH$ = Y$ + Z$:CH = VAL  
(CH$): IF CH < 1 THEN 550
```

• CATPRO (suite et fin)

```

615 IF MID$(FP$(CH),18,3) < > "BAS"
    THEN 550
620 IF CH < 17 THEN V = (CH + 4):H =
    2: GOTO 810
630 IF CH < 33 THEN V = (CH - 12):H
    = 29: GOTO 810
640 IF CH < 47 THEN V = (CH - 28):H
    = 56: GOTO 810
650 REM TABLEAU 2

660 VTAB 5: FOR I = 47 TO 62: GOSUB
    430: NEXT
670 VTAB 5: FOR I = 63 TO 78: HTAB 2
    8: GOSUB 430: NEXT
680 VTAB 5: FOR I = 79 TO 92: HTAB 5
    5: GOSUB 430: NEXT
690 VTAB 20: HTAB 55: INVERSE : PRIN
    T " flèche gauche < RETOUR ": NO
    RMAL
700 VTAB 22: HTAB 78: PRINT " ";: H
    TAB 78: GET Y$: IF ASC (Y$) = 08
    THEN HOME : GOTO 400
710 IF ASC (Y$) = 27 THEN 50
720 IF ASC (Y$) < 52 OR ASC (Y$) > 5
    7 THEN 700
730 PRINT Y$;
740 GET Z$: IF ASC (Z$) < 48 OR ASC
    (Z$) > 57 THEN 740
750 PRINT Z$;:CH$ = Y$ + Z$:CH = VAL
    (CH$): IF CH < 47 THEN 700
755 IF MID$(FP$(CH),18,3) < > "BAS"
    THEN 700
760 IF CH < 63 THEN V = (CH - 42):H
    = 2: GOTO 810

```

```

770 IF CH < 79 THEN V = (CH - 58):H
    = 29: GOTO 810
780 IF CH < 93 THEN V = (CH - 74):H
    = 56: GOTO 810
790 CALL - 1008: GOTO 740
800 REM VOTRE CHOIX

```

```

810 FI$ = MID$(FP$(CH),2,15):PF$ =
    MID$(FP$(CH),29,49)
820 VTAB V: POKE 1403,H: INVERSE : P
    RINT "<";FI$;">": NORMAL
830 POKE 34,23: HOME : HTAB 33: PRIN
    T "ES-CE CORRECT...O/N ";: GET Z
    $: HOME
840 IF Z$ = "O" THEN VP$ = PF$ + FI$
    : PRINT D$;"RUN";VP$
850 IF Z$ = "N" THEN 400
860 IF ASC (Z$) = 27 THEN 50
870 GOTO 830
900 REM ONERR GOTO

```

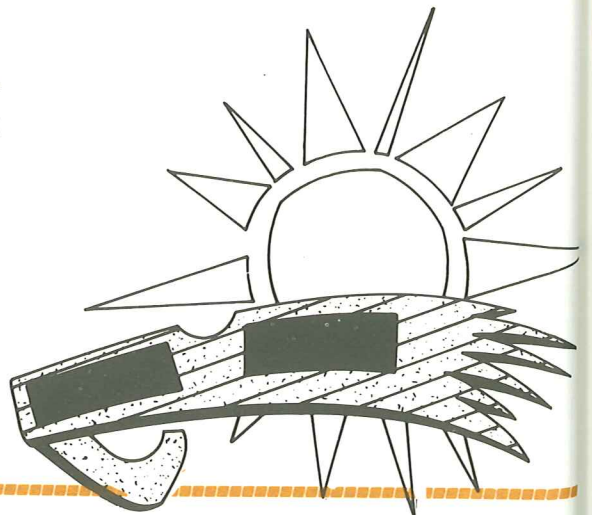
```

910 IF PEEK (222) = 5 THEN FP$(X) =
    "": GOTO 280
920 POKE 32,40: POKE 33,35: POKE 34,
    16: POKE 35,22: INVERSE : HOME
930 POKE 32,41: POKE 33,33: POKE 34,
    17: POKE 35,21: NORMAL : HOME
940 IF PEEK (222) = 8 THEN PRINT SPC
    ( 39)"VERIFIER VOTRE LECTEUR"
950 IF PEEK (222) = 3 THEN PRINT SPC
    ( 39)"VERIFIER INSTALLATION."
960 IF PEEK (222) = 107 THEN PRINT S
    PC( 39)"DESOLE! JE SUIS SATURE"
970 PRINT : PRINT "      presser un
    e touche ";: GET Z$: GOTO 400

```

S I vous avez l'intention de commander la disquette de *TREMLIN MICRO N°9*, soyez aimable de le faire sans tarder :

- vous serez servi par retour de courrier (ce qui est d'ailleurs le cas en règle générale),
- vous faciliterez la tâche de notre service "expéditions"
- vous éviterez les habituels retards dus, pendant le mois d'août, aux nombreux problèmes que rencontrent les PTT : réexpéditions du courrier, pénurie de personnel, etc.



TYPC : une nouvelle commande pour ProDOS

CERTAINS lecteurs connaissent sans doute la commande *TDUMP* ajoutée au *DOS 3.3* pour permettre le listage des fichiers *TEXT* sans en connaître la structure.

Les auteurs de *ProDOS* ont "oublié" cette intéressante fonction lors de la conception de leur système d'exploitation. Plusieurs versions en ont donc été proposées par divers auteurs.

Très proche de celle publiée dans *Beneath Apple ProDOS* ma version en diffère sur deux points importants :

1. La routine qui reloge le module actif entre *ProDOS* et ses buffers n'est pas limitée au traitement de routines de moins de 256 octets.
2. Si votre fichier *TXT* renferme des caractères de contrôle, je vous souhaite bien du plaisir lors d'une sortie papier avec les routines *TYPE* antérieures. Il est possible d'éviter la crise de nerfs avec la carte série *APPLE* en filtrant systématiquement les caractères de contrôle (par *CTRL-I>Z<CR>*) mais dans ce cas il n'y en aura aucune trace. Au contraire, *PRO.TYPC* remplace soigneusement chaque caractère de contrôle par son équivalent majuscule normal encadré par les caractères *<et>* (*CTRL-A-><A>*).

Le programme *DEMO.PRO.TYPC* vous montre comment utiliser *PRO.TYPC* (qui fonctionne aussi en mode immédiat). ATTENTION, il ne faut pas faire plusieurs *BRUNS PRO.TYPC* successifs car *HIMEM* descend d'une page à chaque fois.

La source me semble suffisamment documentée pour ceux qui souhaitent étudier la routine de près.

TYPE Si vous désirez supprimer le recodage des contrôles il vous suffit de taper :
\$6183: 09 80 4C 9F 61, puis \$61C7: 50 avant de sauver la routine par un :
BSAVE PRO.TYPE, A\$6037, L401.

Vous pourrez alors activer *TYPC*, *TYPE*, ou les deux si besoin est.

TYPE pourra sans doute dépanner les utilisateurs d'interface *EPSON* qui ne peuvent imprimer avec *APPLEWORKS*. La vraie solution est de changer de carte mais, en attendant, il est possible d'imprimer sur disque en utilisant la commande ad'hoc de *APPLEWORKS*. Il vous restera à *TYPER* le fichier ainsi créé après avoir quitté *APPLEWORKS*. Utilisez de préférence une disquette ne portant pas *ProDOS* afin de disposer d'un maximum de place car les fichiers "imprimés" sont TRÈS encombrants.

Y.K.

PRO.TYPC

Yvan KOENIG

Si vous n'êtes pas rompu à la pratique de l'Assembleur... ou si vous ne possédez pas d'assembleur, ne vous découragez pas pour autant et rendez-vous directement à la page 38. Vous y trouverez à la fois le programme DEMO en Basic et une liste des codes de la routine LM.

```
1
2 *****
3 *
4 *          PRO.TYPE          *
5 *          PRO.TYPC         *
6 * Yvan KOENIG          10-03-86 * Révisé 20/03/86
7 *
8 *****
9 CTRL      =      1
10 * Avec CTRL = 1 TYPC permet de filtrer les caractères
11 * de controle éventuellement contenus dans le fichier
12 * et de les afficher entre crochets: ctrl A --> <A>
13 *****
14 *
15 * On demande à GETBUFR de nous allouer une page
16 * puis le relogeur copie la routine TYPE (TYPC)
17 * dans la page qui nous est attribuée.
18 *
19 * TYPE nomfichier <,Snumslot> <,Ddrivnum> (ou TYPC)
20 * permet de lire tout fichier de type TEXT
21 * le listage peut être interrompu en frappant une touche
22 * Une autre frappe le relance.
23 * Frapper Ctrl C provoque l'arrêt complet du listage.
24 *
25 *****
26
27 OFFSET   =      2          ;&3
28
29 LENGTH   =      $2F          ;Longueur opérande
30 START    =      $3C          ;Début du bloc à dépl. par MOVE
31 END      =      $3E          ;Fin du bloc pour MOVE
32 DEST     =      $42          ;Début zone dest. pour MOVE
33 PTR      =      $48
34 HIMEM    =      $73          ;&74 Plus haute adresse
                                   disponible en MEM
35 CTLC     =      $83          ;Ctrl C
36
37 TAMPON   =      $200         ;Buffer temporaire pour TYPE
38
39 * Adresses ProDOS
40
41 CI_ENTRY =      $BE00
42 EXTRNCMD =      CI_ENTRY+6
43 ERROUT   =      CI_ENTRY+9
44 XTRNADDR =      $BE50
45 XLEN     =      $BE52
46 XCNUM    =      $BE53
47 PBITS    =      $BE54
48 VPATH1   =      $BE6C
49 GOSYSTEM =      $BE70
50 OSYSBUF  =      $BECE
51 OREFNUM  =      $BED0
52 RWREFNUM =      $BED6
53 RWDATA   =      $BED7
54
55 RWCOUNT =      $BED9
56 RWTRANS  =      $BEDB
57 CFREFNUM =      $BEDE
58 GETBUFR  =      $BEF5
59
60 * Adresses SYSTEME
61 KBD      =      $C000
62 STROBE   =      $C010
63
64 * Adresses MONITEUR
65
```

```

66  INSDS2  =    $F88E    ;Entrée du désassembleur
67  NXTA4   =    $FCB4    ;Av. DEST L/H et tombe en NXTA1
68  CROUT   =    $FD8E    ;Affiche CR
69  COUT     =    $FDED    ;Af. le caractère contenu dans A
70  MOVE     =    $FE2C    ;Déplace (START L/H--END L/H)
                               vers DEST L/H
71
72  *****
73          ORG    $6037
74  *****
75
6037: A9 01   76          LDA    £LAST-PROGRBEG/$100+1
6039: 20 F5 BE 77          JSR    GETBUFR    ;Libère (A) pages & revient avec
                               ;partie haute adresse dans (A)
78
603C: 90 0E   79          BCC    PAGEFND
603E: A0 16   80          LDY    £CANTEND-CANTRELO-1
6040: B9 E9 60 81  CANTLOOP LDA    CANTRELO,Y
6043: 20 ED FD 82          JSR    COUT
6046: 88      83          DEY
6047: 10 F7   84          BPL    CANTLOOP
6049: 4C 00 BE 85          JMP    CI_ENTRY
86
87  PAGEFND
604C: 85 43   88          STA    DEST+1    ;Partie haute adr. zone allouée
604E: AE 08 BE 89          LDX    EXTRNCMD+2 ;Sauve l'ancienne adresse
6051: 8E 3D 61 90          STX    NEXTCMD+2 ;pour le chaînage des commandes
6054: 8D 08 BE 91          STA    EXTRNCMD+2 ;Installe la nouvelle adresse
6057: AE 07 BE 92          LDX    EXTRNCMD+1
605A: 8E 3C 61 93          STX    NEXTCMD+1
94
605D: A9 00   95          LDA    £<PROGRBEG ;Début de zone à déplacer
605F: 85 3C   96          STA    START
6061: A9 61   97          LDA    £>PROGRBEG
6063: 85 3D   98          STA    START+1
99
6065: A9 C1   100         LDA    £<ENDPROG-1 ;Fin de la partie programme
6067: 85 3E   101         STA    END
6069: A9 61   102         LDA    £>ENDPROG-1
606B: 85 3F   103         STA    END+1
104
606D: D8      105         CLD                ;Au cas où
106
606E: A9 00   107         LDA    £0          ;PROGRBEG sur un début de page
6070: 85 42   108         STA    DEST
6072: 8D 07 BE 109         STA    EXTRNCMD+1
6075: 38      110         SEC
6076: E9 00   111         SBC    £<PROGRBEG ;Calcule offset
6078: 85 02   112         STA    OFFSET    ;que l'on ajoutera
607A: A5 43   113         LDA    DEST+1    ;aux adresses
607C: E9 61   114         SBC    £>PROGRBEG ;devant être "relogées"
607E: 85 03   115         STA    OFFSET+1
116
6080: A0 00   117         LDY    £0
118  DORELOC1                ;Ici Y=0

```

6082:	B1 3C	119	LDA	(START),Y	;Opcode
6084:	20 8E F8	120	JSR	INSDS2	;ATTENTION! ne connaît pas les nouveaux opcodes sur Iie
6087:	A6 2F	121	LDX	LENGTH	
6089:	E0 02	122	CPX	£3-1	
608B:	90 05	123	BCC	MOVIT1	;Pas de changement pour instruction 1/2 octet
608D:	C8	124	INY		;-> Y=1
608E:	20 C2 60	125	JSR	CHANGIT?	
6091:	88	126	DEY		;-> Y=0
		127	MOVIT1		;Ici X=LENGTH
6092:	B1 3C	128	MLOOP1	LDA (START),Y	;Copie l'instruction
6094:	91 42	129		STA (DEST),Y	;à sa destination
6096:	20 B4 FC	130	JSR	NXTA4	
6099:	CA	131	DEX		
609A:	10 F6	132	BPL	MLOOP1	
609C:	90 E4	133	BCC	DORELOC1	;Si A1<A2
		134			
609E:	A9 C3	135	LDA	£<ENDADRS-1	;Fin de table d'adresses
60A0:	85 3E	136	STA	END	
60A2:	A9 61	137	LDA	£>ENDADRS-1	
60A4:	85 3F	138	STA	END+1	
		139			
		140	DORELOC2		;Ici Y=0
60A6:	20 C2 60	141	JSR	CHANGIT?	
		142			
60A9:	A2 01	143	MOVIT2	LDX £2-1	;2 octets par adresse
60AB:	B1 3C	144	MLOOP2	LDA (START),Y	
60AD:	91 42	145		STA (DEST),Y	
60AF:	20 B4 FC	146	JSR	NXTA4	
60B2:	CA	147	DEX		
60B3:	10 F6	148	BPL	MLOOP2	
60B5:	90 EF	149	BCC	DORELOC2	;Si l'on n'a pas fini la table
		150			
60B7:	A9 C7	151	LDA	£<ENDALL-1	;Fin de table de valeurs
60B9:	85 3E	152	STA	END	
60BB:	A9 61	153	LDA	£>ENDALL-1	
60BD:	85 3F	154	STA	END+1	
		155			
60BF:	4C 2C FE	156	JMP	MOVE	;Ici Y=0, OK pour MOVE
		157			
		158			* C'est relogé
		159			
		160			*-----
		161			
		162	CHANGIT?		
60C2:	A9 C7	163	LDA	£<ENDALL-1	
60C4:	D1 3C	164	CMP	(START),Y	
60C6:	C8	165	INY		
60C7:	A9 61	166	LDA	£>ENDALL-1	
60C9:	F1 3C	167	SBC	(START),Y	
60CB:	90 1A	168	BCC	CHANGEND	;Pas de changement
		169			

```

60CD: 88      170      DEY
60CE: A9 00   171      LDA    £<PROGRBEG
60D0: D1 3C   172      CMP    (START),Y
60D2: C8      173      INY
60D3: A9 61   174      LDA    £>PROGRBEG
60D5: F1 3C   175      SBC    (START),Y
60D7: B0 0E   176      BCS    CHANGEND ;Pas de changement
                                ;Ici C=0
                                177
60D9: 88      178      DEY
60DA: B1 3C   179      LDA    (START),Y
60DC: 65 02   180      ADC    OFFSET ;Ajuste
60DE: 91 3C   181      STA    (START),Y ;l'adresse
60E0: C8      182      INY
60E1: B1 3C   183      LDA    (START),Y
60E3: 65 03   184      ADC    OFFSET+1
60E5: 91 3C   185      STA    (START),Y
                                186
60E7: 88      187      CHANGEND DEY
60E8: 60      188      RTS
                                189
60E9: 8D 87   190      CANTRELO HEX    8D,87
60EB: A1 C5 C3 191      REV    "IL N'Y PLUS DE PLACE!"
60EE: C1 CC D0 A0 C5 C4 A0 D3
60F6: D5 CC D0 A0 D9 A7 CE A0
60FE: CC C9
                                192      CANTEND
                                193
                                194      ERR    CANTEND/£100*£100-CANTEND
                                                ;Contrôle début page
                                195      PROGRBEG
                                196      *****
                                197      * TYPE commence ici
                                198      *****
6100: D8      199      CLD
6101: AD 6C BE 200      LDA    VPATH1 ;Copie pointeur sur com. entrée
6104: 85 48   201      STA    PTR
6106: AD 6D BE 202      LDA    VPATH1+1
6109: 85 49   203      STA    PTR+1
610B: A0 01   204      LDY    £1
610D: B1 48   205      CHKCMD LDA    (PTR),Y ;Compare la commande entrée
610F: D9 C3 61 206      CMP    CMDNAME-1,Y ; à TYPE
6112: D0 26   207      BNE    NOTYPE ;=> ce n'est pas TYPE
6114: C8      208      INY
6115: C0 04   209      CPY    £ENDALL-CMDNAME
6117: 90 F4   210      BCC    CHKCMD ;Caractère suivant
                                211
6119: 88      212      DEY ;C'est bien TYPE
611A: 8C 52 BE 213      STY    XLEN ;Sauve index dans la commande
611D: A9 00   214      LDA    £0
611F: 8D 53 BE 215      STA    XCNUM ;Marque commande externe
6122: A9 01   216      LDA    £1
6124: 8D 54 BE 217      STA    PBITS ;Réclame un "pathname"
6127: A9 04   218      LDA    £4

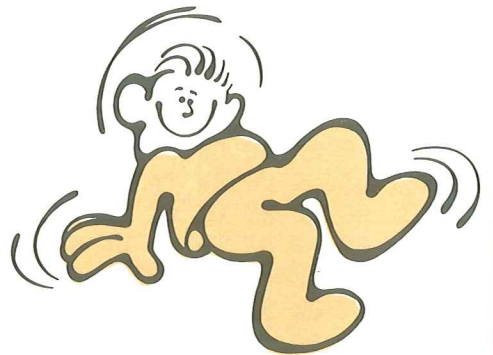
```

6129:	8D 55 BE	219	STA	PBITS+1	;Autorise options Slot Drive
612C:	AD C2 61	220	LDA	TOEXTRN	
612F:	8D 50 BE	221	STA	XTRNADDR	;Donne adresse de routine active
6132:	AD C3 61	222	LDA	TOEXTRN+1	;à BASIC.SYSTEM
6135:	8D 51 BE	223	STA	XTRNADDR+1	
6138:	18	224	CLC		;On a trouvé la commande
6139:	60	225	RTS		;Retour à BASIC.SYSTEM: analyse
		226			;de la commande et exécution.
613A:	38	227	NOTYPE	SEC	;On n'a pas trouvé TYPE
613B:	4C 00 00	228	NEXTCMD	JMP	\$0000 ;Vers l'ancien vecteur externe
613E:	4C 09 BE	229	ERROR	JMP	ERROUT ;Vers BASIC.SYSTEM erreur
		230			
		231			* Quand BASIC.SYSTEM a fini d'analyser la commande,
		232			* il arrive ici pour exécution.
		233			
6141:	A9 00	234	DOTYPE	LDA	£0
6143:	A4 74	235		LDY	HIMEM+1
6145:	8D CE BE	236		STA	OSYSBUF ;Part. bas. adr. buf. de travail
6148:	8C CF BE	237		STY	OSYSBUF+1 ;partie haute
614B:	A9 C8	238		LDA	£C8 ;MLI code OPEN
614D:	20 70 BE	239		JSR	GOSYSTEM ;OPEN fichier
6150:	B0 EC	240		BCS	ERROR ;=> ERREUR
6152:	AD D0 BE	241		LDA	OREFNUM ;Numéro de référence
6155:	8D D6 BE	242		STA	RWREFNUM ;pour READ
6158:	8D DE BE	243		STA	CFREFNUM ;pour CLOSE
615B:	A0 00	244	READ256	LDY	£0
615D:	8C D9 BE	245		STY	RWCOUNT
6160:	8C D7 BE	246		STY	RWDATA ;Marque mode lecture
6163:	C8	247		INY	;-> Y=1
6164:	8C DA BE	248		STY	RWCOUNT+1 ;(RWCOUNT)=\$100
6167:	C8	249		INY	;-> Y=2
6168:	8C D8 BE	250		STY	RWDATA+1 ;(RWDATA)=\$200 adresse du tampon
616B:	A9 CA	251		LDA	£CA ;MLI code READ
616D:	20 70 BE	252		JSR	GOSYSTEM ;READ 256 oct; & copie en \$200
6170:	90 0C	253		BCC	TYPEOUT ;OK on peut les afficher
6172:	C9 05	254		CMP	£5 ;Erreur = FIN DE FICHER ?
6174:	DO C8	255		BNE	ERROR ;NON => erreur fatale
		256			;On est en fin de fichier.
6176:	20 8E FD	257	QUITYPE	JSR	CROUT
6179:	A9 CC	258		LDA	£CC ;MLI code CLOSE
617B:	4C 70 BE	259		JMP	GOSYSTEM ;Ferme le fichier
		260			
617E:	A0 00	261	TYPEOUT	LDY	£0
6180:	B9 00 02	262	TYPELOOP	LDA	TAMPON,Y ;Récupère un caractère
		263			*****
		264		DO	CTRL
6183:	08	265		PHP	
6184:	09 80	266		ORA	£80 ;Sauve état (z=1 si A=\$00)
6186:	28	267		PLP	;Récupère état (z=1 si A valait \$00)
6187:	F0 16	268		BEQ	OUTIT ;On avait un \$00 fin de zone
		269		ELSE	
		270		ORA	£80
		271		JMP	OUTIT

```

272          NOP          ;Pour fixer taille programme
273          FIN
274 *****
6189: C9 8D 275          CMP    £$8D    ;CR ?
618B: F0 12 276          BEQ    OUTIT   ;On traite normalement les CR
618D: C9 A0 277          CMP    £$A0
618F: B0 0E 278          BCS    OUTIT   ;=> Ce n'est pas un contrôle
6191: 69 40 279          ADC    £$40    ;Contrôle -> caractère normal
6193: 48      280          PHA
6194: A9 BC 281          LDA    £"<"
6196: 20 ED FD 282          JSR    COUT    ;Affiche <
6199: 68      283          PLA
619A: 20 ED FD 284          JSR    COUT    ;Récupère le caractère recodé
619D: A9 BE 285          LDA    £">"
619D: A9 BE 286          ;Pour afficher >
619F: 20 ED FD 287  OUTIT  JSR    COUT
61A2: AD 00 C0 288          LDA    KBD
61A5: 10 13 289          BPL    NEXT
61A7: 8D 10 C0 290          STA    STROBE   ;On n'a pas touché le clavier
61AA: C9 83 291          CMP    £CTLC     ;Remise à zéro du clavier
61AC: F0 C8 292          BEQ    QUITYPE  ;Etait-ce ctrl C ?
61AE: AD 00 C0 293  WAIT  LDA    KBD
61B1: 10 FB 294          BPL    WAIT
61B3: 8D 10 C0 295          STA    STROBE   ;OUI, on arrête
61B6: C9 83 296          CMP    £CTLC     ;Attend une nouvelle touche
61B8: F0 BC 297          BEQ    QUITYPE  ;toujours rien
61BA: C8      298  NEXT  INY
61BB: CC DB BE 299          CPY    RWTRANS
61BE: D0 C0 300          BNE    TYPELOOP ;remise à zéro
61C0: F0 99 301          BEQ    READ256 ;ctrl C ?
61C0: F0 99 302          ;OUI, on arrête
303  ENDPROG
304 *****
305
61C2: 41 61 306  TOEXTRN DA  DOTYPE ;Fournir l'adresse au relogeur
307  ENDADRS
308 *****
309
310          DO    CTRL
61C4: 54 59 50 311  CMDNAME ASC  'TYPC'
61C7: 43
312          ELSE
313  CMDNAME ASC  'TYPE'
314          FIN
315  LAST    =    *-1
316  ENDALL  =    *
317 *****

```



MÉMORISATION : BSAVE PRO.TYPC, A\$6037, L401

VÉRIFICATION : T = 0 : FOR I = 24631 TO 24630 + 401 : T = T + PEEK(I) : NEXT : PRINT T

RÉSULTAT : 53714.

PRO.TYPC.DEMO

```
10 PRINT CHR$ (21)
20 TEXT : HOME : PRINT "DEMO PRO.TYPC"
30 VTAB 5: PRINT "PRO.TYPC met en place
une commande qui comble une lacune d
e ProDOS."
40 PRINT "TYPC nomfichier <,Dnumdrive> <
,Snmslot>permet de lire tout fichier
TEXT sans enconnaître la structure."
50 PRINT : PRINT "Si le fichier contient
des caractères de controle, ceux-ci
bénéficieront d'un": PRINT "traitemen
t particulier pour éviter les facéti
es classiques des imprimantes."
60 PRINT : PRINT "ctrl X sera remplacé
par le triplet <X>"
70 PRINT : PRINT "PATIENCE, je crée un p
etit fichier TXT"
80 D$ = CHR$ (4)
90 PRINT D$"OPEN TYPETXT": PRINT D$"WRIT
E TYPETXT"
```

Cette courte démonstration vous montrera, mieux que de longues explications, comment utiliser *TYPC*.

```
100 PRINT "Ce fichier démo contient un ba
ckspace (" CHR$ (8)") et un Gling ("
CHR$ (7)")"
110 PRINT D$"CLOSE TYPETXT"
120 IF PEEK (768) < > 255 THEN PRINT D$"B
RUN PRO.TYPC": POKE 768,255: REM "pou
r éviter les chargements répétitifs"
130 HOME
140 PRINT D$"TYPC TYPETXT"
150 PRINT : PRINT "Vous auriez pu provoqu
er une pause en frappant une touche
. Une seconde frappe aurait relancé l
e listage."
160 PRINT "Taper ctrl C provoque un arrêt
définitif du listage"
170 PRINT "Cette démo est succincte mais v
ous pouvezTYPCr en 80 colonnes ou sur
imprimante, il vous suffit d'activer
le mode d'affi-chage voulu."
180 PRINT : PRINT "Au revoir." ■
```

```
6037: A9 01 20 F5 BE 90 0E A0 16
6040: B9 E9 60 20 ED FD 88 10 F7 4C 00 BE 85 43 AE 08
6050: BE 8E 3D 61 8D 08 BE AE 07 BE 8E 3C 61 A9 00 85
6060: 3C A9 61 85 3D A9 C1 85 3E A9 61 85 3F D8 A9 00
6070: 85 42 8D 07 BE 38 E9 00 85 02 A5 43 E9 61 85 03
6080: A0 00 B1 3C 20 8E F8 A6 2F E0 02 90 05 C8 20 C2
6090: 60 88 B1 3C 91 42 20 B4 FC CA 10 F6 90 E4 A9 C3
60A0: 85 3E A9 61 85 3F 20 C2 60 A2 01 B1 3C 91 42 20
60B0: B4 FC CA 10 F6 90 EF A9 C7 85 3E A9 61 85 3F 4C
60C0: 2C FE A9 C7 D1 3C C8 A9 61 F1 3C 90 1A 88 A9 00
60D0: D1 3C C8 A9 61 F1 3C B0 0E 88 B1 3C 65 02 91 3C
60E0: C8 B1 3C 65 03 91 3C 88 60 8D 87 A1 C5 C3 C1 CC
60F0: D0 A0 C5 C4 A0 D3 D5 CC D0 A0 D9 A7 CE A0 CC C9
6100: D8 AD 6C BE 85 48 AD 6D BE 85 49 A0 01 B1 48 D9
6110: C3 61 D0 26 C8 C0 04 90 F4 88 8C 52 BE A9 00 8D
6120: 53 BE A9 01 8D 54 BE A9 04 8D 55 BE AD C2 61 8D
6130: 50 BE AD C3 61 8D 51 BE 18 60 38 4C 00 00 4C 09
6140: BE A9 00 A4 74 8D CE BE 8C CF BE A9 C8 20 70 BE
6150: B0 EC AD D0 BE 8D D6 BE 8D DE BE A0 00 8C D9 BE
6160: 8C D7 BE C8 8C DA BE C8 8C D8 BE A9 CA 20 70 BE
6170: 90 0C C9 05 D0 C8 20 8E FD A9 CC 4C 70 BE A0 00
6180: B9 00 02 08 09 80 28 F0 16 C9 8D F0 12 C9 A0 B0
6190: 0E 69 40 48 A9 BC 20 ED FD 68 20 ED FD A9 BE 20
61A0: ED FD AD 00 C0 10 13 8D 10 C0 C9 83 F0 C8 AD 00
61B0: C0 10 FB 8D 10 C0 C9 83 F0 BC C8 CC DB BE D0 C0
61C0: F0 99 41 61 54 59 50 43
```

Conseils aux néophytes :

1. Tapez un *CALL-151* (suivi de RETURN) pour passer en mode Moniteur.
2. Rentrez chaque ligne (toujours suivie de RETURN... pour la valider) telle qu'elle se présente ici.
Si vous entendez un BIP, vous avez tapé un mauvais code.
3. **BSAVE PRO.TYPC, A\$6037, L401** et RETURN .
Vous pourrez sortir du mode moniteur par un CTRL-C. ■

GESTION DE LA DATE

avec le système ProDOS

Le système d'exploitation ProDOS, dernier-né de la gamme des langages systèmes, écrit et mis au point par Apple Computer, permet entre autres améliorations par rapport au DOS 3.3, la gestion de la date. Cette dernière commodité n'est possible qu'en association avec une carte horloge du type Thunderclock. Il est désormais possible de dater ses fichiers soit lors de leurs créations, soit lors de leurs modifications. Il est alors possible de différencier deux versions d'un programme, ce qui est un "must" de ProDOS.

Si l'Apple est doté d'une carte horloge, cette datation se fait automatiquement. Pour l'utilisateur non nanti d'une telle carte interface, rien n'est prévu pour la gestion de la date, les fichiers étant alors datés avec la mention NO DATE (pas de date). Le concepteur a néanmoins remarquablement structuré la page globale du fichier ProDOS (\$BF00-\$BFFF). C'est ainsi qu'aux adresses \$BF90-\$BF91 sont sauvegardés les paramètres de la date, selon le format Année/Mois/Jour (deux bytes).

Structure des bits relatifs à la date (2 bytes) :

```
bits F E D C B A 9 8 7 6 5 4 3 2 1 0
      A A A A A A M M M M J J J J
```

A = bits représentant l'année
M = bits représentant le mois
J = bits représentant le jour

Si une carte horloge est en place, les bytes des adresses \$BF92-\$BF93 seront alors réservés pour le temps (heure/minute).

Structure des bits relatifs au temps (2 bytes) :

```
bits F E D C B A 9 8 7 6 5 4 3 2 1 0
      --- h h h h h --- m m m m m
```

— = bits inutilisés
h = bits représentant les heures
m = bits représentant les minutes

Aux adresses \$BF06-\$BF08 de la page globale de ProDOS, se trouve le vecteur d'appel de la routine de lecture de la date et de l'heure (carte interface présente). Si celle-ci est en place, ce vecteur bien particulier de la page globale aura la structure suivante :

```
BF06- 4C 42 F1 JMP $F142
```

où l'adresse \$F142 est le début de l'implantation de la routine GET TIME (\$82) qui a comme mission d'aller lire la date et l'heure au niveau de la carte Thunderclock. Si la carte horloge est absente, la structure de ce vecteur sera la suivante : BF06- 60 EA EA
où 60 est le code RTS du langage machine — retour au sous-programme appelant (en l'occurrence au langage Basic en général). EA est une instruction muette sans aucune action (EA = instruction NOP).

A propos du programme :

Le programme est assemblé suivant les instructions de l'assembleur BIG-MAC. Le listing est largement commenté pour renseigner sur les phases essentielles du déroulement. Une fois assemblé, le programme pourra être sauvegardé sur une disquette ProDOS, où un bloc sera occupé. Pour le rendre opérationnel, il suffira d'insérer une ligne de commande dans le programme STARTUP (HELLO sous DOS 3.3). Cette ligne de commande, la première dans le listing du programme STARTUP, sera de la forme :

```
10 PRINT CHR$(4) ; "BRUN DATE"
```

où DATE sera le nom du programme "GESTION DE LA DATE ProDOS".

Une fois le programme STARTUP lancé, le sous-programme binaire du nom de DATE sera chargé en mémoire centrale, à l'adresse \$6000, puis exécuté. L'écran affichera la ligne suivante : DATE (JJ/MM/AA) où JJ représente le jour, MM le mois et AA l'année. Le curseur attendra une réponse de l'utilisateur, en prenant soin de taper les valeurs dans l'ordre précité. Il faudra rentrer le / (backlast) après chaque valeur afin de bien les différencier entre elles. Le programme testera la valeur de chaque paramètre suivant une table définie dans le programme source (lignes 184-195).

Appel de ProDOS en langage machine

L'utilisation des routines de ProDOS est relativement simplifiée par rapport au bon vieux DOS 3.3 du fait qu'Apple Computer a prévu une procédure bien particulière. En effet, un point d'entrée unique par un appel au MLI (Machine Language Interface) a été prévu dans la page globale de ProDOS (BF00- 4C B7 BF). Les seules données à fournir sont : un code d'opération et l'adresse de la table des paramètres adaptée à l'opération en cours.

Format standard d'appel du MLI :

```
JSR $BF00
DFB Code Opération
DW Adresse de la table des paramètres
```

POUR SAUVER VOTRE PROGRAMME

Si vous ne possédez pas d'assembleur, contentez-vous de taper les codes (en couleur) des pages 40 à 43, suivant le procédé habituel. Vous sauverez votre programme sur disquette par un :

```
BSAVE DATE, A$6000, L$10B (267 octets)
```

GESTION DE LA DATE PRODOS

(ASSEMBLEUR ProCODE)

COPYRIGHT DÉCEMBRE 1985 — Marcel COTTINI (TREMPLEIN MICRO)

```

0
1
2
3
4 CURSEUR EQU $33 ;PROMPT, RECEPTIONNE LES ORDRES. 603
5 BUFFER EQU $0200 ;TAMPON POUR DONNEES DE LA DATE. 603
6 DATE EQU $BF90 ;ADRESSE GLOBALE PAGE DE PRODOS. 603
7 COUT EQU $FDED ;SOUS-PROGR. SORTIE DE CARACTERES. 603
8 GETLIN EQU $FD6A ;ENTREE DE CARACTERES AU CLAVIER
9 ;TAMPON $200 (IN).
10
11 ORG $6000 603
12 603
13 LDA CURSEUR ;CHARGE LA VALEUR DU PROMPT 603
14 STA CURSEUR1 ;ET LE SAUVEGARDE. 603
15 LDA £$A0 ;CHARGE LA VAL.D'UN BLANC (ESP.) 603
16 STA CURSEUR ;& ANNULE VAL. INIT. DU PROMPT.
17 INIT LDX £00 ;INITIALISE LE REGISTRE X,
18 DATAS LDA CURSEUR2,X ;& CHARGE LES CARACTERES ASCII.
19 BEQ SORTIE ;ENTREE DES CARAC. (BUFFER $0200)
20 JSR COUT ;ET VISUALISATION SUR L'ECRAN.
21 INX ;X+1 & POINTE VERS PROCH. CARAC. 60
22 BNE DATAS ;CHARGE LE PROCHAIN CARACTERE. 60
23 SORTIE JSR GETLIN ;STOCKE LE CARAC. DANS LE BUFFER 60
24 60
25 * FORMATE LE BYTE DE LA DATE 60
26 * ET COMPARE SI SA VALEUR DEPASSE 8 BITS. 60
27 60
28 * BITS: 01234567 -- FORMAT: JJ/MM/AA 60
29 60
30 * JJ = JOUR, CODE SUR 2 BITS
31 * / = SEPARATION, CODE SUR UN BIT
32 * MM = MOIS, CODE SUR 2 BITS
33 * / = SEPARATION CODE SUR 1 BIT 60
34 * AA = ANNEE, CODE SUR 2 BITS 60
35 60
36 DEX ;X-1 (COMPTAB. NOMBRE DE CARAC.) 60
37 CPX £7 ;COMPARE SI LE BIT 7 EST CODE; 60
38 BNE INIT ;DEPASSE. DE VALEUR, A REFAIREE. 60
39
40 * UNIQUEMENT CHIFFRES ET "/" ADMISSIBLES EN ENTREE.
41
42 CHAR LDA BUFFER,X ;CHARGE UN CARACTERE DU BUFFER 60
43 AND £$7F ;ET FORCE LE BIT 7 à 0. 60
44 STA DATE,X ;SAUVEGARDE DU RESULTAT, 60
45 CMP £'/' ;EST-CE LE "/" ? 60
46 BEQ GETLIN2 ;OUI, ON CONTINUE. 60
47 CMP £'0' ;EST-CE UN 0 ? 60
48 60
49 * DETERMINE LA POSITION DU PREMIER CHIFFRE DANS LA
50 * TABLE DES CODES ASCII: LE 0 ETANT LE PREMIER CHIFFRE
51
52 BCC INIT ;NON, ON RECOMMENCE.
53 CMP £': ' ;EST-CE ":" ?

```

```

54
55 * DETERMINE LA POSITION DU DERNIER CHIFFRE DANS LA TABLE
56 * DES CODES ASCII: LE DERNIER CHIFFRE AUTORISE EST LE 9
57
6030: B0 D7 58          BCS INIT          ;NON, ON RECOMMENCE
6032: CA 59 GETLIN1 DEX
6033: 10 E9 60          BPL CHAR
6035: 30 0A 61          BMI LOOP          ;DETERMINE SI LA DATE EST CORRECTE
62
63 * PAS DE DEPASSEMENT DANS LE NOMBRE DE JOURS AUTORISES.
64
6037: E0 02 65 GETLIN2 CPX £2
6039: F0 F7 66          BEQ GETLIN1
603B: E0 05 67          CPX £5
603D: F0 F3 68          BEQ GETLIN1
603F: D0 C8 69          BNE INIT          ;ON RECOMMENCE.
70
71 * UNE ERREUR A ETE DETECTEE DANS LA FORMULATION DE LA DATE.
72
73 * TESTE SI LE FORMAT DE LA DATE EST CORRECT.
74
6041: A2 FF 75 LOOP LDX £$FF          ;INITIALISE LE REGISTRE X,
6043: 20 C0 60 76          JSR FORMAT          ;& FORMATE DONNEES DANS 1 BYTE.
6046: 8D EA 60 77          STA JJ              ;SAUVEGARDE DU PARAMETRE "JOUR"
6049: E8 78          INX              ;PREMIER "/".
604A: 20 C0 60 79          JSR FORMAT          ;INCORPORE DONNEES DANS LE BYTE.
604D: 8D EB 60 80          STA MM              ;SAUVEGARDE DU PARAMETRE "MOIS"
6050: E8 81          INX              ;DEUXIEME "/".
6051: 20 C0 60 82          JSR FORMAT          ;INCORPORE DONNEES DANS LE BYTE.
6054: 8D EC 60 83          STA AA              ;SAUVEGARDE DU PARAMETRE "ANNEE"
84
85 * TESTE LE CHIFFRE DE L'ANNEE (0-99).
86
6057: C9 64 87          CMP £100            ;ANNEE SUPERIEURE à 99 ?
6059: 90 02 88          BCC ANNEE          ;NON, ALORS ON CONTINUE
605B: B0 AC 89          BCS INIT          ;ON RECOMMENCE.
605D: C9 00 90 ANNEE CMP £0              ;ANNEE INFERIEURE à 0 ?
605F: B0 02 91          BCS MOIS1          ;NON, ALORS ON CONTINUE.
6061: 90 A6 92          BCC INIT          ;OUI, ON RECOMMENCE.
93
94 * TESTE SI LE CHIFFRE ATTRIBUE AU MOIS EST CORRECT (1-12)
95
6063: AD EB 60 96 MOIS1 LDA MM              ;CHARGE LE CHIFFRE DU MOIS,
6066: C9 01 97          CMP £1              ;EST-IL SUPERIEUR à 0 ?
6068: B0 02 98          BCS MOIS2          ;OUI, ALORS ON CONTINUE.
606A: 90 9D 99          BCC INIT          ;NON, ON RECOMMENCE.
606C: C9 0D 100 MOIS2 CMP £13            ;EST-IL SUPERIEUR à 12 ?
606E: 90 02 101          BCC JOUR1          ;NON, ALORS ON CONTINUE.
6070: B0 97 102          BCS INIT          ;OUI, ON RECOMMENCE.
103
104 * TESTE SI LE CHIFFRE ATTRIBUE AU JOUR EST CORRECT:
105 * DETERMINE PAR LA TABLE DES JOURS LE MOIS CONSIDERE.
106
6072: AA 107 JOUR1 TAX

```

(Suite page 42)

```

6073: CA          108      DEX
6074: AD EA 60    109      LDA JJ          ;CHARGE LA VALEUR DU JOUR,
6077: C9 01       110      CMP £1         ;EST-IL SUPERIEUR à 0 ?
6079: B0 02       111      BCS JOUR2     ;OUI, ALORS ON CONTINUE.
607B: 90 8C       112      BCC INIT      ;NON, ON RECOMMENCE.
607D: DD DE 60    113      CMP TABLE,X ;TEST DU "JOUR" PAR RAPPORT
                                     ;AU MOIS INDIQUE.
6080: 90 05       115      BCC PLACE1    ;PLACE LA VALEUR DANS LA PAGE
                                     ;GLOBALE DE PRODOS.
6082: F0 03       117      BEQ PLACE1
6084: 4C 09 60    118      JMP INIT      ;ERREUR, ON RECOMMENCE.
                                     119
120 * PLACE LES VALEURS "DATE" DANS LA PAGE GLOBALE DE PRODOS,
121 * SUR 2 BYTES, AUX ADRESSES $BF90,$BF91.
122 * FORMAT DES 2 BYTES :
123 * BITS: FEDCBA9876543210
124 *      AAAAAAAMMMJJJJJ  ANNEE/MOIS/JOUR
125
126 * $BF91 ==> PLACE1      $BF90 ==> PLACE2
127 * BITS: FEDCBA98      BITS: 76543210
128 *      AAAAAAAM      MMMJJJJJ
129
6087: AD EC 60    130      PLACE1 LDA AA          ;CHARGE LA VALEUR DE L'ANNEE,
608A: 0A          131      ASL          ;& DECALE D'UN BIT VERS LA GAUCHE.
608B: 8D EE 60    132      STA SAUV1    ;SAUVEGARDE DU RESULTAT.
608E: AD EB 60    133      LDA MM          ;CHARGE LA VALEUR DU MOIS,
6091: 0A          134      ASL          ;DECALE CINQ FOIS VERS LA GAUCHE,
6092: 0A          135      ASL
6093: 0A          136      ASL
6094: 0A          137      ASL
6095: 0A          138      ASL
6096: 8D ED 60    139      STA SAUV3    ;ET SAUVEGARDE DU RESULTAT.
6099: 90 09       140      BCC PLACE2
609B: 18          141      CLC
609C: AD EE 60    142      LDA SAUV1    ;CHARGE LA VALEUR DU BYTE,
609F: 69 01       143      ADC £1       ;ET LUI ADDITIONNE 1 (CORRECTION),
60A1: 8D EE 60    144      STA SAUV1    ;PUIS SAUVEGARDE DU RESULTAT.
60A4: 18          145      PLACE2 CLC      ;ANNULE LA RETENUE.
60A5: AD ED 60    146      LDA SAUV3    ;CHARGE LE RESUL. DU BYTE DU MOIS,
60A8: 6D EA 60    147      ADC JJ       ;LUI INCORP. LA VALEUR DE "JOUR",
60AB: 8D EF 60    148      STA SAUV2    ;ET SAUVEGARDE DU RESULTAT.
                                     149
150 * POSITIONNE LES BITS DANS LE BON SENS:
151 * LBYTE, HBYTE = $BF90, $BF91  MM/JJ, ENSUITE AA/MM
152
60AE: AD EF 60    153      LDA SAUV2    ;CHARGE LA VALEUR DE LBYTE MM/JJ,
60B1: 8D 90 BF    154      STA DATE     ;ET SAUVEGARDE DANS LA PAGE
                                     ;GLOBALE DE PRODOS.
60B4: AD EE 60    156      LDA SAUV1    ;CHARGE LA VALEUR DE HBYTE AA/MM
60B7: 8D 91 BF    157      STA DATE+1   ;ET SAUVEGARDE à L'ADRESSE $BF91.
60BA: AD F0 60    158      LDA CURSEUR1 ;CHARGE VALEUR INITIALE DU PROMPT,
60BD: 85 33       159      STA CURSEUR  ;ET LA SAUVEGARDE à L'ADRESSE $33.
60BF: 60         160      RTS         ;RETOUR AU SOUS-PROG. APPELANT.
                                     161

```

```

162 * FORMATE LA DATE. INCORPORE LES VALEURS DANS UN BYTE.
163
60C0: E8 164 FORMAT INX
60C1: BD 90 BF 165 LDA DATE,X
60C4: 29 0F 166 AND £$0F
60C6: 0A 167 ASL ;MULTIPLIE LE RESULTAT PAR 2.
60C7: 8D ED 60 168 STA SAUV3
60CA: 0A 169 ASL ;MULTIPLIE PAR 2 LE RESULTAT.
60CB: 0A 170 ASL ;MULTIPLIE PAR 2 LE RESULTAT.
60CC: 18 171 CLC ;ANNULE LA RETENUE.
60CD: 6D ED 60 172 ADC SAUV3
60D0: 8D ED 60 173 STA SAUV3 ;SAUVEGARDE DU RESULTAT.
60D3: E8 174 INX
60D4: BD 90 BF 175 LDA DATE,X
60D7: 29 0F 176 AND £$0F
60D9: 18 177 CLC
60DA: 6D ED 60 178 ADC SAUV3
60DD: 60 179 RTS
180
181 * TABLE DES JOURS/MOIS
182 * FIXE LE NOMBRE DE JOURS, D'APRES LE MOIS CONSIDERE.
183
60DE: 1F 184 TABLE DFB 31 ;JANVIER
60DF: 1D 185 DFB 29 ;FEVRIER
60E0: 1F 186 DFB 31 ;MARS
60E1: 1E 187 DFB 30 ;AVRIL
60E2: 1F 188 DFB 31 ;MAI
60E3: 1E 189 DFB 30 ;JUIN
60E4: 1F 190 DFB 31 ;JUILLET
60E5: 1F 191 DFB 31 ;AOUT
60E6: 1E 192 DFB 30 ;SEPTEMBRE
60E7: 1F 193 DFB 31 ;OCTOBRE
60E8: 1E 194 DFB 30 ;NOVEMBRE
60E9: 1F 195 DFB 31 ;DECEMBRE
196
60EA: 00 197 JJ HEX 00 ;JOUR ==> JJ
60EB: 00 198 MM HEX 00 ;MOIS ==> MM
60EC: 00 199 AA HEX 00 ;ANNEE ==> AA
200
60ED: 00 201 SAUV3 HEX 00
60EE: 00 202 SAUV1 HEX 00 ;FORMAT : AA+MM = AA/MM
60EF: 00 203 SAUV2 HEX 00 ;FORMAT : MM+JJ = MM/JJ
204
60F0: 00 205 CURSEUR1 HEX 00 ;SAUVAGARDE DU PROMPT.
60F1: C4 C1 D4
C5 A0 A0
A8 CA CA
AF CD CD
AF C1 C1
A9 A0
6102: 00 206 CURSEUR2 ASC "DATE (JJ/MM/AA) ";FORMAT AFFICHE DE DATE.
6103: 4A 4A 3A 207 HEX 00 ;FIN DE LA TABLE ASCII
4D 4D 3A
41 41 208 DATE1 ASC 'JJ:MM:AA' ;SAUVEGARDE DU FORMAT.

```

CHASSEURS

de bogues et d'octets



Carac.charg

• Quand on utilise cette routine sous ProDOS (T.M n°6, page.31), il convient de modifier les adresses \$AA60-AA61 (longueur du dernier fichier binaire chargé par le DOS). Sous ProDOS, elles deviennent \$BEC8 et \$BEC9. On aura donc :

308 : AD C8 BE et

30D : AD C9 BE

Claude JUPPÉ.

AUTO.COÛT

• Le programme **Auto.Coût** (T.M n°7, page 47) fonctionne bien mais, sous ProDOS, le démarrage est laborieux ! Compléter la ligne 120 par :

: PRINT D\$"CLOSE"

A la ligne 505, remplacer 390 par 395.
Ne pas oublier qu'il faut rentrer une dépense avant de passer à la mémorisation. Il semble souhaitable d'éliminer la ligne 765!

Tremplin Micro.

```

1 *****
2 * SONDAGE SUR 8 BITS *
3 * Le nombre de femmes *
4 *****
5
6 PTR = $6
7 COUNT = $8
8
9 ORG $300
10
11 LDY $0
12 STY PTR
13 STY COUNT
14 STY COUNT+1
15 LDA $>$2000
16 STA PTR+1
17
18 LOOP LDA (PTR),Y
19 BEQ FINI ; Fin de table
20 BMI PLUS ; C'est un homme
21 AND $%01000000+$%00000010
22 BEQ PLUS
23 INC COUNT
24 BNE PLUS
25 INC COUNT+1
26 PLUS INY
27 BNE LOOP
28 INC PTR+1
29 BNE LOOP ; Toujours
30
31 FINI RTS
0300: A0 00
0302: 84 06
0304: 84 08
0306: 84 09
0308: A9 20
030A: 85 07
030C: B1 06
030E: F0 13
0310: 30 0A
0312: 29 42
0314: F0 06
0316: E6 08
0318: D0 02
031A: E6 09
031C: C8
031D: D0 ED
031F: E6 07
0321: D0 E9
0323: 60

```

Combien de femmes ?

• Souvenez-vous du programme **SONDAGE SUR 8 BITS** de T.M. n°7 (page 43) et comparez la solution d'Yvan KOENIG à celle (plus longue) que nous avons publiée.

Il s'agit bien, ici, de **chasse aux octets** superflus.

Avouez que notre ami Yvan KOENIG mérite que son **AMPERELO** bénéficie d'une bonne diffusion !



Peut-on raccourcir cette nouvelle version ?
Là est la question !

VILLES DE FRANCE

Je me suis amusé à taper le programme de M. DEVAUX "Villes de France" paru dans votre n°7. Hélas ! quand j'ai voulu le faire fonctionner sur mon "IIC", celui-ci n'a rien voulu savoir et semblait contrarié par la ligne 510. Je suis novice en matière de programmation, préférant habituellement utiliser des programmes tout faits. Obstiné, j'ai cherché à comprendre ce qui coïncit ; après quelques essais infructueux, j'ai réussi à écrire une ligne 510 qui semble satisfaire mon complice "IIC". Je ne sais pas si c'est l'unique et bonne solution ; en tout cas ça marche. Alors je vous livre le fruit de mes cogitations...

510 PRINT : INPUT " Nom du joueur ? : "; N\$(I)

Enfin nos chers z'enfants vont pouvoir se distraire intelligemment sans faire la guéguerre aux "Martiens" et autres "Lunatiques" et "Drôlatiques", surtout si, sur la même disquette, on enregistre "Fleuves de France" paru dans "POM'S" n°15.

De plus "Villes de France" est intransigeant avec "l'ortographe". De quoi "inculper" une bonne instruction à nos rejetons.

Patrice SILBON.

Pas d'erreur à la ligne 510, mais une confusion (excusable) entre n° " et n" " ... surtout quand on dispose d'un exemplaire un peu trop encré. Naturellement, le Basic n'accepte pas ce genre de fantaisie... et il est bon de le savoir.

Le ton ironique de votre lettre a excité ma curiosité et j'ai relu la liste des villes de France. Je me suis arrêté sur CHALON. Monsieur Devaux a confondu CHALON-SUR-SAÔNE et CHÂLONS-SUR-MARNE. Belle occasion, pour le maître que vous êtes peut-être, de faire une leçon à vos élèves ! Une leçon de géographie et d'orthographe. Hélas ! nul n'est infallible !

DATA.GENERATOR

Dans ce programme (T.M. n°7), les lignes de commandes se mélangent à la ligne de DATA. Ce n'est pas bien grave, mais esthétiquement gênant. Voici ma solution :

730 CALL — 958 : PRINT... etc.

François LHOSTE.

Je vous signale une petite erreur dans le programme DATA.GENERATOR. Il faut remplacer, à la ligne 545, PRINT L par PRINT I, si l'on veut que ce soit effectivement le nombre de données qui s'inscrive en haut et à droite de l'écran. En même temps que ces corrections, j'en ai profité pour ajouter l'affichage du nombre total des DATAs, vous trouverez ci-joint le listage des lignes corrigées.

Guy CALLO.

La ligne 545 n'affiche pas le nombre de données, mais la longueur de la ligne en cours. Par contre, voici les lignes complémentaires que vous proposez :

105 TT = 0 : DIM I(50)

245 IF R% = 19 THEN I(A) = I : TT = TT — 1 + I(A) : GOTO 515

525 IF L1 + L > 239 THEN I(A) = I : TT = TT — 1 + I(A) : L\$(A) = LEFT\$(L\$(A), L — 1) : S = 1 : I = 0 : GOTO 155

545 VTAB 1: HTAB 37: CALL — 868: INVERSE : PRINT I: NORMAL

TM

547 VTAB 2: HTAB 20: PRINT "Total des DATA = " : PRINT " " : HTAB 37: INVERSE : PRINT TT + I: NORMAL

TRAJET MINUTÉ

Une ou deux remarques pour ceux qui se sont intéressés à ce programme (T.M. n°6).

— L'arrondi de la ligne 440 mériterait d'être ajusté à la valeur la plus voisine en écrivant :

INT(VAL(M\$) * 100/60 + .5)

— Essayez de taper, comme durée, 1.0

1.005...

Le résultat est curieux.

Tout cela rentre dans l'ordre avec une ligne 389

IF C\$ = "." AND LT = 1 OR LT > 1 + 3 THEN 280

— Enfin le "PRINT-USING" de la ligne 520 ne prend pas les nombres négatifs. Ce n'est sans doute pas un grand malheur ! Mais plus généralement, on peut déterminer la tabulation à la colonne H par :

HTAB H — FN PO(X)

(ou ? TAB())

où FN PO(X) est au préalable défini par :

DEF FN PO(X) = LEN(STR\$(INT(X))) (a)

— (INT(X) = — 1) (b)

— (INT(X) = 0 AND X < > 0) (c)

La première ligne (a) positionne la partie entière ; (b) corrige pour les nombres négatifs décimaux ; (c) corrige pour les nombres positifs décimaux. Jean GOUTELLE.

TM

Voici une application simple :

10 TEXT : HOME : H = 20

11 DEF FN PO(X) = LEN (STR\$ (INT (X))) - (INT (X) = - 1) - (INT (X) = 0 AND X < > 0)

12 FOR I = 1 TO 10 : READ C : HTAB H - FN PO(C) : PRINT C : NEXT

13 DATA 10,0.10,12.40,20,123,12356,12.50,14.80,200,123456,.0012

ÉCRANS 80 COLONNES

Suite à votre article "créer des écrans sur 80 colonnes", paru dans le numéro 3 de T.M. et ayant utilisé récemment dans un logiciel des écrans créés grâce à votre utilitaire, nous avons constaté un petit problème concernant la sortie sur Imagewriter (I ou II) à partir d'un Apple IIc. Alors que tout fonctionne avant le chargement d'un écran 80 colonnes, notre imprimante refuse d'imprimer quoi que ce soit après ledit chargement. Vous pouvez d'ailleurs essayer de charger un écran à partir de votre utilitaire, puis interrompre ce dernier et faire l'habituel PR&1. Le résultat est surprenant. Nous avons d'ailleurs eu le même problème sur deux Apple IIc différents : cela ne provient donc pas d'une défaillance ou des connexions. Notez que cet ennui ne se pose pas avec un IIe (équipé d'un 6502). Cela proviendrait-il de la gestion mem.principale/mem.auxiliaire du 65C02, vu que l'écran 80 colonnes se décompose en deux parties ? Jean-Luc M et Lionel Z.

TM

Pour revenir en 80 colonnes, utilisez l'une de ces deux versions :

— DOS 3.3 POKE 54,7 : POKE 55,195 : CALL 1002
— ProDOS PRINT CHR\$(4) ; "PR&A\$C307"

Le problème que vous évoquez n'existe pas sur le IIe avec 65C02. Si l'un de nos lecteurs découvre la solution...

QUESTIONS

POKE 243,32

Q : Existe-t-il un moyen simple pour afficher en minuscules un texte qui est normalement en majuscules ?

R : Essayez POKE 243,32 (\$F3). Cette adresse est normalement à 0 quand votre Apple est en mode NORMAL (ou INVERSE) et à \$40 (64) lorsqu'il est en mode FLASH.

Si vous tapez LIST après POKE 243,32, toutes les majuscules deviendront des minuscules. POKE 243,0 rétablira un listage normal. Notez que cette opération ne modifie pas le programme en mémoire. Ça fonctionne aussi sur imprimante. Vous voulez en savoir plus ? Comparez le A au a... en notation binaire (01000001 — 01100001) et vous constaterez que les deux caractères ne diffèrent que par le bit 5. Celui-ci est à 1 pour les minuscules et à 0 pour les capitales, d'où l'utilité du masque \$20 (32) : 00100000 qui, utilisé avec ORA forcera le bit 5 à 1. CQFD.

Précision : avec les caractères accentués, la transformation n'est pas triste !

MID\$ ABRÉGÉ

Q : Est-on obligé d'utiliser deux "pointeurs" dans la commande MID\$?

R : Pas obligatoirement. Ainsi, dans les exemples suivants :

§D\$="ABCDEFGH IJ"	MID\$ joue un
§PRINT MID\$(D\$,5)	rôle voisin de
EFGHIJ	celui de RIGHT\$.
§PRINT MID\$(D\$,7)	Si on ne lui indi-
GHIJ	que pas le nombre
§PRINT MID\$(D\$,1)	de caractères à
ABCDEFGHIJ	prendre dans la chaîne,

l'Applesoft en déduit que l'on désire la fin de la chaîne, à partir du caractère de rang N.

RESET-BOOT

Q : Comment faire pour que chaque CTRL-RESET relance inmanquablement le système ?

R : Les vecteurs de RESET sont bien connus : \$3F2.3F4 (1010-1012). Les valeurs norma-

les, lues à ces adresses sont \$BF-9D-38 (1010-1011-1012).

Un simple POKE 1012,0 aura l'effet souhaité !

OÙ EST LE CURSEUR ?

Q : Comment connaître la position du curseur ? J'ai essayé avec $P = POS(X)$, mais je ne sais pas interpréter le résultat. EXEMPLE : PRINT "TOTO" : $P = POS(X)$: PRINT P... ce résultat est zéro... or la case suivant TOTO est bien en HTAB5 ?

R : Vous n'avez pas mis de point-virgule à la suite de TOTO... donc votre curseur passe à la ligne suivante, en HTAB1, ce qui correspond à POKE 36,0. Essayez les quelques lignes ci-après et n'oubliez pas qu'il faut ajouter 1 au résultat pour le traduire en HTAB. Il en irait de même verticalement : $V = PEEK(37)$ et $VTAB = PEEK(37) + 1$

```
10 HOME
20 VTAB 6: PRINT "TOTO";
30 GOSUB 80
40 VTAB 10: FOR I = 1 TO 45: PRINT "E";: NEXT
50 GOSUB 80
70 END
80 H = PEEK (36): REM H=PEEK(1403) EN 80 COLONNES
90 PRINT: PRINT H: RETURN
```

TOUCHE... À MON CLAVIER !

Q : Quelle instruction utiliser pour savoir quelle touche du clavier a été pressée ?

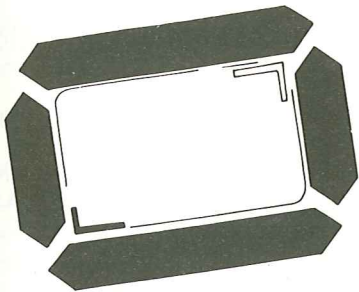
R : L'instruction GET R\$ vous permet de connaître la valeur ASCII de la touche pressée (il suffit de la compléter par $V = ASC(R\$)$). En langage machine, LDA \$C000 joue un rôle identique... ou PEEK(49152)... ou PEEK(—16384) en Basic. Ainsi avec :

```
300: AD 00 C0 LDA $C000
303: 85 06 STA $06
305: 60
```

```
10 GET R$: CALL 768: PRINT PEEK(6): PRINT PEEK(49152): PRINT ASC(R$)
```

RUN suivi d'une action sur la touche A... affichera trois fois la valeur 65.

REponses



Yvan KOENIG
a planché sur...

UN INPUT en mode direct

- On sait qu'un POKE 118,0 autorise un INPUT EN MODE DIRECT (voir le courrier du n°7 de T.M., page 56). Toutefois — nous l'avons précisé — la longueur du mot saisi est limitée à 7 caractères s'il s'agit d'une variable non indiquée... et à 12 (parfois 13) dans l'autre cas.

EXEMPLES : POKE 118,0 : INPUT M\$
? ABCDEFG (7 caractères)... OK
POKE 118,0 : INPUT M\$
? ABCDEFGH (8 caractères)...
SYNTAX ERROR IN 10

POKE 118,0 : FOR I = 1 TO 3 : INPUT
M\$(I) : NEXT
? ABCDEFGHIJKL
? ABCDEFGHIJKLM
? SYNTAX ERROR IN 10276

NOTRE AMI YVAN KOENIG — dont on connaît la perspicacité — a planché sur le problème et voici son intéressante explication :

Votre question concernant la limitation à 13 caractères de l'INPUT direct m'a provoqué. Je pense que ces quelques lignes seront suffisamment explicites.

Lorsque vous tapez POKE 118,0 : INPUT X\$ vous rangez des octets dans le buffer d'entrée, en l'occurrence :

\$200 : 50 4F 4B 45 31 31 38 2C 30 3A 49 4E 50 55 54 58 24

Lorsque vous tapez RETURN un 00 est déposé après le 24 (\$) et la main est donnée à l'APPLESOFT. Celui-ci se livre alors à son occupation favorite : la "tokenisation". Le mot POKE est remplacé par le token \$B9 tandis qu'un INPUT est remplacé par \$84. Enfin on place deux octets nuls simulant une fin de programme juste après le code 24. Attention ces 2 octets nuls ne sont pas consécutifs mais de rang 0 et 2.

Le buffer devient donc :

\$200 : B9 31 31 38 2C 30 3A 84 58 24 00 4E 00 55 54 58 24

Je suis sûr que vous avez compris, le second octet nul a la position 13 dans le buffer. Si votre input dépasse 13 caractères, APPLESOFT ne va pas retrouver son octet fin de programme qui aura été remplacé par votre 13^e caractère et il vous le fera savoir brutalement.

Si vous tenez à faire un input plus long, au lieu d'un seul (:) entre l'instruction POKE et l'instruction INPUT, mettez en vingt et un et vous pourrez entrer $12 + 20 = 32$ caractères.

C'était simple. En fait il m'a fallu une heure pour trouver.

Yvan KOENIG

Moralité : Pour un INPUT DIRECT de 15 caractères, taper 8 (15-7) fois : (plus les : habituels) entre POKE 118,0 et INPUT, comme ceci :

POKE 118,0 :::::::::: INPUT M\$

Des pommes par milliers



REMPLISSEZ vos écrans de pommes... ou de sabliers, bref de chacun des caractères "souris" que vous propose la nouvelle ROM de l'Apple. Pour comprendre parfaitement comment cela marche, recopiez d'abord ce petit programme de démonstration, mais sachez que la routine d'affichage ne comporte que les 16 misérables octets de la ligne 190 (ou, en 80 colonnes, les 19 premières données de la ligne 235). Pour le reste, lisez les explications... et regardez votre écran !

```

100 GOSUB 190
105 NORMAL : TEXT : PRINT CHR$(4)"PR&3": PRINT CHR$(17)
)
110 INVERSE : PRINT CHR$(27): VTAB 1: CALL 768: NORMAL
: GOSUB 205: PRINT : HOME
115 IF PEEK ( - 16384) < > 27 THEN X = X + 1: IF X < 32
THEN POKE 769, PEEK (769) + 1: GOTO 110
120 PRINT CHR$(24)
125 HOME : CALL 784
130 ES = ES + 1: ON ES GOTO 135,140,150,155
135 AF$ = "$FDED (COUT)":P = 237: GOTO 145
140 AF$ = "$FDF0 (COUT1)":P = 240: GOTO 145
145 PRINT : PRINT : PRINT "FAITES UN ESSAI AVEC "AF$: GO
SUB 210:X = 0: POKE 769,64: POKE 771,P: GOSUB 205: P
RINT : GOTO 105
150 PRINT : PRINT : PRINT "REVENONS A $FDF6 (COUT2)": GO
SUB 210:X = 0: POKE 769,64: POKE 771,246: GOSUB 205:
PRINT : GOTO 105
155 PRINT CHR$(21): GOSUB 205
160 PRINT : PRINT CHR$(4)"PR&3": PRINT : VTAB 24: PRINT
"<1> ENCORE <2> 80 COLONNES <3> MENU DE DISQUETTE <
4> FIN ";; CALL - 198: GET R$: PRINT : HOME
165 IF R$ = CHR$(49) THEN RUN
170 IF R$ = CHR$(50) THEN 215
175 IF R$ = CHR$(51) THEN PRINT CHR$(4)"RUN MENU"
180 IF R$ < > CHR$(52) THEN 160
185 END
190 POKE 768,169: POKE 769,64: POKE 770,32: POKE 771,246
: POKE 772,253: POKE 773,174: POKE 774,246: POKE 775
,7: POKE 776,224: POKE 777,160: POKE 778,240: POKE 7
79,246: POKE 780,141: POKE 781,247: POKE 782,7: POKE
783,96

```

100 : Initialise le système (installation des deux petites routines LM).

105 : PRINT CHR\$(17) ramène en 40 colonnes sans désactiver la carte 80 colonnes.

115 : Cette ligne permet de modifier l'adresse \$301... qui contient le caractère souris. On peut ainsi passer en revue tous les caractères.

125 : A l'adresse 784. (\$310), est installée une courte routine permettant d'afficher le désassemblage de A instructions (ligne de pokes 195).

155 : PRINT CHR\$(21) désactive la carte 80 colonnes.

Vous constaterez que \$FDF6 est beaucoup plus rapide que \$FDED ou \$FDF0 et en tirerez sans doute une leçon !

195 : Routine de désassemblage.

205 : Bell et attente.

210 : Si l'on utilise ESCAPE entre deux séquences, on passe immédiatement à la suite du programme.

215 à 235 : Affichage sur 80 colonnes.

```
195 POKE 784,169: POKE 785,0: POKE 786,133: POKE 787,58:
    POKE 788,169: POKE 789,3: POKE 790,133: POKE 791,59
    : POKE 792,169: POKE 793,7: POKE 794,32: POKE 795,99
    : POKE 796,254: POKE 797,96
```

```
200 RETURN
```

```
205 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: PO
    KE - 16368,0: RETURN
```

```
210 VTAB 22: PRINT "ESCAPE POUR BRULER LES ETAPES": RETU
    RN
```

```
215 FOR I = 768 TO 800: READ R: POKE I,R: NEXT :X = 0
```

```
220 INVERSE : PRINT CHR$(27): VTAB 1: CALL 768: NORMAL
    : GOSUB 205: PRINT : HOME
```

```
225 IF PEEK ( - 16384) < > 27 THEN X = X + 1: IF X < 32
    THEN POKE 769, PEEK (769) + 1: GOTO 220
```

```
230 PRINT CHR$(24): POKE 796,8: CALL 787: GOSUB 205: GO
    TO 160
```

```
235 DATA 169,64,32,246,253,174,246,7,224,160,240,246,32,
    246,253,141,247,7,96,169,0,133,58,169,3,133,59,169,8
    ,32,99,254,96
```

AFFICHAGE SUR 40 COLONNES

```
0300- A9 40
0302- 20 F6 FD
0305- AE F6 07
0308- E0 A0
030A- F0 F6
030C- 8D F7 07
030F- 60
```

```
LDA £$40
JSR $FDF6
LDX $07F6
CPX £$A0
BEQ $0302
STA $07F7
RTS
```

Le caractère désiré est placé dans l'accumulateur
Affichage du caractère par COUTZ
Lecture de la case-écran \$7F6 (2038)
Est-ce un espace ?
Si oui, elle est encore vierge : un autre tour !
Ça, c'est un POKE 2039, caractère

```
0310- A9 00
0312- 85 3A
0314- A9 03
0316- 85 3B
0318- A9 07
031A- 20 63 FE
031D- 60
```

```
LDA £$00
STA $3A
LDA £$03
STA $3B
LDA £$07
JSR $FE63
RTS
```

Routine classique de désassemblage.
L'adresse de départ est écrite dans \$3A-3B
(PCL et PCB), puis A est chargé avec le nombre
d'instructions à désassembler avant d'appeler
\$FE63 (LIST2).

AFFICHAGE SUR 80 COLONNES

```
0300- A9 40
0302- 20 F6 FD
0305- AE F6 07
0308- E0 A0
030A- F0 F6
030C- 20 F6 FD
030F- 8D F7 07
0312- 60
```

```
LDA £$40
JSR $FDF6
LDX $07F6
CPX £$A0
BEQ $0302
JSR $FDF6
STA $07F7
RTS
```

Une seule différence : en \$30C, on affiche un
caractère de plus (celui qui est pris en compte
par la mémoire auxiliaire). Par contre, comme
en 40 colonnes, le dernier caractère de la ligne
est "poqué", pour éviter le déroulement des
lignes d'écran vers le haut.

CARACTÈRES SOURIS

Rappelons que, pour afficher les caractères souris, il faut activer la carte 80 colonnes, puis utiliser une séquence du genre
INVERSE : PRINT CHR\$(27). On retourne à l'affichage traditionnel par **NORMAL : PRINT CHR\$(24)**.

Cercle... économique

NOUS avons publié, dans le numéro 3 de *TREMPIN MICRO*, page 47, deux routines permettant de tracer un cercle. La première — classique — était lente, mais la seconde plus rapide... du moins en apparence. Nous la reprenons ici... en vue de l'améliorer. Pourquoi en apparence ? Tout simplement parce qu'elle commence par calculer et mémoriser les paramètres de la circonférence. Dans un second temps, à partir de ces points connus, elle travaille évidemment à vitesse grand V. Il convient de noter qu'un nouveau calcul est inutile si l'on désire effacer, puis redessiner le même cercle... d'où son principal intérêt. Mais elle présente un gros défaut : elle se révèle fort gourmande en mémoire. Pour un tracé grossier, 2×90 variables suffisent, mais c'est déjà beaucoup. Or, il existe une solution à ce problème, nullement pénalisante sur le plan de la rapidité, mais qui permet de diviser par deux le nombre des points mémorisés : NE TRAVAILLER que sur une demi-circonférence. Voyez plutôt, braves gens, ce que donnent ces élucubrations !

```

100 TEXT : PRINT CHR$(4)"PR&3": PRINT : HOME
110 XC = 140:YC = 94: REM Positions Hor & Vert du centre
120 R = 90: REM Rayon du cercle
130 ND = 180: REM Nombre de points
140 DIM SI(ND / 2),CO(ND / 2)
150 PB = 3.1415926 * 2:PN = PB / ND
160 FOR I = 1 TO ND / 2
170 A = PN * I
180 CO(I) = COS (A) * R:SI(I) = SIN (A) * R
190 NEXT
200 HGR : POKE - 16302,0: HCOLOR= 3
210 CALL - 198
220 HPLOT XC + R,YC
230 FOR I = 1 TO ND / 2
240 HPLOT TO XC + CO(I),YC + SI(I)
250 NEXT
260 FOR I = 1 TO ND / 2
270 HPLOT TO XC - CO(I),YC - SI(I)
280 NEXT
290 CALL - 198: GET R$: PRINT : HOME : POKE - 16301,0
300 VTAB 22: HTAB 18: PRINT "<1> ENCORE <2> BASIC <3> ME
    NU DE DISQUETTE -> ";: CALL - 198: GET R$: PRINT : T
    EXT : HOME
310 ON VAL (R$) GOTO 320,330,340: GOTO 300
320 GOTO 200
330 LIST 100,280: END
340 PRINT CHR$(4)"RUN MENU"

```

Notez que ND (ligne 30) peut être réduit à 90... voir à 60 ! Faites des essais.

Un GOTO 200 vous permettra aussi de retracer rapidement votre cercle... économique.

QUESTION

Est-il possible de faire mieux ?

A vos claviers !

Transformation d'une fonte

IMPRIMANTE en fonte GRAPHIQUE

Le numéro 6 de *TREMLIN MICRO* vous a donné le moyen de transformer des fontes de caractères graphiques en polices destinées à votre imprimante, mais il est possible de réussir l'opération inverse... par le truchement de la petite routine contenue dans les lignes 185 et 190 du programme ci-contre.

Après utilisation de *CARAC.FTOOL*, il vous suffira d'appeler *CARAC.VISU* (*T.M. n°6*) pour visualiser votre fonte... transformée.

Attention !
il est indispensable qu'elle soit classique et exactement structurée comme les fontes obtenues à partir de la disquette *TOOLKIT APPLE*.

CARAC.FTOOL

```

100 CLEAR : TEXT : PRINT CHR$ (21): HOME
105 PRINT "FONTE POUR IMPRIMANTE = FONTE GRAPHIQUE -----
-----"
110 VTAB 4: PRINT "ATTENTION! il faut utiliser une polic
e classique (structure TREMLIN MICRO 6)"
115 VTAB 7: PRINT : CALL - 958: INPUT "TITRE (CAT=?) ->
";T$: IF T$ = "?" THEN PRINT CHR$ (4)"CATALOG": GOTO
100
120 ONERR GOTO 150
125 IF T$ = "" THEN 150
130 PRINT CHR$ (4)"BLOAD "T$",A$4000"
135 GOSUB 180
140 CALL 768:L = ( PEEK (8) + PEEK (9) * 256) - 8191
145 PRINT : PRINT "FONTE GRAPHIQUE SAUVEE:";T$ = T$ + ".
G,A$2000,L" + STR$ (L): PRINT : PRINT " - "T$: PRINT
CHR$ (4)"BSAVE" T$
150 VTAB 19: PRINT : PRINT "<1> CARAC.VISU (TM6) <2> AUT
RE FONTE": PRINT "<3> MENU DE DISQUETTE <4> TERMINE
-> ";; GET R$
155 PRINT : IF R$ = "1" THEN PRINT CHR$ (4)"RUN CARAC.VI
SU"
160 IF R$ = "2" THEN 115
165 IF R$ = "3" THEN PRINT CHR$ (4)"RUN MENU"
170 IF R$ < > "4" THEN 150
175 HOME : END

180 RESTORE : FOR I = 768 TO 855: READ A: POKE I,A: NEXT
: RETURN
185 DATA 169,63,133,7,74,133,9,169,247,133,8,169,250,133
,6,165,6,24,105,10,133,6,144,2,230,7,160,0,177,6,201
,32,176,1,96,201,128,176,251,165,8,24,105,8,133,8,14
4,2
190 DATA 230,9,152,153,248,31,200,192,8,208,248,200,162,
8,177,6,10,62,247,31,202,208,249,136,192,1,208,240,1
85,247,31,145,8,200,192,9,208,246,240,183

```

Vous trouverez au verso le désassemblage commenté de ces deux dernières lignes.

CARAC.FTOOL

LA POLICE DE CARACTÈRES IMPRIMANTE est impérativement chargée à partir de l'adresse \$4000.

0300-	A9 3F	LDA	£\$3F
0302-	85 07	STA	\$07
0304-	4A	LSR	
0305-	85 09	STA	\$09
0307-	A9 F7	LDA	£\$F7
0309-	85 08	STA	\$08
030B-	A9 FA	LDA	£\$FA
030D-	85 06	STA	\$06
030F-	A5 06	LDA	\$06
0311-	18	CLC	
0312-	69 0A	ADC	£\$0A
0314-	85 06	STA	\$06
0316-	90 02	BCC	\$031A
0318-	E6 07	INC	\$07
031A-	A0 00	LDY	£\$00
031C-	B1 06	LDA	(£06),Y
031E-	C9 20	CMP	£\$20
0320-	B0 01	BCS	\$0323
0322-	60	RTS	
0323-	C9 80	CMP	£\$80
0325-	B0 FB	BCS	\$0322
0327-	A5 08	LDA	\$08
0329-	18	CLC	
032A-	69 08	ADC	£\$08
032C-	85 08	STA	\$08
032E-	90 02	BCC	\$0332
0330-	E6 09	INC	\$09
0332-	98	TYA	
0333-	99 F8 1F	STA	\$1FF8,Y
0336-	C8	INY	
0337-	C0 08	CPY	£\$08
0339-	D0 F8	BNE	\$0333
033B-	C8	INY	
033C-	A2 08	LDX	£\$08
033E-	B1 06	LDA	(£06),Y
0340-	0A	ASL	
0341-	3E F7 1F	ROL	\$1FF7,X
0344-	CA	DEX	
0345-	D0 F9	BNE	\$0340
0347-	88	DEY	
0348-	C0 01	CPY	£\$01
034A-	D0 F0	BNE	\$033C
034C-	B9 F7 1F	LDA	\$1FF7,Y
034F-	91 08	STA	(£08),Y
0351-	C8	INY	
0352-	C0 09	CPY	£\$09
0354-	D0 F6	BNE	\$034C
0356-	F0 B7	BEG	\$030F

On utilise les adresses \$6 à \$9 (page zéro) pour l'affichage indexé (avec Y). On installe \$3FFA (\$4004, adresse contenant le code ASCII du premier caractère, moins \$A — chaque caractère occupant ici 10 octets) et \$1FF7 (\$1FFF-\$8, chaque caractère de la police graphique occupera 8 octets). Notez que LSR (décalage à droite) transforme \$3F en \$1F. On économise un octet par rapport à LDA \$1F.

Adresse \$6-7 = adresse + \$0A
S'il n'y a pas de retenue, BCC envoie à \$31A.
Sinon l'adresse \$07 est incrémentée.

Registre Y mis à 0
Lecture de l'octet contenu à l'adresse \$6-7 + Y
Est-il égal ou supérieur à \$20 (espace) ?
Si oui on continue
Retour au Basic
Est-il égal ou supérieur à \$80 ?
Si oui, fin de programme.

Le CLC est ici superflu (C=0 après le CMP)
Même traitement que pour \$6-7 (plus haut) mais en ajoutant \$8.

Contenu de Y dans A (sans modifier Y).

8 octets mis à 0 (\$1FF8 à \$1FFF)
pour stockage provisoire.

Y = Y + 1 (il est donc à 1)
Registre X à zéro
Lecture du contenu de \$6-7 + Y
Décalage à gauche
Rotation à gauche
(on récupère le bit poussé par ASL)
Si X plus grand que 0, encore un tour !
Y = Y moins un

Si Y différent de 1, encore un tour !

Copie à partir de \$2000
(fonte graphique, par groupe de 8 octets)

Retour au point de départ.

Conseils aux utilisateurs

d'un Apple //c avec imprimante

Voici un truc pour résoudre le problème posé par les caractères ASCII de l'Apple, dont le bit de poids fort est à 1. Il n'est valable que dans le cas des ports série. Il faut d'abord rappeler que, par une liaisons série, les bits sont transmis en séquence : bit de START, cinq à sept bits de donnée, un bit (facultatif) de parité, un ou deux bits de stop. Rien ne les distingue sinon leur emplacement dans la séquence.

Configurer un port série, c'est (entre autres) définir cette séquence pour qu'elle soit celle que peut décoder l'imprimante.

Le truc consiste à leurrer l'imprimante : on la configure pour huit bits de donnée sans bit de parité, tandis qu'on programme l'Apple pour sept bits de donnée et un bit de parité : ce bit de parité sera pris par l'imprimante comme bit de poids fort du code ASCII. C'est possible et intéressant parce qu'on peut imposer la valeur du bit de parité : parité MARK ou parité SPACE.

Marche à suivre côté Apple //c :

- Démarrer la disquette des utilitaires-systèmes.
- Demander l'option 8 : "Autres Activités".
- Prendre l'option 5 : "Configurer les ports série".
- Choisir le port (pour une imprimante, c'est habituellement le port 1).
- Vous ne connaissez pas le NIP : répondre aux questions :
 - Mode d'impression
 - 7 bits de données/2 de stop
 - Débit selon votre imprimante
 - Parité SPACE pour mettre le bit de poids fort à zéro
 - ... les autres paramètres selon vos besoins.
- Sauver la configuration sur le disque (répondre oui).
- Quitter les utilitaires.

Par la suite, il faudra à chaque fois démarrer par la disquette des utilitaires-systèmes : la configuration

START : démarrage — MARK : caractère — SPACE : espace

ci-contre sera automatiquement restaurée ; puis quitter de suite les utilitaires (sans passer par l'option de configuration). Enfin lancer votre disquette de programme (sans mettre hors de tension : utiliser les trois touches : CTRL — Pomme ouverte — Reset).

Bien sûr, c'est long, mais on ne peut pas programmer de configuration (par défaut), par un startup en Basic, car cela nécessite des accès en mémoire auxiliaire. Par contre, c'est faisable en langage machine : voir plus loin.

Par cette méthode, les ports série sont prêts dès leur ouverture (en Basic, dès l'instruction "PR&1"), mais il est possible de changer le réglage ensuite, par exemple pour envoyer des codes avec 1 comme bit de poids fort au milieu d'autres avec 0 :

Pour changer le bit de parité, écrire Control-I suivi de "nP" (PRINT CHR\$(9) + "nP" en Basic) avec n = 5 pour bit à 1 ou n = 7 pour bit à 0.

Exemple, en Basic, pour imprimante OKI 80 avec interface RS232 connectée au port 1 :

```
10 REM Ouverture du port 1
20 PRINT CHR$(4)"PR&1"
30 REM Les codes 32 à 127 correspondent à des lettres, les codes 128 à 256 à des caractères graphiques genre Télétel.
40 PRINT "Ecriture de caractères texte"
50 REM Changement du bit de poids fort
60 PRINT CHR$(9)"5P"
70 PRINT "Ceci est pris par l'imprimante comme caractère graphique"
80 REM Re-changement
90 PRINT CHR$(9)"7P"
100 PRINT "De nouveau du texte..."
```

Remarque : pour le port 2, remplacer CHR\$(9) par CHR\$(1).

Comme lancer les utilitaires-systèmes à chaque démarrage est long, voici une petite routine en langage machine qui devra être appelée par un Programme STARTUP.

(Suite page 54)

Mode d'emploi :

- Tout d'abord, taper le programme et le sauvegarder par "BSAVE CONFIG, A\$8000, L\$100".
- Configurer la première fois les ports série avec les utilitaires-systèmes comme expliqué ci-dessus.
- Puis, sans mettre hors de tension, lancer la routine par "BRUN CONFIG, A\$8000" en MAINTENANT LA TOUCHE "ESC" APPUYÉE : ainsi la routine se met elle-même à jour.
- Il faut la sauvegarder à nouveau par "BSAVE CONFIG, A\$8000, L\$100".

Par la suite, un "BRUN CONFIG, A\$8000" dans votre programme STARTUP (cette fois-ci sans appuyer sur "ESC") configurera les ports série sans avoir à démarrer les utilitaires.

Pour ceux qui n'ont pas d'assembleur, taper la routine à partir du moniteur : CALL-151. Puis...

```
8000: A0 00 20 65 80 A2 8F A9
8008: 2A 20 A8 FC CA D0 F8 AD
8010: 00 C0 8D 10 C0 10 24 C9
8018: 9B D0 20 A0 3C 20 65 80
8020: A9 78 85 3C A9 04 85 3D
8028: A9 7F 85 3E A9 04 85 3F
8030: A9 DB 85 42 A9 80 85 43
8038: 18 90 19 A9 DB 85 3C A9
8040: 80 85 3D A9 E2 85 3E A9
8048: 80 85 3F A9 78 85 42 A9
8050: 04 85 43 38 AD 18 C0 68
8058: 8D 00 C0 20 86 CF 48 10
8060: 03 8D 01 C0 60 B9 71 80
8068: 10 06 20 ED FD C8 D0 F5
8070: 60 CD E9 F3 E5 A0 C0 A0
8078: EA EF F5 F2 A0 E4 E5 F3
8080: A0 F3 EF F2 F4 E9 E5 F3
8088: A0 F3 E5 F2 E9 E5 F3 A0
8090: BA 8D A0 A0 F0 F2 E5 F3
8098: F3 E5 FA A0 C5 D3 C3 A0
80A0: ED E1 E9 EE F4 E5 EE E1
80A8: EE F4 A0 8D 00 CF CB A0
80B0: BA A0 EE A7 EF F5 E2 EC
80B8: E9 E5 FA A0 F0 E1 F3 BA
80C0: A0 C2 D3 C1 D6 C5 A0 C3
80C8: CF CE C6 C9 C7 AC C1 A4
80D0: B8 B0 B0 B0 AC CC A4 B1
80D8: B0 B0 00 9E 0B 40 50 16
80E0: 0B 01 00 04 60 4C 88 9A
80E8: 4C 76 82 84 78 A5 79 48
80F0: 00 00 00 00 00 00 00 00
80F8: 00 00 00 00 00 00 00 00
```

MENU.TYPE

DOS ou ProDOS

- Est-il indispensable de passer par une routine en langage machine pour obtenir un menu ressemblant à celui des disquettes TREMLIN MICRO ? Non. Bien entendu, l'affichage est un peu moins rapide, mais est-ce tellement important ? Voici, un petit exemple... à compléter et éventuellement à modifier... si le cœur vous en dit !

Bernard DELAMOT

```
100 TEXT : HOME : PRINT : INVERSE
110 ONERR GOTO 200
120 HTAB 14: PRINT " M E N U "
130 VTAB 1: HTAB 14:T$ = "
    ": PRINT T$: VTAB 3: HTAB 14:
    PRINT T$: NORMAL
140 DATA 1. PROGRAMME NUMERO 1,2. PRO
    GRAMME NUMERO 2,3. PROGRAMME NUME
    RO 3,4. FIN DE TRAVAIL,0
150 FOR I = 1 TO 10: READ A$(I): IF A
    $(I) < > "0" THEN NEXT
160 N = I - 1
170 H = 10:V = 5
180 FOR I = 1 TO N: VTAB V + I * 2: P
    RINT TAB( H);A$(I): NEXT :V = 7
190 INVERSE : VTAB V: HTAB H: PRINT A
    $((V - 5) / 2): NORMAL
200 POKE - 16368,0: WAIT - 16384,128,
    127: POKE - 16368,0:T = PEEK ( -
    16384)
210 IF T = 13 THEN 280
220 IF T = 10 AND V = 5 + (N * 2) THE
    N V1 = 7: GOTO 270
230 IF T = 10 THEN V1 = V + 2: GOTO 2
    70
240 IF T = 11 AND V = 7 THEN V1 = 5 +
    (2 * N): GOTO 270
250 IF T = 11 THEN V1 = V - 2: GOTO 2
    70
260 GOTO 200
270 VTAB V: HTAB H: PRINT A$((V - 5)
    / 2):V = V1: GOTO 190
280 IF (V - 5) / 2 = N THEN END
290 VTAB 22: PRINT TAB( H + 3);"RUN "
    :: INVERSE : PRINT "PROGRAMME "(V
    - 5) / 2: NORMAL
```

TREMLIN MICRO

Tous les goûts sont dans la nature et le Basic conserve tous les charmes discrets d'un langage facile à comprendre !

Langage machine

6502-65C02 INITIATION



RÉPONSES à des
QUESTIONS simples

OCTET COUPÉ EN DEUX

Q : Comment ranger, dans les mémoires \$7 et \$8, les deux quartets de l'octet contenu dans la mémoire \$6 ?

R : Il faut d'abord masquer le quartet de poids fort (bits 3210) en utilisant `AND 00001111` (\$0F en hexadécimal). `AND` permet en effet de forcer des bits à zéro. Examinons ce qui va se passer avec deux exemples (`LDA £$AB` place la valeur \$AB dans l'Accumulateur).

1	<code>LDA £\$AB</code>	<code>1010 1011</code>	
	<code>AND £\$0F</code>	<code>0000 1111</code>	Comme on le voit :
	<code>A = \$B</code>	<code>0000 1011</code>	0 et 1 = 0
			1 et 1 = 1
2	<code>LDA £\$40</code>	<code>0100 0000</code>	1 et 0 = 0
	<code>AND £\$0F</code>	<code>0000 1111</code>	0 et 0 = 0
	<code>A = \$0</code>	<code>0000 0000</code>	

Pour extraire le quartet de poids fort (bits 7654), on utilise ensuite l'instruction `LSR`. Celle-ci décale vers la droite (d'un bit), le contenu de l'Accumulateur (ou celui d'une mémoire). Pour chaque `LSR`, un zéro entre à gauche et le bit poussé vers la droite (c'est une image !) tombe dans la retenue où... dans le cas présent, il ne sert à rien. Comme il faut éliminer quatre bits, nous allons user de quatre `LSR`.

Ainsi, partant d'une valeur \$AB, nous obtiendrons successivement :

<code>LSR 0101 0101</code>	soit \$55
<code>LSR 0010 1010</code>	soit \$2A
<code>LSR 0001 0101</code>	soit \$15
<code>LSR 0000 1010</code>	soit \$0A

Ces exemples vous montrent tout l'intérêt que présente la notation binaire (voir à ce sujet la fiche de *Tremplin Micro n°8*). En pratique, cela donne le petit programme de la colonne suivante. On suppose, que la mémoire \$6 (page zéro) a préalablement été chargée par un `POKE 6,171` (\$AB... en notation décimale).

<code>300 :</code>	<code>A5 06</code>	<code>LDA \$06</code>
<code>302 :</code>	<code>29 0F</code>	<code>AND £\$0F</code>
<code>304 :</code>	<code>85 07</code>	<code>STA \$07</code>
<code>306 :</code>	<code>A5 06</code>	<code>LDA \$06</code>
<code>308 :</code>	<code>4A</code>	<code>LSR</code>
<code>309 :</code>	<code>4A</code>	<code>LSR</code>
<code>30A :</code>	<code>4A</code>	<code>LSR</code>
<code>30B :</code>	<code>4A</code>	<code>LSR</code>
<code>30C :</code>	<code>85 08</code>	<code>STA \$08</code>
<code>30E :</code>	<code>60</code>	<code>RTS</code>

*6.8

`0006- AB 0B`
`0008- 0A`

POUR TERMINER,
contrôler
en tapant 6-8...
et RETURN

LDA : MNÉMONIQUE ET CODES

Q : L'instruction `LDA` indique que l'on charge l'accumulateur avec la donnée (ou le contenu de la mémoire) indiquée... mais pourquoi cette abondance de codes : `A9`, `AD`, `A5`... etc. alors que le but reste le même ?

R : On en souhaiterait davantage encore... car chaque code correspond à un mode d'adressage, et c'est le nombre de modes d'adressage qui fait la richesse d'un microprocesseur. Pour programmer en langage machine, il faut absolument posséder un livre donnant la liste de toutes les instructions ainsi que les codes et explications des différents modes d'adressage. Je vous conseille, dans la série intitulée "*La Pratique de l'Apple II*", le volume n°3 : *LANGAGE MACHINE ET ASSEMBLEUR DU 6502* (Nicole Bréaud-Pouliquen, Editions du PSI).

Quand on possède un Assembleur comme `PROCODE`, il n'est pas nécessaire de connaître les codes des instructions, mais il est indispensable de savoir comment utiliser les différents modes d'adressage... d'où l'intérêt de l'ouvrage cité plus haut (il en existe plusieurs autres, et *Tremplin Micro* a publié une critique sur les plus récents).

Guy-HACHETTE.

Vous avez écrit à TREMPLIN MICRO



• Apple LOGO vers imprimante

Peut-on, avec Apple LOGO, envoyer les codes ASCII à une imprimante à partir d'une procédure ? En effet, quand je veux lister une procédure en utilisant • IMPRIMANTE 1, le jeu de caractères est changé et certains codes ASCII modifiés, sans raison apparente (à partir du Basic ou du Pascal, aucun problème).

Ainsi, le * est transformé en ç, le + en souligné, etc. Un lecteur de Tremplin Micro peut-il éclairer ma lanterne ?

P.S. : Mon imprimante est une EPSON RX 80 F avec interface II EPSON-APPLE.

Jean-Marc B. (69000 OULLINS)

TM Nous n'utilisons pas Apple LOGO, mais nous sommes disposés à publier de courts programmes... en LOGO, et notamment une routine d'impression.

• HIGHER TEXT II ?

Je recherche désespérément Higher Text II, une disquette qui aurait semble-t-il été développée par Apple il y a 5 ou 6 ans. Elle comporterait un grand nombre de polices de caractères en haute résolution... polices que je pourrais transformer en autant de fontes pour mon Imagewriter.

Qui peut me dire où acheter cette disquette ?

Jean-Marc C. (26000 VALENCE)

TM Là encore, nous avouons notre profonde ignorance, mais nous ne demandons qu'à apprendre !

• IMP.HGR1

Le programme de Hard Copy IMP.HGR1 paru dans Tremplin Micro n°6 fonctionne parfaitement sur Apple IIc avec mon imprimante Scribe. Il présente toutefois la particularité suivante :

— après l'impression de la page graphique, une suite de caractères : "o" et "x" apparaît en inversé clignotant au milieu de l'écran. Le nombre de caractères parasites correspond au nombre de lignes imprimées.

Ma connaissance de l'Assembleur étant limitée, j'en ai été réduit à ajouter un JMP \$FC58 plutôt brutal avant la ligne 150, ce qui résoud mon problème.

Mais je préférerais une solution plus élégante qui s'attaquerait directement à la source de ce défaut que je n'ai pas réussi à identifier.

Avez-vous une idée sur cette question ?

Guy C. (07500 GRANGES LES VALENCE)

TM Une solution radicale : remplacer le JSR de \$60B5 par un JMP (\$4C au lieu de \$20)... ou bien utiliser la nouvelle version que présente la FICHE N°9 de Tremplin Micro.

L'édition obtenue grâce à IMP.HGR1 correspond au format simple. Peut-on doubler le format d'impression ? Comment ?

G. B. (92100 CLICHY)

TM On pourrait, en envoyant un CHR\$(14) à l'imprimante, éditer le graphique en double largeur... mais la hauteur resterait la même, dans l'état actuel du programme en tout cas.

• Index Tremplin Micro

Bravo pour cet index ! Quelle bonne idée. Je suis sûr qu'elle va inspirer d'autres revues. On retrouve tout ce que l'on veut en moins de temps qu'il n'en faut pour... taper le mot.

Je vois dix applications PERSONNELLES... avec ce programme, mais hélas ! pas de routine de saisie. Alors, par pitié, mon T.M. chéri, fais-moi plaisir...

Nicole G. (28000 CHARTRES)

TM Et moi, Nicole, j'adore te faire plaisir... à tel point que (il y a un décalage, mais je n'y suis pour rien) LA BOÎTE À IDÉES du dernier numéro... est tout à fait ce que tu attendais... sauf que sa parution était prévue bien avant la réception de ton amusante lettre... que tu m'excuseras d'avoir abrégée.

NESTOR.

• La pseudo page 3

J'ai lu quelque part que l'on peut accéder à une PAGE 3 graphique sur n'importe quel Apple II : est-ce exact ? Quelles sont les instructions à utiliser ?

Dominique C. (28100 DREUX)

TM Vous lisez mal votre Tremplin Micro. La réponse se trouve en effet dans notre n°5, page 27. Je vous invite à vous y reporter. La pseudo Page 3 présente un réel intérêt... quand on sait l'utiliser.

• Open, Close, Write et Read, Append, Store et Restore

Un grand nougat de Montélimar si vous réussissez à m'expliquer simplement :

— Comment ouvrir, écrire et lire un fichier séquentiel, avec ProDOS.

— Comment utiliser APPEND, STORE et RESTORE.

Louis D. (26200 MONTÉLIMAR)

TM

D'accord... mais tiendrez-vous parole ?

Premier point, il vous faut une disquette formatée avec votre utilitaire ProDOS. Ensuite, il est indispensable que le Système connaisse le PREFIX de cette disquette... sinon vous obtiendrez un message du type *PATH NOT FOUND* (chemin non trouvé).

Quatre commandes sont utilisées pour écrire et lire les fichiers séquentiels : OPEN — WRITE — READ — CLOSE. Supposons que le préfixe de votre disquette est *TM*, et voyons comment procéder avec :

OPEN

à utiliser en mode programme exclusivement. Ainsi, PRINT CHR\$(4); "OPEN/TM/FIC" va ouvrir un fichier FIC sur la disquette dont le PREFIX est TM. Si le fichier n'existe pas, il sera créé. Quand le PREFIX est connu, il est inutile de le préciser. Un simple PRINT CHR\$(4); "OPEN FIC" suffit.

On ne peut ouvrir un fichier, qui n'a pas été fermé.

CLOSE

ferme le fichier nommément désigné... ou tous les fichiers ouverts si le nom n'est pas précisé. CLOSE peut être utilisé en mode direct et il est recommandé de s'en servir après un "plantage". C'est un remède souverain contre le message d'erreur *FILE(S) STILL OPEN* (Fichier(s) non fermé(s)). Syntaxe : PRINT CHR\$(4); "CLOSE"... ou "CLOSE FIC", dans le cas présent.

WRITE-READ

des exemples vous sont donnés dans le mini-programme démo ci-contre.

APPEND

comme son nom l'indique, permet d'ajouter une donnée à la fin d'un fichier. APPEND ouvre le fichier (OPEN), et écrit la donnée à la suite du dernier enregistrement. Attention ! ne pas oublier d'utiliser CLOSE (voir notre exemple).

STORE et RESTORE

permettent de mémoriser, puis de récupérer, dans un fichier de type VAR, toutes les variables actives du programme. C'est extra... et réellement pratique. Dans certains cas, c'est mieux qu'un fichier séquentiel et parfois préférable à une longue série de DATA(S). L'exemple ci-contre vous montre comment écrire un fichier (120-145), comment utiliser APPEND (155-165), relire ce même fichier (175-190), stocker et récupérer les variables (200 puis

TM

210). Vous noterez que le tableau N\$(X) est remis à 0 par les lignes 170 et 205... pour bien montrer que toutes les instructions fonctionnent.

```
100 TEXT : HOME :D$ = CHR$(4)
105 FOR I = 1 TO 4
110 INPUT "NOM -> ";N$(I)
115 NEXT
120 PRINT D$"OPEN NOM": PRINT D$"CLOSE NOM": PRINT D$"DELETE NOM"
    : PRINT D$"OPEN NOM"
125 PRINT D$"WRITE NOM"
130 FOR J = 1 TO I - 1
135 PRINT N$(J)
140 NEXT
145 PRINT D$"CLOSE NOM"
150 INPUT "NOM EN PLUS ";N$
155 PRINT D$"APPEND NOM"
160 PRINT N$
165 PRINT D$"CLOSE NOM"
170 FOR I = 1 TO 4:N$(I) = "": NEXT I
    :N$ = ""
175 PRINT D$"OPEN NOM"
180 PRINT D$"READ NOM"
185 FOR I = 1 TO 5: INPUT N$(I): NEXT I
190 PRINT D$"CLOSE NOM"
195 FOR I = 1 TO 5: PRINT N$(I): NEXT I
200 PRINT D$"STORE NM"
205 FOR I = 1 TO 5:N$(I) = "": NEXT I
    :N$ = ""
210 PRINT D$"RESTORE NM"
215 FOR I = 1 TO 5: PRINT N$(I): NEXT I
```

Pourquoi "DELETE" à la ligne 120 ? Tout simplement parce que, dans le cas de plusieurs utilisations, un mélange des données se produirait fatalement au niveau d'APPEND (faites un essai).

IMPORTANT

La commande MAXFILES du DOS 3.3 est inconnue de ProDOS.

Lors de l'ouverture d'un fichier, ProDOS lui alloue un tampon (buffer) de 1024 octets.. On ne peut pas ouvrir plus de 8 fichiers en même temps et il est bon de savoir que les commandes CATALOG et EXEC ouvrent aussi un fichier... qui vient évidemment en déduction des 8 indiqués plus haut.

ProDOS ignore INIT (le formatage d'une disquette passe par un utilitaire) et MON, NOMON sont remplacés par TRACE et NOTRACE.

Et maintenant, à nous le nougat... et au dentiste le bénéfice de l'opération !

Vous avez écrit à TREMPLIN MICRO

• BYE... sous ProDOS

Est-il exact qu'il existe une commande BYE, utilisable sous ProDOS ? Malgré de multiples essais et de nombreuses recherches dans les livres et les revues consacrés à Apple, je n'ai pas réussi à obtenir un semblant de réponse à cette question.

Philippe QUEYROU (56170 QUIBERON)

TM Vous n'aviez pas frappé à la bonne porte, tout simplement... à moins que votre ProDOS soit une version antérieure à celle qui autorise l'instruction BYE, ProDOS 1.1.1.

Comment ça marche ? Il suffit de taper BYE en mode direct et un message s'affiche sur un écran préalablement nettoyé par un HOME fort efficace :

ENTER PREFIX (PRESS "RETURN" TO ACCEPT)

ProDOS vous indique le PREFIX qu'il connaît. A vous de l'accepter, comme il vous le propose gentiment, par un RETURN paresseux... ou de taper un autre PREFIX. Sauf erreur, un nouveau message vous demandera...

ENTER PATHNAME OF NEXT APPLICATION

Gageons que vous allez taper le nom de votre application préférée : MENU... et là les choses vont se compliquer. ProDOS attend le nom d'une application de type SYS... et il vous répond instantanément :

NOT A TYPE "SYS" FILE

Vous ne vous en tirerez qu'en tapant... par exemple BASIC.SYSTEM... car, à notre connaissance, si ESCAPE permet de revenir à l'item précédent (choix du PREFIX), rien à faire pour retrouver l'APPLESOFT, à moins de relancer le SYSTÈME.

BYE... BYE !

Nestor.

• Registres A, X et Y

Je "bricole" en langage machine, mais je me pose encore bon nombre de questions... dont celle-ci :

— est-il possible, sans acrobaties compliquées de modifier la valeur des registres A, X et Y ?

Précision utile : je sais que CTRL-E permet d'afficher les registres précités.

Noël GÉRANT (LIÈGE — BELGIQUE)

TM Aucune acrobatie ! immédiatement après l'astérisque, tapez deux points, puis les trois

TM valeurs (A, X, Y), en HEXA bien sûr, chacune d'elles étant séparée de la suivante par un espace... voici un exemple :

CALL-151

*

* CTRL-E RETURN

A=00 X=00 Y=00 P=00 S=98

*:01 02 03 CTRL-E RETURN

A=01 X=02 Y=03 P=00 S=98



• Remplissage d'écran

Oui, je sais, vous avez déjà publié des exemples de remplissage d'écran, mais j'ai besoin d'un conseil. J'ai récemment trouvé, dans un bouquin du PSI, la petite routine suivante :

0300-	20 58 FC	JSR	\$FC58
0303-	20 1B FD	JSR	\$FD1B
0306-	20 F0 FD	JSR	\$FDF0
0309-	D0 FB	BNE	\$0306

seulement impossible d'en sortir ! Connaissez-vous un moyen de remplir complètement l'écran, par cette technique, mais avec retour au Basic ?

Jean-Pierre L. (72130 FRESNAY)

TM Oui, mais il faut l'allonger un peu et prévoir l'affichage du dernier octet (\$7F7) sans scrolling. Vous appellerez cette routine par CALL — 768 : VTAB1. Voici ce que cela donne :

0300-	20 58 FC	JSR	\$FC58
0303-	20 1B FD	JSR	\$FD1B
0306-	20 F0 FD	JSR	\$FDF0
0309-	A6 25	LDX	\$25
030B-	E0 17	CPX	\$17
030D-	90 F7	BCC	\$0306
030F-	A6 24	LDX	\$24
0311-	E0 27	CPX	\$27
0313-	90 F1	BCC	\$0306
0315-	8D F7 07	STA	\$07F7
0318-	60	RTS	

• Gare à CTRL-B !

Je suis sûr d'avoir lu quelque part que le CTRL-B permet de revenir au Basic quand on est en mode Moniteur... mais cette commande m'a fait perdre mon programme en Basic. Depuis, je m'en tiens au CTRL-RESET...

François NÈGRE (13000 MARSEILLE)

TM Vous pouvez utiliser CTRL-C (qui ne vous fera pas perdre le programme en mémoire), mais aussi 3 DØG. Quant à CTRL-B, vous connaissez déjà ses effets !

• Pour accélérer le tri des tableaux

J'ai réalisé un programme dans lequel je suis amené à trier une centaine de variables (en fait, il s'agit d'un tableau de quatre éléments). Je crois savoir qu'il est possible de trier... sans modifier la liste, en utilisant un index, mais je n'y parviens pas... d'où mon appel au secours.

Quand publierez-vous un programme de tri en langage machine, à la fois simple à utiliser et facile à comprendre ?

Comment s'abonner à *Tremplin Micro* quand on est Belge (ne souriez pas : ce n'est pas une histoire... française) ?

Louis S. (BRUXELLES)

```

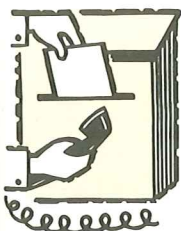
100 TEXT : PRINT CHR$(21): HOME
105 PRINT "ESSAI DE TRI PAR INDEX
-----"
110 PRINT : PRINT "ESSAYEZ avec N=40, puis avec N=100":
PRINT : PRINT "Vous constaterez des DIFFERENCES, mais"
115 PRINT "celles-ci resteraient les MEMES si le fichier
comportait plusieurs ELEMENTS et c'est tout l'avantage
de ce TRI..."
120 PRINT : PRINT "Le tri commence quand retentit le BIP"
"
125 PRINT : PRINT "ENFONCEZ UNE TOUCHE SVP ";; GET R$: PRINT
130 N = 40
135 DIM M$(N),IN%(N)
140 FOR I = 1 TO N: FOR J = 1 TO 6:M$(I) = M$(I) + CHR$(
(65 + INT ( RND (1) * 26)): NEXT : NEXT
145 CALL - 198
150 FOR I = 1 TO N:IN%(I) = I: NEXT
155 FOR I = 1 TO N - 1
160 FOR J = I + 1 TO N
165 IF M$(IN%(I)) < M$(IN%(J)) THEN 175
170 IN% = IN%(I):IN%(I) = IN%(J):IN%(J) = IN%
175 NEXT : NEXT
180 HOME
185 FOR I = 1 TO N: PRINT M$(IN%(I)),: NEXT
190 PRINT : VTAB 24: PRINT "<1> ENCORE <2> MENU DISQUETTE
<3> FIN ";; GET R$: HOME : PRINT
195 IF R$ = "1" THEN RUN
200 IF R$ = "2" THEN PRINT CHR$(4)"RUN MENU"

```

TM Voici un exemple de *tri sur index*. Bien entendu, M\$(N) pourrait être M\$(N,E) et, dans la boucle de la ligne 140, M\$(I) deviendrait M\$(I,E), E étant remplacé par le rang de l'élément sur lequel on trie.

Programme LM : L'idée est dans l'air, mais nous ne disposons que de (TROP) LONGUES routines, multi-usages (les tris de variables numériques n'intéressent pas nos lecteurs).

Abonnement : Vous pouvez procéder comme nos autres lecteurs belges, par virement bancaire, mais la carte VISA doit être acceptée.

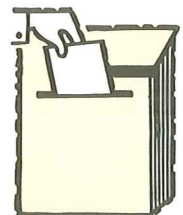


Renseignements téléphoniques

Ne nous téléphonez que dans les cas désespérés. Bien sûr, nous nous efforçons toujours de répondre à vos appels, mais c'est pour nous une perte de temps considérable. Meilleure heure : le matin entre 11 h et 12 h 15. Notez que les bureaux sont fermés le samedi... et le dimanche.

Renseignements par lettre

Nous ne pouvons pas "déboguer" des programmes... écrits par d'autres et il est donc inutile de nous adresser des demandes de cette nature. Dans tous les cas, joignez un timbre-poste pour la réponse. Merci de votre amicale compréhension.



COPIE HGR en HGR2

et vice-versa

COURTE et efficace... cette routine en 32 octets !

La petite démo ci-contre vous en convaincra.

Quant à la partie L.M. je vous laisse le soin de l'étudier.

G.-H.

```
0300-  A9 20    LDA    £$20
0302-  85 1C    STA    $1C
0304-  0A      ASL
0305-  85 1E    STA    $1E
0307-  A9 00    LDA    £$00
0309-  85 1B    STA    $1B
030B-  85 1D    STA    $1D
030D-  A8      TAY
030E-  B1 1B    LDA    ($1B),Y
0310-  91 1D    STA    ($1D),Y
0312-  C8      INY
0313-  D0 F9    BNE    $030E
0315-  E6 1C    INC    $1C
0317-  E6 1E    INC    $1E
0319-  A5 1E    LDA    $1E
031B-  C9 60    CMP    £$60
031D-  D0 EF    BNE    $030E
031F-  60      RTS
```

PAGE1-P2

```
100 TEXT : PRINT CHR$(21): HOME
110 FOR I = 768 TO 799: READ V: POKE
    I,V: NEXT
120 PRINT CHR$(21): HOME : HGR : TEX
    T
130 VTAB 20: INPUT "IMAGE ";I$: IF I$
    = "?" THEN PRINT CHR$(4)"CATALO
    G": PRINT : GOTO 130
140 POKE - 16304,0: POKE - 16302,0: P
    OKE - 16300,0: POKE - 16297,0
150 PRINT CHR$(4)"BLOAD"I$,A$2000"
160 CALL - 198: GET R$: HGR2 : PRINT
    : CALL 768: CALL - 198: GET R$: P
    RINT
170 HOME : TEXT
180 VTAB 6: PRINT "POUR PAGE 2 VERS P
    AGE 1, TAPER:          - POKE 771,
    30:POKE 774,28          - CA
    LL 768"
190 CALL - 198: GET R$: PRINT
200 HGR : POKE 771,30: POKE 774,28: C
    ALL 768
210 PRINT : VTAB 23: PRINT "<1> MENU
    DISQUETTE <2> ENCORE <3> FIN ";:
    GET R$: VTAB 22: PRINT
220 IF R$ = "1" THEN PRINT CHR$(4)"R
    UN MENU"
230 IF R$ = "2" THEN RUN
240 IF R$ < > "3" THEN 210
250 TEXT : HOME : END
260 DATA 169,32,133,28,10,133,30,169,
    0,133,27,133,29,168,177,27,145,29
    ,200,208,249,230,28,230,30,165,30
    ,201,96,208,239,96
```

INVERSION D'IMAGE

HGR ou HGR2

```
0300-  A9 40    LDA    £$40
0302-  85 1C    STA    $1C
0304-  A9 00    LDA    £$00
0306-  85 1B    STA    $1B
0308-  A8      TAY
0309-  B1 1B    LDA    ($1B),Y
030B-  49 FF    EOR    £$FF
030D-  91 1B    STA    ($1B),Y
```

Même programme... à quelques octets près : EOR inverse l'image.

P = 1 ou 2 : POKE 769,32 * P : POKE 791,32 * (P + 1)

```
030F-  C8      INY
0310-  D0 F7    BNE    $0309
0312-  E6 1C    INC    $1C
0314-  A5 1C    LDA    $1C
0316-  C9 60    CMP    £$60
0318-  D0 EF    BNE    $0309
031A-  60      RTS
```

IMP.HGR (en page 3)

0300-	A5 07	LDA	\$07
0302-	69 3A	ADC	£\$3A
0304-	C9 3C	CMP	£\$3C
0306-	F0 02	BEG	\$030A
0308-	A9 1C	LDA	£\$1C
030A-	85 07	STA	\$07
030C-	A5 FE	LDA	\$FE
030E-	C9 1A	CMP	£\$1A
0310-	90 04	BCC	\$0316
0312-	A9 18	LDA	£\$18
0314-	85 FE	STA	\$FE
0316-	A5 FD	LDA	\$FD
0318-	C5 FE	CMP	\$FE
031A-	90 04	BCC	\$0320
031C-	A9 00	LDA	£\$00
031E-	85 FD	STA	\$FD
0320-	A9 01	LDA	£\$01
0322-	20 95 FE	JSR	\$FE95
0325-	A2 17	LDX	£\$17
0327-	BD A8 03	LDA	\$03A8,X
032A-	20 ED FD	JSR	\$FDED
032D-	CA	DEX	
032E-	10 F7	BPL	\$0327
0330-	A2 07	LDX	£\$07
0332-	BD C0 03	LDA	\$03C0,X
0335-	20 ED FD	JSR	\$FDED
0338-	CA	DEX	
0339-	10 F7	BPL	\$0332
033B-	E8	INX	
033C-	86 FF	STX	\$FF
033E-	A5 FD	LDA	\$FD
0340-	20 C1 FB	JSR	\$FBC1
0343-	65 FF	ADC	\$FF
0345-	85 08	STA	\$08
0347-	A5 29	LDA	\$29
0349-	65 07	ADC	\$07
034B-	85 09	STA	\$09
034D-	A0 08	LDY	£\$08
034F-	A2 00	LDX	£\$00
0351-	A1 08	LDA	(\$08,X)
0353-	4A	LSR	
0354-	76 18	ROR	\$18,X
0356-	E8	INX	
0357-	E0 07	CPX	£\$07
0359-	D0 F8	BNE	\$0353
035B-	A9 03	LDA	£\$03

035D-	65 09	ADC	\$09
035F-	85 09	STA	\$09
0361-	88	DEY	
0362-	D0 EB	BNE	\$034F
0364-	A2 FF	LDX	£\$FF
0366-	E8	INX	
0367-	B5 18	LDA	\$18,X
0369-	A4 06	LDY	\$06
036B-	88	DEY	
036C-	F0 02	BEG	\$0370
036E-	49 FF	EOR	£\$FF
0370-	48	PHA	
0371-	09 80	ORA	£\$80
0373-	C9 8D	CMP	£\$8D
0375-	F0 04	BEG	\$037B
0377-	C9 9A	CMP	£\$9A
0379-	D0 05	BNE	\$0380
037B-	68	PLA	
037C-	29 F7	AND	£\$F7
037E-	D0 01	BNE	\$0381
0380-	68	PLA	
0381-	20 ED FD	JSR	\$FDED
0384-	E0 06	CPX	£\$06
0386-	D0 DE	BNE	\$0366
0388-	E6 FF	INC	\$FF
038A-	A5 FF	LDA	\$FF
038C-	C9 28	CMP	£\$28
038E-	D0 AE	BNE	\$033E
0390-	20 8E FD	JSR	\$FD8E
0393-	E6 FD	INC	\$FD
0395-	A5 FD	LDA	\$FD
0397-	C5 FE	CMP	\$FE
0399-	D0 95	BNE	\$0330
039B-	A9 1B	LDA	£\$1B
039D-	20 ED FD	JSR	\$FDED
03A0-	A9 63	LDA	£\$63
03A2-	20 ED FD	JSR	\$FDED
03A5-	4C 93 FE	JMP	\$FE93

Codes envoyés à l'imprimante

03A8-	0D 5A 09 5A 1B 36 31 54
03B0-	1B 6E 1B 30 31 30 4C 1B
03B8-	3E 1B 0D 4E 30 38 09 0D
03C0-	30 38 32 30 47 1B 00 00

BSAVE IMP.HGR A\$300, L\$C8

IMAGEWRITER IMAGEWRITER 2 DMP APPLE

HGR ou HGR2

PAGE 1 : POKE 7,1
PAGE 2 : POKE 7,2

IMPRESSION

Noir sur blanc : POKE 6,1
Blanc dans noir : POKE 6,0
POKE 6,0 convient pour
les photos et POKE 6,1
pour les graphiques.

MARGE

La marge est fixée à 10
(L010). Pour la modifier,
taper :
M\$ = "L020" (par exem-
ple), puis
X = 0 : FOR I = 950 TO
947 STEP -1 : X = X +
1 : POKE I, ASC
(MID\$(M\$, X, 1)) : NEXT

HAUTEUR

POKE 253,0 = ligne 1
POKE 254,24 = ligne 24

CARACTÈRE

Le caractère actuel est n
(camemberts ronds). Pour
le changer : POKE 945,
ASC(N ou E... par
exemple).

QUALITÉ

L'impression peut être
améliorée en mettant
préalablement l'impri-
mante en gras :
PRINT CHR\$(27) ; "!".

OCCUPATION DE LA PAGE ZÉRO PAR LE DOS

BYTE

UTILISATION

• par Marcel COTTINI

24	Position horizontale du curseur — DOS
26-27	Adresse du tampon où sera lu le secteur courant — ROM
28-29	BASL-BASH. Adresse de la première colonne de la ligne courante dans la page texte — DOS
2A	Contenu incorporé dans un compteur temporaire — BOOT, ROM
	Espace réservé en cas de rencontre d'une erreur — RWTS
2B	Boot slot × 16 — ROM
	Espace réservé en cas de rencontre d'une erreur — RWTS
2C	Checksum du champ d'adresses d'un secteur — RWTS
2D	Numéro du secteur lu à partir du champ d'adresses du secteur — RWTS
2E	Numéro de la piste lu à partir du champ d'adresses du secteur — RWTS
2F	Numéro du volume lu à partir du champ d'adresses du secteur — RWTS
33	Caractère solliciteur (prompt) — DOS
35	Numéro du drive incorporé dans le bit de poids fort — RWTS
36-37	CSWL-CSWH. Adresse de la routine de sortie de caractères — DOS
38-39	KSWL-KSWH. Adresse de la routine d'entrée de caractères — DOS
3C	Adresse utilisée pour la sauvegarde et transmission de données
	Byte de travail
	Adresse de la table des caractéristiques du drive, octet de poids faible (device table — DCT) — RWTS
3D	Numéro du secteur — ROM
	Adresse de la table des caractéristiques du drive, octet de poids fort — RWTS
3E-3F	Adresse du sous-programme de lecture d'un secteur — BOOT
	Adresse du tampon du DOS — RWTS
40-41	Adresse de début de l'image du DOS implanté en mémoire centrale — BOOT
	Adresse du tampon fichier — DOS
41	Compteur du nombre de pistes, utilisé lors du formatage de la disquette — RWTS
42-43	Adresse du tampon du DOS — DOS
44-45	Opérande numérique — DOS
46-47	Pointeur de la zone de sauvegarde de paramètres lors de la rencontre d'une erreur — RWTS
48-49	Adresse de la table IOB — RWTS
4A-4B	Adresse de LOMEM avec le Basic Integer actif — DOS
	Pointeur de la zone de travail utilisée lors du formatage de la disquette — RWTS
4C-4D	Adresse de HIMEM avec le Basic Integer actif — DOS
67-68	TXTTAB. Adresse de début du programme Applesoft — DOS
69-6A	LOMEM Basic Applesoft. Adresse de début de la zone des variables simples — DOS
6F-70	FRETOP Basic Applesoft. Adresse de début de la zone des chaînes de caractères et de fin de la zone libre — DOS
73-74	HIMEM Basic Applesoft. Adresse la plus élevée de la mémoire vive (RAM) disponible — DOS
76	CURLIN Basic Applesoft. Numéro de la ligne courante en exécution, HByte — DOS
AF-B0	PRGEN Basic Applesoft. Adresse de la fin de la zone du programme — DOS
CA-CB	Pointeur de début d'un programme du Basic Integer — DOS
CC-CD	Pointeur de la fin des variables du Basic Integer — DOS
D6	Drapeau utilisé pour la protection d'un programme Applesoft — DOS
D8-D9	Numéro de la ligne courante d'un programme Basic Integer — DOS
	Basic Applesoft ONERR — DOS

tremplin micro

SONDAGE du numéro 9

Les renseignements concernant votre identité et votre adresse ne seront en aucun cas communiqués à une société étrangère aux EDITIONS JIBENA.
TREMPILIN 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 6502 65C02

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 :

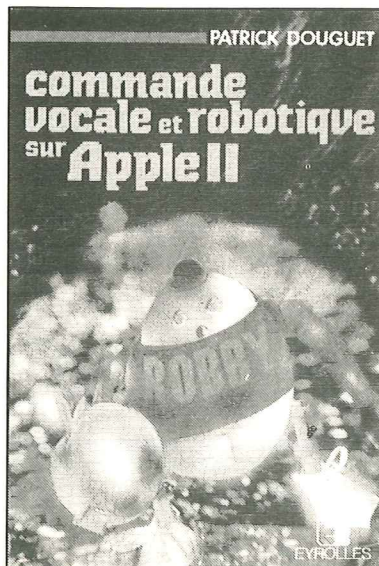
.....
.....

VOTRE BIBLIOTHÈQUE INFORMATIQUE

- **COMMANDE VOCALE ET ROBOTIQUE SUR APPLE II** (Eyrolles)

Si vous vous intéressez à la robotisation, offrez-vous l'ouvrage de Patrick DOUGUET et étudiez, en sa compagnie, programmes et schémas à l'appui, la réalisation d'un robot évolutif... utilisant votre APPLE II comme cerveau. Votre robot se déplacera sous les ordres d'une programmation inspirée du langage LOGO. Relié à l'Apple par radio et infrarouges, il se déplacera en évitant les obstacles et reconnaîtra une voix à laquelle il obéira : la vôtre.

Robby, c'est son nom, saura parler, siffler, détecter la fumée, surveiller la maison. Il jouera évidemment avec les enfants et, étant équipé de bras et de pinces, il pourra même vous apporter le journal ! Du rêve ? bien sûr que non !, mais tout cela n'est pas gratuit !



- **PHOTOGRAPHIE SUR AMSTRAD ET APPLE II** (PSI)

Peut-on s'intéresser à la photo... et à la programmation ? Nous

savons que oui, et c'est assurément le cas de Pierrick MOIGNEAU et de Xavier de la TULLAYE ! Le premier est responsable qualité aux laboratoires FUJI FILM et le second chef de produit chez KODAK-PATHÉ. On peut en déduire qu'ils connaissent leur affaire.

Personnellement, je n'ai pas testé leurs programmes, mais j'ai parcouru leur livre et j'avoue avoir été séduit par la clarté de certaines explications. Quant à la présentation, elle est bien meilleure que celle d'autres ouvrages du P.S.I. Alors, si le cœur vous en dit, apprenez vous aussi à maîtriser l'image à travers les nombreuses routines, en Basic (Amstrad et Applesoft) que vous proposent les auteurs !



- **ENTREPRENDRE ET GÉRER SUR MACINTOSH *** (Cedic/Nathan)

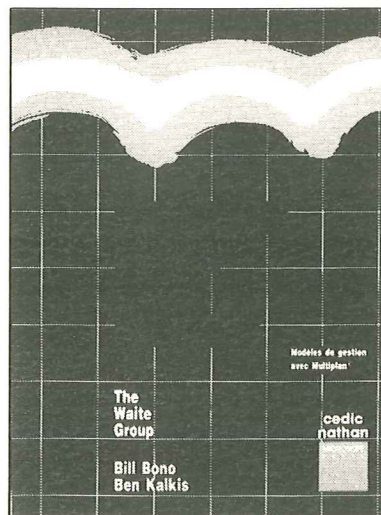
Multiplan est une feuille de calcul très puissante et très facile à utiliser. La facilité avec laquelle on peut quiconque, sans expérience informatique, peut apprendre à se servir de ce logiciel et du

Macintosh pour simplifier l'organisation de sa propre affaire, est réellement impressionnante. Au lieu de devoir "se transformer en informaticien" pour appliquer convenablement la puissance de calcul aux problèmes des affaires, les dirigeants peuvent désormais résoudre leurs problèmes dès qu'ils ont découvert l'interrupteur du Macintosh.

ENTREPRENDRE ET GÉRER SUR MACINTOSH commence par un court apprentissage du logiciel puis explique comment créer un journal de banque, l'analyse du cash flow, l'élaboration d'un budget et la préparation d'un compte de résultat, l'amortissement pour dépréciation, un tableau de remboursement d'emprunt, etc...

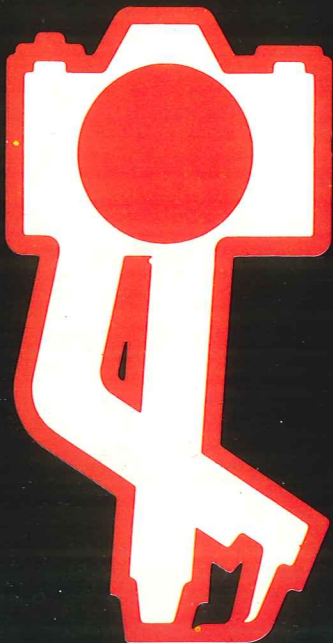
Toutes les feuilles de calcul présentées dans le livre peuvent être adaptées aux exigences propres de l'utilisateur. Cet ouvrage qui présente un système de comptabilité réellement très souple, s'adresse à tous les dirigeants de petites et moyennes entreprises. Ils disposeront ainsi à tout moment d'une mise à jour des informations nécessaires à la bonne marche de leur entreprise.

* Par Bill BONO et Ben KALKIS



Chasseur d'Images

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



Chez votre
marchand de journaux