

tremplin micro

**Du Dos 3.3
à ProDOS**

**TOUJOURS PLUS
POUR VOTRE
APPLE**

Bimestriel — Première année
JUILLET-SEPTEMBRE 1985
Numéro 3 — 30 F

D'abord déroutant, puis absolument captivant



A PROGRAMMATION vous tente, mais vous paraît d'un accès difficile. Ce que vous en savez déjà comporte un aspect déroutant... sinon inquiétant. En clair : vous redoutez l'échec.

Je ne vais pas vous dresser, ici, un tableau idyllique de l'apprentissage du Basic, et moins encore de celui de l'assembleur. L'un et l'autre exigent en effet un effort personnel et, surtout, la volonté de bien en assimiler l'abc.

Aucune revue — et la pédagogie, mot arrangé à toutes les sauces, n'a rien à voir là-dedans — ne fera de vous, simple débutant(e), un(e) champion(ne) de la programmation... si vous n'y mettez pas (un peu) du vôtre.

Mais, croyez-moi — et je ne m'adresse ici qu'aux débutant(e)s — l'effort se révèle toujours payant.

Bien des satisfactions sont réservées à celles et à ceux qui, inlassablement, alignent les précieux octets sur leur écran.

Cela vaut pour le Basic et pour l'assembleur, mais plus encore pour ce dernier, tout à fait incompréhensible quand on en ignore les principes, mais super-puissant et rapide quand, après bien des tentatives infructueuses, on parvient à le dompter.

Mais pourquoi programmer, alors que tant de génies produisent des logiciels et progiciels autorisant toutes les fantaisies ? Peut-être, tout simplement, pour ne pas mourir idiot !

Notre intellect, passif devant l'audiovisuel, habitué aux stéréotypes et aux informations prédigérées, ressent inconsciemment le besoin de se libérer, d'abord en luttant pour comprendre, ensuite en tirant le meilleur profit de ce qu'il a réussi à dominer.

Bonne programmation, et rendez-vous le 20 septembre dans le numéro 4 de Tremplin Micro.

GUY-HACHETTE.

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

BIMESTRIEL

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

EDITIONS JIBENA

Direction-Rédaction :
Editions JIBENA
Guy-HACHETTE

La Petite Motte — Senillé
86100 CHÂTELLERAULT.

Téléphone :
(49) 93-66-66

PUBLICITÉ :
Joelle (même numéro)

Commission paritaire :
Demande en cours.

Sommaire

3

Jouer sur un Apple

- La Personne X 17
- Ouest-Est 38
- Ouvrez l'œil 52

Jeu du numéro 2

- Coder ou décoder ? 13

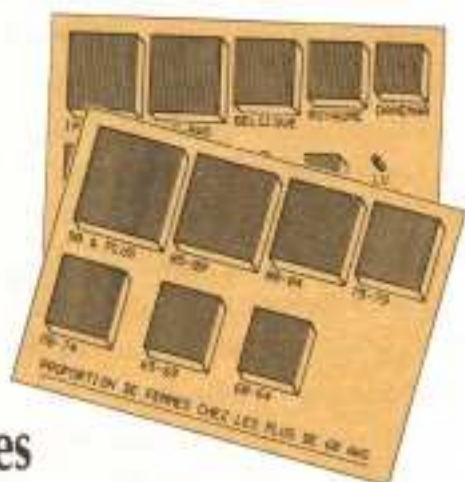
Initiation

- Générateur de programmes 33

SPÉCIAL ProDOS

- Datez vos programmes 31
- ProDOS et la carte langage 43
- ProDOS et vous 44

Une autre manière de visualiser vos nombres



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

21

PROGRAMMES UTILITAIRES

- Boutabout 30
- Ecrans condensés 36
- Lignes Data ou de Pokes 55

Créer des écrans sur 80 colonnes 3

LANGAGE MACHINE

- Saisie d'un texte 45
- Les variables en mémoire 46

Saisie LM assistée 48

LISEZ AUSSI :

- Partez en voyage avec Max 12
- Découvrez l'Assembleur 15
- Dessine-moi un cercle 47
- Quelques notes pour votre Apple 51

- Votre fiche n°3 41
- Le courrier des lecteurs 57
- Bulletin de commande et d'abonnement 62

Le numéro 4 de

TREMPIN MICRO

paraîtra le 20 septembre.

Créer des écrans sur 80 colonnes ?

Bien sûr que c'est possible !

APPLE IIe
APPLE IIc

POURQUOI ALLONGER démesurément certains de vos programmes, alors qu'il est tellement commode de stocker, x pages d'informations sur la disquette ? Cela est possible avec l'écran normal (40 colonnes), et nous avons vu comment dans le numéro 1 de *Tremlin Micro*, mais vous disposerez désormais des mêmes facilités sur 80 colonnes.

Le programme que nous vous proposons fonctionne sur un Apple IIe équipé d'une carte 80 colonnes (64 ou 128 k).

La version destinée à l'Apple IIc comporte un certain nombre de modifications (récapitulées page 11), ainsi qu'un sous-programme supplémentaire, en langage machine (lire page 11).

ECRANS.LM

Commencez par taper la routine ECRANS.LM, différente suivant le matériel dont vous disposez. Elle trace instantanément, et sur 80 colonnes, un cadre en mode inverse, mais elle ne fait pas que cela (les pages 5 et 6 vous la décrivent). Mémorisez cette routine par un

BSAVE ECRANS.LM, A\$300, L\$9D ou
BSAVE ECRANS.LM, A\$300, L\$9C pour le IIc.

ECRANS.BASIC

Passez ensuite à la partie la plus importante du travail : le programme en BASIC.

Dans un premier temps, ajoutez provisoirement une ligne **55 GOTO 80** qui vous évitera de recharger *ECRANS.LM* lors de vos essais (il restera en mémoire si vous n'éteignez pas votre micro, et si vous ne l'écrasez pas avec une autre routine, occupant les mêmes adresses), et escamotera également le chargement de *ECRC1* et *ECRC2* (encore inexistant).

Tapez minutieusement toutes les instructions, en les vérifiant au fur et à mesure.

Sauvez votre programme à plusieurs reprises (**ESSAI.1**, **ESSAI.2...** ou **TOTO.1**, **TOTO.2**, etc.).

Quand vous estimerez que votre travail est bien avancé, copiez aussi sa dernière version sur une autre disquette. Ces précautions vous éviteront de réécrire deux fois le même programme... à la suite de quelque manipulation contraire aux principes de votre Apple chéri. Ne souriez pas : cela arrive !

Tenez : je connais quelqu'un qui, après avoir tapé 200 lignes de programme, s'est aperçu qu'il n'avait pas (ou plus) le DOS. Heureusement, il lui restait la ressource du magnétophone (voir *Tremlin Micro* numéro 2) !

PREMIER ESSAI

Si tout fonctionne normalement, un RUN devrait vider l'écran, afficher un cadre en mode inverse, et un curseur en haut et à gauche. Si vous n'obtenez pas cela, vérifiez que *ECRANS.LM* est encore en mémoire et, surtout, qu'il est rigoureusement exact, puis relisez chaque ligne du programme en BASIC (il n'y a pas d'autre solution).

Si tout va bien, tapez n'importe quoi (un mot suffira), puis essayez la commande **CTRL-S** (**BSAVE**). Donnez *ECRC* comme titre. Continuez par **CTRL-F** (Fin). Profitez de l'occasion pour vérifier qu'un **GOTO 2000** vous restitue l'écran...

Normalement, vous devez maintenant trouver *ECRC1* et *ECRC2* sur votre disquette. Vous pouvez supprimer la ligne 55, puis relancer l'action par un nouveau RUN.

ECRAN DE COMMANDES

Cette fois, vous allez passer aux choses sérieuses. Le lecteur ronronne, émet peut-être des bruits bizarres (horribles !), mais le cadre apparaît. Parfait. Tapez **CTRL-^** (accent circonflexe).

(suite page 4).

ÉCRANS 80 COLONNES

Vous devez obtenir le mot que vous aviez ren-
tré plus haut, avant le BSAVE. C'est votre futur
écran de commandes. Enfoncez une touche :
vous retrouvez un écran vierge (cadre
excepté).

Tapez alors votre écran de commandes (ci-
dessous), en utilisant toutes les ressources
offertes par les divers CTRL.

Ainsi, le centrage s'obtient par CTRL-C, mais
on peut centrer une ligne en se plaçant au
début et en tapant CTRL-P (plus une case) ou
DELETE (moins une case).

Si vous tapez un malencontreux CTRL-O
(effacement du bas de l'écran, sous le cur-
seur), sachez qu'un CTRL-W (comme WRITE)
vous permettra de retrouver l'écran précé-
dent... intact. Pas beau cela ?

PARTICULARITES

La barre d'espacement permet effectivement
d'avancer (de même que la flèche → sur
l'Apple IIe), mais n'efface pas le caractère qui
se trouve sous le curseur. Pour taper un
"espace-effaceur" (ou "marqueur", dans le cas
du mode inverse), il convient de taper CTRL-E
(comme espace).

Et puis, il y a des astuces. Ainsi, pour passer
une ligne en mode inverse, il suffit de placer
le curseur au début, de taper CTRL-V, puis de
passer sur la ligne en actionnant la barre
d'espacement (ou la flèche →, sur l'Apple
IIe).

Autre technique : faire un CTRL-C (centrage)
qui inverse directement toute la ligne.

APPLE IIe SOUS ProDOS : Une seule modification, l'ajout d'une ligne 1295,
afin de pouvoir charger un écran à l'adresse \$400
(Texte de cette ligne : 1295 POKE 48984,0)

VOTRE PREMIER ÉCRAN SERA CELUI-CI (SOIGNEZ-LE : IL REVIENDRA SOUVENT !)

ÉCRAN DE COMMANDES

(POURRA ÊTRE RAPPELÉ PAR UN CTRL-~)

1 DÉPLACEMENTS : UTILISER LES FLÈCHES DU CLAVIER, MAIS

- CTRL-A ramène à la première ligne de l'écran
- CTRL-Z conduit à la fin de la dernière ligne de l'écran
- CTRL-D positionne le curseur au début de la ligne en cours
- CTRL-I (c'est la flèche TAB, sous ESC) le met en bout de ligne

FIN =
CTRL-F

NE PAS UTILISER LES FONCTIONS ESCAPE DE LA CARTE 80 COLONNES

2 ÉCRITURE : Tous les caractères sont utilisables, mais l'ESPACE n'efface pas un caractère existant. Pour imprimer un espace, taper un CTRL-E

- CTRL-G efface la page, à partir du curseur.
- CTRL-C permet de centrer une ligne et CTRL-O de l'effacer.
- CTRL-P crée un espace pour ajouter un caractère et la touche Del supprime au
contraire le caractère qui se trouve sous le curseur.

CTRL-W = Rappel de l'écran mémoire CTRL-V = INVERSE CTRL-N = NORMAL

CTRL-L = LOAD CTRL-S = BSAVE CTRL-T = SAVE LIGNES CTRL-X = IMPRESSION

ÉCRANS.LM (POUR APPLE IIe)

300 :	20 58 FC	JSR \$FC58	Home	
303 :	8D 01 C0	STA \$C001	Mise à 1 de 80 STORE (utilisation MEM.AUX)	
306 :	20 0C 03	JSR \$030C	ROUTINE n°1	
309 :	4C 2F 03	JMP \$032F	Entrée n°2	
ROUTINE N°1				
30C :	A2 00	LDX £\$00	Boucle initialisée à zéro	
30E :	A9 20	LDA £\$20	Accumulateur chargé avec \$20 (espace)	
310 :	8D 55 C0	STA \$C055	Ecriture en mémoire auxiliaire	
313 :	20 F0 FD	JSR \$FDF0	COUT1 : affiche le caractère dont le code ASCII est dans A	
316 :	20 10 FC	JSR \$FC10	BS : recule le curseur	
319 :	8D 54 C0	STA \$C054	Ecriture en mémoire résidente	
31C :	20 F0 FD	JSR \$FDF0	COUT1 de nouveau	
31F :	E8	INX	Incrémentation de la boucle	
320 :	E0 28	CPX £\$28	Sommes-nous à 40 (\$28) ?	
322 :	D0 EC	BNE \$0310	Un autre tour s'il n'y a pas égalité	
324 :	60	RTS	Retour au basic	
ROUTINE N°2				
325 :	A2 27	LDX £\$27	Registre X chargé avec \$27 (39)...	
327 :	86 24	STX \$24	... et versé dans CH (\$24), position horizontale du curseur.	
329 :	8D 54 C0	STA \$C054	Ecriture mémoire résidente	
32C :	20 F0 FD	JSR \$FDF0	COUT1	
ENTREE N°2				
32F :	20 62 FC	JSR \$FC62	CR : retour chariot	
332 :	A9 20	LDA £\$20	Espace chargé dans l'Accumulateur	
334 :	8D 55 C0	STA \$C055	Ecriture en mémoire auxiliaire	
337 :	20 F0 FD	JSR \$FDF0	COUT1	
33A :	A4 25	LDY \$25	Position verticale du curseur (CV) chargée dans Y	
33C :	C0 17	CPY £\$17	Est-elle égale à \$17 (23) ?	
33E :	D0 E5	BNE \$0325	Sinon, vers routine n°2	
340 :	20 10 FC	JSR \$FC10	BS : recule le curseur	
343 :	F0 C7	BEQ \$030C	Vers routine numéro 1 (tracé du bas de l'écran)	
345 :	AD 59 03	LDA \$0359	Recopie de la valeur poquée depuis le Basic	ROUTINE N°3 (lecture d'une ligne écran)
348 :	8D 63 03	STA \$0363		
34B :	AD 5A 03	LDA \$035A		
34E :	8D 64 03	STA \$0364		
351 :	A2 00	LDX £\$00	Initialisation	
353 :	A0 00	LDY £\$00		
355 :	8D 55 C0	STA \$C055	Page 2 commutée	
358 :	BD 00 00	LDA \$0000,X	Lecture (adresse poquée depuis le Basic)	
35B :	99 00 48	STA \$4800,Y	Ecriture pour constituer la variable X\$	
35E :	C8	INY	Plus 1 pour mémoire	
35F :	8D 54 C0	STA \$C054	Page 1 commutée	
362 :	BD 00 00	LDA \$0000,X	Lecture	
365 :	99 00 48	STA \$4800,Y	Ecriture (suite variable X\$)	
368 :	C8	INY	Plus 1 pour mémoire	
369 :	E8	INX	Plus 1 pour la boucle	
36A :	E0 28	CPX £\$28	A-t-on lu 2 fois 40 caractères ?	
36C :	D0 E7	BNE \$0355	Un autre tour si ce n'est pas terminé	

SUITE PAGE 6

ÉCRANS.LM (POUR APPLE IIe)

(suite)

36E : A5 69	LDA \$69	LOMEM pour l'adresse de X\$	
370 : 18	CLC	Annulation de la retenue	
371 : 69 02	ADC £\$02	Plus 2 pour obtenir l'adresse	
373 : 8D 83 03	STA \$0383	Préparation de la boucle (partie basse)	
376 : A5 6A	LDA \$6A	LOMEM + 1 partie haute	
378 : 69 00	ADC £\$00	Plus retenue éventuelle	
37A : 8D 84 03	STA \$0384	Préparation de la boucle (partie haute)	
37D : A2 00	LDX £\$00	Mise à zéro	
37F : 8D 88 03	LDA \$038B,X	Lecture	
382 : 9D 00 00	STA \$0000,X	Ecriture	
385 : E8	INX	Plus un	
386 : E0 03	CPX £\$03	Retour au Basic si on a lu 3 adresses	
388 : D0 F5	BNE \$037F	Encore un tour ?	
38A : 60	RTS	BASIC	
38B : 50 00	BVC \$038D	Longueur de la variable (\$50 = 80) et	
38D : 48	RTS	adresse (parties basse et haute)	
38E : 38	SEC	Retenue à zéro ou 1 (poquée depuis le Basic)	ROUTINE N°4
38F : 20 11 C3	JRS \$C311	AUXMOVE	(AUXMOVE)
392 : 60	RTS	Retour au Basic	
393 : 68	PLA	POUR TERMINER :	ROUTINE N°5
394 : A8	TAY	BSAVE ECRANS.LM,	(traitement
395 : 68	PLA	A\$300, L\$9D	des erreurs)
396 : A6 DF	LDX \$DF		
398 : 9A	TXS		
399 : 48	PHA		
39A : 98	TYA	VÉRIFICATION Tapez la ligne suivante, en mode direct :	
39B : 48	PHA	T = 0 : FOR I = 768 TO 924 : T = T + PEEK (I) : NEXT : PRINT T	
39C : 60	RTS	Le résultat doit être 18234 .	

ÉCRANS.Lm2c POUR APPLE IIc

La routine pour
Apple IIc est
différente.
La voici,
condensée :

SAUVEGARDE AVEC

BSAVE ECRANS.LM,
A\$300, L\$9C

* 300 : 20 58 FC A9 20 20 0B 03 20 16 03 A2 00 20 F0 FD E8 E0
50 D0 F8 60 20 F0 FD A2 4F 86 24 20 F0 FD A4 25 C0 16 D0 F0
60 AD 3B 03 8D 45 03 AD 3C 03 8D 46 03 A2 00 A0 00 8D 55
C0 BD 00 00 20 6D 03 C8 8D 54 C0 BD 00 00 20 6D 03 C8 E8 E0
28 D0 E7 RETURN, PUIS DE NOUVEAU :

* 350 : A5 69 18 69 02 8D 65 03 A5 6A 69 00 8D 66 03 A2 00
BD 8A 03 9D 00 00 E8 E0 03 D0 F5 60 C9 E0 30 06 09 80 C9 E0
10 0B 09 C0 C9 E0 30 09 49 40 4C 86 03 C9 FF F0 F7 99 00 48 60
50 00 48 18 20 11 C3 60 68 A8 68 A6 DF 9A 48 98 48 60
... ET UN DERNIER RETURN !

VÉRIFICATION Tapez, en mode direct, la ligne suivante :

T = 0 : FOR I = 768 TO 923 : T = T + PEEK (I) : NEXT : PRINT T
Le résultat devra être **17044**.

ECRANS.BAS

Version Apple IIe

Pour un Apple IIc, même programme, mais avec les corrections indiquées page 11.

A propos de la variable X\$, relisez la page 44 du numéro 2 de Tremplin Micro.

Cette ligne installe une fenêtre à l'intérieur du cadre.

Si l'on tape CTRL-à, c'est-à-dire CHR\$(0), R\$ est égal... à rien.

Si R\$ est une lettre, on passe immédiatement à l'affichage, ce qui permet de gagner du temps, en évitant la longue série des "IF".

La ligne 200 a été ajoutée pour la même raison. Si R est plus grand que 14, il est intéressant de sauter par-dessus les lignes inutiles.

```
10 LOMEM: 19000
20 X$ = " ": REM VARIABLE A NE PAS DEPLACER
30 FOR I = 1 TO 20:TR$ = TR$ + "_____": NEXT
40 TEXT : PRINT CHR$(4)"PRÉ3": HOME : PRINT
50 DIM L$(24):LL = 837:ER = 915:MO = 910
60 PRINT CHR$(4)"BLOAD ECRANS.LM": PRINT
70 PRINT CHR$(4)"BLOADECRC1,A$2000": PRINT CHR$(4)
)"BLOADECRC2,A$2400": PRINT
80 CALL 768: GOSUB 90: GOTO 150
90 POKE 34,2: POKE 35,23: POKE 32,2: POKE 33,76: RETU
RN

100 :
110 REM *****
120 REM ** SAISIE DE VOTRE ECRAN **
130 REM *****
140 :
150 PRINT :V = 3:H = 0
160 IF V < 3 THEN V = 3: CALL - 198
170 IF V > 22 THEN V = 22: CALL - 198
180 IF H < 8 THEN H = 72:V = V - 1: GOTO 160
190 IF H > 72 THEN H = 0:V = V + 1: GOTO 170
200 VTAB V: POKE 1403,H: PRINT "": GET R$
210 IF R$ = " " THEN GOSUB 1500: TEXT : GOTO 80
220 R = ASC (R$)
230 IF R = 127 THEN 550: REM DEL, SUPPRESSION
240 IF R = 5 THEN R$ = " ": GOTO 270: REM CTRL-E, ESPA
CE MATERIALISE
250 IF R = 32 OR R = 21 THEN H = H + 1: GOTO 190
260 IF R < 32 THEN 280
270 PRINT R$:H = H + 1: GOTO 160
280 IF R > 14 THEN 400
290 IF R = 1 THEN H = 0:V = 3: GOTO 200
300 IF R = 3 THEN 710: REM CTRL-C,CENTRAGE
310 IF R = 4 THEN H = 0: GOTO 200
320 IF R = 6 THEN GOSUB 1500: GOTO 1730: REM CTRL-F,
FIN
330 IF R = 7 THEN GOSUB 1500: NORMAL : CALL - 958: G
OTO 200: REM CTRL-G, EFFACEMENT A PARTIR ET SOUS LE
CURSEUR
340 IF R = 8 THEN H = H - 1: GOTO 180
350 IF R = 9 THEN H = 72: GOTO 200
360 IF R = 10 THEN V = V + 1: GOTO 170
370 IF R = 11 THEN V = V - 1: GOTO 160
380 IF R = 12 THEN GOTO 1270: REM CTRL-L,LOAD ECRAN
390 IF R = 14 THEN NORMAL :ET = 0: GOTO 200: REM CTRL
-I,NORMAL
400 IF R = 15 THEN H = 0: VTAB V:H = 0: POKE 1403,H: N
ORMAL : CALL - 868: GOTO 200: REM CTRL-O, EFFACER
LA LIGNE EN COURS
```

(SUITE PAGE 8)

ÉCRANS.BAS

(SUITE)

```
410 IF R = 16 THEN 630: REM CTRL-P, LETTRE EN PLUS
420 IF R = 19 THEN 1090: REM CTRL-S,BSAVE ECR1,ECR2
430 IF R = 20 THEN 1180: REM CTRL-7,SAVE FICHER TEXT
440 IF R = 22 THEN INVERSE :ET = 1: GOTO 200: REM CTR
L-V,INVERSE
450 IF R = 23 THEN GOSUB 1580: GOTO 150: REM CTRL-W,
RAPPEL DE L'ECRAN EN MEMOIRE AUXILIAIRE (DE DERNIER
)
460 IF R = 24 THEN 1380: REM CTRL-X,EDITION SUR IMPRIM
ANTE
470 IF R = 26 THEN H = 72:V = 22: GOTO 200
480 IF R = 30 THEN GOSUB 1500: GOSUB 1660: GOTO 200:
REM CTRL-FLEX, ECRAN DE COMMANDES
490 GOTO 160
500 :
510 REM *****
520 REM ** SUPPRESSION DE CARACTERE **
530 REM *****
540 :
550 GOSUB 840
560 VTAB V: POKE 1403,H: PRINT MID$(L$,H + 2) + " "
570 GOTO 200
580 :
590 REM *****
600 REM ** CARACTERE SUPPLEMENTAIRE **
610 REM *****
620 :
630 GOSUB 840
640 VTAB V: POKE 1403,H: PRINT " " + MID$(L$,H + 1,7
2 - H)
650 GOTO 200
660 :
670 REM *****
680 REM ** CENTRAGE D'UNE LIGNE **
690 REM *****
700 :
710 GOSUB 840:L = LEN (L$)
720 ONERR GOTO 1800
730 IF ASC ( MID$(L$,L,1)) = 160 THEN L = L - 1: GOT
O 730
740 L$ = LEFT$(L$,L)
750 FOR I = 1 TO L: IF ASC ( MID$(L$,I,1)) = 160 THE
N NEXT
760 NORMAL : VTAB V: POKE 1403,0: CALL - 868: IF ET =
1 THEN INVERSE
770 L = L + 1 - I::H = INT (74 - L) / 2: POKE 1403,H:
PRINT RIGHT$(L$,L)
780 POKE 216,0: GOTO 200
790 :
800 REM *****
810 REM ** LECTURE D'UNE LIGNE **
820 REM *****
830 :
```

◀ Les commandes de l'imprimante (voir plus loin) concernent IMAGE WRITER ou DMP Apple.

◀ CTRL-(Accent circonflexe) permet de revoir l'écran de commandes (attention ! il faut d'abord l'avoir tapé... et mémorisé !).

◀ Si l'on est en mode **INVERSE**, les effets sont très... spéciaux : toute la ligne passe en mode inverse, espaces compris, à partir du caractère à supprimer. Pour que tout rentre dans l'ordre, passer en mode **NORMAL** (CTRL-N), placer le curseur immédiatement à la suite du dernier caractère utile... et retaper DEL (Delete). On assiste au même phénomène avec le caractère supplémentaire (ligne 630). On pourrait faire précéder les deux GOSUB (550 et 630) par "NORMAL :", mais les lignes en mode INVERSE passeraient alors en mode NORMAL.

◀ **NOTA** : CTRL-C (centrage) fait passer une ligne normale en mode INVERSE si l'on a pris la précaution de taper au préalable un CTRL-V (INVERSE)... et rassurez-vous, sur votre écran, tout cela deviendra beaucoup plus clair !

Le programme vérifie l'existence d'un fichier et vous prévient si le titre existe déjà.

Un conseil : "LOCKEZ" votre écran de commandes dès qu'il sera installé. Mieux vaut un BREAK... qu'un fâcheux écrasement de cette belle page !

```

840 RESTORE : FOR I = 1 TO V: READ L: NEXT
850 POKE LL + 20,L - INT (L / 256) * 256: POKE LL + 2
  1, INT (L / 256): CALL LL:T$ = MID$ (X$,1)
860 L$ = MID$ (X$,5,73): RETURN
870 :
880 REM *****
890 REM ** TITRE D'UN FICHER **
900 REM *****
910 :
920 NORMAL : TEXT : VTAB 17: CALL - 958: PRINT : PRIN
  T TR$
930 VTAB 20: POKE 1403,0: PRINT "TITRE PROPOSE (? POUR
  OBTENIR LE MENU) ";: CALL - 198: INPUT "":F$: IF
  F$ = "" THEN RETURN
940 IF F$ = "?" THEN HOME : PRINT CHR$ (4)"CATALOG":
  GET R$: POKE - 16368,0: HOME : GOTO 920
950 IF ASC (F$) < 65 OR ASC (F$) > 90 THEN 930
960 ONERR GOTO 1010
970 G$ = F$ + "1"
980 PRINT : PRINT CHR$ (4)"VERIFY"G$: IF LT = 1 THEN
  LT = 0: GOTO 1000
990 VTAB 20: CALL - 950: INVERSE : PRINT F$;; NORMAL
  : CALL - 198: PRINT " EXISTE -- 1) TITRE MAINTENU
  <2> AUTRE TITRE ";: GET R$: POKE - 16368,0: IF R$
  < > "1" THEN 920
1000 POKE 216,0: RETURN
1010 CALL ER: POKE 216,0: IF PEEK (222) < > 6 THEN 17
  90
1020 IF LT = 0 THEN RETURN
1030 VTAB 20: CALL - 958: INVERSE : PRINT F$;; NORMAL
  : CALL - 198: PRINT " N'EXISTE PAS - APPUYEZ SUR U
  NE TOUCHE SVP ";: GET R$: POKE - 16368,0: GOTO 920
1040 :
1050 REM *****
1060 REM ** SAUVEGARDE DE L'ECRAN **
1070 REM *****
1080 :
1090 GOSUB 1500:LT = 0: GOSUB 920: GOSUB 1500
1100 GOSUB 1580: IF F$ = "" THEN 150
1110 F1$ = F$ + "1":F2$ = F$ + "2"
1120 PRINT : VTAB 2: PRINT CHR$ (4)"BSAVE"F1$,A$400,L
  1016": POKE - 16299,0: PRINT CHR$ (4)"BSAVE"F2$,
  A$400,L1016": POKE - 16300,0: GOTO 200
1130 :
1140 REM *****
1150 REM ** SAUVEGARDE EN FICHER T **
1160 REM *****
1170 :
1180 N = V:V = 0:F$ = "ECR.T"
1190 IF V < 23 THEN V = V + 1: GOSUB 840:L$(V) = T$: GO
  TO 1190
1200 V = N: VTAB 2: PRINT : PRINT CHR$ (4)"OPEN"F$: PRI
  
```

(SUITE PAGE 10)

Vous mémorisez deux écrans (chacun d'eux occupe 5 secteurs sur la disquette).

Cette partie du programme ne présente pas une réelle utilité. Elle mémorise, sous forme de fichier T, 23 lignes d'écran.

ÉCRANS.BAS

(SUITE)

```
NT CHR$ (4)*WRITE"F$ : PRINT 23: FOR I = 1 TO 23: P
RINT CHR$ (34) + L$(I): NEXT : PRINT CHR$ (4)*CLO
SE"F$
1210 GOTO 200
1220 :
1230 REM *****
1240 REM **      CHARGER UN ECRAN      **
1250 REM *****
1260 :
1270 GOSUB 1500:LT = 1: GOSUB 920
1280 IF F$ = "" THEN GOSUB 1580: GOTO 1320
1290 F1$ = F$ + "1":F2$ = F$ + "2"
1300 POKE - 16300,0: VTAB 2: PRINT CHR$ (4)"BLOAD"F1$
  ",A$400": POKE - 16299,0: PRINT CHR$ (4)"BLOAD"F2
  $",A$400": POKE - 16300,0
1320 GOSUB 90: GOTO 150
1330 :
1340 REM *****
1350 REM **      COPIE DE L'ECRAN 80      **
1360 REM *****
1370 :
1380 N = V:V = 0: NORMAL
1390 IF V < 23 THEN V = V + 1: GOSUB 840:L$(V) = T$: GO
  TO 1390
1400 GOSUB 1500
1410 PRINT CHR$ (21): HOME : PRINT CHR$ (4)"PRÉ1": PR
  INT CHR$ (27)*"E": PRINT CHR$ (27)"L010": PRINT C
  HR$ (27)"!": PRINT CHR$ (9)"BIN"
1420 FOR I = 1 TO 23: PRINT L$(I): NEXT : PRINT : PRINT
  CHR$ (12): PRINT CHR$ (4)"PRÉ0"
1430 PRINT CHR$ (4)"PRÉ3": HOME : PRINT : GOSUB 1580
1440 V = N: GOTO 200
1450 :
1460 REM *****
1470 REM **      TRANSFERT ECRAN MEM AUX      **
1480 REM *****
1490 :
1500 POKE 60,56: POKE 60,0: POKE 61,4: POKE 62,248: POK
  E 63,7: POKE 66,0: POKE 67,64: CALL M0
1510 POKE - 16299,0: POKE 60,0: POKE 61,4: POKE 62,248
  : POKE 63,7: POKE 66,0: POKE 67,68: CALL M0: POKE
  - 16300,0
1520 RETURN
1530 :
1540 REM *****
1550 REM **      RECUPERATION DE L'ECRAN      **
1560 REM *****
1570 :
1580 POKE 60,24: POKE 60,0: POKE 61,64: POKE 62,248: PO
  KE 63,67: POKE 66,0: POKE 67,4: CALL M0
1590 POKE - 16299,0: POKE 60,0: POKE 61,68: POKE 62,24
  8: POKE 63,71: POKE 66,0: POKE 67,4: CALL M0: POKE
  - 16300,0
```

Avec ProDOS, prévoyez une ligne 1295, ainsi rédigée (quel style !):

◀ 1295 POKE 48984,0

Cela évitera à votre Apple de vous faire une scène, consistant tout bonnement à refuser de charger votre écran !

◀ DMP Apple ou IMAGE WRITER. Avec d'autres marques, s'ent tenir à PRINT CHR\$(9) "BIN", ou "80N", selon les cas.

Attention ! avec un Apple IIc, plusieurs lignes sont à modifier et d'autres à ajouter. Lisez le détail ci-contre.

Les routines en assembleur reprennent des programmes utilisés (et parfois expliqués) dans les deux premiers numéros de Tremplin Micro.

```

1600 GOTO 98
1610 :
1620 REM *****
1630 REM ** AFFICHAGE ECRAN COMMANDES **
1640 REM *****
1650 :
1660 POKE MO,56: REM RETENUE A 1
1670 POKE 60,0: POKE 61,32: POKE 62,248: POKE 63,35: PO
KE 66,0: POKE 67,4: CALL MO
1680 POKE MO,56: REM RETENUE A 1
1690 POKE - 16299,0: POKE 60,0: POKE 61,36: POKE 62,24
8: POKE 63,39: POKE 66,0: POKE 67,4: CALL MO: POKE
- 16300,0
1700 CALL - 198: POKE - 16368,0: WAIT - 16384,128,12
7: POKE - 16368,0: GOTO 1580
1710 :
1720 DATA 1024,1152,1280,1408,1536,1664,1792,1920,1064,
1192,1320,1448,1576,1704,1832,1960,1104,1232,1360,1
488,1616,1744,1872,2000
1730 TEXT : HOME : END
1740 :
1750 REM *****
1760 REM ** AUTRES ERREURS **
1770 REM *****
1780 :
1790 PRINT : VTAB 20: CALL - 958: PRINT "ERREUR NUMERO
* PEEK (222)" LIGNE " PEEK (218) + PEEK (219) * 2
56" -- PRESSEZ UNE TOUCHE " : CALL - 198: GET R#:
POKE - 16368,0: RETURN
1800 CALL 915: POKE 216,0: GOTO 200
2000 POKE 51,0: GOSUB 1580: GOTO 200

```

Si vous vous plantez, ou si vous faites un intempestif CTRL-F (FIN), tapez un GOTO 2000... et vous retrouverez à la fois votre écran... et le DOS (sauf erreur !).

ROUTINE ECR.MEM2C

Rentrez et sauvez ce petit programme en utilisant la technique habituelle : CALL-151, etc puis BSAVE ECR.MEM2C, A\$1E40, L\$70

* 1E40.1EAF

```

1E40: A2 00 BD 78 04 9D 00 1E
1E48: BD F8 04 9D 08 1E BD 78
1E50: 05 9D 10 1E BD F8 05 9D
1E58: 18 1E BD 78 06 9D 20 1E
1E60: BD F8 06 9D 28 1E BD 78
1E68: 07 9D 30 1E BD F8 07 9D
1E70: 38 1E E8 E0 08 D0 CB 60
1E78: A2 00 BD 00 1E 9D 78 04
1E80: BD 08 1E 9D F8 04 BD 10
1E88: 1E 9D 78 05 BD 18 1E 9D
1E90: F8 05 BD 20 1E 9D 78 06
1E98: BD 28 1E 9D F8 06 BD 30
1EA0: 1E 9D 78 07 BD 38 1E 9D
1EA8: F8 07 E8 E0 08 D0 CB 60

```

APPLE IIc (MODIFICATIONS DE ÉCRANS.BAS)

```

50 DIM L$(24): LL = 807: ER = 914: MO = 909
55 M1 = 7744: M2 = 7800: PRINT CHR$(4) "BLOAD
ECR.MEM2C"
90 POKE 34,3: POKE 35,22: POKE 32,3: POKE 33,76: RETURN
100 IF H = 0 THEN H = 73: V = V - 1: GOTO 160
190 IF H = 73 THEN H = 0: V = V + 1: GOTO 170
SUPPRIMER LES LIGNES 240 - 250
350 IF R = 9 THEN H = 73: GOTO 200
435 IF R = 21 THEN H = H + 1: GOTO 190
860 L$ = MID$(X$,4,73): RETURN
1295 CALL M1 + ":POKE 48984,0" sous ProDOS
1305 CALL M2
1400 IM = 1: GOSUB 1580
1515 IF IM = 1 THEN IM = 0
1665 CALL M1
1695 CALL M2

```

Dans le tableau de commandes, CTRL-E devient inutile. Sur l'Apple IIc, l'espace joue son rôle normal et la flèche droite provoque un déplacement vers la droite. Attention ! Sauvegardez votre écran sur disquette avant de l'imprimer, car il sera ensuite effacé !

Si votre anglais laisse encore à désirer...

Partez en voyage avec MAX !

Où en est votre anglais ? Sauriez-vous vous faire comprendre si vous étiez parachuté à Londres ? Si la réponse est oui, *MAX The Globe Trotter* ne vous intéresse pas, encore que, possesseur d'un *Apple*, il ne saurait vous laisser totalement indifférent... pour la simple, mais excellente raison qu'il utilise votre ordinateur personnel, dont il tire d'ailleurs le meilleur parti.

Pour utiliser *MAX* — 4 disquettes, 2 cassettes, et un *Microlingua Book* de 180 pages — il suffit de disposer d'un *Apple IIc* ou *IIe* équipé d'une carte 80 colonnes, ainsi que d'un lecteur de cassettes.

Si vous désirez une bonne qualité sonore, je vous conseille le lecteur de votre radio, préférable, et de loin, à votre magnétocassette portable.

Si vous n'avez vraiment aucune connaissance en anglais, *MAX* ne deviendra pas votre ami sans de sérieux efforts de votre part.

Par contre, si votre vocabulaire (anglais) de base atteint déjà une centaine de mots, demandez à *MAX*, soit de vous rafraîchir la mémoire, soit de vous en apprendre davantage, ce qui revient au même.

Que l'on ne s'y trompe pas : il s'agit d'une initiation sérieuse à l'anglais (et à l'américain) de tous les jours, et non d'une approche plus ou moins réussie de l'étude de cette langue.

Avec *MAX*, vous apprendrez réellement à vous débrouiller, par le son, le texte et l'image (celle-ci ne constitue d'ailleurs pas l'intérêt numéro 1 de ce logiciel).

Avec la *Computercassette* (c'est aussi la première), vous accompagnerez *MAX* en bateau, dans le train, en avion, à l'hôtel et au restaurant. Dans le même temps, votre *Apple* chéri vous invitera à répondre à des



questions plus ou moins difficiles, tout cela avec un accompagnement musical et un certain nombre de bruitages.

Par le truchement du mode inverse, votre attention sera attirée sur les mots pouvant poser un problème de comprenette. Ils vous seront bien sûr expliqués... dans un anglais moins obscur.

Sketches, exercices, jeux : le cours n'est pas seulement didactique, mais se veut ludique. Un bon point pour lui. A noter que vous n'êtes jamais obligé de subir le programme, la touche *ESCAPE* vous permettant de revenir au menu si tel est votre bon plaisir... et aussi de recommencer à tel ou tel niveau du cours.

Le manuel de référence paraît complet, mais je n'ai pas aimé ses couleurs (beaucoup trop vives), sans doute destinées à interdire la photocopie. C'est un peu dommage car il est bien conçu. Ceci dit, le *Microlingua Book* reste un bon ouvrage d'accompagnement, où chaque exercice est clairement expliqué en français. Il est indispensable aux vrais débutants.

Nestor.

Vous pouvez recevoir une disquette *Démo*, fort bien présentée d'ailleurs, en utilisant le bulletin de commande de la page 62. (*MAX The Globe Trotter* est vendu 1200 F T.T.C., prix justifié).

Jeu du numéro 2

Cryptotexte

CRYPTOTEXTE permet de décrypter ou coder un programme préalablement transformé en fichier EXEC (voir la fiche n°1 de *Tremplin Micro*). Nous vous donnons, page suivante, le listage de la petite routine en assembleur qui correspond à la ligne 450 (DATA) du programme en BASIC.

TEXTE ORIGINAL

```
10 TEXT : HOME
20 PRINT "VOULEZ-VOUS DEVINER UN NOMBRE CHOISI PAR L'ORDINATEUR ?"
30 VTAB 4: HTAB 1: INPUT "OUI OU NON ? ";R$
40 IF R$ = "NON" OR R$ = "N" THEN 160
50 IF R$ < > "OUI" AND R$ < > "O" THEN 30
60 N = 1 + INT ( RND (1) * 100)
70 PRINT : PRINT "CE NOMBRE EST COMPRIS ENTRE "; INVERSE : PRINT 1
  ;: NORMAL : PRINT " ET "; INVERSE : PRINT 100; NORMAL
80 VTAB 10: HTAB 1: CALL - 860: INPUT "VOTRE NOMBRE -> ";N$
90 X = VAL (N$): IF X = 0 THEN 80
100 IF X = N THEN R$ = " VOUS AVEZ GAGNE !": GOTO 120
110 R$ = "PERDU! TROP PETIT!": IF X > N THEN R$ = "PERDU! TROP GRAND!"
  "
120 C = C + 1
130 VTAB 13: HTAB 1: CALL - 860: PRINT R$;" - ESSAI NUMERO "; INVE
  RSE : PRINT C: NORMAL
140 IF X < > N THEN 80
150 PRINT : PRINT : PRINT "PRESSEZ UNE TOUCHE SVP "; GET R$: GOTO 10
160 PRINT : PRINT : PRINT "DOMMAGE, NOUS NOUS SERIONS BIEN AMUSES!"
```

```
10 TEXT : PRINT CHR$(21): HOME
20 D$ = CHR$(4)
30 DIM L$(150)
40 GOTO 440
50 :
60 REM *****
70 REM * LECTURE ET RE...CREATION *
80 REM *****
90 :
100 L = 0:N = 0:L$ = ""
110 ONERR GOTO 240
120 PRINT : VTAB 15: PRINT D$"OPEN"F1$
130 PRINT D$"READ"F1$
140 GET CA$
150 CALL 760
160 CA = PEEK (6): IF CA = 13 THEN 180
170 L$ = L$ + CHR$(CA): GOTO 140
180 L = L + 1:L$(L) = L$:L$ = "": GOTO 140
190 :
200 REM *****
210 REM * ECRITURE DU FICHIER F2$ *
220 REM *****
230 :
240 IF PEEK (222) = 5 THEN PRINT : PRINT D
  $"CLOSE"F1$: POKE 216,0: GOTO 260
250 VTAB 22: CALL - 860: PRINT "PROBLEME NO
  N PREVU AU PROGRAMME": STOP
260 PRINT D$"PR#3": PRINT : HOME : FOR I = 1
  TO L: PRINT L$(I): NEXT
270 PRINT D$"OPEN"F2$: PRINT D$"WRITE"F2$: F
  OR I = 1 TO L: PRINT L$(I): NEXT : PRINT
```

```
D$"CLOSE"F2$: HOME : RUN
280 :
290 REM *****
300 REM * DECRYPTER OU CRYPTER *
310 REM *****
320 :
330 INVERSE : PRINT " CRYPTER OU DECRYPTER
  UN FICHIER (T) "; NORMAL
340 FOR I = 1 TO 10:T$ = T$ + " _": NEXT
350 VTAB 20: PRINT T$
360 VTAB 22: HTAB 2: PRINT "<1> CRYPTER (2
  ) DECRYPTER (3) FIN "; GET R$: PRINT
  R$:R = ASC (R$) - 48: IF R = 1 OR R = 2
  THEN 390
370 IF R < > 3 THEN 360
380 TEXT : HOME : END
390 IF R = 1 THEN F1$ = "JEU":F2$ = "JEU":V
  $ = "AEIOUDAUEI": GOTO 410
400 F2$ = "JEU":F1$ = "JEU":V$ = "DAUEIAEIOU
  "
410 FOR I = 797 TO 806: POKE I, ASC ( MID$(
  V$,I - 796,1)): NEXT
420 VTAB 6: PRINT "ON PART DE "; INVERSE :
  PRINT F1$; NORMAL : PRINT " POUR CREER "
  ;: INVERSE : PRINT F2$: NORMAL
430 GOTO 100
440 FOR I = 760 TO 796: READ R: POKE I,R: NE
  XT : RESTORE : GOTO 330
450 DATA 173,0,2,201,13,208,2,240,10,162,0,
  221,29,3,200,6,189,34,3,133,6,96,232,224,
  5,200,240,240,246
```

Cryptotexte

Les données de la section encadrée correspondent (en hexadécimal) aux valeurs décimales de la ligne 450 (DATA) du programme en Basic.

0300 :	AD	00	02
0303 :	C9	0D	
0305 :	D0	02	
0307 :	F0	0A	
0309 :	A2	00	
030B :	DD	1D	03
030E :	D0	06	
0310 :	BD	22	03
0313 :	85	06	
0315 :	60		
0316 :	E8		
0317 :	E0	05	
0319 :	D0	F0	
031B :	F0	F6	
031D :	C1	C5	C9
	CF	D5	
0322 :	CF	C1	D5
	C5	C9	

IDENTIFICATION DES VOYELLES AFIN DE LES PERMUTER ENTRE ELLES (CRYPTER)

Guy-hachette Février 85

ASSEMBLAGE PAR ProCODE

0				
1				
2				
3				
4				
5				
6				
7				
8	ORG	ORG	\$300	
9				
10	BUFFER	EQU	\$200	; GET à partir d'un fichier T
11	CSR	EQU	\$06	; Case-réponse "poquée" en basic
12				
13				
14		LDA	BUFFER	
15		CMP	£\$0D	; Si "return", retour au basic !
16		BNE	INIT	
17		BEQ	REPONSE	; ...si la réponse est positive
18				
19	INIT	LDX	£\$00	; Pour tester les 5 voyelles
20	BOUCLE	CMP	ASC1,X	; Vraie comparée au contenu de A
21		BNE	INC	; Si négatif, on continue
22		LDA	ASC2,X	; Fausse est mise dans A
23				
24	REPONSE	STA	CSR	; Et c'est la réponse...
25		RTS		; ... suivie du retour
26				
27	INC	INX		; Plus 1 dans la boucle
28		CPX	£\$05	; Il y a 5 comparaisons à faire
29		BNE	BOUCLE	
30		BEQ	REPONSE	; Ce n'est pas AEIOU
31				
32	ASC1	ASC	"AEIOU"	; Les vraies
33	ASC2	ASC	"OAUEI"	; Les fausses
34				

TABLE DES SYMBOLES ASC1.....\$031D ASC2.....\$0322 BOUCLE ... \$030B BUFFER \$0200

ORDRE ALPHABÉTIQUE CSR\$0006 INC\$0316 INIT\$0309 REPONSE ... \$0313

A partir du même canevas, il est évidemment possible d'utiliser une autre clé pour coder un texte. Si le nombre des caractères à permuter est différent de 5, il faut alors modifier la valeur indiquée à la ligne 28 (octet \$318). Les adresses \$31D et \$322, ici fournies par ProCODE, seront facilement recalculées si l'on ne dispose pas d'un assembleur. Je vous rappelle que créer une ligne de DATA à partir d'une routine en assembleur est absolument enfantin. Prenons l'exemple de cette page (adresse 768 à 796). Il suffit de taper, en mode direct :

PRINT "]"450 DATA" ; : FOR I = 768 TO 796 : PRINT PEEK (I) ; "," ; : NEXT, puis un RETURN.

Une ligne 450 va s'afficher. Passer en mode ESCAPE, et la valider en allant de la première à la dernière lettre, en utilisant la flèche droite —>. Effacer la dernière virgule avant le RETURN final.

NESTOR.

Nota : Le] (clavier américain) correspond au § du clavier AZERTY. Qu'on se le dise dans les chaumières !

INITIATION

Découvrez l'assembleur

avec Madeleine HODÉ,
l'auteur de **GRIBOUILLE**

Programmer en assembleur, c'est le jeu le plus merveilleux que l'on puisse imaginer.

Après une pareille affirmation, beaucoup de lecteurs ont tourné la page. Je peux supposer que nous sommes seuls, vous qui êtes tenté par cette "récréation", moi qui voudrais vous aider à l'apprendre.

Toutefois, jeu n'est pas toujours synonyme de facilité, comme vous le savez déjà. D'accord ? Alors, nous pouvons commencer.

Utilisez d'abord le mini-assembleur

Le mini-assembleur vous est expliqué dans "La pratique de l'Apple II". C'est une facilité que donne le système pour écrire de petites routines et les tester rapidement, sans passer par tout le circuit des assembleurs. Circuit qui implique, outre l'écriture du programme, son enregistrement, et son assemblage avec manipulation de disquettes, erreurs variées à corriger, etc.

Le mini-assembleur est très simple, et il vous suffira amplement pour vérifier par de petits tests les mécanismes que vous voudrez étudier. Il présente pour le débutant un triple avantage :

1. il nécessite très peu de

connaissances (deux ou trois pages d'explications, alors qu'un manuel d'assembleur peut en compter 50 à 80).

2. il vous donne l'habitude de tester en direct vos idées avant de les intégrer à un programme en forme, plus ou moins lourd et plus ou moins rigide.

3. enfin, vous n'avez pas à l'acheter ; il est dans votre machine.

Quand vous serez sûr de votre vocation, achetez un assembleur. ProCODE, par exemple. Cela dit, votre liberté de choix reste entière, et personne ne se permettrait, ici, de vous influencer.

Matériel

Si vous êtes réellement tenté par l'assembleur, si

vous êtes en fonds, et si vous voulez que votre équipement soit vraiment complet, achetez-vous :

- une calculette de programmeur, qui fonctionne en hexadécimal aussi bien qu'en décimal, et qui traite les opérations logiques (AND ORA EOR).

Sachez tout de même que votre Apple fait les additions et les soustractions en hexadécimal. Sur un seul octet, et avec seulement deux nombres, certes, mais c'est déjà mieux que vos doigts. Comment ? Tapez CALL-151. Vous avez le moniteur. Tapez alors votre opération. Par exemple 79 + 32 RETURN vous amène le résultat = AB. A vous de penser aux retenues et d'en tenir compte.

- si vous avez un Apple IIe, (suite page 16)



achetez le kit qui lui donnera le processeur, la ROM et le décodeur de caractères du IIc. Vous aurez ainsi une machine plus rapide, plus puissante, avec un mini-assembleur très bien conçu et très commode qui vous fera passer sur votre clavier des nuits réellement magiques.

Pratique et exercices

Choisissez entre deux méthodes :

- utiliser un manuel d'initiation qui comporte beaucoup d'exercices. Si vous trouvez que les ouvrages qui sont indiqués ci-dessous

n'en présentent pas suffisamment, demandez conseil à votre vendeur.

- explorer, difficilement, lentement, mais fructueusement, les routines du moniteur.

Et pourquoi tout cet effort ?

Pour le plaisir. Pour la paix de l'âme, pour la vigueur et la santé du neurone.

Mais encore ?

Tout dépend de votre imagination. Si vous concevez facilement tout ce qu'un ordinateur pourrait faire pour vous amuser, ou

pour vous aider, si vous avez des envies, des besoins, des projets, aucun problème. Vous finirez par écrire vos programmes, et vous verrez que l'assembleur, pour certaines routines, est l'outil le plus adéquat, le plus élégant et le plus rapide.

Vous passerez avec virtuosité du Basic ou du Pascal à l'assembleur, selon des règles et des proportions que vous déterminerez à votre guise.

Et ceux qui n'ont pas cette forme d'imagination ? Ceux-là, il y a longtemps, n'est-ce pas, qu'ils ont tourné la page !...

Bibliographie

"Pratique des Apple" volume II, par A. Andrieux et G. Creuzet. Editions Radio.

"**Programmation du 6502**" de Rodney Zaks, éditions Sybex. (Vous pouvez vous arrêter avant le chapitre "*Techniques d'entrées-sorties*").

Pour passer des connaissances théoriques données par Rodney Zaks au pilotage de l'assembleur sur un Apple II: "**La pratique de l'Apple II**", 3^e volume, de Nicole Bréaud-Pouliquen et Daniel-Jean David, éditions du P.S.I.

Voilà déjà de bons pavés sur la route de l'assembleur. Il y a certainement d'autres excellents ouvrages d'initiation. Mais ne les connaissant pas, je n'en parlerai pas.

Il y a aussi les ouvrages spécialisés. Voici quelques titres, mais gardez-vous de les acheter avant d'avoir assimilé l'un des précédents :

- **Manuels de référence Apple IIe ou IIc** édités par Apple.
- **Manuels ProDOS** édités par Apple (ProDOS User's Kit etc.).

"**Beneath Apple ProDOS**", by Don Worth and Pieter Lechner, éditions Quality Software. L'ouvrage précédent des mêmes auteurs ("*Beneath Apple Dos*"), a livré les secrets du Dos 3.3 à tous les programmeurs de ma génération, et tous le connaissent — ou devraient le connaître — par cœur. Il reste l'ouvrage fondamental pour le Dos 3.3, dont la documentation est moins abondante que celle du ProDOS.

D'une manière générale, les ouvrages techniques édités par Apple sont clairs et complets. Ils apportent au programmeur expérimenté une satisfaction proche de la béatitude. Mais ils sont difficiles pour des débutants. Attendez un peu. Ainsi, quand vous les aborderez, vous découvrirez qu'ils répondent à des problèmes que vous vous serez déjà posés. Vous les comprendrez plus facilement, alors que leur intérêt vous échappera en grande partie si vous les attaquez trop tôt.

Il y a enfin tous les ouvrages que vous irez feuilleter dans les boutiques, et par lesquels vous vous laisserez tenter, à tort ou à raison. A dose modérée, les achats stupides sont quand même le sel de l'existence. ■

La Personne X

Logique élémentaire

IX personnes (désignées par leur prénom, c'est plus sympa !) sont ceci... et ne sont pas cela. D'après les renseignements, le programme trace le portrait-robot de la personne X, qu'il s'agit évidemment d'identifier.

Ce jeu repose sur une logique élémentaire qu'il n'est pas interdit de développer (à vous de jouer !).

Le programme actuel peut être complété par une page d'explications et par d'autres lignes de données (DATA).



Il est plus simple d'utiliser plusieurs séries de variables qu'un seul tableau.

On peut choisir de jouer avec les données du programme : c'est plus rapide que de les saisir.

CALL-198 actionne la cloche (CTRL-G).

Aucune logique n'est possible si on donne le même trait de caractère à plusieurs personnes.

```

10 TEXT : PRINT CHR$(21): HOME : CLEAR
20 DIM S$(6),S1$(6,3),S2$(6,3):SB = 3
30 FOR I = 1 TO 10:T$ = T$ + "----": NEXT
40 FOR I = 1 TO 19:V$ = V$ + " ": NEXT
50 :
60 REM *****
70 REM * SAISIE OU DONNEES EN DATA *
80 REM *****
90 :
100 VTAB 8: HTAB 8: INVERSE : PRINT 1;: NORMAL : PRINT
    " SAISIE DES ELEMENTS": PRINT : PRINT : HTAB 8: IN
    VERSE : PRINT 2;: NORMAL : PRINT " ELEMENTS DU PROG
    RAMME"
110 GOSUB 190: IF W = 1 THEN 420
120 IF W = 2 THEN GOSUB 1160:S = 6: GOTO 690
130 GOTO 110
140 :
150 REM *****
160 REM * ATTENTE SANS CURSEUR *
170 REM *****
180 :
190 CALL - 198: POKE - 16368,0: WAIT - 16384,128,12
    7: POKE - 16368,0:W = PEEK ( - 16384) - 48: RETUR
    N
200 :
210 REM *****
220 REM * CHASSE AUX REPETITIONS *
230 REM *****
240 :
250 TR$ = "MOT DEJA UTILISE - A CHANGER"
260 IF LEN (S1$(S,S1)) > 10 OR LEN (S2$(S,S2)) > 10
    THEN TR$ = "UN SYNONYME DE MOINS DE 11 LETTRES SVP"
    : GOTO 360
270 FOR I = 1 TO S: FOR J = 1 TO SB: IF PR = 3 THEN 32
    0
  
```

(suite au verso)

```

280 IF S = I AND S1 = J THEN 300
290 IF S1$(S,S1) = S1$(I,J) THEN 360
300 IF S1$(S,S1) = S2$(I,J) THEN 360
310 GOTO 350
320 IF S = I AND S2 = J THEN 340
330 IF S2$(S,S2) = S2$(I,J) THEN 360
340 IF S2$(S,S2) = S1$(I,J) THEN 360
350 NEXT : NEXT : RETURN
360 PRINT : VTAB 22: CALL - 958: PRINT TR$: POP : ON
PR GOTO 510,510,550
370 :
380 REM *****
390 REM * SAISIE *
400 REM *****
410 :
420 S = 0:S1 = 0:S2 = 0
430 HOME : INVERSE : PRINT "LE SUJET";: NORMAL : PRINT
" (UN SEUL MOT-6 SUJETS AU TOTAL)"
440 VTAB 2: PRINT T$
450 S1 = 0:S2 = 0
460 S = S + 1:PR = 1: PRINT : VTAB 4: CALL - 958
470 PRINT : VTAB 4: INVERSE : CALL - 198: PRINT S;: N
ORMAL : INPUT " -> ";S$(S): IF S$(S) = "" THEN 470
480 GOSUB 610
490 PRINT : VTAB 4: CALL - 958: INVERSE : PRINT LEFT
$(S$(S),13);: NORMAL : PRINT " EST":PR = 2
500 S1 = S1 + 1
510 PRINT : VTAB S1 + 5: CALL - 198: PRINT S1" - ";:
INPUT " ";S1$(S,S1): IF S1$(S,S1) = "" THEN 510
520 GOSUB 250: GOSUB 610
530 IF S1 < 3 THEN 500
540 S2 = S2 + 1
550 PRINT : VTAB 4: HTAB 20: INVERSE : PRINT LEFT$(S
$(S),10);: NORMAL : PRINT " N'EST PAS":PR = 3
560 PRINT : VTAB S2 + 5: HTAB 20: CALL - 198: PRINT S
2" - ";: INPUT " ";S2$(S,S2): IF S2$(S,S2) = "" THEN
560
570 GOSUB 250: GOSUB 610
580 IF S2 < 3 THEN 540
590 IF S < 6 THEN 450
600 GOTO 690
610 POKE - 16368,0: VTAB 22: CALL - 868: CALL - 198
: INVERSE : PRINT "RETURN";: NORMAL : PRINT " POUR
CONTINUER OU ESCAPE ";: GET R$:R = ASC (R$): IF R
= 13 THEN RETURN
620 PRINT : IF R < > 27 THEN PRINT : GOTO 610
630 POP : ON PR GOTO 470,510,550
640 :
650 REM *****
660 REM * LA PERSONNE X PREND FORME *
670 REM *****
680 :
690 Q1 = 1 + INT ( RND (1) * S):Q2 = 1 + INT ( RND (1
) * SB)

```

← La chasse aux répétitions doit tenir compte des deux séries de renseignements : EST — N'EST PAS.

← PR est un pointeur qui renvoie exactement où il faut.

← CALL-958 vide le bas de l'écran.

← Si le sujet comporte plus de 13 caractères, ceux-ci ne sont pas affichés.

La longueur des renseignements est limitée à 10 caractères. Elle est contrôlée au début du sous-programme "Chasse aux répétitions", ligne 260.

← Pour chaque mot rentré, il est demandé de confirmer par un RETURN.

Q1 et Q2 déterminent quel sera le renseignement permettant d'identifier la personne X.

Copie de l'écran,
obtenue à l'issue
de la saisie
des données.



autre choix aléatoire ➔

Ces quelques lignes éta-
blissent le portrait-robot
de la personne X.

```

700 X = INT ( RND (1) * 2):ER = 0:IC = ER:IP = ER:FIN
    = ER
710 FOR I = 1 TO 3:H = 1 + INT ( RND (1) * SB)
720 X = X + 1: IF X = 2 THEN X = 0
730 IF Q1 < > I THEN 760
740 IF X = 0 THEN IP = IP + 1:S1$(0,IP) = S1$(1,Q2): G
    OTO 780
750 IC = IC + 1:S2$(0,IC) = S2$(1,Q2): GOTO 780
760 IF X = 1 THEN IC = IC + 1:S2$(0,IC) = S1$(1,H): GO
    TO 780
770 IP = IP + 1:S1$(0,IP) = S2$(1,H)
780 NEXT I
790 :
800 REM *****
810 REM * IL S'AGIT DE L'IDENTIFIER *
820 REM *****
830 :
840 TEXT : HOME
850 HTAB 8: INVERSE : PRINT " L A   P E R S O N N E   X
    ": NORMAL : PRINT T$
860 VTAB 4: INVERSE : PRINT V$;; HTAB 22: PRINT V$: VT
    AB 14: PRINT V$;; HTAB 22: PRINT V$: FOR I = 5 TO 1
    3: VTAB I: HTAB 1: PRINT " ";; HTAB 19: PRINT " ";;
    HTAB 22: PRINT " ";; HTAB 40: PRINT " ": NEXT
870 VTAB 6: HTAB 3: PRINT "EST:";; HTAB 24: PRINT "N'E
    ST PAS:" : NORMAL
880 FOR I = 1 TO 3: VTAB 6 + I * 2: HTAB 3: PRINT S1$(
    0,I);; HTAB 24: PRINT S2$(0,I): NEXT
890 POKE 34,15: PRINT : HOME
900 VTAB 16: PRINT "QUEL EST LE NOM DE ";; INVERSE : P
    RINT "X": NORMAL : PRINT LEFT$(T$,20)
910 FOR I = 1 TO 3: VTAB 17 + I: HTAB 1: INVERSE : PRI

```

(suite au verso)

L'affichage obtenu ici est
celui dont nous vous
donnons une reproduction
en haut de la page.

LA PERSONNE X (suite et fin)

```

NT I;; HTAB 19: PRINT I + 3;; NORMAL : HTAB 3: PRIN
T S$(I);; HTAB 21: PRINT S$(I + 3): NEXT
920 VTAB 16: HTAB 22: FLASH : PRINT " ";; NORMAL : CAL
L - 198: GET R$: POKE - 16368,0:R = ASC (R$) - 4
8: IF R < 1 OR R > 6 THEN 920
930 VTAB 16: HTAB 22: IF R = Q1 THEN CALL - 198: PRI
NT "VOUS AVEZ GAGNE!":FIN = 1: GOTO 950
940 PRINT "HELAS! NON":ER = ER + 1
950 CALL - 958: PRINT : PRINT "CAR "S$(R): PRINT : PR
INT "- EST: ";; FOR I = 1 TO 3: PRINT S1$(R,I)" ";;
NEXT : PRINT "": PRINT "- N'EST PAS: ";; FOR I = 1
TO 3: PRINT S2$(R,I)" ";; NEXT : PRINT ""
960 VTAB 23: PRINT "AVEC "ER" ERREUR";: IF ER > 1 THEN
PRINT "S"
970 GOSUB 190
980 IF FIN = 1 THEN 1010
990 IF ER < 6 THEN 890
1000 FIN = 1: VTAB 16: HTAB 22: CALL - 868: INVERSE : P
RINT S$(Q1): NORMAL :R = Q1: GOTO 950
1010 GOSUB 190
1020 :
1030 REM *****
1040 REM * A SUJURE OU FIN DE JEU *
1050 REM *****
1060 :
1070 VTAB 22: PRINT : CALL - 198: INVERSE : PRINT 1;;
NORMAL : PRINT " AUTRE VERSION ";: INVERSE : PRINT
2;; NORMAL : PRINT " JEU COMPLET ";: INVERSE : PRIN
T 3;; NORMAL : PRINT " TERMINE";: GET R$
1080 POKE - 16368,0:R = ASC (R$) - 48: IF R < 1 OR R
> 3 THEN 1070
1090 ON R GOTO 690,10,1100
1100 TEXT : HOME : END
1110 :
1120 REM *****
1130 REM * POUR TOUS LES PARESSEUX *
1140 REM *****
1150 :
1160 FOR I = 1 TO 6: READ S$(I): FOR J = 1 TO 3: READ S
1$(I,J): NEXT J: FOR K = 1 TO 3: READ S2$(I,K): NEX
T K: NEXT I: RETURN
1170 DATA LOUISE,TIMIDE,GRANDE,FERMIERE,POITEVINE,RAPID
E,BRUNE,ALBERTINE,MENTEUSE,FACETIEUSE,RIEUSE,PARISI
ENNE,RIDEE,TOUCHANTE
1180 DATA ANTOINETTE,BLONDE,GAIE,BELLE,SENSIBLE,AFFABLE
,REVEUSE,EULALIE,PAUVRE,FAIBLE,DOUCE,GROSSE,ANGEVIN
E,MARIEE
1190 DATA COLETTE,DUBLIEUSE,INSENSIBLE,CORIACE,RANCUNIE
RE,EPICIERE,SECHE,CHRISTINE,GENTILLE,MEDECIN,POTELE
E,ROUSSE,CURIEUSE,VELLEITAIRE

```

◀ On comptabilise som-
mairement les erreurs.

◀ En 190 se trouve la bou-
cle d'attente.

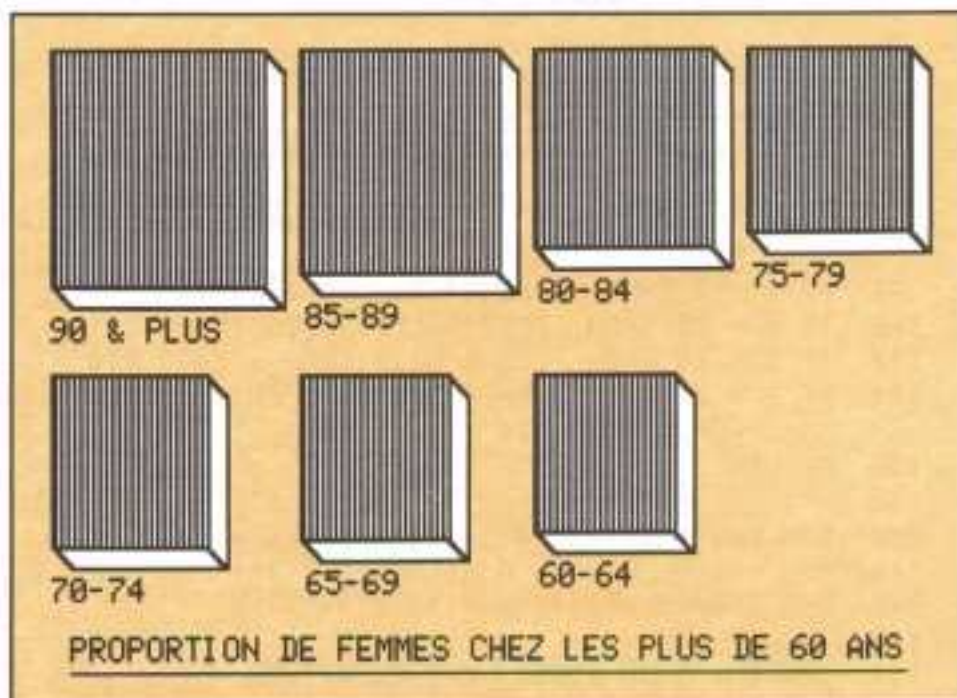
◀ Si le nombre d'erreurs
atteint 6, il faut en tout
cas afficher la réponse.

◀ Le jeu peut recommen-
cer autant de fois que
l'on veut avec les mêmes
données. L'identification de
la Personne X n'en devient
pas, pour autant, enfantine
(contrairement à ce que l'on
pourrait penser).

Naturellement, vous pouvez
modifier et compléter cette
partie du programme. Il
serait alors possible d'aller
aléatoirement à la pêche
aux renseignements !

VISUBLOCS

**Une autre
manière
de
visualiser
vos
nombres**



SORTEZ de l'ordinaire et donnez de nouvelles dimensions aux chiffres qui émaillent vos rapports... ou, plus simplement, à ceux qui traduisent vos dépenses annuelles (les loisirs par exemple). Le programme **VISUBLOCS** travaille dans la page graphique numéro 2 (HGR2), à l'adresse \$4000 (16384), et nous avons installé la FONTE de caractères à l'adresse 12288 (\$3000). En supprimant tous les REM, il est possible d'utiliser la page HGR, mais cela ne présente pas d'intérêt particulier sur les APPLE actuels.

Vous comprendrez facilement le fonctionnement de VISUBLOCS, au fur et à mesure de la copie de son listage.

Un conseil : Plutôt que de suivre l'ordre logique des lignes de programme, essayez de taper les instructions dans l'ordre exact où elles vont se dérouler. Ainsi, après la ligne 20, passez à l'adresse 2060, puis à la ligne 310, etc. C'est une démarche apparemment désordonnée, mais qui a le mérite de respecter la structure (bonne ou mauvaise) du programme que l'on étudie.

Saisie des données : Le menu ne comporte pas d'option "Correction de données", mais celle-ci existe, en passant par "Créer un fichier de données". Si l'on choisit l'option "Modifications" (ligne 1270), les données en mémoire sont visualisées et on peut ou bien les modifier, ou les conserver par un simple RETURN.



VISUBLOCS.2

```

10 LOMEM: 24580
20 GOSUB 2060: GOTO 310
30 :
40 REM *****
50 REM *** ATTENTE SANS CURSEUR ***
60 REM *****
70 :
80 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: POKE - 16368,0:
  B = PEEK ( - 16384)
90 IF B = 9 THEN POKE 34,21: GOTO 1990
100 IF B = 10 THEN POKE 34,21: GOTO 1980
110 IF B = 18 THEN GOSUB 2120: POP : GOTO 310
120 IF B = 63 THEN HOME : PRINT : PRINT D$"CATALOG": GET R$: PRINT :R$ =
  "": POP : GOTO 310
130 RETURN
140 :
150 REM *****
160 REM *** BOUCLE D'ATTENTE ***
170 REM *****
180 :
190 FOR E = 1 TO 1500: NEXT E: RETURN
200 :
210 REM *****
220 REM *** CONFIRMER LE CHOIX ***
230 REM *****
240 :
250 PRINT "Confirmez votre choix par un " CHR$ (34)"oui" CHR$ (34): RET
  URN
260 :
270 REM *****
280 REM *** MENU GENERAL ***
290 REM *****
300 :
310 GOSUB 1110
320 PRINT : HOME : VTAB 1: INVERSE : PRINT "          QUEL EST VOTRE CHOIX
  ?          ";; NORMAL : PRINT "-----"
  -"
330 FOR I = 1 TO M: HTAB 1: VTAB 2 + I * 2: INVERSE : PRINT I;; NORMAL :
  PRINT "> " MID$(M$,I + 30 - 29,30): NEXT
340 FOR I = 1 TO M: HTAB 32: VTAB 2 + I * 2: PRINT ".....": NEXT
350 CALL - 950
360 IF R = 1 THEN R = 0: GOTO 420
370 VTAB 22: PRINT " * TAPER UN (<; INVERSE : PRINT "?"; NORMAL : PR
  INT "> POUR LE CATALOGUE"
380 GOSUB 80:C = B - 48: IF C < 1 OR C > M THEN 380
390 VTAB 2 + (B - 48) * 2: HTAB 38: PRINT "<-"; FLASH : PRINT C: NORMAL
  : VTAB 22: HTAB 4: GOSUB 250: GOSUB 80: IF B < > ASC ("0") AND B <
  > ASC ("o") THEN 340
400 VTAB 22: HTAB 1: PRINT T$
410 SPEED= 150: INVERSE : FOR I = 37 TO 32 STEP - 1: HTAB 1: VTAB 2 + C
  * 2: PRINT " "; NEXT : NORMAL : SPEED= 255
420 ON C GOTO 1230,1760,440,430,1640,510,2230,2300

```

| VISUBLOCS.3 |

```

430 GOSUB 1050: HTAB 1: VTAB 22: CALL - 958: INVERSE : PRINT "PAGE GRAPH
    IQUE EFFACEE": NORMAL : GOTO 340
440 IF HG = 0 THEN 2230
450 GOSUB 990: GOSUB 80: GOSUB 1110: GOTO 340
460 :
470 REM *****
480 REM *** PAGE GRAPHIQUE (TRACE) ***
490 REM *****
500 :
510 IF N < 2 THEN VTAB 20: HTAB 5: INVERSE : PRINT "DONNEES TOTALEMENT I
    NEXISTANTES": NORMAL : GOSUB 1170: GOTO 340
520 HGR2 : HCOLOR= 3: VTAB 19: CALL - 958
530 G = INT (N / 2): IF G < > N / 2 THEN G = G + 1
540 J = 278: FOR I = 1 TO G - 1: J = J - 11.5: NEXT
550 F = 0
560 FOR I = 1 TO N - 1: IF VAL (A$(I,1)) > = VAL (A$(I + 1,1)) THEN 58
    0
570 A$ = A$(I,1):A$(I,1) = A$(I + 1,1):A$(I + 1,1) = A$:A$ = A$(I,0):A$(I,
    0) = A$(I + 1,0):A$(I + 1,0) = A$:A$ = A$(I,2):A$(I,2) = A$(I + 1,2):A
    $(I + 1,2) = A$:F = 1
580 NEXT : IF F = 1 THEN 550
590 T = 0:S = T:X = T:Y = G: FOR I = 1 TO N STEP 2:X = X + 1:Y = Y + 1
600 B$(I,0) = A$(X,0):B$(I,2) = A$(X,2):B$(I,1) = A$(I,1): IF Y < = N THE
    N B$(I + 1,0) = A$(Y,0):B$(I + 1,2) = A$(Y,2):B$(I + 1,1) = A$(I + 1,1
    )
610 T = T + VAL (B$(I,2)): NEXT
620 U = J / T:K = 0:O = - 10:P = 10:Q = 1:U = 0:W = 1:Z = 0
630 K = K + 1
640 S = ( VAL (B$(K,2))) * U: IF K = 1 AND S + 35 + ( VAL (B$(K + 1,2))) *
    U > 165 THEN P = P + .10:U = U - .10: GOTO 640
650 HCOLOR= 2:U = U + 1: IF U = 1 THEN O = O + Q + P:W = 1
660 IF U = 2 THEN W = Z + 25:U = 0
670 FOR I = W TO W + S - 1 STEP .41: HPLLOT O + S - 1,I TO O,I: NEXT
680 HCOLOR= 7: IF K = G THEN SS = O + S + 7
690 HPLLOT O,W TO O,W + S - 1 TO O + S - 1,W + S - 1 TO O + S - 1,W TO O,W
700 :
710 REM *****
720 REM *** LEGENDES DES BLOCS ***
730 REM *****
740 :
750 VT = W + S + 8:L = LEN (B$(K,0)): IF L > (S + 7) / 6 THEN L = (S + 7)
    / 6:B$ = LEFT$(B$(K,0),L): GOTO 770
760 B$ = B$(K,0)
770 DE = S - L * 6:DE = DE + L * 6
780 IF O > 1 THEN H = O + 2 + S - DE: GOTO 800
790 H = 1 + S - DE: IF H < 1 THEN H = 1
800 FOR I = 1 TO L:AS = ASC (MID$(B$,I,1)) - 31: DRAW AS AT H,VT:H = H
    + 6: NEXT
810 HPLLOT O,W + S - 1 TO O + 6,W + S + 5 TO O + S + 5,W + S + 5 TO O + S
    - 1,W + S - 1
820 HPLLOT O + S + 5,W + S + 5 TO O + S + 5,W + 6 TO O + S - 1,W
830 IF S > Z THEN Z = S

```



I VISUBLOCS.4 I

```

1310 N = 8
1320 N = N + 1: IF N > 12 THEN N = N - 1: GOTO 1530
1330 VTAB 3 + N: E$ = " ": IF N > 9 THEN E$ = ""
1340 CALL - 868: PRINT E$N" -> .....";: IF MO < > 0 THEN HTAB 7:
    PRINT A$(N,0);
1350 CALL - 198: HTAB 7: INPUT ";A$: IF A$ = "FIN" OR A$ = "fin" OR A$ =
    "Fin" THEN N = N - 1: GOTO 1530
1360 IF A$ = "*" AND N > 1 THEN N = N - 2: GOTO 1320
1370 IF A$ = "***" OR (N = 1 AND A$ = "*") THEN 320
1380 IF A$ = "" AND MO < > 0 THEN A$ = A$(N,0): GOTO 1410
1390 IF A$ = "" THEN 1330
1400 IF LEN (A$) > 12 THEN A$ = LEFT$ (A$,12)
1410 VTAB 3 + N: HTAB 7: CALL - 868: PRINT A$;: HTAB 25: PRINT ".....
    .";: IF MO < > 0 THEN HTAB 25: PRINT A$(N,1);
1420 CALL - 198: HTAB 25: INPUT ";B$: IF B$ = "" AND MO < > 0 THEN B$ =
    A$(N,1)
1430 IF B$ = "" OR B$ = "*" THEN 1330
1440 FOR I = 1 TO LEN (B$): IF ASC ( MID$ (B$,I,1)) < 48 OR ASC ( MID$
    (B$,I,1)) > 57 THEN VTAB 3 + N: HTAB 24 + I: FLASH : PRINT MID$ (B$,
    I,1): NORMAL : CALL - 198: CALL - 198: GOSUB 190: GOTO 1410
1450 NEXT
1460 VTAB 20: HTAB 1: GOSUB 250: GOSUB 80: IF B < > ASC ("0") AND B < >
    ASC ("o") THEN 1410
1470 VTAB 20: CALL - 868
1480 HTAB 24: VTAB 3 + N: CALL - 868: HTAB 24 + 10 - LEN (B$): PRINT B$:
    A$(N,0) = A$:A$(N,1) = B$:A$ = "":B$ = ""
1490 T = 0: FOR I = 1 TO N:T = T + VAL (A$(I,1)): NEXT I
1500 FOR I = 1 TO N:A$(I,2) = STR$ ( INT ( VAL (A$(I,1)) / T * 10000) / 1
    00): NEXT : IF A = 1 THEN A = 0: RETURN
1510 FOR I = 1 TO N: VTAB 3 + I: HTAB 35: CALL - 868: HTAB 38 - LEN ( ST
    R$ ( INT ( VAL (A$(I,2))))): PRINT A$(I,2): NEXT
1520 GOTO 1320

1530 R = 1: GOSUB 1640
1540 VTAB 20: HTAB 1: CALL - 868:F$ = ""
1550 INPUT "TITRE DU FICHER ";F$: IF F$ = "" THEN VTAB 20: HTAB 1: INV
    ERSE : PRINT "CONFIRMEZ NON-ENREGISTREMENT PAR " CHR$ (34)"NON" CHR$
    (34): NORMAL : GOSUB 80: IF B < > ASC ("N") AND B < > ASC ("n") T
    HEN 1540
1560 IF F$ = "" THEN 1580
1570 VTAB 20: HTAB 1: CALL - 868: INVERSE : PRINT "ENREGISTREMENT "F$: NO
    RMAL : VTAB 19: PRINT D$"OPEN"F$: PRINT D$"WRITE"F$: PRINT N: PRINT C$
    : FOR I = 1 TO N: PRINT A$(I,0): PRINT A$(I,1): NEXT : PRINT D$"CLOSE"
    F$
1580 D = 1: GOTO 310
1590 :
1600 REM *****
1610 REM *** TITRE DU FICHER ***
1620 REM *****
1630 :
1640 HTAB 1: VTAB 20: CALL - 958: INVERSE : PRINT "TITRE DU GRAPHIQUE": N
    ORMAL
1650 VTAB 22: HTAB 4: FOR I = 1 TO 11: PRINT "...";: NEXT : PRINT

```

VISUBLOCS.5

```
1660 VTAB 22: HTAB 1: PRINT "-> "C$;: VTAB 22: HTAB 4: INPUT "";A$: IF A$
= "" THEN 310
1670 IF LEN (A$) > 44 THEN 1640
1680 VTAB 20: HTAB 1: CALL - 950: INVERSE : PRINT A$: NORMAL : VTAB 22: H
TAB 1: GOSUB 250: GOSUB 80: IF B < > ASC ("0") AND B < > ASC ("o")
THEN 1640
1690 C$ = A$:A$ = "": IF R = 1 THEN R = 0: RETURN
1700 GOTO 310
1710 :
1720 REM *****
1730 REM *** CHARGER UN FICHIER ***
1740 REM *****
1750 :
1760 VTAB 20: HTAB 1: CALL - 950: INVERSE : PRINT "TITRE DU FICHIER";: NO
RMAL : INPUT " ";F$: IF F$ = "?" THEN TEXT : HOME : PRINT : PRINT
CHR$ (4)"CATALOG": GOSUB 80: HOME :R = 1:C = 2: GOTO 310
1770 IF F$ = "" THEN 310
1780 PRINT : VTAB 20: HTAB 1: CALL - 950: INVERSE : PRINT "CHARGEMENT DE
"F$: NORMAL : PRINT : ONERR GOTO 1820
1790 VTAB 18: PRINT : PRINT D$"OPEN"F$: PRINT D$"READ"F$: INPUT N: INPUT C
$: FOR I = 1 TO N: INPUT A$(I,0): INPUT A$(I,1): NEXT : PRINT D$"CLOSE
"F$
1800 A = 1: GOSUB 1490: POKE 216,0
1810 D = 1: GOTO 340
1820 IF PEEK (222) = 5 THEN PRINT CHR$ (4)"DELETE"F$: VTAB 20: HTAB 1:
CALL - 950: FLASH : PRINT "FICHIER "F$" INEXISTANT": NORMAL : GOSUB 1
170: GOSUB 80
1830 POKE 216,0: GOTO 1760
1840 :
1850 REM *****
1860 REM *** FONTE DE CARACTERES ***
1870 REM *****
1880 :
1890 IF DL = 1 THEN RETURN
1900 AD = 12288: ROT= 0: SCALE= 1:F$ = "VISUFONTE"
1910 VTAB 21: PRINT : PRINT D$"BLOAD"F$,A*AD:DL = 1
1920 POKE 232,AD - INT (AD / 256) * 256: POKE 233, INT (AD / 256): RETURN
1930 :
1940 REM *****
1950 REM *** EDITION (DMP+CARTE BIP)***
1960 REM *****
1970 :
1980 PRINT : PRINT D$"PRÉ1": PRINT CHR$ (27)"L005": PRINT CHR$ (9)"GLED2
": GOTO 2000
1990 PRINT : PRINT D$"PRÉ1": PRINT CHR$ (27)"L020": PRINT CHR$ (9)"GLE2"
2000 PRINT : PRINT CHR$ (4)"PRÉ0": TEXT : GOTO 340
2010 :
2020 REM *****
2030 REM *** INITIALISATION ***
2040 REM *****
2050 :
2060 TEXT : PRINT CHR$ (12); CHR$ (21): HOME : HGR2
```



| VISUBLOCS.6 |

```

840 IF U = 1 THEN Q = S
850 IF K = 2 THEN VM = VT
860 IF K < N THEN 630
870 IF C$ = "" THEN 930
880 L = LEN (C$):H = (279 - L * 6) / 2:VT = VM + 17
890 H PLOT H,VT + 10 TO H + L * 6,VT + 10
900 FOR I = 1 TO L:AC = ASC (MID$ (C$,I,1)) - 31: IF AC = 32 THEN H = H
    + 6: GOTO 920
910 DRAW AC AT H,VT:H = H + 6
920 NEXT
930 HG = 1: GOSUB 80: GOSUB 1110: GOTO 340
940 :
950 REM *****
960 REM *** HGR2 SANS EFFACEMENT ***
970 REM *****
980 :
990 X = PEEK (49232) + PEEK (49234) + PEEK (49237) + PEEK (49239): RET
    URN
1000 :
1010 REM *****
1020 REM *** EFFACEMENT HGR2 $4000 ***
1030 REM *****
1040 :
1050 POKE 230,64: HCOLOR= 4: H PLOT 0,0: CALL - 3082:HG = 0: RETURN
1060 :
1070 REM *****
1080 REM *** RETOUR MODE TEXT ***
1090 REM *****
1100 :
1110 TEXT : RETURN
1120 :
1130 REM *****
1140 REM *** BRUIT DE BALLE ***
1150 REM *****
1160 :
1170 FOR I = 0 TO 30: FOR J = 49200 TO 49230 - I:X = PEEK (J): NEXT J,I:
    RETURN
1180 :
1190 REM *****
1200 REM *** SAISIE DES DONNEES ***
1210 REM *****
1220 :
1230 GOSUB 1110
1240 HOME : VTAB 1: INVERSE : PRINT " SAISIE DES DONNEES
    ";: NORMAL : PRINT "-----"
1250 POKE 34,3:MO = 0
1260 HTAB 1: INVERSE : PRINT "MAXIMUM=12": NORMAL : GOSUB 1170: HOME
1270 VTAB 22: PRINT "<0> MODIFICATIONS <1> NOUVELLES DONNEES": GOSUB 80: I
    F B = 49 THEN 1300
1280 IF B < > 48 THEN PRINT : GOTO 1270
1290 MO = N
1300 VTAB 22: PRINT T$: HTAB 8: PRINT "Taper " CHR$ (34)"fin" CHR$ (34)" p
    our terminer"

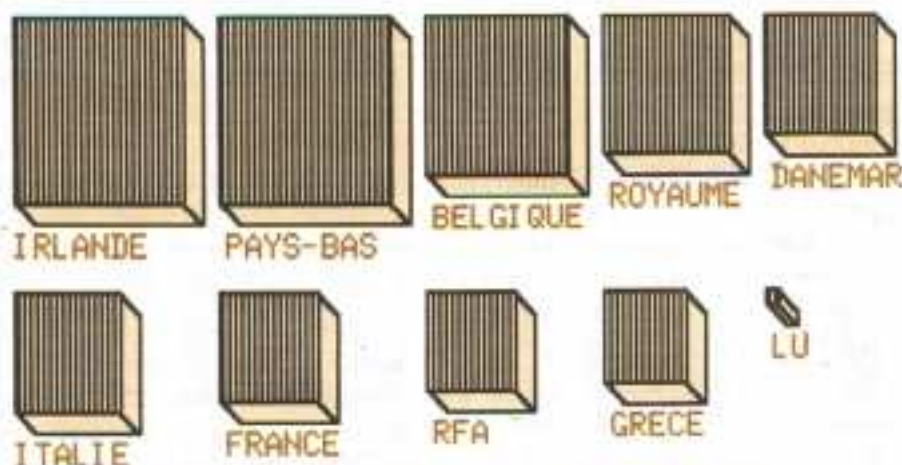
```

| VISUBLOCS.7 |

```

2070 DIM A$(13,2),B$(13,2)
2080 C$ = "....."
2090 FOR I = 1 TO 10:T$ = T$ + "____": NEXT
2100 O = - 10:Z = 1:P = 10:D$ = CHR$(4)
2110 M$ = "CREER UN FICHER DE DONNEES    CHARGER UN FICHER EXISTANT    VOIR
        LA PAGE GRAPHIQUE              EFFACER LA PAGE GRAPHIQUE    CHANGER LE TIT
RE DE LA PAGE    TRACER LA PAGE GRAPHIQUE    ENREGISTRER LE GRAPHIQUE
        FIN DE TRAVAIL"
2120 TEXT : HOME :MB = R:M = 8
2130 VTAB 8: PRINT "CTRL-I PERMET D'EDITER LA PAGE GRAPHIQUE          EN
        FORMAT SIMPLE              CTRL-J PERMET DE REALISER CETTE EDITION
        EN FORMAT SIMPLE"
2140 PRINT : PRINT "PENDANT LA SAISIE DES DONNEES          -----
        -----              UN * RENVOIE A LA DONNEE PRECEDENTE    ET *
* RENVOIE AU MENU GENERAL, SANS POUR    AUTANT ANNULER CE QUI A ETE
SAISI."
2150 PRINT : PRINT "CTRL-R REAFFICHE CETTE PAGE DE FONCTIONS"
2160 GOSUB 1160: GOSUB 80: IF MB = 18 THEN MB = 0: RETURN
2170 TEXT : GOSUB 1890: RETURN
2180 :
2190 REM *****
2200 REM *** ENREGISTRER LA PAGE 2 ***
2210 REM *****
2220 :
2230 IF HG = 0 THEN VTAB 20: HTAB 8: INVERSE : PRINT "PAGE GRAPHIQUE ENCO
RE VIERGE": NORMAL : GOSUB 1170: GOTO 340
2240 VTAB 20: PRINT : INPUT "TITRE ";G$: IF G$ = "" THEN 340
2250 ONERR GOTO 2280
2260 VTAB 20: PRINT : PRINT D$"BSAVE"G$,A$4000,L8192"
2270 VTAB 20: PRINT : PRINT "PAGE 2 SAUVEGARDEE SUR DISQUETTE": GOSUB 190:
POKE 216,0: GOTO 340
2280 POKE 216,0: VTAB 20: PRINT "ERREUR NUMERO * PEEK (222): GOSUB 190: GO
TO 340
2290 POKE 51,0: GOTO 310
2300 TEXT : HOME : VTAB 20: PRINT "$GOTO2290 VOUS RENVOIE AU MENU": VTAB 1
9: END

```



TAUX DE CHOMAGE COMPARE (EN 1984)

VISUBLOCS.8

VISUFONTE *TAPEZ CALL-151, suivi de RETURN, puis enregistrez la totalité des données (attention ! chaque enregistrement doit commencer par le numéro de ligne, suivi de deux points, comme indiqué ci-après). Exemple :*

*** 3000 : 64 01 D0 00 D9 00 E0 00 (puis RETURN)**

3000- 64 01 D0 00 D9 00 E0 00
 3008- E8 00 F9 00 0A 01 1A 01
 3010- 29 01 2F 01 39 01 42 01
 3018- 53 01 5E 01 65 01 6E 01
 3020- 74 01 7F 01 91 01 9A 01
 3028- A8 01 B5 01 C2 01 D1 01
 3030- E2 01 ED 01 FE 01 0F 02
 3038- 15 02 1C 02 30 02 3B 02
 3040- 4F 02 5A 02 6A 02 7C 02
 3048- 8D 02 9B 02 AA 02 B9 02
 3050- C7 02 D5 02 E7 02 F1 02
 3058- FC 02 0B 03 13 03 25 03
 3060- 37 03 48 03 56 03 68 03
 3068- 79 03 89 03 93 03 A4 03
 3070- B4 03 C6 03 D6 03 E2 03
 3078- EE 03 D0 00 D0 00 D0 00
 3080- D0 00 D0 00 D0 00 D0 00
 3088- D0 00 D0 00 D0 00 D0 00
 3090- D0 00 D0 00 D0 00 D0 00
 3098- D0 00 D0 00 D0 00 D0 00
 30A8- D0 00 D0 00 D0 00 D0 00
 30B8- D0 00 D0 00 D0 00 D0 00
 30C0- D0 00 F8 03 F8 03 F8 03
 30C8- FD 03 22 01 00 00 00 00
 30D0- 09 09 92 92 00 00 92 0A
 30D8- 00 36 36 16 4E 00 00 00
 30E0- 0D FE 6E 96 52 01 00 3E
 30E8- 69 FE 17 2D 2D 1E 1F 17
 30F0- 2D 2D 1E 1F 6E 4E 01 00
 30F8- 3E 09 8D 3F BF 0D 15 3F
 3100- 0E 0D 1E 3F 77 71 49 00
 3108- 00 6D 6D 11 DF 77 09 1E
 3110- 17 4D 3A DF 4E 29 4E 00
 3118- 00 2D A9 1F 6E 1E 17 0D
 3120- 8D 1E DF 8E 6D 4E 00 00
 3128- 2D 36 96 52 01 00 00 09
 3130- 1E 17 36 15 0E 4E 00 00
 3138- 35 15 0E 36 1E 17 4E 09
 3140- 00 35 09 8D 1F 1F 0E 2D
 3148- 1E 17 2D 15 1F 1F 4E 4E
 3150- 09 00 FF 09 6E 3A 3F 77
 3158- 31 56 49 00 00 92 92 0A
 3160- F6 4D 00 00 37 52 89 3F

3168- B7 52 49 01 00 00 92 92
 3170- 4E 00 00 00 49 11 17 1E
 3178- 17 1E 56 49 09 00 35 29
 3180- AD DF 33 4D 35 1F 1F 2E
 3188- 4D FE 1B 0E 2D 4E 01 00
 3190- 6D 31 77 36 BE 2D 4E 00
 3198- 00 00 29 AD DF 73 49 1E
 31A0- BF 1E 2E 2D 75 01 00 DF
 31A8- 2D 2D BE 3E 4E FE 1B 0E
 31B0- 2D 4E 01 00 3E 49 3E 17
 31B8- 0D FE 33 2D 2D 1E 76 09
 31C0- 00 35 2D 2D DE 1B 2E 2D
 31C8- 15 36 DF 73 2D 4E 01 00
 31D0- 11 09 2D DE BB 4D 3A 3F
 31D8- 6E 09 FE 1B 0E 2D 4E 01
 31E0- 00 2E 2D 2D BE 1E 17 36
 31E8- 4E 49 00 00 29 29 AD DF
 31F0- 33 4D F1 3F 17 4D 31 DF
 31F8- 73 2D 4E 01 00 2E 29 AD
 3200- DF 33 4D 31 3F 77 09 1E
 3208- DF 2A 75 49 00 00 52 12
 3210- 16 96 09 00 3E 52 B2 F6
 3218- 4D 00 00 B7 49 3E 17 2D
 3220- 2D 35 3F 3F 3F 0E 2D 2D
 3228- F5 1B 77 4E 49 00 00 09
 3230- 12 2D 2D DE 9B 2D 2D 96
 3238- 09 00 2A 49 15 FF 13 2D
 3240- 2D AD 3F 3F 3F 2E 2D 2D
 3248- 1E 37 4E 49 00 00 09 29
 3250- AD DF 73 09 1E B6 4E 09
 3258- 00 09 09 15 9F 2D 35 DF
 3260- 33 4D 31 FF 73 6D 4E 00
 3268- 00 1E 09 15 1F 17 4D 31
 3270- DF 33 2D 2D FE 1B 6E 09
 3278- 4E 00 00 09 2D AD DF 33
 3280- 4D F1 3F 37 4D 31 DF 33
 3288- 2D 75 09 00 4A 29 AD DF
 3290- 33 36 4D 11 DF 73 2D 4E
 3298- 01 00 09 2D AD DF 6E 31
 32A0- DF 6E 31 DF 17 2D 75 09
 32A8- 00 4A 2D 2D DE 1B 6E 11
 32B0- 3F 37 36 2D 2D 4E 00 00
 32B8- 4A 2D 2D DE 1B 6E 11 3F
 32C0- 37 36 4E 49 01 00 49 29
 32C8- AD DF 33 36 4D 35 DF 73

| VISUBLOCS.9 |

```

32D0- 2D 75 01 00 4A 4D 31 DF
32D8- 33 4D 31 3F 3F 6E 09 FE
32E0- 1B 6E 09 4E 00 00 09 2D
32E8- 1E 36 36 17 2D 4E 00 00
32F0- 09 49 31 36 36 DF 73 2D
32F8- 4E 01 00 09 4D F1 DF 6E
3300- 1E 37 0D 15 DF 6E 09 4E
3308- 00 00 09 36 36 36 2D 75
3310- 01 00 00 4D 31 FF 37 0D
3318- 0D FE 1F 6E 09 FE 1B 6E
3320- 09 4E 00 00 49 4D 31 DF
3328- 33 6D 31 1F 1F 6E 29 FE
3330- 1B 6E 09 4E 00 00 09 29
3338- AD DF 33 4D 31 DF 33 4D
3340- 31 DF 73 2D 4E 01 00 09
3348- 2D AD DF 33 4D F1 3F 37
3350- 36 4E 49 01 00 09 29 AD
3358- DF 33 4D 31 DF 33 0D 0D
3360- 1E DF 0E 6D 4E 00 00 09
3368- 2D AD DF 33 4D F1 3F 37
3370- 0D 15 DF 6E 09 4E 00 00
3378- 09 29 AD DF 33 4D 3A 77
    
```

```

3380- 09 FE 1B 0E 2D 4E 01 00
3388- 09 2D 2D DE 36 36 76 49
3390- 00 00 4D 4D 31 DF 33 4D
3398- 31 DF 33 4D 31 DF 73 2D
33A0- 4E 01 00 09 4D 31 DF 33
33A8- 4D 31 DF 33 4D F1 1F 0E
33B0- 4E 09 00 49 4D 31 DF 33
33B8- 4D 31 1F 1F 6E 0D 3E 1F
33C0- 37 4D 71 01 00 92 4D 31
33C8- DF 73 0D 1E 17 0D 15 DF
33D0- 33 4D 71 01 00 4A 4D 31
33D8- DF 73 0D 1E 36 76 49 00
33E0- 00 2D 2D 2D BE 1E 17 1E
33E8- 2E 2D 75 01 00 4A 29 3E
33F0- 2E 3E 2E 3E 2E 3E 4D 00
33F8- 40 C0 40 43 00 19 00 00
3400- 00 00 00 00 00 00 00 00
3408- DB 1B 3F 2E 2D 0D AD 3F
3410- 1F 3F 4E 09 2D 35 3F DF
3418- DB 2A 6D 09 2D 3E 3F 1F
3420- 3F 77 2D 0D 2D FE 11 3F
3428- 3F
    
```

*Pour sauver ce programme, faites un **BSAVE VISUFONTE, AS3000, LS429***

CONTRÔLE DE SAISIE DE VISUFONTE

Après avoir enregistré VISUFONTE sur disquette, offrez-vous le luxe de le contrôler rapidement. Comment ? Tapez NEW, suivi de RETURN, puis :

```
10 FOR I = 12288 TO 12288 + 1064 : C = C + PEEK (I) : NEXT : PRINT C
```

Ensuite : BLOAD VISUFONTE (et RETURN, bien sûr), puis RUN (et RETURN). Après quelques secondes un total devra apparaître sur votre écran. Si ce total est 75665... Bravo ! Dans le cas contraire, vérifiez chaque ligne de VISUFONTE...

VISUBLOCS fonctionne sous DOS 3.3 et sous ProDOS et comprend deux parties :

- VISUBLOCS (BASIC), de la page 22 à la page 27 ;
- VISUFONTE (LM), pages 28-29.

IMPRESSION : Les instructions des lignes 1980-1990 concernent l'imprimante APPLE DMP, équipée d'une interface-buffer de la société BIP. Elles devront être revues et corrigées, en fonction du matériel dont vous disposez. Notez que vous avez la possibilité (dans le menu) de sauver vos VISUBLOCS sur disquette, puis de les éditer, par la suite, en utilisant le programme de votre choix. En général, l'interface livrée avec les imprimantes EPSON comporte une copie d'écran graphique. Chez APPLE, une routine d'édition est fournie sur la disquette accompagnant la documentation (cas d'IMAGE WRITER, notamment).

LIMITATIONS : Il est évident que certaines données se prêtent mieux que d'autres à une transformation en VISUBLOCS.

Par ailleurs, on comprendra facilement qu'un nombre pair de blocs est visuellement plus satisfaisant qu'un nombre impair.

On remarquera que, dans le cas d'un nombre impair, c'est toujours la première rangée qui accueille le bloc "supplémentaire". Etant donné qu'un tri place les valeurs les plus grandes en tête, une modification du programme pourrait au contraire renvoyer le bloc supplémentaire en bas de l'écran...

NESTOR.

UTILITAIRE

DOS 3.3

ProDOS

BOUTABOUT

Tapez ces 55 octets
derrière un CALL-151,
et sauvez-les par un
**BSAVE BOUTABOUT,
A5300, L537**

0300:	A2 00
0302:	BD 2F 03
0305:	9D 01 08
0308:	E8
0309:	E8 08
030B:	D8 F5
030D:	A9 09
030F:	85 67
0311:	A9 08
0313:	85 68
0315:	60
0316:	A5 AF
0318:	38
0319:	E9 02
031B:	85 67
031D:	A5 08
031F:	E9 08
0321:	85 68
0323:	60
0324:	A9 01
0326:	85 67
032B:	A9 08
032A:	85 68
032C:	4C 66 D5
032F:	09 08 08
	08 BC 3A
	08 00

```

*****
*                                     *
*                                     *
*                                     *
* Taper successivement:              *
*   - NEW (attention au programme!) *
*   - BRUN BOUTABOUT                *
*   - LOAD PROGRAMME1 et CALL 798   *
*   - LOAD PROGRAMME2 et CALL 804   *
*****
*                                     *
11      ORG      $300
12 LIGNE0 EQU   $32F      ;Ligne 0 (LIST)
13 BASIC  EQU   $801      ;Début du programme Basic
14 DEBUT  EQU   $67       ;Adresse du programme Basic
15 FIN    EQU   $AF       ;Adresse de la fin du programme
16 RUN    EQU   $D566     ;Run, tout simplement!
17 *-----*
18      LDX     £$00      ;Initialisation de la boucle
19 BOUCLE LDA   LIGNE0,X  ;Installation de la ligne 0
20      STA   BASIC,X    ;
21      INX     ;
22      CPX     £$08      ;Il n'y a que 8 caractères
23      BNE   BOUCLE     ;S'il n'y a pas égalité...
24      LDA     £$09      ;On trompe i'Applesoft
25      STA   DEBUT      ;pour qu'il n'écrase pas la
26      LDA     £$08      ;ligne 0, lorsqu'il va charger
27      STA   DEBUT+1    ;le PROGRAMME1
28      RTS     ;Vers LOAD PROGRAMME1
29 *-----*
30      LDA   FIN        ;Où prend fin Programme1 ?
31      SEC     ;Attention! on va soustraire
32      SBC     £$02      ;2 petits octets à soustraire
33      STA   DEBUT      ;pour tromper de nouveau
34      LDA   FIN+1     ;ce brave Apple
35      SBC     £$00      ;sans oublier la retenue,
36      STA   DEBUT+1    ;et la partie haute de l'adresse
37      RTS     ;Vers LOAD PROGRAMME2
38 *-----*
39      LDA   £$01      ;Et quant à tromper son meilleur
40      STA   DEBUT      ;ami, autant le faire une
41      LDA   £$08      ;troisième (à cause du dicton),
42      STA   DEBUT+1    ;mais dernière fois!
43      JMP   RUN       ;Voici la liste de PROGRAMME1-2
44 *-----*
45      HEX   09080808BC3A8000 ;Fausse ligne 0

```

**ASSEMBLAGE
PAR ProCODE**

SI VOUS ÊTES DÉBUTANT, mettez votre Apple en route en utilisant votre disquette MASTER. Tapez NEW, puis CALL-151, et :
* 300 : A2 00 BD 2F 03 9D 01 08 E8 E0 08 D0 F5 A9 09 85 67 A9 08 85 68 60 A5 AF 38 E9 02 85 67 A5 B0 E9 00 85 68 60 A9 01 85 67 A9 08 85 68 4C 66 D5 09 08 00 00 BC 3A 80 00, et un RETURN final. Sauvez ensuite votre programme par un : **BSAVE BOUTABOUT, A5300, L537**

IMPORTANT : Les lignes des deux programmes à mettre bout à bout doivent s'enchaîner.

Datez vos programmes



I l existe maintenant tout un arsenal d'utilitaires ProDOS, et la plupart d'entre eux permettent une mise à jour de la date... mais on ne peut pas toujours s'offrir un utilitaire... d'où la raison d'être du programme ci-après. Si vous le lisez attentivement (mieux : si vous le disséquez), vous constaterez qu'il est possible d'obtenir le même résultat en 12 lignes, en éliminant les saisies et affichages sophistiqués, de même que les barrages prévus pour interdire les erreurs de manipulation. On est... ou pas perfectionniste. C'est l'affaire de chacun !

NESTOR.

```

10 TEXT : PRINT CHR$(12); CHR$(21); HOME : INVERSE
20 AN$ = "JANFEVMARAVRMAIJUNJULAUSEPOCTNOVDEC"
30 FOR I = 1 TO 10:TR$ = TR$ + "____": NEXT
40 FOR I = 1 TO 5: VTAB I: HTAB 1: PRINT " "; IF I = 1 OR I = 5 THEN FOR
   J = 2 TO 39: PRINT " "; NEXT J
50 HTAB 48: PRINT " ": NEXT : NORMAL : VTAB 3: HTAB 7: PRINT "QUELLE EST
   LA DATE DU JOUR ?"
60 VTAB 6: POKE 34,7
70 PRINT CHR$(4)"PREFIX": INPUT PR$: PRINT CHR$(4)"PREFIX"PR$
80 HOME : PRINT "PREFIX"; INVERSE : PRINT PR$: NORMAL : PRINT
90 D1 = PEEK (49040) - INT ( PEEK (49040) / 32) * 32:D2 = INT ( PEEK (4
   9041) / 2):D3 = ( PEEK (49041) - D2 * 2) * 8 + INT ( PEEK (49040) / 32
   )
100 IF D3 ( ) 0 THEN M$ = MID$(AN$,D3 * 3 - 2,3)
110 PRINT "DATE DU SYSTEME : ";
120 PRINT RIGHT$( ("0" + STR$(D1)),2) + "-" + M$; IF M$ = "" THEN PR:
   NT "...";
130 PRINT "-" + RIGHT$( ("0" + STR$(D2)),2)
140 VTAB 20: PRINT TR$: VTAB 22: PRINT "<1> CHANGER DE DATE <2> CONTINUER
   -) ";
150 CALL - 198: GET R$: IF ASC (R$) < 49 OR ASC (R$) > 50 THEN 150
160 IF R$ = "1" THEN 180
170 GOTO 400
180 J$ = ""
190 VTAB 10: HTAB 19: CALL - 198: PRINT J$; GET R$: POKE - 16368,0: IF
   ASC (R$) = 127 AND J$ = ( ) "" THEN J$ = "": GOTO 190
200 IF J$ = "" AND R$ > "3" THEN 190
210 IF R$ < "0" OR R$ > "9" THEN 190
220 IF J$ = "" THEN J$ = J$ + R$: GOTO 190
230 IF VAL (J$ + R$) < 1 OR VAL (J$ + R$) > 31 THEN 190
240 PRINT R$:J$ = J$ + R$
250 M = M + 1: IF M = 13 THEN M = 1
260 VTAB 10: HTAB 22: CALL - 198: PRINT MID$(AN$,M * 3 - 2,3); HTAB 22
   : GET R$: POKE - 16368,0
270 R = ASC (R$): IF R = 13 THEN 300
280 IF R = 127 OR R = 8 THEN M = M - 2: IF M < 0 THEN M = 11
290 GOTO 250
300 A$ = ""

```

(SUITE AU VERSO).

DATEZ VOS PROGRAMMES (suite et fin)

```

310 VTAB 10: HTAB 26: CALL - 198: PRINT A$: GET R$: POKE - 16368,0
320 IF A$ < > "" AND (ASC (R$) = 8 OR ASC (R$) = 127) THEN 300
330 IF R$ < "0" OR R$ > "9" THEN 310
340 PRINT R$:A$ = A$ + R$: IF LEN (A$) < 2 THEN 310
350 VTAB 12: HTAB 1: CALL - 198: INPUT "HEURE ARRONDIE : ";R$:H = VAL (R
$: IF H < 1 OR R > 24 THEN 350
360 POKE 49041, VAL (A$) * 2 + (M > 7)
370 IF M < 8 THEN POKE 49040,M * 32 + VAL (J$): GOTO 390
380 POKE 49040,(M - 8) * 32 + VAL (J$)
390 POKE 49043,H: POKE 49042,0
400 PRINT : VTAB 22: CALL - 198: PRINT "<1> CHANGER DE PREFIX <2> APPLESO
FT -) "; GET R1$
410 IF R1$ = "2" THEN 490
420 IF R1$ = "1" THEN CALL - 198: VTAB 12: HTAB 1: INPUT "NOUVEAU PREFIX
: ";R$: IF R$ = "" THEN 400
430 FOR I = 1 TO LEN (R$):R = ASC (MID$ (R$,I,1)): IF (R > 45 AND R < 5
8) OR (R > 64 AND R < 90) THEN NEXT I: GOTO 450
440 GOTO 420
450 ONERR GOTO 400
460 IF LEFT$ (R$,1) < > "/" THEN R$ = "/" + R$
470 PRINT CHR$ (4)"PREFIX"R$: GOTO 70
480 POKE 216,0: IF PEEK (222) = 6 THEN VTAB 14: PRINT "PREFIX INCONNU":
GOTO 400
490 TEXT : HOME

```

ATTENTION ! Ce programme ne fonctionne qu'en version ProDOS.

Cette routine permet, dans un menu par exemple, d'afficher une ligne dont seules les MAJUSCULES sont en INVERSE.

L'essayer, c'est probablement l'adopter. Nous la devons à Yvan KOENIG, un expert (amateur) pour lequel le langage machine n'a pas plus de secrets que la mosaïque.

```

5 PRINT CHR$ (4)"BLOAD IN/MAJ"
10 TEXT : HOME
20 A$ = "Tout Recettes Sorties
CHOIX < >"
30 CALL 768,A$: HTAB 35
40 POKE 49168,0: GET X$: IF X$ <
"R" OR X$ > "T" THEN CALL -
1052: GOTO 40
50 PRINT X$

```

0300-	20 BE DE	JSR	\$DEBE
0303-	20 7B DD	JSR	\$DD7B
0306-	20 FD E5	JSR	\$E5FD
0309-	05 60	STA	\$60
030B-	A0 00	LDY	£\$00
030D-	B1 5E	LDA	(£5E),Y
030F-	C9 41	CMP	£\$41
0311-	90 00	BCC	\$031B
0313-	C9 5B	CMP	£\$5B
0315-	B0 04	BCS	\$031B
0317-	29 3F	AND	£\$3F
0319-	10 02	BPL	\$031D
031B-	09 00	ORA	£\$00
031D-	20 ED FD	JSR	\$FDED
0320-	C8	INY	
0321-	C4 60	CPY	\$60
0323-	90 E8	BCC	\$030D
0325-	60	RTS	

Générateur de programmes

S I vous désirez écrire une petite routine, destinée à saisir x valeurs (numériques ou alphanumériques), puis à les mémoriser dans un fichier, créé sur disquette, utilisez plutôt ce modeste générateur de programmes !

Il présente l'avantage d'être réutilisable à l'infini, pour d'autres applications.

CREATEFIC (c'est son nom de baptême) se charge en effet d'écrire autant de clones que vous le désirez, et en quelques minutes.

FONCTIONNEMENT Après le RUN habituel, répondez aux quelques questions qui vous seront posées, puis laissez votre Apple tourner. Un fichier EXEC, intitulé *PROGRAMME 1*, sera automatiquement installé sur votre disquette, puis chargé en mémoire, à la place de *CREATEFIC* (qui sera effacé... d'où l'impérieuse nécessité de le sauvegarder en priorité, avant la première utilisation).

Faites alors un *SAVE FIC* (ou tout autre titre, choisi par vous), puis un *RUN*. Et c'est parti !

G.H.

GÉNÉRATEUR DE PROGRAMMES (suite)

```
1000 TEXT : PRINT CHR$ (12); CHR$ (21): HOME
1010 D$ = CHR$ (4):G$ = CHR$ (34)
1020 DIM L$(25)
1030 T$ = " CREATION D'UN PROGRAMME DE SAISIE ": VTAB 1: HTAB 21 - LEN (T$)
/ 2: PRINT T$
1040 FOR I = 1 TO 10:TR$ = TR$ + " ____ ": NEXT : VTAB 2: PRINT TR$
1050 VTAB 4: HTAB 1: CALL - 950: PRINT "DESIREZ-VOUS ENREGISTRER ": VTAB
6: HTAB 2: CALL - 198: PRINT "<1> DES CHIFFRES <2> DES LETTRES ? "; G
ET R$
1060 NA = ASC (R$) - 48: IF NA < 1 OR NA > 2 THEN 1050
1070 V$ = "E": IF NA = 2 THEN V$ = "E$"
1080 INVERSE : PRINT R$: NORMAL
1090 GOSUB 1330
1100 VTAB 10: HTAB 1: PRINT "VOTRE VARIABLE EST DONC "; INVERSE : PRINT E$
: NORMAL
1110 R = 1: GOSUB 1280: IF W = 78 THEN 1050
1120 VTAB 12: HTAB 1: CALL - 950: CALL - 198: INPUT "TITRE DE VOTRE FICHI
ER ? ";TI$
1130 IF TI$ = "?" THEN R = 2: GOSUB 1280: GOTO 1120
1140 IF TI$ = "*" THEN 1050
1150 TI = LEN (TI$): IF TI = 0 THEN 1120
1160 FOR I = 1 TO TI:A = ASC ( MID$ (TI$,I,1)): IF (A < 65 OR A > 90) AND
(A < 48 OR A > 57) AND A < > ASC (".") THEN TI$ = "?": GOTO 1130
1170 NEXT
1180 VTAB 12: HTAB 26: INVERSE : PRINT TI$: NORMAL :R = 1: GOSUB 1280: IF W
= 78 THEN 1120
1190 VTAB 14: HTAB 1: CALL - 198: INPUT "NOMBRE D'ENREGISTREMENTS ? ";NE$
1200 IF NE$ = "?" THEN R = 3: GOSUB 1280: GOTO 1190
1210 IF NE$ = "*" THEN 1120
1220 NE = VAL (NE$): IF NE < 1 THEN 1190
1230 GOSUB 1330
1240 VTAB 3: CALL - 950: PRINT : PRINT "FICHIER "; INVERSE : PRINT TI$;;
NORMAL : PRINT " DIM "; INVERSE : PRINT E$: NORMAL
1250 PRINT : PRINT "* L'enregistrement ZERO sera le pointeur"
1260 R = 1: GOSUB 1280: IF W = 78 THEN 1190
1270 GOTO 1420
1280 VTAB 18: HTAB 1: CALL - 950: PRINT TR$
1290 ON R GOTO 1300,1310,1320,1005
1300 INVERSE : PRINT "O";: NORMAL : PRINT "UI OU "; INVERSE : PRINT "N";:
NORMAL : PRINT "ON EN CAS DE DESACCORD": GOTO 1360
1310 PRINT "AUTORISES: CAPITALES, CHIFFRES ET POINT": GOTO 1360
1320 PRINT "ATTENTION! LA MEMOIRE A DES LIMITES!": GOTO 1360
1330 IF NE$ = "" THEN NE$ = "X"
1340 E$ = V$ + "(" + NE$
1350 E$ = E$ + ")": RETURN
1360 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: POKE - 16368,0:W
= PEEK ( - 16384): RETURN
1370 :
```

```

1380 REM *****
1390 REM *   PROGRAMME COMPLET DE SAISIE   *
1400 REM *****
1410 :
1420 L$(1) = "10 TEXT:PRINT CHR$(12);CHR$(21):HOME"
1430 L$(2) = "20 ONERR GOTO 230"
1440 L$(3) = "30 DIM" + E$ + " :D$=CHR$(4):F$=" + G$ + TI$ + G$ + " :N=" + NE$
1450 L$(4) = "40 FORI=1TO10:TR$=TR$+" + G$ + "_____" + G$ + " :NEXT"
1460 L$(5) = "50 T$=" + G$ + "SAISIE DES VALEURS DE " + G$ + "+F$"
1470 L$(6) = "60 HTAB 21-(LEN(T$)/2):PRINT T$:PRINT TR$:POKE 34,3"
1480 L$(7) = "70 I=0:VTAB20:PRINT TR$:VTAB22:PRINT " + G$ + "?=CORRECTION -
  *=TERMINE - &=8 DERNIERS" + G$ + " :POKE35,19"
1490 L$(8) = "80 I=I+1:VTAB5:HTAB1:CALL-958:INVERSE:PRINT I:NORMAL"
1500 L$(9) = "90 VTAB7:HTAB1:INPUT " + G$ + "--> " + CHR$(7) + G$ + " ;E$:I
  FE$=" + G$ + G$ + " THEN 90"
1510 L$(10) = "100 IF I>1 AND E$=CHR$(63)THEN I=I-1:VTAB5:HTAB1:CALL-958:INV
  ERSE:PRINTI:NORMAL:VTAB7:HTAB5:PRINT E$(I):GOTO90"
1520 L$(11) = "110 IF E$=CHR$(42) THEN I=I-1:GOTO240"
1530 L$(12) = "120 IF E$(<)CHR$(38)THEN 150"
1540 L$(13) = "130 HOME:R=I-8:IFR<1THENR=1"
1550 L$(14) = "140 V=4:FORJ=RT01-1:VTABV:HTAB1:INVERSE:PRINTJ;;NORMAL:PRINT
  TAB(4);" + V$ + "<J):V=V+2:NEXT:CALL-198:POKE-16368,0:WAIT-16384,128:PO
  KE-16368,0:PRINT:HOME:I=I-1:GOTO80"
1560 IF NA = 1 THEN L$(15) = "150 " + V$ + "(I)=VAL(E$):E$=" + G$ + G$ + " :
  IF I<N THEN 80": GOTO 1580
1570 L$(15) = "150 E$(I)=E$:E$=" + G$ + G$ + " :IF I<N THEN 80"
1580 L$(16) = "160 GOTO 240"
1590 L$(17) = "170 PRINT D$;" + G$ + "OPEN " + TI$ + G$ + " :PRINT D$;" + G$
  + "WRITE " + TI$ + G$ + " :PRINT I:FORE=1TOI:PRINT " + V$ + "(E):NEXT:PR
  INT D$;" + G$ + "CLOSE " + G$
1600 L$(18) = "180 TEXT:HOME:PRINT D$;" + G$ + "CATALOG" + G$
1610 L$(19) = "190 GETR$:PRINT:HOME:PRINT " + G$ + "RELECTURE DU FICHIER " +
  G$ + " ;F$ :PRINT TR$:POKE34,3:PRINT:LIST200:GOTO 210"
1620 L$(20) = "200 PRINT D$;" + G$ + "OPEN " + TI$ + G$ + " :PRINT D$;" + G$
  + "READ " + TI$ + G$ + " :INPUT I:FORE=1TOI:INPUT " + V$ + "(E):NEXT:PRI
  NT D$;" + G$ + "CLOSE " + G$
1630 L$(21) = "210 PRINT:PRINT TR$:PRINT" + G$ + "ET MAINTENANT, A VOUS DE J
  OUER!" + G$
1640 L$(22) = "220 CALL-198:POKE-16368,0:WAIT-16384,128:POKE-16368,0:END"
1650 L$(23) = "230 I=I-1:IFPEEK(222)<)255THEN VTAB18:PRINT" + G$ + "PROBLEME
  IMPREVU!" + G$ + " :END"
1660 L$(24) = "240 VTAB18:CALL-868:CALL-198:PRINT" + G$ + "1) ENREGISTRER -
  2) DERNIER SAISI " + G$ + " ;:GETR$:PRINTR$:IFR$=CHR$(49)THEN 170"
1670 L$(25) = "250 I=I+1:E$=CHR$(63):GOTO100"
1680 VTAB 17: PRINT
1690 PRINT D$"OPEN PROGRAMME1": POKE 33,33: PRINT D$"WRITE PROGRAMME1": FOR
  I = 1 TO 25: PRINT L$(I): NEXT : PRINT D$"CLOSE": POKE 33,40
1700 HOME : PRINT : PRINT CHR$(4)"EXEC PROGRAMME1"
1710 DEL 1000,1710

```

SUGGESTION Compléter cet embryon de «générateur de programmes», de manière à ce qu'il prévoie un menu, ainsi que la lecture du fichier créé.

ECRANS CONDENSÉS

A QUOI bon mémoriser, sur une disquette, les 64 octets inutiles d'un écran qui, dans la réalité, n'en exige que 960 (24 fois 40) ? Le mini-programme ci-dessous n'est qu'une démonstration, destinée à vous montrer comment fonctionne la routine ECRANLM de la page suivante.

```
10 TEXT : HOME : PRINT CHR$(12); CHR$(21)
20 PRINT CHR$(4)"BLOAD ECRANLM"
30 PRINT "ECONOMISONS LES SECTEURS DE NOS DISQUETTES EN
  NE MEMORISANT QUE LA PARTIE TEXTE DE NOS ECRANS (EN 40
  COLONNES).
  COMMENT ? VOICI UNE SOLUTION: -----
  -"
40 VTAB 22: PRINT *  ** APPUYEZ SUR UNE TOUCHE SVP **
  *
50 GOSUB 240
60 FOR I = 1 TO 23: VTAB I: FOR J = 1 TO 39: PRINT CHR$(
  I + 64);: NEXT : IF I < > 23 THEN PRINT CHR$(I +
  64): NEXT
70 POKE 1911,215: NEXT
80 FOR I = 2000 TO 2039: POKE I,216: NEXT
90 VTAB 1: HTAB 1: GOSUB 240
100 CALL 768
110 FOR I = 8 TO 14: VTAB I: HTAB 5: PRINT "
  ": NEXT
120 VTAB 10: HTAB 7: PRINT "L'ECRAN PLEIN DE LETTRES EST"
130 VTAB 11: HTAB 7: PRINT "ACTUELLEMENT MEMORISE A UNE "
140 VTAB 12: HTAB 7: PRINT "ADRESSE CONNUE: $2000 ET ON "
150 VTAB 13: HTAB 7: PRINT "PEUT L'ECRIRE SUR DISQUETTE."
160 PRINT "": VTAB 14: HTAB 7: PRINT CHR$(4)"BSAVE ECRD
  EMO,A$2000,L$3C0"
170 GOSUB 240
180 HOME : VTAB 12: PRINT " APPUYER SUR UNE TOUCHE POUR L
  E REVOIR": GOSUB 240
190 CALL 816: GOSUB 240
200 HOME
210 VTAB 12: PRINT "CELA EST POSSIBLE GRACE A UN PETIT PR
  O- GRAMME EN ASSEMBLEUR (76 OCTETS), QUI SECHARGE DES
  TRANSFERTS."
220 PRINT : PRINT "IL FAUT EVIDEMMENT LE CHARGER EN MEMOI
  RE(A$300,L$4C), PUIS PREVOIR : - CALL 7
  68 vers $2000 - BSAVE ECRAN,A$2000,L
  $3C0 - BLOAD ECRAN pour le relire
  - CALL 816 pour l'afficher."
230 GOSUB 240: HOME : END
240 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127:
  POKE - 16368,0: RETURN
```

Il reste que cette façon de faire présente non seulement l'avantage d'économiser un secteur par écran* mais celui de ne plus heurter la susceptibilité de l'Apple... en l'obligeant à ingurgiter 64 octets d'informations incompatibles avec sa fonction du moment.

En effet, il ne faut pas oublier que les 64 octets en question représentent notamment les N fois 8 octets de MEV réservés à chaque carte interface de périphérique.

* On économise en tout cas 1 secteur en sauvant un écran texte avec L1016 (qui escomote les 8 derniers octets, inutilisés pour l'affichage).

ECRANLM

Ce programme lit les 24 lignes de l'écran en 40 colonnes, et les recopie à partir de l'adresse \$2000 (qui peut évidemment être modifiée). On sauve l'écran par CALL 768,

puis BSAVE ECRDEMO, A\$2000, L\$3C0.

Bien entendu, il devra être rechargé à la même adresse... à moins de modifier la routine ci-dessous. Un simple BLOAD ECRDEMO provoquera l'effet souhaité, puis un CALL 816 entraînera l'affichage.

0300-	A9 20	LDA	£\$20] On utilise les mémoires 06 et 07 de la Page zéro comme boîte à lettres, pour y stocker l'adresse à laquelle va être transféré l'écran.
0302-	85 07	STA	\$07	
0304-	A9 00	LDA	£\$00	
0306-	85 06	STA	\$06	
0308-	85 08	STA	\$08] — \$25 = CV, position verticale du curseur — Calcule l'adresse de la ligne courante — Début de boucle — En \$28 et \$29 se trouve l'adresse de la ligne — Réécriture à l'adresse \$6 — 7 + Y — Plus un dans la boucle — A-t-on atteint 40 ? — Non ? alors, un autre tour !
030A-	85 25	STA	\$25	
030C-	20 C1 FB	JSR	\$FBC1	
030F-	A0 00	LDY	£\$00	
0311-	B1 28	LDA	(<\$28), Y] — En \$28 et \$29 se trouve l'adresse de la ligne — Réécriture à l'adresse \$6 — 7 + Y — Plus un dans la boucle — A-t-on atteint 40 ? — Non ? alors, un autre tour !
0313-	91 06	STA	(<\$06), Y	
0315-	C8	INY		
0316-	C0 28	CPY	£\$28	
0318-	D0 F7	BNE	\$0311] L'adresse de réécriture est incrémentée de 40 (\$28), longueur de la ligne qui vient d'être lue.
031A-	A5 06	LDA	\$06	
031C-	18	CLC		
031D-	69 20	ADC	£\$28	
031F-	85 06	STA	\$06] En \$8 est stocké le nombre de lignes — Est-on à 24 ? — Non ? on recommence
0321-	A5 07	LDA	\$07	
0323-	69 00	ADC	£\$00	
0325-	85 07	STA	\$07	
0327-	E6 08	INC	\$08] Réaffichage. Cette partie du programme modifie les adresses, puisqu'il faut lire à partir de \$2000, et réécrire à partir de \$400. Ensuite, on remet les choses en place.
0329-	A5 08	LDA	\$08	
032B-	C9 18	CMP	£\$18	
032D-	D0 DB	BNE	\$030A	
032F-	60	RTS] Réaffichage. Cette partie du programme modifie les adresses, puisqu'il faut lire à partir de \$2000, et réécrire à partir de \$400. Ensuite, on remet les choses en place.
0330-	A9 06	LDA	£\$06	
0332-	80 12 03	STA	\$0312	
0335-	A9 28	LDA	£\$28	
0337-	80 14 03	STA	\$0314] Réaffichage. Cette partie du programme modifie les adresses, puisqu'il faut lire à partir de \$2000, et réécrire à partir de \$400. Ensuite, on remet les choses en place.
033A-	20 00 03	JSR	\$0300	
033D-	A9 28	LDA	£\$28	
033F-	80 12 03	STA	\$0312	
0342-	A9 06	LDA	£\$06] Réaffichage. Cette partie du programme modifie les adresses, puisqu'il faut lire à partir de \$2000, et réécrire à partir de \$400. Ensuite, on remet les choses en place.
0344-	80 14 03	STA	\$0314	
0347-	A9 00	LDA	£\$00	
0349-	85 25	STA	\$25	
034B-	60	RTS		

Pour rentrer cette routine, tapez CALL-151 et RETURN, puis

* 300 : A9 20 85 07 A9 00 85 06 85 08 85 25 20 C1 FB A0 00 B1 28 91 06
C8 C0 28 D0 F7 A5 06 18 69 28 85 06 A5 07 69 00 85 07 E6 08 A5 08 C9
18 D0 DB 60 A9 06 8D 12 03 A9 28 8D 14 03 20 00 03 A9 28 8D 12 03 A9
06 8D 14 03 A9 00 85 25 BSAVE ECRANLM, A\$300, L\$4C.

VÉRIFICATION (mode direct)

FOR I = 768 TO 843 : E = E + PEEK (I) :
NEXT : PRINT E

Le résultat doit être 6955.



Ouest-Est

VOTRE entourage (vos élèves, peut-être ?) connaît-il sa géographie ? Rien ne vous interdit de le soumettre à un petit test, rapidement mis au point sur votre ordinateur personnel.

```

10 TEXT : PRINT CHR$ (12); CHR$ (21); HOME
20 INVERSE : FOR I = 10 TO 14: HTAB 10: VTAB I: PRINT
   " ";: IF I = 10 OR I = 14 THEN FOR J = 11 TO 29:
   PRINT " ";: NEXT
30 HTAB 30: PRINT " ";: NEXT : NORMAL
40 B$ = " O U E S T ";A$ = " E S T "
50 GOTO 190
60 :
70 ::::: REM *****
80 ::::: REM **      SOUS-PROGRAMMES      **
90 ::::: REM *****
100 :
110 FOR B2 = 1 TO F:X = PEEK (49200): IF PEEK (49249
   ) > 128 AND F ( ) < 5 THEN FIN = 1
120 NEXT : RETURN
130 FOR I = 1 TO 3000: NEXT : RETURN
140 :
150 ::::: REM *****
160 ::::: REM **      AFFICHAGE DU TITRE      **
170 ::::: REM *****
180 :
190 VTAB 12: HTAB 12: PRINT A$ + " ";: INVERSE : PRINT
   B$
200 F = 30: GOSUB 110
210 VTAB 12: HTAB 12: PRINT B$;: NORMAL : PRINT " " +
   A$: IF FIN = 1 THEN 230
220 GOSUB 110:A = A + 1: IF A < 8 THEN 190
230 A = 0: VTAB 16: PRINT "Voulez-vous jouer avec moi :
   Oui - Non ?"
240 VTAB 16: HTAB 40: CALL - 198: GET R$
250 IF ASC (R$) = 79 THEN 330
260 IF ASC (R$) = 78 THEN PRINT R$: PRINT " Dommage
   ! ce sera pour une autre fois! ": END
270 PRINT "": VTAB 18: PRINT " OUVREZ MIEUX LES YEUX S
   'IL VOUS PLAIT!": VTAB 16: HTAB 29: INVERSE : PRINT
   "O";: HTAB 35: PRINT "N": NORMAL : GOTO 240
280 :

```

◀ Cadre mode inverse

◀ Bruit dont la durée dépend de la variable F.

◀ Boucle d'attente.

◀ Une manière plus ou moins élégante (ça dépend des goûts !) de déplacer un mot. Si FIN = 1 (Pomme ouverte enfoncée ; ligne 110), on passe immédiatement à la suite.

◀ Rappel à l'ordre... sans aménité.

CALL-958 vide l'écran, à partir et au-dessous du curseur.

Un bruit accompagne l'affichage de chaque lettre.

CALL-913 fait remonter la totalité de l'écran d'une ligne.

Mélange des trois villes en utilisant la fonction RND de l'Apple.

Si R=63, c'est que la touche Del a été enfoncée, et on revient en arrière.

On stocke dans la variable RX\$ les réponses successives aux trois questions.

SPEED = 255, vitesse maximum d'affichage.

```
290 ::::: REM *****
300 ::::: REM **          LA PARTIE          **
310 ::::: REM *****
320 :
330 PRINT : VTAB 16: CALL - 958
340 T$ = "Le programme va successivement vous pro-poser
      10 fois trois noms de villes... Cesera à vous de d
      ésigner la ville qui est approximativement située en
      tre les deux autres, et les positions de celles-ci.
      "
350 F = 3: FOR I = 1 TO LEN (T$): PRINT MID$ (T$,I,1)
      ; GOSUB 110: NEXT I: PRINT
360 F = 20: FOR I = 1 TO 9: GOSUB 110: CALL - 912: NEX
      T : REM SCROLL (Déroulement vers le haut)
370 POKE 34,6: HOME
380 JEU = JEU + 1: VTAB 8: INVERSE : PRINT "QUESTION " J
      EU;: NORMAL : PRINT TAB( 15);PT;" POINT";: IF PT >
      1 THEN PRINT "S";
390 PRINT TAB( 28)"ERREUR = Del!": PRINT
400 FOR I = 1 TO 3: READ V$(I): NEXT
410 V(1) = 1 + INT ( RND (1) * 3);V(2) = 1 + INT ( RN
      D (1) * 3); IF V(1) = V(2) THEN 410
420 V(3) = 6 - V(1) - V(2): FOR I = 1 TO 3:X$(I) = V$(V
      (I)): NEXT
430 FOR I = 1 TO 3: INVERSE : PRINT CHR$ (I + 64);: N
      ORMAL : PRINT " X$(I): PRINT : NEXT
440 PRINT : VTAB 20: CALL - 958: FOR I = 1 TO 10: PRI
      NT "___":; NEXT : PRINT
450 PRINT "TAPEZ LES LETTRES-CLES DANS L'ORDRE QUE
      VOUS ESTIMEZ ETRE LE BON"
460 X = 1:RX$ = ""
470 VTAB 17: HTAB 1: INVERSE : PRINT "REPONSE:":; NORM
      AL : CALL - 198: PRINT " RX$;: GET R$: POKE - 16
      368,0:R = ASC (R$) - 64
480 IF R = 63 AND X = 2 THEN X = 1:RX$ = "": GOTO 470
490 IF R = 63 AND X = 3 THEN X = 2:RX$ = LEFT$ (RX$,1
      ): GOTO 470
500 IF R < 1 OR R > 3 THEN 470
510 IF X = 1 THEN 540
520 FOR I = 1 TO X - 1: IF MID$ (RX$,I,1) = R$ THEN 4
      70
530 NEXT I: PRINT R$
540 RX$ = RX$ + R$: VTAB 8 + R * 2: HTAB 2: CALL - 868
      : HTAB - 9 + X * 12: PRINT X$(R)
550 IF X < 3 THEN X = X + 1: GOTO 470
560 VTAB 17: HTAB 14
570 FOR I = 1 TO 3: IF V$(I) = X$( ASC ( MID$ (RX$,I,1
      )) - 64) THEN NEXT : GOTO 610
580 PRINT "REPONSE INEXACTE": GOSUB 130
590 SPEED= 40: FOR I = 1 TO 3: VTAB 8 + I * 2: HTAB 2:
      CALL - 868: HTAB - 9 + V(I) * 12: PRINT X$(I): N
      EXT
```

(suite au verso)

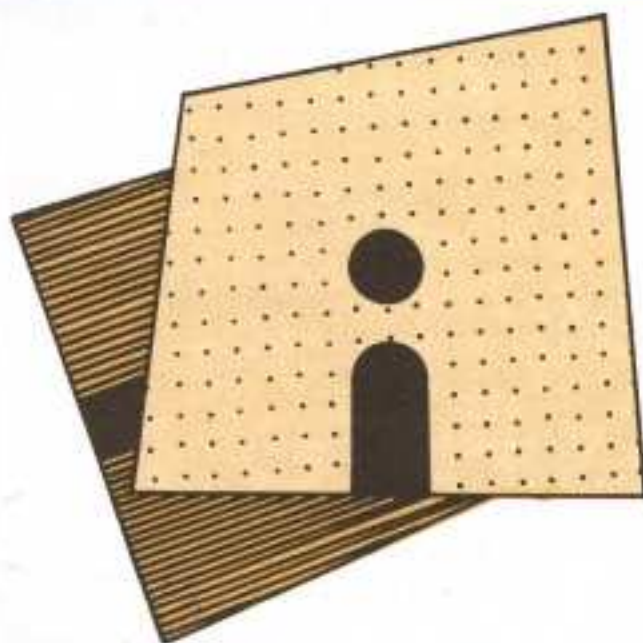
QUEST-EST (SUITE et FIN)

```
600 SPEED= 255: GOSUB 130: GOTO 620
610 SPEED= 60: PRINT "REPONSE EXACTE: + 1 POINT!":PT =
    PT + 1: SPEED= 255: GOSUB 750
620 IF JEU < 10 THEN HOME : GOTO 380
630 HOME : PRINT : PRINT : INVERSE : PRINT PT" POINT";
    : IF PT > 1 THEN PRINT "S";
640 NORMAL : PRINT : PRINT : PRINT : IF PT > 8 THEN P
    RINT "TRES BON SCORE": GOTO 700
650 IF PT > 5 THEN PRINT "RESULTAT HONORABLE": GOTO 7
    00
660 IF PT > 3 THEN PRINT "PEUT MIEUX FAIRE": GOTO 700
670 IF PT > 1 THEN PRINT "DOIT MIEUX FAIRE": GOTO 700
680 PRINT "OUILLE! UN VRAI DESASTRE! A REFAIRE!": GOSU
    B 130
690 HOME : CLEAR : GOTO 380
700 VTAB 22: HTAB 1: PRINT "<1> JE RECOMMENCE <2> C'E
    ST TERMINE ";; CALL - 198: GET R$
710 IF R$ = "1" THEN 690
720 TEXT : HOME : END
730 DATA LE MANS,ORLEANS,MONTARGIS,PERIGUEUX,BRIVE,AUR
    ILLAC,CAEN,LISIEUX,VERNON,BOURGES,NEVERS,AUTUN,CHAT
    ELLERAULT,CHATEAUROUX,VERNON
740 DATA DINAN,FOUGERES,ALENCON,ANGERS,TOURS,BLOIS,DAX
    ,AUCH,CASTRES,MONTLUCON,VICHY,ROANNE,CHAUMONT,EPINA
    L,COLMAR,AUXERRE,VESOUL,BELFORT
750 FOR S = 760 TO 786: POKE S, VAL ( MID$ ("001730481
    92136200004190007240008202208246166006076000003096"
    ,(S - 767) * 3,3)): NEXT S: FOR U = 10 TO 40 STEP 5
    : POKE 6,U: POKE 7,15: CALL 760: NEXT U
760 RETURN
```

Pour les commentaires, donnez libre cours à votre inspiration personnelle, sûrement meilleure...

✦ CLEAR permet de remettre toutes les variables à zéro et évite la commande RESTORE.

✦ C'est une utilisation du "son" de notre page 51. Normalement, il est inutile de la relancer à chaque fois, mais cela vaut bien une boucle de temps !



Offrez-vous les disquettes
de programmes de

TREMPAIN MICRO

Au choix : DOS 3.3 ou ProDOS.
(ProDOS ne fonctionne qu'après avoir
lancé le système avec une disquette
MASTER).

Utilisez le bulletin de commande de la
page 62... ou recopiez-le.

CARTE LANGAGE

APPLE IIe et IIc (DOS 3.3)
Avec ProDOS, on ne peut utiliser que MEV2

TRANSFERT ET RÉCUPÉRATION DE DEUX ÉCRANS TEXTE

La routine n°1 transfère un écran de 1024 octets en MEV, Bloc 1, à partir de l'adresse D000 (il suffit de taper **CALL 768**). La récupération est obtenue par un **CALL 806**. La routine n°2 utilise la même adresse, mais vers le Bloc 2 (transfert : **CALL 841** ; retour : **CALL 879**).

Utilisable à partir d'un programme en Basic, ou en mode direct. Un RESET est sans effet sur les écrans ainsi mémorisés.

Restriction : il ne faut évidemment pas avoir chargé le BASIC INTEGER sur la carte langage.

1. MEV1

```

0300 : AD 89 C0      LDA  $C089
0303 : AD 89 C0      LDA  $C089
0306 : A9 00         LDA  £$00
0308 : 85 3C         STA  $3C
030A : A9 04         LDA  £$04
030C : 85 3D         STA  $3D
030E : A9 D0         LDA  £$D0
0310 : 85 3E         STA  $3E
0312 : A9 07         LDA  £$07
0314 : 85 3F         STA  $3F
0316 : A9 00         LDA  £$00
0318 : 85 42         STA  $42
031A : A9 D0         LDA  £$D0
031C : 85 43         STA  $43
031E : 38           SEC
031F : 20 11 C3      JSR  $C311
0322 : AD 8A C0      LDA  $C08A
0325 : 60           RTS
0326 : AD 88 C0      LDA  $C088
0329 : A9 00         LDA  £$00
032B : 85 3C         STA  $3C
032D : A9 D0         LDA  £$D0
032F : 85 3D         STA  $3D
0331 : A9 D0         LDA  £$D0
0333 : 85 3E         STA  $3E
0335 : A9 D3         LDA  £$D3
0337 : 85 3F         STA  $3F
0339 : A9 00         LDA  £$00
033B : 85 42         STA  $42
033D : A9 04         LDA  £$04
033F : 85 43         STA  $43
0341 : 18           CLC
0342 : 20 11 C3      JSR  $C311
0345 : AD 8A C0      LDA  $C08A
0348 : 60           RTS
    
```

2. MEV2

```

0349 : AD 81 C0      LDA  $C081
034C : AD 81 C0      LDA  $C081
034F : A9 00         LDA  £$00
0351 : 85 3C         STA  $3C
0353 : A9 04         LDA  £$04
0355 : 85 3D         STA  $3D
0357 : A9 D0         LDA  £$D0
0359 : 85 3E         STA  $3E
035B : A9 07         LDA  £$07
035D : 85 3F         STA  $3F
035F : A9 00         LDA  £$00
0361 : 85 42         STA  $42
0363 : A9 D0         LDA  £$D0
0365 : 85 43         STA  $43
0367 : 38           SEC
0368 : 20 11 C3      JSR  $C311
036B : AD 82 C0      LDA  $C082
036E : 60           RTS
036F : AD 80 C0      LDA  $C080
0372 : A9 00         LDA  £$00
0374 : 85 3C         STA  $3C
0376 : A9 D0         LDA  £$D0
0378 : 85 3D         STA  $3D
037A : A9 D0         LDA  £$D0
037C : 85 3E         STA  $3E
037E : A9 D3         LDA  £$D3
0380 : 85 3F         STA  $3F
0382 : A9 00         LDA  £$00
0384 : 85 42         STA  $42
0386 : A9 04         LDA  £$04
0388 : 85 43         STA  $43
038A : 18           CLC
038B : 20 11 C3      JSR  $C311
038E : AD 82 C0      LDA  $C082
0391 : 60           RTS
    
```

ÉCRITURE SUR DISQUETTE :

BSAVE MEV1, A\$300, L\$49

BSAVE MEV2, A\$349, L\$49

Si vous réunissez les deux routines : **BSAVE MEV1.2, A\$300, L\$92**

UTILISATION : BLOAD MEV1, ou BLOAD MEV2 (dans un programme : **PRINT CHR\$(4) ; "BLOAD MEV1"**).

CARTE LANGAGE

TRANSFERT D'ÉCRANS TEXTE

Avec ProDOS, on ne peut utiliser que MEV2

Il n'est évidemment pas nécessaire d'utiliser les deux routines (exposées au recto de cette fiche) pour transférer plusieurs écrans sur la carte langage, mais quelques pokes doivent alors précéder chaque opération.

Le programme en BASIC ressemblera à ceci (nous avons escamoté les numéros des lignes) :

ECRAN 1 VERS MEV1 (\$D000-D3D0)

POKE 769,137 : POKE 772,137 : POKE
803,138 : POKE 795,208 : CALL 768

MEV1 VERS ECRAN (\$0400-07D0)

POKE 807,136 : POKE 838,138 : POKE
814,208 : POKE 822,211 : CALL 806

ECRAN 3 VERS MEV1 (\$D400-D7D0)

POKE 769,137 : POKE 772,137 : POKE
803,138 : POKE 795,212 : CALL 768

MEV1 VERS ECRAN (\$0400-07D0)

POKE 807,136 : POKE 838,138 : POKE
814,212 : POKE 822,215 : CALL 806

ECRAN 2 VERS MEV2 (\$D000-D3D0)

POKE 769,129 : POKE 772,129 : POKE
803,130 : POKE 795,208 : CALL 768

MEV2 VERS ECRAN (\$0400-07D0)

POKE 807,128 : POKE 838,130 : POKE 814,
208 : POKE 822,211 : CALL 806

ECRAN 4 VERS MEV2 (\$D400-D7D0)

POKE 769,129 : POKE 772,129 : POKE
803,130 : POKE 795,212 : CALL 768

MEV2 VERS ECRAN (\$0400-07D0)

POKE 807,128 : POKE 838,130 : POKE
814,212 : POKE 822,215 : CALL 806

Quant à votre routine unique, la voici, à taper après le CALL-151 habituel (☐ = RETURN)

```
300 : AD 89 C0 AD 89 C0 A9 00 85 3C A9 04 85 3D A9 D0 85 3E A9 07 85 3F A9
00 85 42 A9 D0 85 43 38 20 11 C3 AD 8A C0 60 AD 88 C0 A9 00 85 3C A9 D0 85
3D A9 D0 85 3E A9 D3 85 3F A9 00 85 42 A9 04 85 43 18 20 11 C3 AD 8A C0 60 ☐
BSAVE MEV, A$300, L$49 ☐
```

Telle qu'elle se présente au départ, et après un *BLOAD MEV*, cette routine ne transfère qu'un écran, vers MEV1. Pour les écrans 2, 3 et 4, les lignes de POKES sont obligatoires. Notez que vous modifiez alors votre programme, d'où la nécessité d'utiliser TOUS LES POKES, y compris pour l'écran numéro 1.

RAPPEL DES INSTRUCTIONS UTILISÉES ICI

HEX.	DEC.	OBJET	INSTRUCTIONS
C089	49289	ECRITURE BLOC 1	LDA \$C089 LDA \$C089
C088	49288	LECTURE BLOC 1	LDA \$C088
C08A	49290	DECONNECTEE EN LECTURE-PROTEGEE EN ECRITURE	LDA \$C08A
C081	49281	ECRITURE BLOC 2	LDA \$C081 LDA \$C081
C080	49280	LECTURE BLOC 2	LDA \$C080
C082	49282	MEME EFFET QUE C08A	LDA \$C08A

ProDOS et la carte langage

ProDOS va se loger sur la carte langage, et il n'est donc pas possible d'utiliser la MEV1 de celle-ci (notre Fiche n°3). Le programme DEMO ci-après nous montre comment transférer 4 écrans en MEV2 (avec ProDOS ou DOS 3.3 d'ailleurs). Son apparente complexité est due au simple fait qu'il est destiné à une démonstration... sur écran. Ne vous affolez pas !

```
5 PRINT CHR$(4)*"BLOADMEV2" (C'est la petite routine MEY, de la fiche ci-contre).
10 TEXT : PRINT CHR$(12); CHR$(21): HOME
20 PRINT "TRANSFERT DE 4 ECRANS TEXTE EN D000-D7D0";
30 T$ = " "; FOR I = 1 TO 10:T$ = T$ + " ____ "; NEXT : PRINT T$
40 VTAB 10: LIST 50: LIST 210
45 :
46 REM ECRAN 1 EN $D000-D3D0 (MEV1) - ECRAN 2 EN $D000-D3D0 (MEV2)
47 REM ECRAN 3 EN $D400-D7D0 (MEV1) - ECRAN 4 EN $D400-D7D0 (MEV2)
48 :
50 GOSUB 210: POKE 795,208: CALL 768: REM TRANSFERT ECRAN 1
60 GOSUB 220: VTAB 10: LIST 70: LIST 210
70 GOSUB 210: POKE 795,212: CALL 768: REM TRANSFERT ECRAN 2
80 GOSUB 220: VTAB 10: LIST 90: LIST 210
90 GOSUB 210: POKE 795,216: CALL 768: REM TRANSFERT ECRAN 3
100 GOSUB 220: VTAB 10: LIST 110: LIST 210
110 GOSUB 210: POKE 795,220: CALL 768: REM TRANSFERT ECRAN 4
115 :
116 REM RAPPEL DES 4 ECRANS A L'ADRESSE $0400-07D0
117 :
120 GOSUB 220: VTAB 12: PRINT "RECUPERATION DES ECRANS 1 2 3 ET 4
-----"
130 GOSUB 220: LIST 140: LIST 230: LIST 235: GOSUB 225
140 GOSUB 230: GOSUB 235: CALL 806: REM RETOUR ECRAN 1
150 GOSUB 220: LIST 160: LIST 230: LIST 240: GOSUB 225
160 GOSUB 230: GOSUB 240: CALL 806: REM RETOUR ECRAN 2
170 GOSUB 220: LIST 180: LIST 230: LIST 245: GOSUB 225
180 GOSUB 230: GOSUB 245: CALL 806: REM RETOUR ECRAN 3
185 GOSUB 220: LIST 190: LIST 230: LIST 250: GOSUB 225
190 GOSUB 230: GOSUB 250: CALL 806: REM RETOUR ECRAN 4
195 GOSUB 220: END
210 POKE 769,129: POKE 772,129: POKE 803,130: GOSUB 255: RETURN
220 CALL - 198: VTAB 22: PRINT "APPUYEZ SUR UNE TOUCHE POUR CONTINUER"
: POKE - 16368,0: WAIT - 16384,128,127: POKE - 16368,0: VTAB 3: CALL
- 958: RETURN
225 CALL - 198: VTAB 22: INVERSE : PRINT "PRESSEZ UNE TOUCHE POUR LE R
EVOIR": NORMAL : POKE - 16368,0: WAIT - 16384,128,127: POKE - 163
68,0: VTAB 3: RETURN
230 POKE 807,128: POKE 838,130: RETURN
235 POKE 814,208: POKE 822,211: RETURN
240 POKE 814,212: POKE 822,215: RETURN
245 POKE 814,216: POKE 822,219: RETURN
250 POKE 814,220: POKE 822,223: RETURN
255 X = X + 1: VTAB 20: HTAB 33: INVERSE : PRINT "ECRAN "X: NORMAL : RETURN
```

Dans un programme, tout ce blablabla est inutile.

Même remarque que ci-dessus. Une boucle bien organisée fera votre bonheur !

Un détail : le disque virtuel /RAM vous permet, sous ProDOS, de mémoriser... puis de récupérer instantanément vos écrans...

ProDOS et VOUS

PETIT à petit, le système d'exploitation ProDOS remplacera le bon vieux DOS 3.3, archi connu et non moins tolérant, mais moins évolué que son cadet.

Puisque vous avez le privilège de posséder un Apple, apprenez donc à utiliser le petit nouveau, Pro comme professionnel et DOS comme... son ancêtre.



RAM Premier avantage de ProDOS — important, à mes yeux du moins — l'emploi de la mémoire auxiliaire de la carte 80 colonnes étendue, sous forme d'un disque-mémoire de 61 K. Comment y accéder ? par la commande `/RAM`, simplement...

Ainsi, vous pouvez stocker un écran sur ce troisième drive en tapant `BSAVE/RAM/ECRAN, A$400, L$300`, puis le récupérer par `BLOAD /RAM/ECRAN`. Ça ne marche pas ? (`NO BUFFERS AVAILABLE`) Normal. Faites précéder le `BLOAD` par `POKE 48984,0...` et vous verrez bel et bien votre écran. Détail qui a son importance, `/RAM` est accessible à tout moment, y compris lorsqu'on se trouve dans un sous-volume (ou répertoire) de type `DIR`. Pratique, non ?

CATALOG La commande `CATALOG` ne présente d'intérêt que si l'on est en 80 colonnes. Sinon, lui préférer `CAT`, surtout si l'on est paresseux. `CAT` ou `CATALOG` présente le catalogue de la disquette dont ProDOS a enregistré le `PREFIX`. Si la disquette a été changée au profit d'une autre, le système va la rechercher en vain... et il y aura un message d'erreur (`PATH NOT FOUND`). Par contre `CAT,D1` ou `CAT,D2`, de même que `CAT/RAM`, comblent ProDOS de joie (mais il n'oublie pas l'ancien `PREFIX` pour autant).

Comment accéder directement, sans changer de `PREFIX`, au sous-catalogue d'une disquette de préfixe `TM2` ? Pas compliqué (si les titres sont

courts !) : `CAT/TM2/TOTO`. Notez que l'indication du `DRIVE` est inutile : ProDOS le cherche... et le trouve. Quelle belle intelligence !

RUN et BRUN Ce qui suit est également valable pour `LOAD` et `BLOAD`. Quand le programme que vous désirez charger ou lancer n'est pas dans un sous-volume (`DIR`), vous l'obtenez en tapant directement `RUN NOM` ou `BRUN NOM...` ou encore — `NOM` (efficace dans un cas comme dans l'autre). Ceci dit, rien ne vous empêche d'aller charger le programme d'un sous-volume... sans changer de `PREFIX`, mais n'oubliez jamais que ProDOS ne conserve en mémoire que le `PREFIX` qui lui a formellement été indiqué par une commande de type `PREFIX,D1` ou `PREFIX /NOM` (`nom` = préfixe en question). Si vous chargez un programme `LILI` dans le sous-volume `TOTO`, alors que le `PREFIX` fourni actuellement à ProDOS est celui de la disquette (`TM1`, par exemple), vous allez taper `LOAD/TM1/TOTO/LILI`, et cela va marcher. Toutefois, attention ! Si votre programme fait appel à un fichier écrit dans le sous-volume `TOTO` (c'est fréquent), il va se planter... à moins que votre commande en Basic n'utilise elle-même la formule ci-dessus `/TM1/TOTO /FICHIER`, par exemple.

Il est donc souhaitable, avec ProDOS et les sous-catalogues `DIR`, de changer de `PREFIX` avant de taper un `RUN`. Dans le cas évoqué ci-dessus, commencez par `PREFIX/TM2/TOTO` puis — `LILI...` et tout se passera bien.

Si vous rencontrez des problèmes en travaillant avec ou sous ProDOS, n'hésitez pas à m'écrire. La consultation est gratuite (sauf pour les jeunes filles, dont j'attends une bise charitable) !

NESTOR.

Saisie d'un texte, suivie de son affichage

COMMENT vider votre écran, matérialiser la longueur maximum du texte à rentrer par 255 points, saisir ce texte (avec sa ponctuation), puis le réafficher instantanément ? Rien de plus facile, avec le petit programme ci-dessous.

UN DÉTAIL : le contenu du buffer est transféré à l'adresse \$2000, mais ce choix est arbitraire et on peut évidemment utiliser une autre adresse.

RÉCUPÉRATION : dans l'état actuel du programme, on ne peut pas récupérer le texte saisi (et mémorisé à l'adresse \$2000) sous forme de variable, mais il est possible de l'écrire sur une disquette, sous forme de fichier B, par exemple en tapant un *BSAVE TEXT, A\$2000, L\$100**.

Un *BLOAD TEXT* lui fera retrouver sa place à l'adresse \$2000 (ou à une autre, car on peut taper *BLOAD TEXT, A\$3000*, par exemple).

* Pourquoi \$100 alors que la longueur maximum du texte est de \$FF (255 caractères) ? Pour ne pas perdre les guillemets (\$22), qui indiquent la fin du texte.

300 — 20 58 FC	JSR \$FC58	\$FC58, c'est le HOME du Basic.
303 — A9 04	LDA £\$04	\$04 est mis dans l'Accumulateur A, puis écrit à l'adresse \$25 (qui correspond à VTAB)
305 — 85 25	STA \$25	
307 — 20 8E FD	JSR \$FD8E	Saut à la ligne suivante
30A — A9 AE	LDA £\$AE	L'Accumulateur est chargé avec \$AE(.)
30C — A0 FF	LDY £\$FF	Boucle de \$FF (255 caractères)
30E — 20 F6 FD	JSR \$FDF6	Affichage du contenu de A
311 — 88	DEY	Moins 1 pour Y
312 — D0 FA	BNE \$030E	Si Y est différent de zéro, un autre tour !
314 — A9 04	LDA £\$04	Sinon on charge A avec la valeur \$04
316 — 85 25	STA \$25	... pour placer le curseur au début des lignes de points que l'on a tracées
318 — 20 8E FD	JSR \$FD8E	
31B — 20 6F FD	JSR \$FD6F	C'est parti pour un INPUT
31E — A2 00	LDX £\$00	X est mis à zéro (Boucle)
320 — BD 00 02	LDA \$0200,X	Lecture du buffer (\$200 + valeur de X)
323 — C9 8D	CMP £\$8D	Est-ce le caractère RETURN ?
325 — F0 06	BEQ \$032D	Si oui, terminé : on va à \$32D
327 — 9D 00 20	STA \$2000,X	Ecriture de A à l'adresse \$2000 + valeur de X
32A — E8	INX	X est incrémenté de 1
32B — D0 F3	BNE \$0320	Retour au début de la boucle
32D — A9 22	LDA £\$22	Guillemets (\$22) dans A
32F — 9D 00 20	STA \$2000,X	Ils vont limiter le texte à l'adresse \$2000 + X
332 — A9 0C	LDA £\$0C	\$0C dans A (12 en décimal)
334 — 85 25	STA \$25	... pour VTAB
336 — 20 8E FD	JSR \$FD8E	Toujours le retour chariot
339 — A9 00	LDA £\$00	
33B — A0 20	LDY £\$20	On place l'adresse du texte à afficher dans A (partie basse) et Y (partie haute)
33D — 20 3A DB	JSR \$DB3A	\$DB3A affiche une chaîne pointée par Y, A, et se terminant par " ou 0.
340 — 60	RTS	

CE PROGRAMME N'EST UTILISABLE QU'A PARTIR DU BASIC PAR UN CALL 768

Les variables dans la mémoire de l'Apple

LANGAGE MACHINE

On sait comment est organisée la mémorisation des variables et il est donc facile de les retrouver, grâce à une série de pointeurs, à des adresses parfaitement connues. Il existe pour cela divers procédés. Nous nous intéresserons ici au plus simple, qui utilise, en la modifiant (à peine) et en la complétant, la routine de la page précédente.

Saisir un texte est certes intéressant, mais son utilisation reste acrobatique s'il n'est pas transformé en variable. Pallions donc cet inconvénient !

Dans notre exemple, le programme Basic pourrait ressembler à ceci :

```
10 A$ = " "
20 CALL 768
```

IMPORTANT :

A\$ (ou un autre nom) doit absolument être la première variable du programme en Basic, ce qui, nous allons le voir immédiatement, simplifie la recherche.

```
332 — 86 08      STX $08
334 — A9 0C      LDA £$0C
336 — 85 25      STA $25
338 — 20 8E FD   JSR $FD8E
33B — A9 00      LDA £$00
33D — A0 20      LDY £$20
33F — 20 3A DB   JSR $DB3A
342 — A5 69      LDA $69
344 — 85 06      STA $06
346 — A5 6A      LDA $6A
348 — 85 07      STA $07
34A — A0 02      LDY £$02
34C — A5 08      LDA $08
34E — 91 06      STA ($06),Y
350 — C8         INY
351 — A9 00      LDA £$00
353 — 91 06      STA ($06),Y
355 — C8         INY
356 — A9 20      LDA £$20
358 — 91 06      STA ($06),Y
35A — 60         RTS
```

La longueur du texte saisi est contenue en X et il convient de la stocker en page zéro, à une adresse disponible (en effet, le JSR \$FD8E va détruire la valeur de X).

On lit ici l'adresse, partie basse de la variable A\$...

... et on la mémorise page zéro

Maintenant, c'est l'adresse de la partie haute...

... également stockée page zéro

Y est mis à \$02

Récupération de la longueur

Ecriture à l'adresse lue en 6\$ et 7\$ + Y

Incrémentation de Y

Partie basse de l'adresse \$2000

Ecriture à l'adresse lue en 6\$ et 7\$ + Y

Incrémentation de Y

Partie haute de l'adresse \$2000

Ecriture à l'adresse lue en 6\$ et 7\$ + Y

Retour au Basic.

RÉPONSES À DEUX QUESTIONS

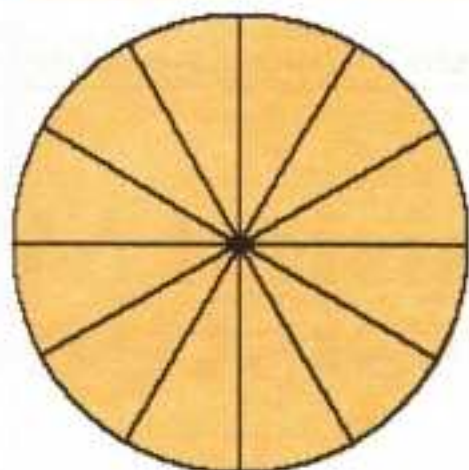
1. Pourquoi est-on sûr de lire l'adresse de la variable A\$ en \$69 — \$6A ?

Parce qu'elle est la première du programme, sinon, il faudrait la chercher... ce que nous étudierons plus tard.

2. Pourquoi donner une valeur de \$02 à Y (ligne 34A) ?

Le descriptif de chaque variable occupe 7 octets. En \$06 — \$07, on a l'adresse (inversée) du premier. Or, celui-ci ne nous intéresse pas. Il contient en effet le nom de la variable (ici : \$41). Le suivant pas davantage. Par contre, le troisième contient sa longueur, le quatrième et le cinquième son adresse (inversée).

Relisez à ce sujet les explications données pages 44 et 45 du N°2 de TREMPIN MICRO.



DESSINE-MOI UN CERCLE

LE BASIC de l'Apple n'est pas le LOGO, et il ne comporte pas de commande permettant de tracer directement une circonférence.

Ce programme de démonstration vous propose deux solutions. La première est lente, mais économe en mémoire. La seconde est rapide, mais grande consommatrice de variables numériques. Nous avons en outre partagé le gâteau en 12 !

HGR initialise la page graphique et POKE -16302,0 permet de supprimer la fenêtre TEXT.

L'unité de mesure d'angle est le radian. Or, la longueur d'une circonférence de rayon 1 vaut 2π radians ($PB = \pi * 2$).

CE est l'angle (en radians) correspondant à chaque secteur, ou part du gâteau.

Calcul préalable des 12 angles.
Initialisation.

Quand on obtient la valeur de l'angle du nouveau secteur, on trace sa limite.

En inversant X et Y, le cercle est tracé à l'envers (sens contraire des aiguilles d'une montre).

La valeur de ND pourrait être 90, pour un tracé plus grossier... ou 360 (mais gare à la place-mémoire !)

Comme vous pouvez le constater, on mémorise les valeurs de COSINUS et SINUS ce qui va permettre (lignes 290 et 300), de gagner un temps considérable par la suite.

INITIALISATION

```
10 HGR : POKE -16302,0 : HCOLOR = 3
20 S = 12
30 DIM C(S)
40 PI = 3.1415926
50 PB = PI * 2
60 CE = PB / S
70 XC = 140 : YC = 94
80 R = 90
```

S	=	Nombre de parts
PI	=	Valeur habituelle
R	=	Rayon du cercle
XC	=	Position horizontale du centre
YC	=	Position verticale

MÉTHODE LENTE

```
90 FOR J = 0 TO PB STEP CE : C(I) = J : I = I + 1 : NEXT
100 HPLOT XC, YC TO XC, YC : I = 1
110 FOR A = 0 TO PB STEP CE
120 IF A >= C(I) THEN HPLOT XC, YC TO XC + X, YC + Y : I
    = I + 1
130 Y = R * SIN (A)
140 X = R * COS (A)
150 HPLOT TO XC + X, YC + Y
160 NEXT
170 NS = 1 : GOSUB 350
```

MÉTHODE RAPIDE

```
200 HGR : POKE -16302,0 : NS = 0
210 ND = 180
220 DIM SI(ND) : DIM CO(ND)
230 FOR I = 1 TO ND
240 A = PB / ND * I
250 CO(I) = COS(A) * R : SI(I) = SIN(A) * R
260 NEXT
270 HPLOT XC + R, YC
280 FOR I = 1 TO ND
290 X = XC + CO(I)
300 Y = YC + SI(I)
310 HPLOT TO X, Y
320 NS = NS + 1 : IF NS = ND / S THEN NS = 0 : HPLOT XC,
    YC TO X, Y
330 NEXT
340 NS = 0
```

CALCUL FINAL DES COORDONNÉES DE CHAQUE POINT DE LA CIRCONFÉRENCE

ATTENTE

```
350 CALL-198 : POKE -16368,0 : WAIT -16384,128,127 : POKE
    -16368,0 : HOME : TEXT : IF NS <> 0 THEN RETURN
```




Saisie LM assistée

Recopier les 768 octets d'une fonte de caractères deviendra plus facile, pour vous, avec cette petite routine inédite.

FONCTIONNEMENT

La routine vous demande l'adresse de départ de votre programme en langage machine. Si celle-ci est \$300, pas de problème. Par contre, si elle est comprise entre \$800 et \$1770, (emplacement occupé par Saisie

LM assistée) convenez d'une autre adresse. Il est facile de modifier l'adresse d'un programme enregistré sur disquette (lire plus loin).

Toutes les huit lignes, le programme transfère les 64 octets qui viennent d'être saisis à leur adresse définitive. A la fin, le bloc entier est automatiquement sauvé sur disquette et l'on vous annonce la couleur : Adresse et longueur.

LIMITATIONS

La mémoire de l'Apple, bien sûr, constitue la première limitation, mais il en existe une autre : au fur et à mesure de la saisie, les variables utilisées dans le programme deviennent envahissantes et risquent d'écraser vos précieux octets. Comment éviter cela ? en prenant la précaution (élémentaire) de revoir la valeur de LOMEM (ligne 10) et, éventuellement, en utilisant HIMEM.

Exemple : vous désirez recopier DIA.CARAC de la page 57 (\$6100 à \$63FF). Fixez carrément un HIMEM : 24576 ou remontez LOMEM à 25600 (\$6400). Dans un cas comme dans l'autre, vos données seront ainsi mises à l'abri d'un quelconque «écrasement».

TRANSFERT

Nous l'avons évoqué plus haut, il peut être intéressant de saisir un programme à une autre adresse que celle à laquelle il est prévu. Si cela vous arrive, notez les indications qui apparaîtront sur votre écran et procédez de la manière suivante (ceci concerne les néophytes). Supposons que vos notes donnent ceci : BSAVE TOTO, A\$2000, L768 \$22FF, mais que, pour des raisons personnelles, vous teniez à avoir TOTO à l'adresse \$801. Deux solutions :

1. Tapez CALL-151 (RETURN), puis, immédiatement à la suite de l'astérisque :
 * 801<2000.22FFM, et RETURN, puis
 * BSAVE TOTO1, A\$801, L768 ... RETURN et CTRL-C...RETURN.
2. Tapez simplement BLOADTOTO,A\$801 (RETURN), puis BSAVE TOTO1, A\$801, L768 (RETURN). Bien entendu, la routine SAISIE LM ASSISTÉE (qui était à l'adresse 801) a disparu, mais elle est sur disquette.

Pourquoi TOTO1 ? Pour conserver la version originale (on ne sait jamais). Si tout a bien marché, il sera toujours possible d'effacer TOTO et de TAPER un *RENAME TOTO1, TOTO*, pas vrai ?

```

10 LOMEN: 6000
20 TEXT : PRINT CHR$ (12); CHR$ (21)
   : HOME
30 D$ = CHR$ (4): GOSUB 1140
40 FOR I = 1 TO 10:T$ = T$ + " ____ ";
   NEXT
50 INVERSE : PRINT "- SAISIE ASSISTEE
   D'UN PROGRAMME EN LM -": NORMAL
60 LIGNES$ = "153616641792192010641192
   13201440157617041832196011041232"
70 FOR I = 1 TO 27 STEP 9: VTAB 2 + (
   I + 8) / 9: HTAB 32: PRINT MID$ ("
   ADRESSE DE DEPARTSVP ?",1,9): NEXT
   : CALL - 198
80 GOSUB 140: GOTO 320
90 :
100 REM *****
110 REM *** PRIORITE AUX GOSUB ***
120 REM *****
130 :
140 VTAB 18: PRINT : CALL - 958: PRIN
   T T$
150 PRINT : VTAB 21: CALL - 860: RETU
   RN
160 VTAB 21: HTAB 12: INVERSE : PRINT
   "RETURN POUR STOPPER": NORMAL : RET
   URN
170 SA$ = SA$ + S$ + CHR$ (32):S$(V -
   8) = S$: RETURN
180 :
190 REM *****
200 REM *** PLUS HUIT OCTETS ***
210 REM *****
220 :
230 IF LI = 1 THEN 260
240 P6 = PEEK (6):P7 = PEEK (7)
250 CALL C1:N$ = "": FOR I = 1823 TO 1
   826:N$ = N$ + CHR$ ( PEEK (I) - 12
   8): NEXT :N$ = RIGHT$ (( "000" + N$
   ),4)
260 VTAB 8: HTAB 32: CALL C2: RETURN
270 :
280 REM *****
290 REM *** ADRESSE DE DEPART ***
300 REM *****
310 :
320 N$ = "0000":N = 1: GOTO 370
330 N$ = "": FOR I = 1616 TO 1619:N$ =
   N$ + CHR$ ( PEEK (I) - 128): NEXT
340 IF N > 5 THEN N = 5: CALL - 198
350 IF N < 4 THEN 370
360 GOSUB 160
370 VTAB 20: PRINT : PRINT N$ " " : HTA
   B N: GET R$:R = ASC (R$): IF R = 1
   3 THEN 420
380 IF (R > 47 AND R < 58) OR (R > 64
   AND R < 71) THEN N = N + 1: PRINT R
   $: GOTO 330
390 IF N > 0 AND (R = 60 OR R = 8) THE
   N N = N - 1: GOTO 330

```

```

400 IF R = 21 OR R = 62 THEN N = N + 1
   : GOTO 330
410 POKE N + 1615,176: GOTO 330
420 VTAB 21: HTAB 12: FLASH : PRINT "C
   ONFIRMEZ": NORMAL : PRINT " PAR (O
   UI) SVP " : CALL - 198: GET R$: IF
   R$ < > "0" AND R$ < > "o" THEN
   GOSUB 150:N = 1: GOTO 370
430 AD$ = N$:G$ = "06:" + RIGHT$ (N$,2
   ) + " " + LEFT$ (N$,2) + " N0823G"
440 FOR I = 1 TO LEN (G$): POKE 511 +
   I, ASC ( MID$ (G$,I,1)) + 128: NEX
   T : POKE 72,0: CALL - 144
450 VTAB 5: HTAB 32: INVERSE : PRINT N
   $: NORMAL : PRINT " " : CALL - 9
   58
460 :
470 REM *****
480 REM *** SAISIE D'UNE LIGNE ***
490 REM *****
500 :
510 GOSUB 140: GOSUB 160:V = 8:LI = 0:
   FOR I = 1 TO 8:S$(I) = "00 00 00 0
   0 00 00 00 00": NEXT
520 H = 5:E = 1:LI = LI + 1
530 VTAB 7: HTAB 32: GOSUB 230
540 IF V = 8 THEN D1$ = N$
550 P = VAL ( MID$ (LI$,V * 4 - 19,4))
   + 5
560 S$ = S$(V - 7):N = 1: GOTO 630
570 IF E = 0 THEN N = N - 1:E = 2: IF
   N < 1 THEN N = 1:E = 1
580 GOTO 630
590 IF E = 3 THEN N = N + 1:E = 1:
600 IF N > 23 THEN N = 23:E = 2
610 IF N = 23 THEN CALL - 198: CALL
   - 198
620 S$ = "": FOR I = P TO P + 22:S$ = S
   $ + CHR$ ( PEEK (I) - 128): NEXT
630 VTAB V: HTAB 1: PRINT N$:"":S$: H
   TAB H + N: GET R$
640 R = ASC (R$): IF R = 13 THEN 690
650 IF (R > 47 AND R < 58) OR (R > 64
   AND R < 71) THEN N = N + 1: PRINT R
   $:E = E + 1: GOTO 590
660 IF N > 0 AND (R = 60 OR R = 8) THE
   N N = N - 1:E = E - 1: GOTO 570
670 IF R = 21 OR R = 62 THEN N = N + 1
   :E = E + 1: GOTO 590
680 POKE N + P - 1,176: GOTO 620
690 GOSUB 150
700 VTAB 21: HTAB 6: PRINT "<": INVER
   SE : PRINT "C": NORMAL : PRINT ">"
   POUR CONTINUER LA SAISIE " : CALL
   - 198: GET R$: IF R$ < > "C" AND R
   $ < > "c" THEN 730
710 V = V + 1: GOSUB 170: IF V > 15 THE
   N 820
720 GOTO 520

```

(SUITE PAGE 50)

SAISIE LM ASSISTÉE (suite)

```

730 VTAB 21: HTAB 5: PRINT "<"; INVER
SE : PRINT "0"; NORMAL : PRINT ">"
POUR STOPPER VOTRE SAISIE ";; CALL
- 198: GET R#: IF R# < > "0" AND
R# < > "o" THEN 690
740 IF N = 1 THEN N = 23:LI = LI - 1:
POKE 6,P6: POKE 7,P7: GOTO 760
750 GOSUB 170
760 F = 1
770 :
780 REM *****
790 REM *** SAUVEGARDER HUIT LIGNES ***
800 REM *****
810 :
820 G$ = D1$ + ":" + SA$ + "ND823G":SA$
= ""
830 FOR I = 1 TO LEN (G$): POKE 511 +
I, ASC ( MID$ (G$,I,1)) + 128: NEX
T : POKE 72,0: CALL - 144
840 IF F < > 1 THEN V = 8: GOTO 520
850 IF N = 23 THEN L = LI * 8:N = 8: G
OTO 920
860 N = (N - 1) / 3:L = (LI * 8 - 8) +
N: IF L < = 8 THEN END
870 :
880 REM *****
890 REM *** ECRITURE DU FICHIER ***
900 REM *****
910 :
920 GOSUB 150: VTAB 21: HTAB 5: INPUT
" TITRE DU FICHIER A CREER ? ":TI$:
IF TI$ = "" THEN 920
930 IF ASC (TI$) < 65 OR ASC (TI$) >
90 THEN 920 ProDOS : Lire notre
rectificatif page 68.
940 PRINT D$"MONC,I,0"
950 GOSUB 140: VTAB 21: PRINT D$"BSAVE
"TI$,"A$"AD$,"L"L: POKE 50,128
960 PRINT D$"NDMON C,I,0": POKE 50,255
970 N = N - 1
980 POKE 5984,N:D2$ = "": VTAB 21: POK
E 36,38: CALL C1: FOR I = 1646 TO 1
649:D2$ = D2$ + CHR$ ( PEEK (I) -
128): NEXT
990 :
1000 REM *****
1010 REM *** IMPRESSION EVENTUELLE ***
1020 REM *****
1030 :
1040 CALL - 198: POKE - 16398,0: WAIT
- 16384,128,127: POKE - 16368,0:
GOSUB 140
1050 VTAB 21: HTAB 5: PRINT "<"; INVER
SE : PRINT "I"; NORMAL : PRINT ">"
POUR IMPRESSION DE SAISIE ";; CALL
- 198: GET R#: IF R# < > "I" AND
R# < > "i" THEN 1070
1060 PRINT : PRINT D$"PREI"

```

- SAISIE ASSISTEE D'UN PROGRAMME EN LM -

ADRESSE
DE DEPART
0300

0338
524

```

0300:00 00 00 00 00 00 00 00
0300:00 00 00 00 00 00 00 00
0310:00 00 00 00 00 00 00 00
0310:00 00 00 00 00 00 00 00
0320:00 00 00 00 00 00 00 00
0320:00 00 00 00 00 00 00 00
0330:00 00 00 00 00 00 00 00
0338:00 00 00 00 00 00 00 00

```

<C> POUR CONTINUER LA SAISIE

```

1070 G$ = AD$ + "." + D2$ + " N17260": F
OR I = 1 TO LEN (G$): POKE 511 + I
, ASC ( MID$ (G$,I,1)) + 128: NEXT
: POKE 72,0: CALL - 144
1080 HOME : END
1090 :
1100 REM *****
1110 REM *** ROUTINE ADDITION HEXA ***
1120 REM *****
1130 :
1140 POKE 5980,165: POKE 5981,6: POKE 5
982,24: POKE 5983,185: POKE 5984,8:
POKE 5985,133: POKE 5986,6: POKE 5
987,178: POKE 5988,144
1150 POKE 5989,2: POKE 5910,230: POKE 5
911,7: POKE 5912,164: POKE 5913,7:
POKE 5914,32: POKE 5915,64: POKE 59
16,249: POKE 5917,96
1160 POKE 5918,165: POKE 5919,7: POKE 5
920,166: POKE 5921,6: POKE 5922,32:
POKE 5923,36: POKE 5924,237: POKE
5925,96
1170 POKE 5926,32: POKE 5927,251: POKE
5928,218: POKE 5929,32: POKE 5930,1
47: POKE 5931,254: POKE 5932,32: POK
E 5933,88: POKE 5934,252: POKE 593
5,76: POKE 5936,288: POKE 5937,3
1180 C1 = 5980:C2 = 5918: RETURN

```

NOTE

Les lignes 1140 et 1170 peuvent être remplacées par 38 octets en langage-machine. Pour les saisir et les sauvegarder, respecter la procédure ci-après (☐ signifie RETURN).

CALL-151 ☐ (l'astérisque doit apparaître sur votre écran).

* 170C : A5 06 18 69 08 85 06 AA 90 02 E6 07 A4 07
20 40 F9 60 A5 07 A6 06 20 24 ED 60 20 FB DA 20
93 FE 20 58 FC 4C D0 03 ☐

* BSAVE 51, A\$170C, L38 ☐

* CTRL-C (OU RESET) vous fera retrouver le Basic.

Tapez ensuite : 1140 PRINT CHR\$(4)"BLOAD 51" ☐
DEL 1150, 1170 ☐ ■



QUELQUES NOTES POUR VOTRE APPLE

FAITSON.LM

0300-	A9 0A	LDA	£#0A
0302-	85 06	STA	#06
0304-	A9 0E	LDA	£#0E
0306-	85 07	STA	#07
0308-	AD 30 C0	LDA	#C030
030B-	88	DEY	
030C-	D8 04	BNE	#0312
030E-	C6 07	DEC	#07
0310-	F0 07	BEQ	#0319
0312-	CA	DEX	
0313-	D8 F6	BNE	#030B
0315-	A6 06	LDX	#06
0317-	D0 EF	BNE	#0308
0319-	A5 06	LDA	#06
031B-	C9 28	CMP	£#28
031D-	F0 07	BEQ	#0326
031F-	18	CLC	
0320-	69 05	ADC	£#05
0322-	85 06	STA	#06
0324-	D8 DE	BNE	#0304
0326-	60	RTS	

BSAVE FAITSON.LM, A\$300, L\$27

L ES enfants sont particulièrement sensibles aux manifestations sonores (les adultes ne le sont pas moins d'ailleurs !), et il est recommandé, dans un jeu, de ponctuer chaque bonne réponse par une petite musique. Celle que je vous propose est assez particulière... et je vous la laisse découvrir !

Vous l'obtiendrez directement, dans un programme, par un simple mais efficace **PRINT CHR\$(4) ; "EXEC SONDIR"**. Comment créer ce fichier EXEC ? Comme ceci :

SONDIR

```
10 D$ = CHR$(4):G$ = CHR$(34): PRINT D$"OPENSNDIR": POKE 33,33:
  PRINT D$"WRITESNDIR"
11 PRINT "FOR$=768TO786:POKE$,VAL(MID$(" + G$ + "001730481921362080
  84198007240008202208246166006076000003096" + G$ + ",(S-767)*3,3))
  :NEXT$ :FORU=10TO40STEP5:POKE$,U:POKE7,15:CALL768:NEXTU"
12 PRINT D$"CLOSE": POKE 33,40
```

Mais il n'est pas interdit d'inclure une ligne 10000 (par exemple) dans ce même programme, ligne que vous ne retaperez jamais plus (quelle belle économie, en vérité !) si vous créez, là encore, un utilitaire EXEC que vous titrerez SONPRO (PRO = programme, intelligent, non ?)

SONPRO

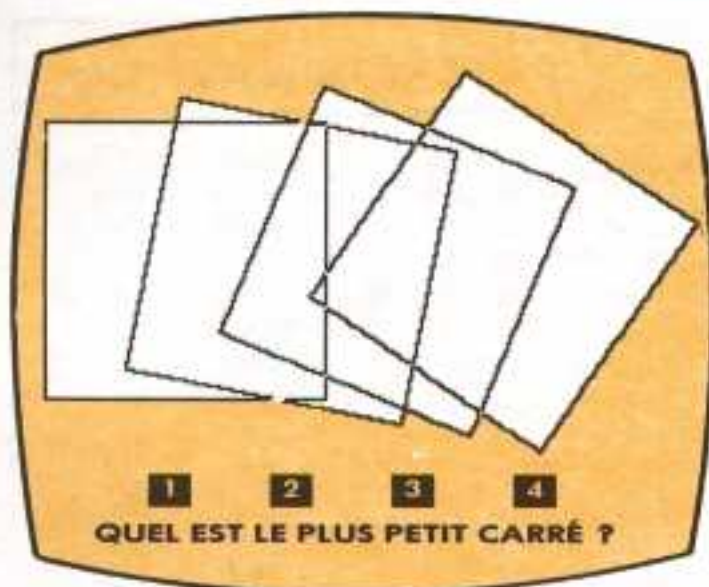
```
10 D$ = CHR$(4):G$ = CHR$(34): PRINT D$"OPENSNDIR": POKE 33,33:
  PRINT D$"WRITESNDIR": LIST 10000: PRINT D$"CLOSE": POKE 33,40:
  END
10000 FOR S = 768 TO 786: POKE S, VAL ( MID$ ("0017304819213620800419
  8007240008202208246166006076000003096", (S - 767) * 3,3)): NEXT S:
  FOR U = 10 TO 40 STEP 5: POKE $,U: POKE 7,15: CALL 768: NEXT U
```

Evidemment, si vous êtes musicien... ou musicienne, votre PMA (Petite Musique Apple) sera la bienvenue... si elle est sur disquette (on vous la retournera, foi de NESTOR !)

Ouvrez l'œil !

Ce court programme, modifiable à volonté, se contente d'afficher 4 carrés sur votre écran, puis de tester l'exactitude de votre réponse à une seule et unique question :

"Quel est le plus petit carré ?". La routine en langage machine ne compte que 71 octets. Elle peut être comprise sans une connaissance approfondie de l'assembleur.



```

10 TEXT : PRINT CHR$(21): HOME
20 PRINT CHR$(4)"BLOAD OUVREZ.LM"
30 T$ = " OUVREZ L'OEIL! "
40 POKE 32,12: POKE 33,28: PRINT
50 INVERSE : FOR I = 8 TO 12: VTAB I: PRINT "
    ": NEXT
60 VTAB 8: PRINT "": HTAB 16: PRINT "": VTAB 10: P
RINT T$: VTAB 12: PRINT "": HTAB 16: PRINT "": N
ORMAL
70 POKE 32,4: POKE 33,33: PRINT : PRINT : PRINT "QUAT
RE CARRES VONT APPARAITRE SUR": PRINT "VOTRE ECRAN.
IL S'AGIT DE TROUVER": PRINT "    QUEL EST LE PLU
S PETIT."
80 TEXT : GOSUB 220
90 X = INT ( RND (1) * 4): POKE 7,X: HOME
100 X = X + 1: CALL 760
110 VTAB 21: FOR I = 8 TO 32 STEP 8: HTAB I: INVERSE :
    PRINT I / 8:; NEXT : NORMAL : PRINT
120 VTAB 23: HTAB 6: PRINT "QUEL EST LE PLUS PETIT CAR
RE ?": GOSUB 220
130 IF R < 1 OR R > 4 THEN VTAB 22: PRINT : GOTO 120
140 VTAB 22: CALL - 950: VTAB 23: INVERSE : IF R = X
THEN HTAB 15: PRINT "C'EST EXACT!": GOTO 160
150 HTAB 8: PRINT "ERREUR! C'EST LE NUMERO "X
160 FLASH : VTAB 21: HTAB X * 8: PRINT X: NORMAL : GOS
UB 220
170 HOME : VTAB 22: INVERSE : PRINT 1:; NORMAL : PRINT
    " UN NOUVEAU TEST ? ":; INVERSE : PRINT 2:; NORMAL
    : PRINT " MERCI! C'EST FINI": GOSUB 220
180 IF R = 1 THEN 90
190 IF R < > 2 THEN 170
200 TEXT : HOME : END
210 :
220 CALL - 198: POKE - 16368,0: WAIT - 16384,128,12
7: POKE - 16368,0: R = PEEK ( - 16384) - 48: RETUR
N

```

✦ POKE 32,12 correspond à HTAB13. Cela évite une répétition de HTAB dans le programme.

✦ Le numéro du plus petit carré est déterminé aléatoirement. On l'incrémente de 1 car le numérotage du programme en langage machine commence à zéro.

Le programme ne prévoit pas de mémorisation des résultats, et peut donc être amélioré en ce sens.

Si vous ne possédez pas d'assembleur, contentez-vous de taper ces valeurs, après un CALL-151

0300: 20 E2 F3
0303: A9 00
0305: 05 06
0307: 05 00
0309: A9 40
030B: 05 09
030D: A5 09
030F: AA
0310: 18
0311: 69 2B
0313: 05 09
0315: A9 4C
0317: A0 00
0319: 20 11 F4
031C: A9 39
031E: A4 06
0320: C4 07
0322: D0 03
0324: 38
0325: E9 02
0327: 05 E7
0329: A5 00
032B: A2 41
032D: A0 03
032F: 20 5D F6
0332: A5 06
0334: C9 03
0336: F0 00
0338: E6 06
033A: E6 00
033C: E6 00
033E: D0 CD
0340: 60
0341: 21 3F 36
20 1C 00

```

0 *****
1 *
2 * OUVREZ L'OEIL! G-H 05 *
3 *
4 *****
5 ORG $300
6 *-----*
7 NBR EQU $06 ;Pointeur pour nombre de carrés
8 ROT EQU $08 ;Rotation
9 HOR EQU $09 ;Position hor. de base du carré
10 PETIT EQU $07 ;Numéro d'ordre du petit carré X
11 SCALE EQU $E7 ;Scale
12 HGR EQU $F3E2 ;Passage en mode HGR avec texte
13 HPOSN EQU $F411 ;Pos. le curseur hte résolution
14 DRAW EQU $F65D ;C'est la commande DRAW du Basic
15 FORME EQU $341 ;La forme (ici, un carré)
16 *-----*
17 JSR HGR ;Avec effacement
18 LDA £$00 ;Initialisation générale
19 STA NBR ;Pointeur (ira de 0 à 3)
20 STA ROT ;Rotation = 0 pour commencer
21 LDA £$40 ;Pour initialiser la pos. horiz.
22 STA HOR
23 BOUCLE LDA $09 ;Lecture position horizontale
24 TAX ;mise dans le registre X
25 CLC ;Annulation de la retenue
26 ADC £$2B ;Position augmentée de $2B
27 STA HOR ;et réécrite à l'adresse $09
28 LDA £$4C ;Position verticale
29 LDY £$00 ;Registre Y à zéro
30 JSR HPOSN ;Initialisation du curseur
31 LDA £$39 ;Fact. d'échelle de DRAW (Scale)
32 LDY NBR ;De quel numéro s'agit-il ?
33 CPY PETIT ;Comparer avec X (Basic)
34 BNE REPRISE ;S'il n'y a pas égalité, saut...
35 SEC ;Retenue à 1 pour soustraire
36 SBC £$02 ;Scale = scale -2 (petit carré)
37 REPRISE STA SCALE ;Ne pas oublier
38 LDA ROT ;On récupère la rotation
39 LDX £$41 ;Adr. de la forme (partie basse)
40 LDY £$03 ;Adresse partie haute
41 JSR DRAW ;On trace le carré
42 LDA NBR ;Où en sommes-nous ?
43 CMP £$03 ;Si on est à 3...
44 BEQ BASIC ;...retour au programme Basic
45 INC NBR ;Sinon on incrémente le pointeur,
46 INC ROT ;...et le pointeur de rotation,
47 INC ROT ;2 fois pour faire bonne mesure!
48 BNE BOUCLE ;Et ça repart pour un autre carré
49 BASIC RTS ;Retour au programme Basic
50 *-----*
51 HEX 213F362D1C00

```

SAUVEGARDE SUR DISQUETTE : BSAVE OUVREZ.LM, A768, L71

ÉCRITURE SUR PAGES GRAPHIQUES

```
100 REM **** ECRITURE EN PAGE GRAPHIQUE AVEC LES CARACTERES DU DOS TOOLKIT
110 HIMEM: 34816: TEXT : HOME
120 VTAB 24: PRINT : PRINT : PRINT "COMBIEN DE CARACTERES A UTILISER: "; GET NS$: PRINT NS$:NS = VAL
L (NS$): PRINT : IF NS < 1 OR NS > 3 THEN PRINT CHR$ (7): GOTO 120
130 DIM AD(NS),CS$(NS):D$ = CHR$ (13) + CHR$ (4)
140 FOR CS = 1 TO NS
150 PRINT "CHAR SET NO "CS;" : "; INPUT " ";CS$(CS)
160 IF RIGHT$(CS$(CS),4) = ".SET" THEN CS$(CS) = LEFT$(CS$(CS), LEN (CS$(CS)) - 4)
170 NEXT CS
180 FOR CS = 1 TO NS
190 AD(CS) = 35096 + 768 * (CS - 1)
200 PRINT D$;"BLOAD "CS$(CS)".SET,A";AD(CS)
210 NEXT CS
220 PRINT : INPUT "NOM DE LA PAGE GRAPHIQUE A CHARGER DANS LE DRIVE 2: ";FI$: IF FI$ = "" THEN 230
230 POKE - 16368,0: PRINT "QUELLE PAGE HGR: "; GET PA$: PRINT PA$:PA = VAL (PA$): IF PA < > 1 AND
PA < > 2 THEN 230
240 IF PA = 2 THEN PA$ = "4000": HGR2 : GOTO 260
250 PA$ = "2000": HGR : POKE - 16302,0
260 IF FI$ < > "" THEN PRINT D$;"BLOAD";FI$;" ,D2"
270 CS = 1:X = 1:Y = 1
280 POKE - 16368,0: GET CH$
290 IF ASC (CH$) = 43 THEN CS = CS + 1: GOTO 450
300 IF ASC (CH$) = 45 THEN CS = CS - 1: GOTO 450
310 IF ASC (CH$) = 13 THEN Y = Y + 1:X = 1: GOTO 430
320 IF ASC (CH$) = 27 THEN 480
330 IF ASC (CH$) = 32 THEN X = X + 1: GOTO 420
340 IF ASC (CH$) = 8 THEN X = X - 1: GOTO 420
350 IF ASC (CH$) = 21 THEN X = X + 1: GOTO 430
360 QU = INT (Y / 8):RM = Y - QU * 8
370 B = 128 * RM + 40 * QU + X + 8192 * PA
380 FOR L = 0 TO 7
390 POKE B + 1024 * L, PEEK (AD(CS) + (ASC (CH$) - 32) * B + L)
400 NEXT L
410 X = X + 1
420 IF X = 40 THEN X = 1:Y = Y + 1
430 IF Y = 25 THEN Y = 0
440 IF X < 1 THEN X = 1
450 IF CS > 3 THEN CS = 1
460 IF CS < 1 THEN CS = 3
470 GOTO 280
480 POKE - 16368,0: GET A$: TEXT
490 PRINT "VOULEZ-VOUS SAUVER LA PAGE GRAPHIQUE: "; POKE - 16368,0: GET A$: PRINT A$: IF A$ < > "0
" THEN END
500 PRINT D$;"BSAVE";FI$ + "2,A$";PA$;" ,L$2000,D2"
510 PRINT "PAGE GRAPHIQUE: "; INVERSE : PRINT FI$ + "2"; NORMAL : PRINT " SAUVEE EN "; INVERSE : PR
INT "HGR";PA$; NORMAL : PRINT " SUR LE DRIVE "; INVERSE : PRINT "2": NORMAL
```

PROGRAMME
de Robert FAVERGE
Professeur
au Lycée Thiers
MARSEILLE

Si vous ne disposez pas du DOS TOOLKIT APPLE, vous pouvez utiliser les fontes parues dans *TREMLIN MICRO n°1*, page 22 (FAIRE.F1) et *TREMLIN MICRO n°2*, page 57 (DIA.CARAC). A propos de celle-ci, profitez de l'occasion pour corriger une erreur typographique : **BSAVE DIA.CARAC, A\$6100, L\$300** (au lieu de \$400).

Parfois, dans un programme en Basic, quelques lignes DATA ou de POKES sont préférables à l'appel (par BLOAD ou BRUN) d'une routine en assembleur. Cet utilitaire permet de les générer automatiquement, sous forme de fichier EXEC. Pour les inclure à votre programme, il vous suffira de taper EXEC TITRE pour qu'elles viennent s'insérer à l'emplacement prévu.

Lignes DATA ou lignes de POKES peuvent remplacer une routine en langage machine

```

10 TEXT : PRINT CHR$(12); CHR$(21); HOME
20 HI = PEEK (115) + PEEK (116) * 256
30 T$ = " LIGNES DE POKES OU DE DATAS "
40 INVERSE : HTAB 21 - LEN (T$) / 2; PRINT T$; NORMAL
50 T$ = ""; FOR I = 1 TO 10:T$ = T$ + "____": NEXT I; VTAB 2; PRINT T$; POK
  E 34,3
60 :
70 REM ***** Attention ! sous ProDOS, les lignes
80 REM **      PARAMETRES      **      140 et 200 doivent être modifiées
90 REM ***** (voir REM 135 et 195).
100 :
110 HOME : GOSUB 630; PRINT " (<?) POUR LE CATALOGUE DE LA DISQUETTE"
120 VTAB 4; CALL - 868; CALL - 198; INPUT "TITRE DU PROGRAMME LM ? ";TI$
130 IF TI$ = "?" THEN HOME : POKE 44452,13; POKE 44605,12; PRINT CHR$(4)
  )"CATALOG"; GET R$; PRINT : GOTO 110
135 REM AVEC PRODOS:140 IF TI$="?" THEN HOME:PRINT CHR$(4)"CAT":GETR$:PRIN
  T:GOTO110
140 IF TI$ = "" THEN 710
150 IF ASC (TI$) < 65 OR ASC (TI$) > 90 THEN GOSUB 650; GOTO 120
160 ONERR GOTO 710
170 PRINT CHR$(4)"BLOAD"TI$
180 POKE 216,0
190 GOSUB 630; PRINT " LE (<?) RENVOIE A L'ITEM PRECEDENT"
195 REM AVEC PRODOS:200 AD=PEEK(48825)+PEEK(48826)*256; LA=PEEK(48840)+PEEK
  (48841)*256
200 AD = PEEK (43634) + PEEK (43635) * 256;LA = PEEK (43616) + PEEK (43
  617) * 256
210 PRINT : VTAB 6; PRINT "FICHIER INSTALLE: A";AD; ", L";LA; GOSUB 670
220 GOSUB 630; INVERSE : PRINT 1;: NORMAL : PRINT " ADRESSE CORRECTE - ";:
  INVERSE : PRINT 2;: NORMAL : PRINT " A MODIFIER ";: GET R$: PRINT : IF
  R$ = CHR$(49) THEN 380
230 IF R$ = "?" THEN 120
240 IF R$ < > CHR$(50) THEN 220
250 VTAB 8; HTAB 1; CALL - 958; CALL - 198; INPUT "NOUVELLE ADRESSE ? ";
  AD$: IF AD$ = "" THEN 220
260 IF ASC (AD$) = 36 THEN 290
270 IF VAL (AD$) + LA > HI THEN GOSUB 630; GOSUB 680; GOTO 250
280 GOTO 360
290 AD$ = RIGHT$(AD$, LEN (AD$) - 1);LA = LEN (AD$); IF LA > 4 THEN GOS
  UB 640; GOTO 250
300 R$ = "45:" + RIGHT$(("<?00" + LEFT$(AD$,LA - 2)),2) + " " + RIGHT$(
  AD$,2) + " N ED246 N D8236"
310 VTAB 8; HTAB 20; CALL - 868

```

(Suite page 56)


```

320 FOR I = 1 TO LEN (R$): POKE 511 + I, ASC ( MID$ (R$,I,1)) + 128: NEXT
   : POKE 72,0: CALL - 144
330 AD$ = "": FOR I = 1939 TO 1943: X = PEEK (I) - 128: IF X < 48 OR X > 59
   THEN 350
340 AD$ = AD$ + CHR$ (X)
350 NEXT I
360 AD = VAL (AD$)
370 PRINT : PRINT CHR$ (4)"BLOAD"TI$,A*AD: GOTO 200
380 IF AD > 6000 THEN HIMEM: AD - 1: GOTO 400
390 IF AD > 2000 THEN LOMEM: AD + LA + 1
400 DIM L$(25):G$ = CHR$ (34)
410 C$ = "DATA":S$ = ",":CH = 1
420 PRINT : VTAB 10: CALL - 958: CALL - 198: INVERSE : PRINT "D";: NORMA
L : PRINT "ATAS - ";: INVERSE : PRINT "P";: NORMAL : PRINT "OKES ? ";:
GET R$: PRINT : IF R$ = "?" THEN RUN
430 IF R$ < > "D" AND R$ < > "P" THEN 420
440 IF R$ = "P" THEN CH = 2:C$ = "POKE":S$ = "":
450 PRINT : VTAB 12: CALL - 868: CALL - 198: INPUT "NUMERO DE PREMIERE L
IGNE = ";R$: IF R$ = "" THEN 410
460 IF R$ = "?" THEN 420
470 :
480 REM *****
490 REM ** ECRITURE DATAS OU POKES **
500 REM *****
510 :
520 L = VAL (R$):AF = AD + LA - 1:N = 1
530 GOSUB 690
540 FOR I = AD TO AF: IF LEN (L$(N)) > 220 THEN L$(N) = LEFT$ (L$(N), LE
N (L$(N)) - 1):N = N + 1: GOSUB 690
550 ON CH GOTO 560,570
560 L$(N) = L$(N) + STR$ ( PEEK (I)) + S$: NEXT I: GOTO 580
570 L$(N) = L$(N) + C$ + STR$ (I) + "," + STR$ ( PEEK (I)) + S$: NEXT I
580 L$(N) = LEFT$ (L$(N), LEN (L$(N)) - 1)
590 HOME : FOR I = 1 TO N: PRINT L$(I): PRINT : NEXT
600 TI$ = C$ + "EXEC": PRINT : PRINT "FICHIER EXEC CREE: ";: INVERSE : PRIN
T TI$: NORMAL
610 PRINT : PRINT CHR$ (4)"OPEN"TI$: POKE 33,33: PRINT CHR$ (4)"WRITE"TI
$: FOR I = 1 TO N: PRINT L$(I): NEXT : PRINT CHR$ (4)"CLOSE": POKE 33,
48
620 END
630 VTAB 19: PRINT : CALL - 958: PRINT T$: VTAB 22: RETURN
640 GOSUB 630: PRINT "QUATRE VALEUR MAXIMUM POUR DE L'HEXA": GOSUB 670: RE
TURN
650 GOSUB 630: PRINT G$ + LEFT$ (TI$,1) + G$;: IF LEN (TI$) > 1 THEN PR
INT RIGHT$ (TI$, LEN (TI$) - 1);
660 PRINT " EST INCORRECT": RETURN
670 CALL - 198: FOR I = 1 TO 2500: NEXT : CALL - 198: RETURN
680 PRINT "HAUTEUR MAXIMUM: ";: INVERSE : PRINT HI - LA: NORMAL : GOSUB 67
0: RETURN
690 L$(N) = STR$ (L - 10 + N * 10): IF CH = 1 THEN L$(N) = L$(N) + C$
700 RETURN
710 CALL - 198: POKE 216,0:ER = PEEK (222): GOSUB 630: PRINT "CE FICHIER
EST INCONNU": GOTO 120

```

CONSEIL :

N'allez pas au-delà de quatre ou cinq lignes

Avec des lignes DATA, il faut prévoir d'installer le programme par une boucle telle que celle-ci :

FOR I = X TO Y : READ R : POKE I, R : NEXT (X = adresse début ; Y = adresse fin)

Vous avez écrit à TREMPLIN MICRO



* Je viens de découvrir votre n°1 chez mon marchand de journaux. J'y ai relevé des idées intéressantes, mais aussi des "erreurs" qui se baladent de publications en publications.

Pourquoi sauver un écran texte avec L1024 alors que L1016 marche aussi bien et économise un secteur sur disque ? Même chose en HGR où L8184 sauve un écran en épargnant à nouveau un secteur. Pour poquer une valeur sur 2 octets, il est inutile de faire 2 fois la division Val/256 puisque l'on peut faire POKE ADR+1, VA/256 : POKE ADR, VA*256 + PEEK (ADR+1).

Pour les manipulations d'écran, au lieu d'utiliser la routine MOVE du moniteur qu'il est souhaitable d'accompagner par un appel à SAUVE-MEVEC, je propose le programme MOVE-SCREEN ci-joint qui, par un seul CALL, sauve ou rappelle un écran en évitant de perturber les précieux vecteurs. L'utilisation étant évidente, je ne joins pas de programme de démo.

Veillez agréer, Messieurs, mes salutations distinguées.

KOENIG Yvan (06220 VALLAURIS)

TM **Ecrans TEXT et HGR :** vous avez évidemment raison, et la plupart des utilisateurs d'Apple II+ connaissent (et emploient) ces valeurs, mais les manuels n'ont pas changé — eux — et nous évitons dans un premier temps, de dérouter les néophytes.

Poquer une valeur : bien sûr, mais cela ne change pas grand-chose, pas vrai ?

Manipulation d'écran : votre MOVE-SCREEN est intéressant, mais pas aussi claire (aux yeux du débutant) que vous le pensez. La voici :

```

0300 : A9 3C 8D 24 03 A9 3E D0
0308 : 07 A9 3E 8D 24 03 A9 3C
0310 : 8D 26 03 A9 04 85 3D A9
0318 : 00 85 3E A9 50 85 3F A0
0320 : 00 84 3C B1 3E 91 3C C8
0328 : C0 F8 D0 F7 A5 3D C9 07
0330 : F0 0E E6 3D 98 18 65 3E
0338 : 85 3E 90 E3 E6 3F D0 DF
0340 : 60
    
```

CALL 768 transfère l'écran à l'adresse \$5000.
CALL 777 le récupère.

Tracé d'un cadre : nous avons donné deux ver-

sions du programme... pour aider les nouveaux amateurs de langage machine à le comprendre, mais nous publions volontiers votre arrangement (pas de musique, mais il y a les paroles !)

```

0300 : A9 20 D0 07 A9 A0 D0 03
0308 : 20 0C FD 85 06 A0 00 84
0310 : 24 84 25 84 28 A0 04 84
0318 : 29 A0 27 91 28 88 D0 FB
0320 : A5 06 20 F0 FB A5 06 A0
0328 : 27 84 24 20 F2 FB A5 25
0330 : C9 17 D0 EC A5 06 A2 28
0338 : 9D D0 07 CA 10 FA 60
    
```

CALL 776 avec un caractère du clavier. **CALL 772** pour effacer. **CALL 768** pour cadre en mode inverse.

* Je suis débutante, et je me perds dans les A-B-T du CATALOG. Comment savoir s'il faut employer LOAD ou RUN, ou BLOAD ? Pourriez-vous publier les adresses des Clubs Micro Apple ?

Mme D.C. (FRANCONVILLE)

TM Rassurez-vous, dans ce domaine, à un moment ou à un autre, nous sommes tous débutants !
LOAD charge un programme Basic, mais ne le lance pas. **BLOAD** charge une routine en langage machine, mais ne l'active pas. **RUN** charge et lance un programme en Basic. **BRUN** charge et active une routine en assembleur, ce qui n'est pas toujours souhaitable (lire le texte du programme Basic utilisant la routine).
Quand vous faites CATALOG, A = Applesoft = BASIC (load ou run)
B = Binaire (d'où BLOAD ou BRUN)
T = Fichier (généralement lu à partir d'un programme Basic, mais pouvant parfois être un fichier EXEC... que l'on lance alors par EXEC... TOTO, par exemple).

CLUBS : A eux de se manifester, s'ils estiment que TREMPLIN MICRO mérite leur confiance, bien sûr !

SUITE DU COURRIER (PAGE 58)

Vous avez écrit à TREMLIN MICRO



* Bravo ! Je suis du genre à utiliser des programmes tout faits, et à être conservateur. A cause de TREMLIN MICRO, je me suis mis à ProDOS, et à l'assembleur. Comme je ne connais pas l'assembleur, je suis de près votre initiation. Mais l'exemple de soustraction de TREMLIN MICRO n°2 me laisse perplexe. Je vous fais une contre-proposition, inspirée d'un bouquin bien sûr, mais dont je saisis mieux la logique. Cet exemple ne marche que si le nombre en \$06-\$07 est plus grand que le nombre \$08-\$09. Je vous soupçonne de le faire exprès pour voir si vos lecteurs suivent... Mais ça ne rompt pas le charme de votre revue. A bientôt.

Jean CONVERT (26000 VALENCE)

TM Rassurez-vous, je ne suis pas du genre «canailou»... encore que... Notre programme DEMO (soustraction) se contentait de soustraire (successivement) le contenu des mémoires 8 et 9 (deux valeurs isolées) du nombre de deux octets placé, à l'envers comme il se doit, dans les mémoires 6-7. Votre petite routine, tout à fait classique, soustrait 8-9 (deux octets) de 6-7 (deux octets). C'est une vraie soustraction. La voici ; pour que l'on en profite dans les chaumières :

0300-	38	SEC	
0301-	A5 06	LDA	\$06
0303-	E5 08	SBC	\$08
0305-	85 06	STA	\$06
0307-	A5 07	LDA	\$07
0309-	E5 09	SBC	\$09
030B-	85 07	STA	\$07
030D-	A9 01	LDA	£\$01
030F-	B0 02	BCS	\$0313
0311-	A9 00	LDA	£\$00
0313-	85 18	STA	\$18
0315-	60	RTS	

* Vous allez sans doute me prendre pour un néophyte, mais je ne peux m'empêcher de vous soumettre un problème que je n'arrive pas à résoudre. Par précaution, j'ai sauvegardé sur une disquette un long programme de plusieurs parties, sous des noms différents. Seulement maintenant, je ne trouve pas de solution pour supprimer les noms de ces programmes et les mettre les uns à la suite des autres pour n'en former plus qu'un seul. Pourriez-vous m'indiquer la solution ?

G. C. (24000 PÉRIGUEUX)

TM La page 30 du présent numéro (BOUTABOUT)

TM devrait répondre à votre attente. Notez que TOOL-KIT (APPLE) permet de mettre des programmes bout à bout (entre autres facilités fort intéressantes, telles que le numérotage).

* 1° Comment lister un programme en langage machine ?

2° Y-a-t'il un façon «simple» d'afficher dans un programme une page graphique sauvegardée avec un brave nom, A8192, L8192 sans utiliser Image Writer Tool Kit ?

Vincent BLEGER (67400 ILLKIRCH)

TM 1° Nous avons publié un programme de LISTAGE ET EDITION d'un programme en langage machine (p.14). Il devrait répondre à votre attente. Si vous désirez seulement l'examiner sur votre écran, tapez CALL-151 et RETURN. Ensuite, à côté de l'astérisque, tapez l'adresse hexadécimale du début de la routine, plus la lettre L. (*300L, par exemple). Votre Apple vous désassemblera aussitôt 20 instructions. Un autre L vous permettra de continuer, et ainsi de suite.

2° Vous n'avez aucune raison d'utiliser Image Writer Tool Kit pour afficher une page graphique. Il faut seulement la rappeler par BLOAD NOM, A\$2000 ou A\$4000 (page 1 ou page 2). Si vous avez d'abord programmé HGR... ou HGR2, elle apparaîtra sur votre écran. Plusieurs programmes de Tremplin Micro vous ont déjà indiqué comment réafficher une page graphique (Exemple, la ligne 990 de VISUBLOCS).

* Comment obtenir une impression de 132 caractères par ligne avec imprimante IMAGE WRITER ? Comment programmer les tabulations horizontales ?

TM 1° PRINT CHR\$(4) "PR£1" : PRINT CHR\$(27) "Q" : PRINT CHR\$(9) "132N"

2° Voici un exemple qui imprime 6 colonnes de nombres aléatoires (évidemment alignés à gauche... l'Apple ne connaissant pas le PRINT USING, mais cela se programme aussi !)

```
55 PRINT CHR$(27) "(010, 030, 050, 070, 090,
110.)" : PRINT CHR$(27) "u" : CHR$(9) :
60 FOR I = 1 TO 5 : FOR J = 1 TO 6
65 PRINT CHR$(27) "u" : CHR$(9) :
70 PRINT 9000 + INT (RND (1) * 10000) ; : NEXT :
PRINT "" : NEXT
```

Autre procédé, avec POKE 36, X

```
50 PRINT CHR$(9) "132N" : FOR I = 1 TO 5
60 FOR J = 1 TO 6 : POKE 36, J * 20 : PRINT 9000
+ INT (RND (1) * 10000) ; : NEXT : PRINT "" :
NEXT
```

* Voici une solution pour imprimer les HISTOGRAMMES (TREMPIN MICRO n°1) sur IMAGE WRITER.

Laurent DANIEL (68740 FESSENHEIM)

TM

Extraire la routine PICDMP de IMAGEWRITER TK

- Charger IMAGEWRITER TK
- Taper 6 —» sortir du programme
- Taper BSAVE PICDMP A\$9000, L\$5B0
- Incorporer PICDMP au basic de Histogramme (liste suivante).

```
60 PRINT : PRINT D$"BLOAD HISTO.LM"  
65 PRINT : PRINT D$"BLOAD PICDMP"  
70 PRINT : PRINT CHR$(13) + CHR$(4) "BLOAD  
FAIRE.F1,A$6100" : RETURN  
80 POKE 24659,234 : POKE 24660,234 : CALL  
24576  
90 RETURN
```

EDITION GRAPH

```
40 HOME  
42 INVERSE : PRINT "EDITION GRAPHIQUE" :  
NORMAL : PRINT : PRINT : PRINT  
43 PRINT "1 — SIMPLE DIMENSION, INVERSE" :  
PRINT : PRINT "2 — SIMPLE DIMENSION,  
NORMAL" : PRINT : PRINT "3 — DOUBLE  
DIMENSION, INVERSE" : PRINT : PRINT "4 —  
DOUBLE DIMENSION, NORMAL" : PRINT :  
PRINT  
44 PRINT "VOTRE CHOIX : " : GET CH$  
45 CH = VAL (CH$) : IF CH = 1 OR CH = 4 THEN  
GOTO 1240  
47 CH = CH - 1  
50 PRINT CHR$(13) : CHR$(4) : "PR$1" : PRINT  
CHR$(27) "N" : POKE 49239,0 : POKE 49234,0  
: POKE 49232,0 : POKE 49236,0 : POKE 6,1 :  
POKE 7, CH : CAL 36864 : PRINT : PRINT CHR$(  
4) : "PR$0"  
60 GOTO 870
```

* Je suis emballé par les deux premiers numéros et disquettes de TREMPIN MICRO qui m'ont fait découvrir, programmes et commentaires à l'appui, de nombreuses et instructives facilités de programmation que j'ai aussitôt essayées sur mon Apple IIc. J'ai eu quelques difficultés à exploiter l'utilitaire FAIRE.ECRAN (TM n°1) pour lequel je n'ai pas encore réussi à comprendre l'usage de la commande CTRL-W. Je souhaiterais, à ce propos, que le mode d'emploi de vos utilitaires soit plus explicite. Le programme ELECTRICITE (TM n°1) comporte une erreur : en saisie les années évoluent de 2. Pour un fonctionnement normal, il suffit de corriger la ligne 270 en supprimant ND=ND+1. J'ai aussi trouvé très instructif le programme MENU associé au STARTUP de la disquette ProDOS TM2. Je me suis empressé de l'adapter à mes disquettes ProDOS.

L'initiation au langage machine est bien faite, continuez, avec un peu plus d'explications, peut-être. Vous mentionnez des logiciels utilitaires (ex. TOOLKIT). Pouvez-vous en publier une liste avec références et fournisseurs ?

Guy CALLO (07500 GRANGES-LES-VALENCE)

TM

FAIRE.ECRAN.— La commande CTRL-W vous permet de transformer votre écran TEXT en écran GRAPHIQUE (avec utilisation de la fonte de caractères).

LOGICIELS UTILITAIRES.— Ceux d'Apple sont souvent les moins onéreux et présentent un avantage non négligeable : la compatibilité.

* Quelques réflexions en vrac :

— J'ai pu noter avec plaisir que les informations sur les programmes (utilisation et adaptabilité) augmentaient dans le n°2. Merci et j'espère que le mouvement va continuer car je ne comprends que 30 à 40% de ce qui se passe dans les programmes qui utilisent le langage machine ; certainement faute de temps ainsi que la volonté de tout vouloir exploiter tout de suite, vu la richesse de la revue.

— Pourquoi ne pas proposer un abonnement revue-disquette.

— D'où viennent ces «bips» au listage de certains programmes.

P.C. (37520 LE THEILLEMENT)

TM

Proposer un abonnement aux disquettes reviendrait à imposer l'achat de TOUTES les disquettes (y compris celles qui ne les intéressent pas) à nos lecteurs... et nous n'en sommes pas partisans. — BIP : Pour insérer un BIP dans une ligne de programme, il suffit de taper un CTRL-G (touches CTRL et G pressées simultanément). Exemple :

```
10 PRINT "TOTO CTRL-G" vous permettra  
d'entendre un BIP quand TOTO sera affiché...  
mais aussi en cours de listage.
```

Ce procédé peut aussi être utilisé dans les REMARQUES, pour attirer l'attention et faciliter un meilleur repérage, toujours lors d'un listage sur écran (sur imprimante, il n'y a aucune trace).

* Quelle ne fut pas ma surprise et ma joie, lorsque je découvris, dans un rayon de journaux, une revue sur la programmation et qui, de plus, était pour mon Apple IIc. Cette revue, c'était TREMPIN MICRO. Je l'achetai et me précipitai au clavier de mon Apple. (Je précise que je possède mon Apple IIc depuis deux ans avec un moniteur vert et un drive de disquettes 5 1/4 pouces. Je programme en basic et, depuis peu, je m'initie au langage machine ainsi qu'à l'assembleur ; c'est pas facile !!!). Je copiai le premier programme, (pour commencer : votre spécial bonjour), et lançai son exécution. Première surprise ! Lorsque j'exécutais l'option deux, (RUN), pour un programme de mon catalogue...

(suite page 60).

Vous avez écrit à TREMLIN MICRO



* (suite de la page 59) — la ligne «Catalogue utilitaire Guy Hachette» restait écrite. Ce problème fut réglé en corrigeant la ligne 15 du programme (intervertir PRINT CHR\$(4) : \$\$: CS (F) avec NEXT). Avec le MENU de TREMLIN MICRO n°1, il est impossible de se servir du programme figurant dans le catalogue à la lettre "O", comme programme de démarrage : en effet, après avoir choisi "O", on aboutit infailliblement aux options !

D'autre part, dans le programme de la page 8 ("l'envers ne vaut pas l'endroit") la ligne de présentation qui apparaît dans la fenêtre, n'arrête pas de défiler. Comment s'en sortir ?

C.C. (12000 RODEZ)

TM **MENU.**— Le programme figurant à la lettre "O" est parfaitement accessible, mais vous commettez une erreur de manipulation.

Quand le catalogue est affiché, il ne faut pas immédiatement taper la lettre-clé (encore que cela soit sans importance... jusqu'à N), mais presser sur la barre d'espacement. Ensuite, choisir l'option 2 (Run, etc.), puis la lettre-clé... et notamment O. Dans tout autre cas, O est prioritaire et renvoie effectivement aux options. Si cela vous gêne vraiment (vous êtes le seul), changez-la pour P, en modifiant le programme, tout simplement.

L'ENVERS NE VAUT PAS L'ENDROIT.— Puisque vous avez tapé vous-même le programme, la ligne 300 aurait dû vous renseigner. PEEK (49249) permet de savoir si la touche Pomme ouverte a été pressée.

MORALITÉ.— Notre programme n'était pas suffisamment documenté.

* Comment m'informer sur ProDOS ?

(nombreuses lettres)

TM Nous consacrerons plusieurs pages à ProDOS dans chaque numéro de T.M. Les Editions du P.S.I. ont, d'autre part, publié un livre qui est certes incomplet, mais permet aux utilisateurs de mieux comprendre les commandes de ce nouveau système d'exploitation. Titre : **ProDOS sur Apple IIe et IIc**, par Francis Verscheure.

T.M. vous signalera, au fur et à mesure de leur parution, tous les bons ouvrages sur ProDOS.

* Je suis lecteur d'Hebdomagiciel et j'ai noté que, dans le numéro 82, du 10 mai, le programme GDOS (de DAN DERY) utilise (listing 4) les mêmes données, à un octet près, que FAIRE.FI, de Tremplin Micro n°1. Le listing 3 (même programme) est quant à lui identique à la pre-

mière partie de Faire.LM (même numéro de Tremplin Micro). Je vous joins la page. Comment expliquez-vous cela ?

J.-L. M. (49000 ANGERS)

TM Nous n'avons rien à expliquer. Tremplin Micro a paru le 3 mars et Hebdomagiciel le 10 mai. D'autre part, la fonte FAIRE.FI utilise des caractères français accentués (â, é, à) que l'on ne trouve pas dans les disquettes commercialisées. Concluez vous-même ! La Rédaction de notre confrère (dont je suis personnellement un fidèle lecteur) a été trompée... mais comme cela nous arrivera aussi ! Quant à la première partie du programme, elle est américaine (BEAGLE BROSI)... ce que nous avons déjà signalé.

Nestor.

Opinion de Guy-Hachette : à partir du moment où je publie un programme personnel dans Tremplin Micro, j'estime qu'il ne m'appartient plus... puisque je le mets à la disposition des Lectrices et Lecteurs de la revue, lesquels peuvent le signer et le modifier si tel est leur bon plaisir. Enfin, quand un auteur fait un emprunt (important) à un autre, il est plus correct de citer la source. C'est cela, la vraie confraternité, mais personnellement, cité ou non, je suis flatté (oh ! le vilain défaut !) d'être copié... avec ou sans citation à l'appui. Si ce n'est pas du nombrilisme, alors qu'est-ce que c'est ? Quant à s'approprier — et sous sa signature — le travail des autres, je laisse le Lecteur seul juge !

Guy-Hachette.

* Je vous ai cherché (et même réclamé) à APPLE EXPO, mais Tremplin Micro y était, semble-t-il, inconnu (sauf de plusieurs exposants, qui m'ont confirmé que vous n'aviez pas de stand). Était-ce délibéré ?

R.L. (54300 LUNEVILLE)

TM Nous aimons bien Apple. La preuve : nous existons, mais Apple France visiblement ignore notre existence, ce qui ne nous empêchera pas de continuer — et objectivement — à travailler pour aider les utilisateurs d'ordinateurs personnels de cette marque. Il y faut d'ailleurs un certain courage puisque la Commission Paritaire des Publications et Agences de Presse refuse son agrément à une revue qui, comme la nôtre, ne s'intéresse qu'à une sorte de matériel... tout en n'ayant avec son constructeur aucun lien (la preuve : Apple Expo).

SAISIE LM ASSISTÉE (page 50)

```
En ProDOS, remplacer les lignes 940 à 960 par :  
940 GOSUB 140 : VTAB 21 : PRINT "BSAVE" TI$ ",A$ "AD$ ",  
L" L  
950 PRINT : VTAB 21 : PRINT D$ "BSAVE" TI$ ",A$ "AD$ ",L" L  
960 :
```

INDEX des articles déjà parus

Additions binaires (p. 19)..... <i>Jeu didactique pour apprendre à composer en binaire</i>	N° 2	Programme en EXEC (Fiche 1) (p. 34).....	N° 1
Affichage accéléré en langage machine (p. 63).....	N° 1	Programmer ? Bien sûr que cela en vaut la peine ! (p. 39).....	N° 1
Amortissement (p. 38).....	N° 1	Que va-t-il se passer ? (p. 54).....	N° 1
Avec ProCODE (p. 52).....	N° 1	Saisie avec tri immédiat (p. 49).....	N° 1
Boucles simples en langage machine (p. 50).....	N° 1	Soustraction en langage machine (p. 43).....	N° 2
Charabia à décrypter (p. 10)..... <i>Jeu dont la solution a paru dans le numéro 3</i>	N° 2	<i>La suite de l'initiation à l'assembleur</i>	
Copie d'écran avec carte 80 colonnes (p. 40).....	N° 1	Spécial Bonjour (p. 3).....	N° 1
Créez vos écrans en vous amusant (p. 17).....	N° 1	Tracer un cadre en langage machine (p. 64).....	N° 1
<i>Raccourcissez vos programmes en mémorisant, sur la disquette, les écrans de présentation</i>		Transfert d'un écran de 80 colonnes en mémoire auxiliaire (Fiche 2) (p. 41).....	N° 2
De bons livres, et en français (p. 61).....	N° 1	<i>Une fiche indispensable</i>	
De nouvelles perspectives pour vos histogrammes (p. 30).....	N° 1	Un accessoire utile : le magnétophone (p. 31).....	N° 2
<i>Un programme utilitaire indispensable</i>		Un certain jour de la semaine (p. 12).....	N° 1
Des octets en déplacement (p. 26).....	N° 1	<i>Quel jour êtes-vous né(e) ?</i>	
<i>Utilisez mieux la mémoire auxiliaire de votre carte 80 colonnes</i>		Un drive pour 1 500 F ? Banco ! (p. 46).....	N° 2
Deux trucs pour votre Apple (p. 40).....	N° 2	<i>Pourquoi payer votre matériel au prix fort ?</i>	
Disquettes à votre nom (p. 6).....	N° 1	Un moyen de savoir si un fichier existe (p. 36).....	N° 1
<i>Personnalisez vos disquettes en DOS 3.3</i>		Une petite révision d'anglais (p. 28).....	N° 1
Disséquer un nombre (p. 38).....	N° 2	<i>La première leçon (à vous d'écrire les suivantes !)</i>	
Double listage en 80 colonnes (p. 11).....	N° 2	Valeur ASCII après GET (p. 48).....	N° 1
<i>Basic et langage machine</i>		Vos chiffres valent bien un fromage ! (p. 49).....	N° 2
Ecrire en page 2 (HGR2) (p. 25).....	N° 1	<i>La suite logique des histogrammes, avec une certaine compatibilité</i>	
Effacement d'écran par la gauche (p. 62).....	N° 1	Vos menus à la carte (p. 5).....	N° 2
Evidemment qu'elle vous intéresse... la mémoire de votre Apple ! (p. 42).....	N° 1	<i>En Basic et langage machine... pour un affichage ultra-rapide</i>	
Fichiers programmes (Fiche 1) (p. 34).....	N° 1	Vous êtes au courant, mais cela vous coûte combien ? (p. 46).....	N° 1
Fichiers EXEC (Fiche 1) (p. 34).....	N° 1	<i>Petit utilitaire de gestion personnelle</i>	
Gribouille (p. 3).....	N° 2	Vous avez emprunté ? (p. 27).....	N° 1
<i>Présentation de l'original traitement de texte de Madeline Hodé</i>		Vous dites addition ? (p. 51).....	N° 1
Hexadécimal ou décimal, et vice versa (p. 44).....	N° 1	<i>Initiation au langage machine</i>	
HTAB en 80 colonnes (p. 48).....	N° 1		
Imprimantes et graphisme (p. 56).....	N° 2		
Imprimez des listages formatés de vos programmes (p. 27).....	N° 2		
<i>Éditez correctement vos meilleurs programmes</i>			
Inversion de HGR1 ou HGR2 (p. 37).....	N° 1		
Jeu (erreur) (p. 45).....	N° 1		
L'envers ne vaut pas l'endroit (p. 8).....	N° 1		
<i>Jeu simple, et améliorable</i>			
La fenêtre (Fiche 1) (p. 33).....	N° 1		
La page zéro de l'Apple IIe (Fiche 1) (p. 33).....	N° 1		
Le curseur (Fiche 1) (p. 33).....	N° 1		
Les caractères «souris» de l'Apple IIc (p. 34).....	N° 2		
Les fichiers aléatoires (p. 15).....	N° 2		
<i>Enregistrements de taille fixe (initiation)</i>			
Les nombres décimaux et votre micro (p. 16).....	N° 1		
List avec l'ampersand (p. 48).....	N° 1		
Listage et édition en langage machine (p. 14).....	N° 1		
<i>Utilitaire fort pratique</i>			
Listage-contrôle d'un fichier «T» (p. 44).....	N° 2		
Logic-grilles (p. 57).....	N° 1		
<i>Jeu visuel</i>			
Ma chère vieille pomme (p. 41).....	N° 1		
Onerr Goto (p. 38).....	N° 2		
Pair ou impair avec quelques octets (p. 35).....	N° 2		
<i>Une petite étude des nombres aléatoires... pour le plaisir</i>			
Plus loin avec l'assembleur (p. 39).....	N° 2		
Pokes utiles (p. 5).....	N° 1		
Pour oublier le jeu du pendu (p. 32).....	N° 2		
<i>Une solution : le jeu de l'homme libre !</i>			
Print using (p. 48).....	N° 1		

PROMO SPÉCIALE POUR ProCODE

Nous avons déjà écrit (TREMPIN MICRO n°1) tout le bien que nous pensons de ProCODE, logiciel en français, permettant une programmation facile en Assembleur (sans souris, mais avec un menu extra !).

Deux bonnes nouvelles :

- Nous vous proposons une disquette DEMO de ProCODE (qui fonctionne... mémorisation exceptée).
- Pendant les mois de juillet et d'août, ProCODE est vendu 840 F HT, mais avec, par-dessus le marché, le très bon bouquin de Nicole BRÉAUD-POULIQUEN et Daniel-Jean DAVID (Editions du P.S.I.). C'est pas beau, tout cela ?

Guy-Hachette pense à vos loisirs !



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