

tremplin micro

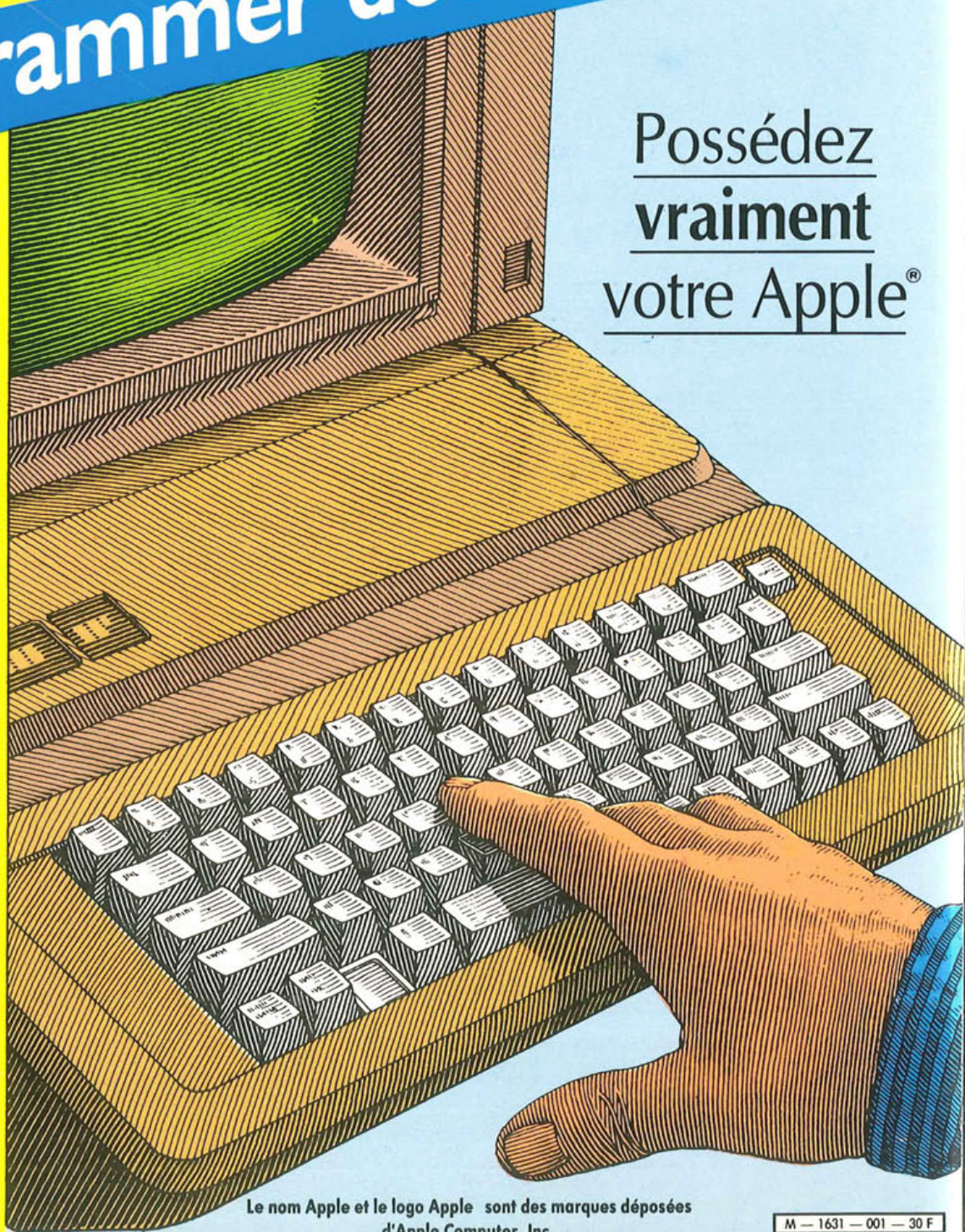
1

Programmer devient un jeu

FICHE
PRATIQUE
DÉTACHABLE

L'éternelle
jeunesse
du 6502

PREMIÈRE ANNÉE
BIMESTRIEL
MARS-AVRIL 1985
Le numéro : 30 F



Possédez
vraiment
votre Apple®

Le nom Apple et le logo Apple sont des marques déposées
d'Apple Computer, Inc.

M - 1631 - 001 - 30 F

Possédez-vous vraiment votre Apple ?



T d'abord, qu'en faites-vous ? Seulement du traitement de texte ? Ce n'est pas suffisant. Vous utilisez aussi un tableur ? Bien, mais encore ? Avez-vous déjà dialogué avec votre Apple autrement que par logiciel (du commerce) interposé ? Non : quel dommage... pour vous, en tout cas !

Découvrez donc, avec nous, en programmant vous-même de petites applications, la face cachée de votre ordinateur personnel. Possédez-le deux fois : financièrement, bien sûr (à moins que vous ne l'ayez volé !) et... moralement, si je puis m'exprimer ainsi !

Il vous rendait déjà la vie plus facile, mais il ne demande qu'à vous aider davantage. Son microprocesseur ne se fatiguera pas, même si vous l'entraînez parfois dans des boucles sans fin.

Peut-être attendiez-vous tout de lui ? Désormais, vous saurez que certains de ses services passeront d'abord par votre imagination... et votre vivacité d'esprit.

Programmer est un jeu. L'une des récréations les plus captivantes qui soient. L'ordinateur personnel est devenu — il faut s'en féliciter — un partenaire infatigable, presque infailible, et quasi irremplaçable quand on le connaît bien.

Pour vous, ce partenaire se nomme Apple, et ses promoteurs, deux jeunes américains géniaux, en ont fait un fruit à consommer toute l'année.

GUY-HACHETTE.

Le nom Apple et le logo Apple sont des marques déposées d'Apple Computer, Inc.

BIMESTRIEL

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

EDITIONS JIBENA

Administration : Editions JIBENA, 4 rue de la Cour-des-Noues 75020 PARIS

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

Jouer sur un Apple

- L'envers ne vaut pas l'endroit 8
- Logic-grilles 57

Apprendre en s'amusant

- Une petite révision d'anglais 28
- Que va-t-il se passer ? 54

Vie pratique

- Un certain jour de la semaine 12
- Combien devez-vous payer ? 27

De nouvelles perspectives pour vos histogrammes

— PAGE 30 —

- Amortissement 38
- Consommation d'électricité 46

Pour commencer simplement votre initiation au langage machine **PAGE 50**

PROGRAMMES UTILITAIRES

- Spécial «Bonjour» 3
- Disquettes à votre nom 7
- Edition en langage machine 14
- Créez vos écrans 17
- Ecrire en page 2 25
- Inversion de HGR1 ou HGR2 37
- Copie d'écran en 80 colonnes 40
- Affichage de la mémoire 42
- Conversion hexa-décimal 44
- Saisie avec tri immédiat 49
- Effacement de l'écran 62
- Un cadre pour vos menus 63

LISEZ AUSSI :

- Programmer ? bien sûr que cela en vaut la peine ! 39
- Lettre ouverte à une vieille pomme .. 41
- ProCODE, éditeur assembleur pleine page 52
- De bons livres 61

TRUCS ET ASTUCES

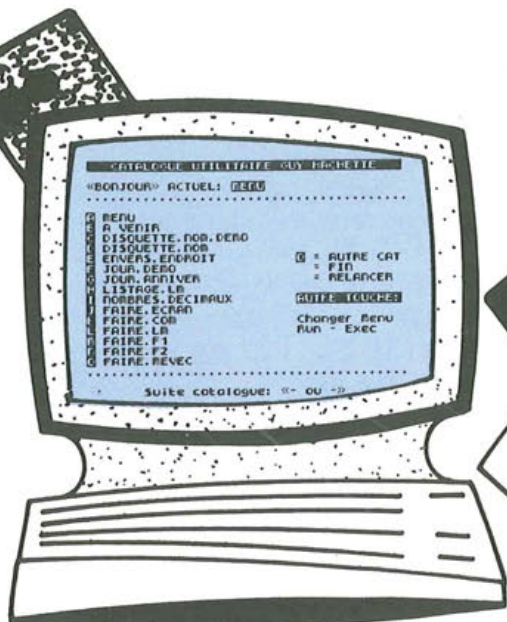
- Pokes utiles 5
- Nombres de secteurs libres 5
- Les nombres décimaux 16
- Sauvez 64 précieux octets 24
- Pokez le langage machine 25
- Octets en déplacement 26
- Transfert en mémoire auxiliaire 27
- Ce fichier existe-t-il ? 36
- Print using 48

Le numéro 2 de **TREPLIN MICRO**

paraîtra le 3 mai

Pour commencer : votre spécial «Bonjour» !

Il permet, entre autres facilités que vous découvrirez en le tapant et en l'utilisant, de modifier à volonté le nom du programme de démarrage.



```

10 CALL 62361: PRINT CHR$(21): CALL 64600:
   CALL 62071: PRINT " CATALOGUE UTILITAIRE
   GUY-HACHETTE ": CALL 62067
15 POKE 0,1: GOTO 575
20 IF K = 21 THEN P = P + 1: IF P > L% THEN P
   = 1
25 IF K = 8 THEN P = P - 1: IF P < 1 THEN P =
   L%
30 CALL 831
35 A = (P - 1) * B: FOR X = 1 TO B: J = A + X:
   IF J > C THEN 45
40 INVERSE : PRINT CHR$(64 + X);: NORMAL :
   PRINT " C$(J): NEXT
45 H = 63 + X: VTAB 5: CALL - 868: RETURN
50 :
55 REM *****
60 REM *** Affichage centré ***
65 REM *****
70 :
75 HTAB FN T(C): IF RIGHT$(T$,8) = "RELANC
   ER" THEN PRINT LEFT$(T$, LEN(T$) - 8);:
   FLASH : PRINT "RELANCER": NORMAL : RETURN
80 PRINT T$: RETURN
85 :
90 REM *****
95 REM *** Petites routines ***
100 REM *****
105 :
110 POKE 826,2: GOTO 120
115 POKE 826,1
120 CALL 805: POKE 72,0: RETURN
125 POKE 818,1: POKE 819,9: RETURN
130 WAIT 6,128: POKE G1,0: K = PEEK (G): IF K
   = 79 AND C% < > 1 THEN 475

```

```

135 RETURN
140 POKE 34,2: POKE 35,20: HOME : TEXT
145 VTAB 22: CALL - 958: RETURN
150 GOSUB 130
155 IF K < > 13 THEN 620
160 IF RL THEN RETURN
165 :
170 REM *****
175 REM *** Lecture du catalogue ***
180 REM *****
185 :
190 CALL 64600: T$ = "Lecture du catalogue" + U
   $: VTAB 23: GOSUB 75
195 FOR X = 15 TO 1 STEP - 1: POKE 819,X: GOS
   UB 120: D = - 34
200 FOR Y = M TO N STEP 35: D = D + 35
205 Z = PEEK (Y + 33): IF Z = 0 THEN 245
210 C = C + 1: IF PEEK (Y) = 255 THEN C = C -
   1: GOTO 240
215 Z = PEEK (Y + 2): IF Z > 16 THEN Z = Z - 1
   28
220 IF Z > 4 THEN C = C - 1: GOTO 240
225 C(C) = 6: IF Z = 0 THEN C(C) = 20: GOTO 235
230 IF Z = 4 THEN C(C) = 52
235 C$(C) = MID$(I$,D,30)
240 NEXT : NEXT
245 L% = C / B: IF C > L% THEN L% = L% + 1
250 IF C / B = INT (C / B) THEN L% = L% - 1
255 GOSUB 125: GOSUB 120: C$ = MID$(I$,104,30
   ): GOSUB 145
260 VTAB 3: HTAB 1: CALL - 868: PRINT "<BONJ
   UR> ACTUEL: ";: FOR Q = LEN (C$) TO 1 STEP
   - 1: IF ASC (MID$(C$,Q,1)) < > 160 TH
   EN C$ = LEFT$(C$,Q): CALL 62071: PRINT C$

```

```

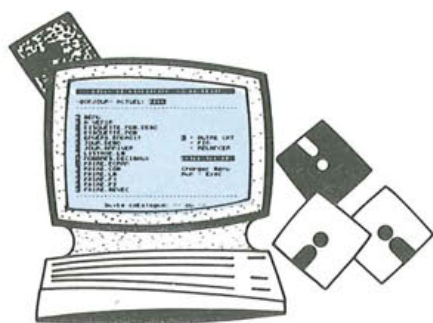
: CALL 62067: GOTO 275
265 NEXT Q
270 FOR Q = 30 TO 1 STEP - 1: IF MID$(C$,Q,
1) < > " " THEN C$ = LEFT$(C$,Q + 1): IN
VERSE: PRINT C$: NORMAL: GOTO 280
275 IF R% = 1 THEN RETURN
280 VTAB 4: PRINT P$: GOSUB 30
285 GOSUB 145: IF LZ = 1 THEN 335
290 T$ = "Suite catalogue: <- ou ->": GOSUB 145
: VTAB 23: GOSUB 75: GOSUB 130: IF K = 13 T
HEN 335
295 IF K = 63 THEN 475
300 IF K < > 8 AND K < > 21 THEN 335
305 GOSUB 20: GOTO 285
310 :
315 REM *****
320 REM *** Changer le Bonjour ***
325 REM *****
330 :
335 T$ = "<1> CHANGER " + CHR$(34) + "BONJOUR
" + CHR$(34) + " <2> RUN/EXEC/BRUN": VTAB
23: GOSUB 75: GOSUB 130
340 IF K = 63 THEN 290
345 IF K = 49 THEN 360
350 IF K < > 50 THEN 335
355 T$ = "Quel est votre choix ?": C% = 1: GOSU
B 145: VTAB 23: GOSUB 75: GOTO 365
360 T$ = "Indiquez le nouveau <BONJOUR>": GOSUB
145: VTAB 23: GOSUB 75
365 GOSUB 130: IF LZ > 1 AND (K = 21 OR K = 8)
THEN GOSUB 20: GOTO 365
370 IF K = 63 THEN 290
375 IF K < 65 OR K > 8 THEN 365
380 F = B * (P - 1) + K - 64: V = K - 64 + 5: IF
C$(F) = C$ THEN 365
385 VTAB V: HTAB 3: CALL 62071: PRINT C$(F): C
ALL 62067: IF C% = 1 THEN 530
390 GOSUB 145: T$ = "Installer nouveau Bonjour
(O/N)": VTAB 23: GOSUB 75
395 C% = 1: GOSUB 130: IF K = 79 THEN 415
400 IF K = 63 THEN 290
405 IF K < > 78 THEN 395
410 VTAB V: HTAB 3: PRINT C$(F): GOTO 285
415 GOSUB 145: T$ = "INSTALLATION" + U$: VTAB 2
3: GOSUB 75
420 GOSUB 125: GOSUB 115
425 X0% = 16500: X1% = X0% + 1: X2% = X0% + 30: X3
% = 16450
430 C$ = C$(F): FOR X = X1% TO X2%
435 POKE X, ASC (MID$(C$,X - X0%,1)): NEXT :
GOSUB 110
440 POKE 818,0: POKE 819,13: GOSUB 115: POKE X
3%,C$(F): GOSUB 110

```

```

445 VTAB V: HTAB 3: PRINT C$:R% = 1: GOSUB 260
:R% = 0: GOTO 285
450 :
455 REM *****
460 REM *** Autres options ***
465 REM *****
470 :
475 GOSUB 140: VTAB 23: PRINT "<C> CATALOGUE <
R> RELANCER <Q> QUITTER"
480 GOSUB 130: IF K < > 67 AND K < > 81 AND
K < > 82 THEN 480
485 GOSUB 140: IF K = 67 THEN GOTO 575
490 IF K = 81 THEN T$ = "MERCI": VTAB 23: GOSU
B 75: VTAB 19: END
495 R$ = "RELANCER": RL = 1: D$ = "1": VTAB 14: G
OSUB 600: GOSUB 140
500 T$ = "SYSTEME RELANCE": GOSUB 75: VTAB 3: P
RINT: CALL - 1370
505 :
510 REM *****
515 REM *** Run - Exec ou Brun ***
520 REM *****
525 :
530 GOSUB 145: VTAB 23: C% = 0: IF C(F) = 6 THE
N S$ = "RUN": GOTO 545
535 IF C(F) = 20 THEN S$ = "EXEC": GOTO 545
540 IF C(F) = 52 THEN S$ = "BRUN"
545 T$ = S$ + U$: GOSUB 75: PRINT CHR$(4); S$:
C$(F): TEXT: HOME: END
550 :
555 REM *****
560 REM *** Affichage préliminaire ***
565 REM *****
570 :
575 CLEAR: I$ = "": GOSUB 650: CALL 768: D$ =
STR$(PEEK(816))
580 G = 49152: M = 16395: N = 16605: G1 = 49168: D
IM C$(105),C(105)
585 DEF FN T(C) = (41 - LEN(T$)) / 2: P = 1:
B = 15
590 POKE 34,2: FOR X = 1 TO 10: P$ = P$ + "....
": NEXT
595 U$ = " - PATIENCE SVP!": O$ = " <O> POUR OPT
IONS": R$ = "DEBUT"
600 T$ = "INSEREZ LA DISQUETTE DANS LE DRIVE "
+ D$: VTAB 10: GOSUB 75
605 T$ = "& PRESSEZ <RETURN> POUR " + R$: VTAB
12: GOSUB 75: IF K = 82 THEN 615
610 IF PEEK(0) = 1 THEN GOSUB 690
615 VTAB 21: PRINT P$: T$ = O$ + " " + R$: VTAB
23: GOSUB 75: POKE 35,20
620 GOTO 150
625 :

```



Et aussi des Pokes utiles !

Ils sont connus, mais il est bon de les avoir sous la main, en cas de besoin.

POKE 44452,N + 1 : POKE 44605, N

N est le nombre de lignes du catalogue que vous désirez afficher à la fois. La valeur normale de N est 21. Si vous lui donnez par exemple une valeur de 10, vous pourrez consulter les 10 premières lignes de votre catalogue, puis les 10 suivantes, etc.

POKE 44460,88 : POKE 44461,252

Inutile de taper un HOME pour vider votre écran : le DOS va s'en charger dans la joie. Les valeurs habituelles sont 47 et 174.

POKE 44505,234 : POKE 44506,234

Même si vous avez effacé un fichier par DELETE, il va réapparaître comme par enchantement dans votre catalogue... s'il n'a pas été écrasé, depuis, par un autre. Valeurs normales : 48 et 74.

NOMBRE DE SECTEURS LIBRES DE VOTRE DISQUETTE

Pour connaître le nombre de secteurs libres d'une disquette (avec DOS 3.3 non modifié), tapez ce petit programme puis sauvez-le par un BSAVE CATFREE, A\$300, L\$84.

Pour l'utiliser, deux solutions :

BRUN CATFREE, et & (return) ou

BLOAD CATFREE : CALL 768, puis & (return).

```
0300 — A9 4C 8D F5 03 A9 27 8D
0308 — F6 03 A9 03 8D F7 03 60
0310 — 01 60 01 00 11 00 91 03
0318 — 00 95 00 00 01 00 00 60
0320 — 01 00 01 EF D8 00 00 A9
0328 — 03 A0 10 20 D9 03 A2 00
0330 — 8E 25 03 8E 26 03 A0 38
0338 — B9 00 95 A2 08 0A 90 08
0340 — EE 25 03 D0 03 EE 26 03
0348 — CA D0 F2 C8 C0 C4 90 E8
0350 — AE 25 03 AD 26 03 C0 03
0358 — B0 07 C0 01 D0 03 8A A2
0360 — 00 20 24 ED A2 00 BD 77
0368 — 03 20 ED FD E8 E0 0D 90
0370 — F5 A9 8D 20 2D FD 60 A0
0378 — D3 C5 C3 D4 AE A0 0C 09
0380 — 02 12 05 13
```

CATALOGUE UTILITAIRE GUY-HACHETTE

«**BONJOUR**» ACTUEL: **MENU**

```

MENU
A VENIR
DISQUETTE. NOM. DEMO
DISQUETTE. NOM
ENVERS. ENDROIT           [O] = AUTRE CAT
JOUR. DEMO                 = FIN
JOUR. ANNIVER              = RELANCER
LISTAGE. LM
NOMBRES. DECIMAUX
FAIRE. ECRAN
FAIRE. COM
FAIRE. LM
FAIRE. F1
FAIRE. F2
FAIRE. MEVEC

```

AUTRE TOUCHE:

Changer Menu
Run - Exec

Suite catalogue: <- ou ->

```

630 REM *****
635 REM *** Routine en langage.mac ***
640 REM *****
645 :
650 FOR A = 768 TO 850: READ B: POKE A,B: NEXT
    : RETURN :B = 0
655 DATA 160,2,165,105,133,6,165,106,133,7,169
    ,240,145,6,200,169,14,145,6,200,169,64,145,
    6,173,233,183,141,47,3,173,234,183,141,48,3
    ,96,169,3,160,46,32,217,3,96
660 DATA 0,1,96,1,0,17,15,251,183,0,64,0,0,1,0
    ,254,96,1,169,5,133,34,169,20,133,35,32,88,
    252,169,2,133,34,169,24,133,35,96
665 :
670 REM *****
675 REM *** Quel Apple travaille ? ***
680 REM *****
685 :
690 VTAB 16:A = PEEK (64435):O = PEEK (64448
    ): IF A = 6 AND (O > 223 AND O < 240) THEN
    T$ = "Vous disposez d'un Apple IIe": GOSUB
    75
695 IF A = 56 OR A = 234 THEN T$ = "Vous dispo
    sez d'un Apple II ou II+": GOSUB 75
700 IF A = 6 AND O = 0 THEN T$ = "Vous dispose
    z d'un Apple IIc": GOSUB 75
705 POKE 0,0: RETURN

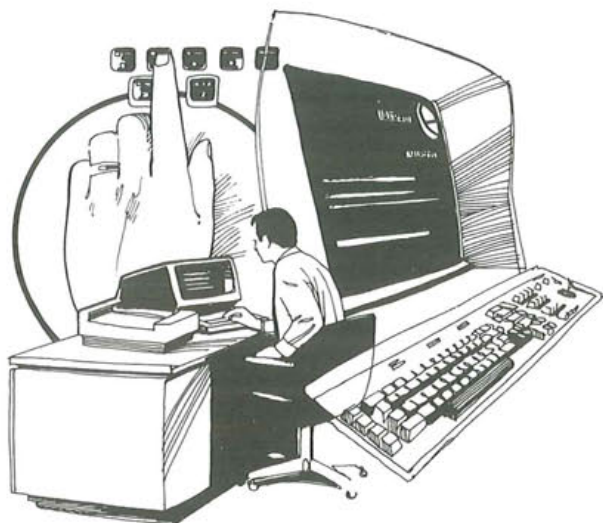
```

Les flèches ← et → permettent de voir la fin du catalogue, ou d'en revoir le début. Il suffit de presser une autre touche pour accéder aux fonctions RUN — EXEC — BRUN.

Le ? renvoie généralement au catalogue en cours, bien qu'il n'en soit pas fait état (je l'ai rajouté).

Eliminez les REM : vous accélérerez le programme !

Des disquettes à votre nom ?



Le **DISK VOLUME** de vos disquettes ne vous inspire pas et vous aimeriez remplacer cette mention par votre nom, ou par un raccourci du titre d'un programme ?

Rien de plus facile avec le court utilitaire de la page 7 (on peut l'abrégier en éliminant le blablabla et l'affichage).

Une restriction : il ne fonctionne complètement (c'est-à-dire avec écriture sur disquette) que sous DOS 3.3.

Mais vous pouvez vous offrir des disquettes personnalisées sans taper de programme, et en respectant le détail des opérations suivantes, en mode direct :

Attention ! pour modifier votre **DISK VOLUME** de cette manière, vous devrez ensuite initialiser une disquette vierge (pas celle sur laquelle vous travaillez) puis recopier vos programmes dessus.

Entrez dans le MONITEUR par un :

\$CALL-151... suivi de RETURN et *B3AFL... puis RETURN

```
B3AF-  A0 C5      LDY  £$C5
B3B1-  CD D5 CC   CMP  $CCD5
B3B4-  CF         ???
B3B5-  D6 A0     DEC  $A0,X
B3B7-  CB         ???
B3B8-  D3         ???
B3B9-  C9 C4     CMP  £$C4
```

Seules ces lignes nous intéressent. En effet :

```
A0 C5 CD D5 CC CF D6 A0 CB D3 C9 C4
E M U L O V   K S I D
```

Si vous désirez remplacer **DISK VOLUME** par **G. HACHETTE**, vous tapez seulement en mode direct :

```
$A$="G. HACHETTE"... suivi d'un RETURN
$FORI=46010 TO 46000 STEP -1: M=M+1: POK
EI,ASC(MID$(A$,M,1))+128:NEXT
```

Après cette opération, si vous faites un **CALL-151**, puis **B3AFL**, vous obtiendrez :

```
B3AF-  A0 C5      LDY  £$C5
B3B1-  D4         ???
B3B2-  D4         ???
B3B3-  C5 C8     CMP  $C8
B3B5-  C3         ???
B3B6-  C1 C8     CMP  ($C8,X)
B3B8-  A0 AE     LDY  £$AE
B3BA-  C7         ???
```

```
A0 C5 D4 D4 C5 C8 C3 C1 C8 A0 AE C7
E T T E H C A H   . G
```

VOICI LES VALEURS HEXA DES CARACTERES :

A	B	C	D	E	F	G	H	I	J
C1	C2	C3	C4	C5	C6	C7	C8	C9	CA
K	L	M	N	O	P	Q	R	S	T
CB	CC	CD	CE	CF	D0	D1	D2	D3	D4
U	V	W	X	Y	Z		-		■
D5	D6	D7	D8	D9	DA	AE	AD	A0	■

Implantation, par **POKES**,
du sous-programme en
langage-machine (DATA).

On charge la piste 2, sec-
teur 2 de la disquette dans
une mémoire tampon : ici,
c'est \$2000 (8192 en
DÉCIMAL).

DISK VOLUME (qui nous
intéresse va de \$20AF à
\$20BA, 8367 à 8378)

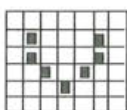
On utilise 8 adresses de
la page zéro pour trans-
mettre les données.

On modifie aussi **DISK**
VOLUME réel (\$B3AF.
\$B3BA) de manière à pou-
voir éventuellement initia-
liser une disquette avec ce
nouveau DOS.

Un détail : on écrit à l'en-
vers, de \$BA à \$AF (46010
à 45999).

```
10 TEXT : NORMAL : PRINT CHR$(21): HOME
15 FOR I = 768 TO 822: READ R: POKE I,R: NEXT
20 FOR I = 1 TO 10:T$ = T$ + "____": NEXT
25 INVERSE : PRINT "    INSCRIRE SON NOM SUR UNE DISQUE
   TTE    ": NORMAL
30 VTAB 5: PRINT "Pas plus de 11 caractères!"
35 PRINT : VTAB 20: PRINT T$: CALL - 958
40 VTAB 8: HTAB 1: PRINT "NOM .....           "; HTA
   B 5: INPUT " ";N$: IF N$ = "" THEN 135
45 IF LEN (N$) > 11 THEN 40
50 N$ = LEFT$(N$ + "           ",12)
55 VTAB 8: HTAB 5: INVERSE : PRINT N$: NORMAL
60 VTAB 22: HTAB 1: PRINT "<1> BON A ENREGISTRER <2> A
   REFAIRE => ";: GET R$: IF R$ = "2" THEN 35
65 IF R$ < > "1" THEN 60
70 :
75 REM *****
80 REM ** TRANSMISSION DES DONNEES PAR **
85 REM ** DES POKE AUX ADRESSES $6-$0B **
90 REM *****
95 :
100 POKE 6,2: POKE 7,2: POKE 8,1: POKE 9,96: POKE 10,0:
   POKE 11,32
105 REM $6:PISTE 02 - $7:SECTEUR 02 - $8:LECTEUR 1 - $9
   :SLOT 6 (*16)
110 REM $0A & $0B:ADRESSE DECIMALE DE LA MEMOIRE TAMPON
   ($2000, soit $0020 à l'envers...$20=32)
115 POKE 227,1: CALL 768: REM $E3 LECTURE = 01
120 X = 0: FOR I = 8378 TO 8367 STEP - 1:X = X + 1: POK
   E I, ASC ( MID$(N$,X,1)) + 128: NEXT
125 X = 0: FOR I = 46010 TO 45999 STEP - 1:X = X + 1: P
   OKE I, ASC ( MID$(N$,X,1)) + 128: NEXT
130 POKE 227,2: CALL 768: REM $E3 ECRITURE = 02
135 VTAB 22: HTAB 1: PRINT "<1> ECRIRE VOTRE NOM <2> T
   ERMINE => ";: GET R$: IF R$ = "1" THEN 35
140 IF N$ = "" THEN 160
145 PRINT : VTAB 22: CALL - 958
150 PRINT "    VOUS POUVEZ INITIALISER TOUTES VOS    DI
   SQUETTES (<";: INVERSE : PRINT "VIERGES";: NORMAL : P
   RINT ") AVEC VOTRE NOM!"
155 POKE - 16368,0: WAIT - 16384,128
160 TEXT : HOME : END
165 :
170 REM *****
175 REM **    INITIALISE LA TABLE IOB    **
180 REM **    DU RWTS ET APPELLE CELUI-CI    **
185 REM *****
190 :
195 DATA 169,0,141,235,183,165,9,141,233,183,165,8,141,
   234,183,165,6,141,236,183,165,7,141,237,183,165,227
200 DATA 141,244,183,165,10,141,240,183,165,11,141,241,
   183,169,183,160,232,32,217,3,144,5,173,245,183,133,1
   2,96
```


L'envers ne vaut pas vraiment l'endroit !



OICI un exemple de programme capable d'intéresser jeunes et moins jeunes.

Le principe est simple : l'ordinateur retourne, en escamotant une lettre (ou un espace) le texte que lui a tapé le joueur.

Et les secondes s'écoulent tandis que l'on cherche la lettre litigieuse... Enfantin ? Pas sûr... quand celui qui pose la colle (on peut être deux) a tapé du charabia !...

Naturellement, cette version peut être améliorée, par exemple en mémorisant les parties... ou encore en utilisant le mode graphique.

Ce programme peut être facilement adapté pour tourner sur n'importe quelle machine, y compris les plus petites... (si l'on élimine les REM, sauf celle de la ligne 20, il n'occupe qu'une place réduite en mémoire et n'utilise que quelques variables).

ECRAN DE PRESENTATION

LE JOUEUR TAPE UN GROUPE DE MOTS (OU DE LETTRES) DANS UNE FENETRE (24 LETTRES AU MAXIMUM).

L'APPLE RESTITUE AUSSITOT LA LIGNE, MAIS A L'ENVERS... ET EN ELIMINANT UNE LETTRE QU'IL FAUT DECOUVRIR DANS UN MINIMUM DE TEMPS.

ON NE TAPE QUE DES LETTRES CAPITALES ET PAS DE PONCTUATION. UTILISER LES FLECHES <- ET -> POUR LES ESPACES ET LES CORRECTIONS, LA TOUCHE <ESC> POUR TERMINER.

QUAND LA PHRASE EST TROP COURTE, ON EST PENALISE ET ON EST EGALEMENT PENALISE SI LE TEMPS MOYEN DE REFLEXION DEPASSE HUIT SECONDES PAR COUP. S'IL EST PLUS COURT, IL Y A BONIFICATION...

PRESSER UNE TOUCHE

← Cette page s'affiche entre chaque partie. Il suffit d'ajouter une ligne puis d'en modifier une autre pour qu'elle ne reparaisse pas :

```
40 IF PEEK (0) <> 0 then  
    GOSUB 960  
45 POKE 0, 0 : HOME :  
    GOTO 220
```

PROGRAMME

```
10 CLEAR : TEXT : PRINT CHR$ (21) : HOME  
20 REM "TAPEZ ICI UN TEXTE DE 24 SIGNES MAXIMUM - TOUCHE  
    POMME OUVERTE POUR COMMENCER **** "  
30 FOR I = 1 TO 10 : T$ = T$ + "____" : NEXT : T1$ = LFN$  
    T$ (T$, 30)  
40 GOSUB 960 : HOME : GOTO 220  
50 :
```

← Attention ! ne pas modifier les deux premières lignes (10 et 20) car l'adresse de la seconde est utilisée par PEEK un peu plus loin.

Utilisé pendant la saisie, quand la ligne est complète (depuis 490).

Blocage et nettoyage de la fenêtre de saisie.

On écrit par **POKE** sur la ligne 6 de l'écran (adresse : 1794)

P = 278 représente la différence entre **DE** (écran, ligne 6, position 2) et le texte lu en mémoire, dans le programme lui-même (ligne 20, REM "TAPEZ ICI, ... etc.).

ESCAPE permet de terminer la saisie à n'importe quel moment. Il serait prudent d'interdire la sortie si $H < 2$

```
430 IF A = 27 AND H > 1 THEN
    H = H - 1 : GOTO 560
```

```
60 REM *****
70 REM ** QUELQUES SOUS-PROGRAMMES **
80 REM *****
90 :
100 TEXT : GOSUB 120: INVERSE : PRINT "ESCAPE SVP";: N
    ORMAL : PRINT "": GET M$:A = ASC (M$): IF A = 27
    THEN H = H - 1: GOSUB 120: GOTO 560
110 PRINT : GOTO 100
120 VTAB 21: HTAB 1: CALL - 958: RETURN
130 POKE 32,2: POKE 33,26: POKE 34,6: POKE 35,10: HOME
    : POKE 35,7: RETURN
140 FOR S = 1 TO 4:X = PEEK (49200): NEXT : RETURN
150 VTAB 6: HTAB 33: INVERSE : GOSUB 140: FOR I = 1 TO
    25: POKE - 16368,0: WAIT - 16384,127: POKE - 16
    368,0:X = PEEK ( - 16384): IF X > 64 AND X < 91 TH
    EN 690
160 NEXT I:T = T + 1: PRINT T: NORMAL : GOTO 150
170 :
180 REM *****
190 REM ** LIGNE ANNONCANT LE DEBUT **
200 REM *****
210 :
220 VTAB 4: PRINT T1$: VTAB 12: PRINT T1$
230 INVERSE : FOR I = 5 TO 12: HTAB 1: VTAB I: PRINT "
    "": HTAB 30: PRINT " ": NEXT : NORMAL
240 GOSUB 130
250 DE = 1794:P = 278
260 FOR I = DE TO 1819:X = PEEK (49200): POKE I, PEEK
    (I + P) + 128
270 NEXT
280 IF P < 337 THEN P = P + 1
290 GOSUB 140
300 ARRET = PEEK (49249): IF AR > 127 THEN 380
310 IF P = 337 THEN GOSUB 140: GOTO 250
320 GOTO 260
330 :
340 REM *****
350 REM ** SAISIE DE LA LIGNE DE TEXTE **
360 REM *****
370 :
380 TEXT : VTAB 10: HTAB 33: PRINT "ESCAPE": VTAB 11:
    HTAB 33: PRINT "POUR": VTAB 12: HTAB 33: PRINT "FIN
    IR"
390 GOSUB 130
400 HOME :H = 1:PL = 0
410 PRINT "..... ";
420 VTAB 7: HTAB H: PRINT "": GET M$:A = ASC (M$)
430 IF A = 27 THEN H = H - 1: GOTO 560
440 IF A = 8 AND H > 1 THEN H = H - 1: POKE H + 1793,1
    74: GOTO 420
450 IF A = 32 OR A = 21 THEN A = 96: GOTO 470
460 IF A < 65 OR A > 90 THEN 420
470 GOSUB 140
480 POKE H + 1793,A - 64
```

```

490 H = H + 1: IF H = 25 THEN 100
500 GOTO 420
510 :
520 REM *****
530 REM ** LETTRE ELIMINEE ET AFFICHAGE **
540 REM *****
550 :
560 SU = 1194 + INT ( RND (1) * H)
570 MH = H
580 H = H + 1793: FOR I = 1194 TO 1193 + MH
590 IF I = SU THEN PL = 1:EL = PEEK (H) + 64: GOTO 610
600 POKE I - PL, PEEK (H) + 128
610 H = H - 1
620 NEXT I
630 TEXT : VTAB 14: HTAB 1: PRINT "QUELLE LETTRE A ETE
    ESCAMOTEE ? ";; GOTO 150
640 :
650 REM *****
660 REM ** QUELLE LETTRE A ETE ESCAMOTEE **
670 REM *****
680 :
690 NORMAL : VTAB 14: HTAB 1: CALL - 868
700 RE$ = "GAGNE!":GA = GA + 1: IF X < > EL THEN GA =
    GA - 1:PE = PE + 1:RE$ = "PERDU!": GOTO 720
710 GOSUB 1120
720 VTAB 14: HTAB 8: PRINT "C'ETAIT UN" ;; INVERSE : HT
    AB 1: PRINT RE$;; HTAB 19: PRINT CHR$ (EL): NORMAL
730 :
740 REM *****ACTUALISATION DU SCORE **
750 :
760 TJ = GA + PE:LT = LT + MH:SC = GA * 10 + TJ * 8 - T
    - TJ * 24 + LT
770 VTAB 1: HTAB 1: CALL - 868: PRINT "VOUS AVEZ TAPE
    ";; INVERSE : PRINT LT;; NORMAL : PRINT " LETTRES
    - SCORE:";; INVERSE : PRINT SC: NORMAL
780 VTAB 3: HTAB 1: PRINT "-TEMPS: -----PERDU: --
    ----GAGNE: -"
790 VTAB 3: INVERSE : HTAB 8: PRINT T;; HTAB 23: PRINT
    PE;; HTAB 38: PRINT GA: NORMAL
800 :
810 REM *****
820 REM ** AUTRE TOUR/AUTRE PARTIE/FIN **
830 REM *****
840 :
850 VTAB 19: HTAB 1: PRINT T$
860 VTAB 21: HTAB 1: PRINT "<1> SUITE <2> NOUVELLE PAR
    TIE <3> FIN ";; GET R$
870 GOSUB 120
880 ON VAL (R$) GOTO 390,10,900
890 GOTO 860
900 TEXT : HOME : END

```

← **GOTO 100** pour indiquer au joueur qu'il doit presser la touche **ESCAPE**.

← Pourquoi **1194** ? parce que l'on va écrire (à l'envers) sur la ligne 9 de l'écran, dont l'adresse décimale est 1192 (mais on saute deux espaces, ce qui donne 1194).

← **GOSUB 1120** envoie à l'air du vainqueur... en fin de liste.

TJ = Total des coups Joués.
GA = coups **GA**gnés.
PE = coups **PER**dus.
LT = Lettres utilisées (Total)
MH = Longueur de la ligne en cours.
SC = Score Cumulé.

MARQUE

10 points par coup **GAGNÉ** + ou - x points, en fonction du temps passé **moyen**.

- x points si le nombre **moyen** de lettres est inférieur à 24.

Ce texte emploie les capitales afin d'être utilisable sur n'importe quel APPLE... mais il est modifiable.

Naturellement, il convient de respecter minutieusement les espaces.

Il fallait bien que le facteur chance intervienne. Quand l'ordinateur escamote un espace, le joueur ne perd finalement que le temps de réflexion... s'il dépasse 8 secondes. Il suffit d'appuyer immédiatement sur n'importe quelle lettre pour avoir une bonification de temps !

Avec un enfant, l'air du vainqueur joue un rôle stimulant indéniable !

```

910 :
920 REM *****
930 REM **   TEXTE DE PRESENTATION   **
940 REM *****
950 :
960 T$(1) = "LE JOUEUR TAPE UN GROUPE DE MOTS (OU DELET
      TRES) DANS UNE FENETRE (24 LETTRES AUMAXIMUM)."

```

CE PROGRAMME TOURNE SUR APPLE II_e et APPLE II_c



Un certain jour de la semaine...

Si quelqu'un prétend que le 1^{er} Janvier 1583 était un samedi, demandez à ce programme de vous le confirmer.

La France a adopté le calendrier grégorien depuis le 9 décembre 1582... qui fut suivi du lundi 20 décembre 1582. Le 1^{er} Janvier 1583 était bien un samedi.

Est bissextile toute année dont le millésime est divisible par 4, sauf si elle est terminée par deux 0 et non divisible par 400 (1600 a été la seule année séculaire bissextile. La prochaine sera l'an 2000 !).

**QUEL JOUR TOMBE
LE 1^{er} JANVIER
1984 ?**

ANNÉE = 1984 - 1

JOUR = ANNÉE + 1 - 1582 + INT (AN/4) + INT (AN/400) - INT (AN/100)

Le premier et le dernier jours d'une année normale sont identiques tandis que l'année bissextile finit un jour plus tard. Ça coule de source, pas vrai ?

PREM = JOUR - INT (JOUR/7) * 7 (Le 0 est dimanche, le 1 lundi, etc.)

VOTRE JOUR ANNIVERSAIRE

```
10 TEXT : PRINT CHR$(21): HOME
15 XD$ = "00/00/1900"
20 T$ = "": FOR I = 1 TO 10:T$ = T$ + "___": NEXT :B$ = "
"
25 PRINT T$: PRINT
30 J$ = "DIMANCHELUNDI MARDI MERCREDIJEUDI VENDREDIS
AMEDI ":L = LEN (J$)
35 GOTO 70
40 FOR A = 1 TO 20:BRUIT = PEEK (49200): NEXT : RETURN
45 :
50 REM *****
55 REM ** ECRAN DE SAISIE **
60 REM *****
65 :
70 FOR I = 1 TO L STEP 8: PRINT MID$(J$,I,8): NEXT
75 POKE 32,29: PRINT : INVERSE : VTAB 4: PRINT B$: PRINT
" QUEL JOUR ": PRINT " ETES-VOUS ": PRINT " NE(E) ? "
: PRINT B$: NORMAL
80 POKE 32,0
85 VTAB 12: PRINT T$: VTAB 20: PRINT T$
90 GOTO 350
95 FOR I = 1 TO 8:M1$ = MID$( " DLMMJVS",I,1):M2$ = MID
$( " DLMMJVS",I + 1,1): VTAB I + 3: HTAB 1: INVERSE : P
RINT M2$: NORMAL : VTAB I + 2: HTAB 1: PRINT M1$
100 GOSUB 40
105 NEXT
110 X = PEEK (49249): REM TEST POMME OUVERTE
115 IF X < 128 THEN 95
120 :
```

PREM = 0

(donc un dimanche)

POKE 32,29 réduit la fenêtre d'écran et permet d'écrire 10 caractères à partir de HTAB 31. Pour éviter de sortir de l'écran, il est bon d'ajouter un POKE 33,10. Attention ! le total (29 + 10) ne doit en aucun cas dépasser 39. Un PRINT est indispensable à la suite de cette instruction. POKE 32,0 (et éventuellement POKE 33,40) rétablit l'écran normal.

Il faut maintenir le doigt sur la touche POMME OUVERTE.

Dans un programme plus long, éliminez les **REM** qui occupent de la place en mémoire.

Lecture des caractères en mode inverse et réaffichage en mode normal (quand on presse la touche ←, qui fait reculer d'une case).

Les **IF** interdisent un certain nombre de saisies erronées, du genre «32/13/2380».

VARIABLES

AN = Année
MO = Mois
VJ = Votre Jour
J\$ = Les jours de la semaine (8 caractères ou espaces chacun)

H et **V** sont utilisées pour **HTAB** et **VTAB**

Ce programme tourne sur
APPLE II® & APPLE II®

```

125 REM *****
130 REM **      DATE DE NAISSANCE      **
135 REM *****
140 :
145 H = 1:V = 14:VD$ = "":VJ = 0
150 VTAB V: HTAB H: PRINT "VOTRE DATE DE NAISSANCE : "XD$:
    GOTO 170
155 IF H = 1 THEN VD$ = "": RETURN
160 VD$ = LEFT$(VD$,H - 1): RETURN
165 PRINT S$
170 VTAB V: HTAB 26 + H: GET S$:S = ASC(S$)
175 IF H = 1 OR S < > 8 THEN 190
180 H = H - 1: POKE 1729 + H, PEEK(1729 + H) + 128: GOSUB
    155: IF PEEK(1729 + H) = 47 THEN 175
185 GOTO 170
190 IF S < 48 OR S > 59 THEN 170
195 GOSUB 40
200 IF H = 1 AND S > 51 THEN 165
205 IF H = 2 AND VAL(VD$ + S$) > 31 THEN 165
210 IF H = 4 AND S > 49 THEN 165
215 IF H = 5 AND VAL(RIGHT$(VD$,1) + S$) > 12 THEN 165
220 IF H = 8 AND S < 53 THEN 165
225 GOTO 230
230 IF H = 2 OR H = 5 THEN S$ = S$ + "/":H = H + 1
235 IF H = 6 THEN S$ = S$ + "1":H = H + 1
240 VD$ = VD$ + S$
245 INVERSE : PRINT S$: NORMAL
250 IF H < 10 THEN H = H + 1: GOTO 170
255 :
260 REM *****
265 REM ***  CALCUL DU JOUR ANNIVERSAIRE  **
270 REM *****
275 :
280 AN = VAL(RIGHT$(VD$,4)):MO = VAL(MID$(VD$,4,2))
    :JO = VAL(LEFT$(VD$,2))
285 IF MO < 3 THEN MO = MO + 12:AN = AN - 1
290 VJ = VJ + JO + AN
295 VJ = VJ + 2 * MO + INT(.6 * (MO + 1)) + INT(AN / 4)
    - INT(AN / 100) + INT(AN / 400) + 2
300 VJ = INT((VJ / 7 - INT(VJ / 7)) * 7 + .5)
305 VTAB V + 2: HTAB 1: PRINT "CE JOUR-LA ETAIT UN "; IN
    VERSE
310 IF VJ = 0 THEN VJ = 7
315 PRINT MID$(J$, (VJ * 8) - 7, 8): NORMAL
320 :
325 REM *****
330 REM **  ON RECOMMENCE OU ON ARRETE  **
335 REM *****
340 :
345 PRINT : VTAB 22: CALL - 868: PRINT "<1> POUR UNE AUTR
    E DATE <2> POUR FINIR "; GET R$: IF R$ < > "1" THEN
    TEXT : HOME : END
350 PRINT : VTAB 22: PRINT "POMME OUVERTE POUR SAISIR UNE
    AUTRE DATE": GOTO 95

```

Listage et édition en langage machine

Désassembler et éditer 20 instructions en langage machine est facile. Il suffit de passer en mode moniteur par **CALL-151** et **RETURN**, puis de taper **CTRL-D PRÉ1, RETURN**, et l'adresse (par exemple * 300 L, 300 LL pour 20 ou 40 instructions, et ainsi de suite... mais toujours par groupe de 20 lignes désassemblées).

Notre programme permet d'éditer une routine, à raison de 60 lignes par page, quel que soit le nombre de ses instructions. Il suffit de connaître son adresse en **HEXADÉCIMAL**, et l'adresse suivant son dernier octet.

On utilise pour cela un sous-programme de l'**APPLE** (à l'adresse \$ FE63) qui désassemble le nombre de lignes stockées dans l'**ACCUMULATEUR**, et pas davantage.

```
10 LOMEM: 4350
15 TEXT : NORMAL : PRINT CHR$ (21): HOME
20 L1 = 4320:L2 = L1 + 21:L3 = L2 + 5
25 FOR I = L1 TO L3: READ R: POKE I,R: NEXT
30 PI$ = "n072N080E096q120Q136P160p144"
35 INVERSE : HTAB 5: PRINT "LISTAGE DE N INSTRUCTIONS EN
    LM": NORMAL
40 T$ = "": FOR I = 1 TO 10:T$ = T$ + "____": NEXT : PRINT
    T$: POKE 34,3
45 :
50 REM *****
55 REM *      SAISIE DES PARAMETRES      *
60 REM *****
65 :
70 VTAB 20: PRINT T$: VTAB 22: HTAB 1: PRINT "<?> PERMET
    DE REVENIR A L'ITEM PRECEDENT": POKE 35,19
75 VTAB 4: HTAB 1: PRINT "SLOT DE L'IMPRIMANTE = ";; GET
    G$: PRINT G$:SL% = ASC (G$) - 48: IF SL% < 1 OR SL% >
    7 OR SL% = 6 OR SL% = 3 THEN 75
80 VTAB 6: HTAB 1: PRINT "TYPE DE CARACTERE (nNEqQPp) =
    ";; GET G$: IF G$ = "?" THEN 75
85 FOR I = 1 TO 32 STEP 4: IF G$ < > MID$ (PI$,I,1) THE
    N NEXT : GOTO 80
90 LM% = VAL ( MID$ (PI$,I + 1,3)):C$ = G$: PRINT C$-"LM
    %
95 VTAB 8: HTAB 1: PRINT "LARGEUR DE LA MARGE (MAX="LM% -
    40") = ";; INPUT "":G$:MA% = VAL (G$): IF MA% > LM% -
    40 THEN 95
100 M$ = "L" + RIGHT$ (<"000" + STR$ (MA%)),3)
105 IF G$ = "?" THEN 80
110 VTAB 10: HTAB 1: PRINT "DEBUT DE PROGRAMME EN HEXA =
    0000";; HTAB 30: CALL - 657
115 AD = 1: GOSUB 270
120 IF A% = 63 THEN 95
125 IF S = 1 THEN 110
130 VTAB 12: HTAB 1: PRINT "& FIN DE PROGRAMME EN HEXA =
    0000";; HTAB 30: CALL - 657
135 AD = 2: GOSUB 270
```

← **LOMEM** protège les 27 octets du sous-programme en langage machine.

← Paramètres de l'imprimante **APPLE**.

← Naturellement, si vous utilisez une imprimante différente, certaines de ces lignes seront à revoir.

← Il faut taper 4 caractères **HEXA**. Exemples: 0010, 0300, 0A1F.

Ce programme tourne
sur
APPLE II. & APPLE IIc

IMPORTANT

Quand vous aurez tapé votre programme, faites en mode direct, un

PRINT PEEK (105) +
PEEK (106) * 256

Si le résultat dépasse 4319, modifiez d'autant la valeur de L1 (ligne 20), puis celle de LOMEM, qui devra être supérieure à L3 d'au moins 1.

NL est le nombre de lignes par page. Vous pouvez le modifier.

On écrit à l'adresse \$3A — \$3B (58 et 59) l'adresse de départ du programme à éditer.

```
140 IF A% = 63 THEN I10
145 IF S = 1 THEN 130
150 IF AD = 1 THEN 130
155 VTAB 14: HTAB 1: PRINT "IMPRESSION DE "; INVERSE : PRINT A(1);: NORMAL : PRINT " A ";: INVERSE : PRINT A(2): NORMAL
160 VTAB 16: PRINT "TITRE DU PROGRAMME (OU RETURN)
.....": VTAB 18: HTAB 1: INPUT " ";TI$
165 IF TI$ = "?" THEN 130
170 VTAB 16: CALL - 958: PRINT "TITRE ";: INVERSE : PRINT TI$;: NORMAL : PRINT " ";: GET G$: PRINT " "; IF G$ = "?" THEN 160
175 VTAB 18: INVERSE : PRINT "L'IMPRIMANTE EST-ELLE PRETE ? ";: GET G$: PRINT " "; NORMAL
180 :
185 REM *****
190 REM * EDITION SUR IMPRIMANTE APPLE *
195 REM *****
200 :
205 POKE 34,22: POKE 50,128: PRINT CHR$ (4)"PR£"SL%: PRINT CHR$ (27) + C$;: PRINT CHR$ (27) + M$
210 IF TI$ < > "" THEN PRINT CHR$ (27)"X" CHR$ (14)TI$ CHR$ (27)"Y" CHR$ (15): PRINT
215 NX = 0:NL = 60: IF TI$ < > "" THEN NL = 60 - 4
220 CALL L2:NX = NX + 1: IF NX > NL THEN PRINT CHR$ (12):NX = 1:NL = 60
225 IF ( PEEK (58) + PEEK (59) * 256) < A(2) THEN 220
230 PRINT CHR$ (12): PRINT CHR$ (4)"PR£0"
235 POKE 50,255: VTAB 22: CALL - 868: PRINT "<1> AUTRE LI STAGE <2> FIN DE PROGRAMME ";: GET G$: PRINT " "; IF G$ = "1" THEN 45
240 TEXT : HOME : END
245 :
250 REM *****
255 REM * ADRESSE HEXA MISE SUR 2 OCTETS *
260 REM *****
265 :
270 S = 0: CALL - 10951: REM MISE A ZERO DU BIT 7 DES CARACTERES DU BUFFER
275 FOR X = 512 TO 515:A% = PEEK (X)
280 IF A% > 47 AND A% < 60 THEN POKE X - 506,A% - 48: GOT 0 295
285 IF A% < 65 OR A% > 70 THEN S = 1: RETURN
290 POKE X - 506,A% - 55
295 NEXT : IF X < > 516 THEN S = 1: RETURN
300 CALL L1
305 A(AD) = PEEK (6) + PEEK (7) * 256
310 IF AD = 1 THEN POKE 58, PEEK (6): POKE 59, PEEK (7)
315 IF AD = 2 AND A(2) < A(1) THEN S = 1
320 RETURN
325 DATA 165,6,10,10,10,10,101,7,133,7,165,8,10,10,10,10,101,9,133,6,96
330 DATA 169,1,32,99,254,96
```


Les nombres décimaux et votre micro

Rares sont les amateurs de programmation qui cherchent à réinventer la poudre en concoctant quelque fade imitation de logiciels qui ont depuis longtemps prouvé leur efficacité : Visicalc et Magicalc.

Pourtant, vous avez certainement pondé, et assurément avec un réel talent, de petits programmes dans lesquels le problème de l'affichage de NOMBRES DÉCIMAUX vous a donné du fil à retordre.

Votre ordinateur personnel chéri élimine les zéros les plus à gauche de la partie entière des nombres ($0.49 = .49$), ce qui n'est pas trop grave, et il se comporte bizarrement avec les chiffres qui sont à droite de la virgule (pardon : du point). Heureusement, il est possible de pallier la plupart de ses défaillances (il possède tellement de qualités par ailleurs, qu'on ne va tout de même pas lui reprocher de rares défauts !).

EXEMPLES : A = 1.499 B = 1.501 C = 1.999

ARRONDIR AU FRANC SUPERIEUR

- PRINT INT (A) " " INT (B) " " INT (C)
 - Affiche : "1 1 1"
-
- PRINT INT (A + 0.5) " " INT (B + 0.5) " " INT (C + 0.5)
 - Affiche : "1 2 2" (arrondi correct)

PAS PLUS DE DEUX DECIMALES

Les formules " A = INT (A * 100) / 100 et
A = INT (A * 10) / 10 "
sont terriblement efficaces.

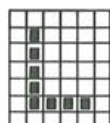
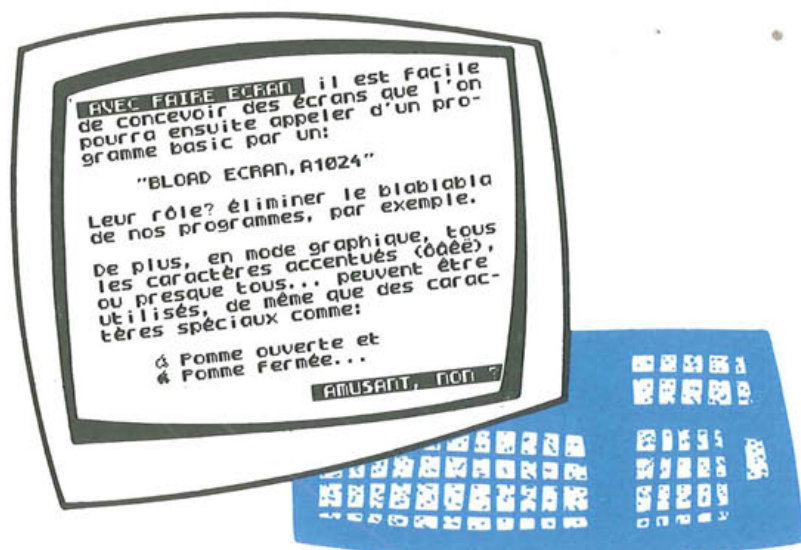
- PRINT INT (A * 100) / 100 ; " — " ; INT (B * 100) / 100 ;
" — " ; INT (C * 100) / 100
- 1.49 — 1.5 — 1.99
- PRINT INT (A * 10) / 10 ; " — " ; INT (B * 10) / 10 ; " — " ;
INT (C * 10) / 10
- 1.4 — 1.5 — 1.9

◀ **Attention !** Si le chiffre à arrondir est négatif, le résultat est surprenant. Ainsi INT (- 4.33) = - 5. Plus de problème si l'on améliore la formule : INT (ABS (- 4.33)) / SGN (- 4.33) = - 4

◀ Pour 3 décimales, on modifierait ainsi la formulation :

- A = 1.1234
- PRINT INT (A * 1000) / 1000
- 1.123

Créez vos écrans en vous amusant !



ES écrans consacrés aux divers menus et explications concernant le programme occupent souvent autant de place, en mémoire, que les algorithmes imaginés par l'auteur, et justifiant seuls le logiciel.

Or, il est facile de se débarrasser de ces pages inutiles en confiant leur mémorisation à la disquette, où il sera toujours possible de les retrouver par un **PRINT CHR\$(4) "BLOAD ECRAN, A1024"** (ou **A\$400**).

Ce programme comprend :

- 1 FAIRE.ECRAN (Basic \$800)
- 2 FAIRE.LM (\$6000 — \$60EB)
- 3 FAIRE.MEVEC (\$340 — \$3AF)
- 4 FAIRE.F1 (\$6100 — \$63FF)

Vous créez vous-même le tableau des fonctions :

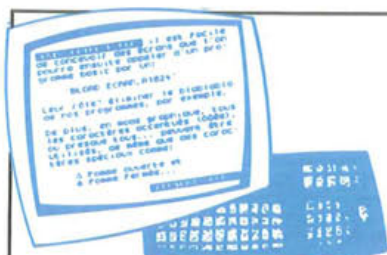
- 5 FAIRE.COM (\$400 — \$7FF)

S'il s'agit d'un écran de commandes (c'est le cas de **FAIRE.COM**, dans le présent programme), il est intéressant de le ranger à une adresse préalablement choisie (la mémoire auxiliaire, par exemple), et de l'afficher instantanément par un petit sous-programme en langage-machine.

La partie **\$60D9** à **\$60EB** de **FAIRE.LM**, est utilisée pour de tels transferts.

Vous savez comment opérer ? Par exemple :

- 1 \$400 — \$7FF vers \$5000
 - POKE 6,0 : POKE 7,04 (\$04/00, partie basse, puis partie haute, en décimal... mais 4 vaut toujours 4).
 - POKE 8,0 : POKE 9,80 (\$50/00, mais 50 hexa = 80 en décimal).
 - CALL 24793 (ici, adresse décimale de \$60CF, notre sous-programme).
- 2 \$5000 — \$57FF vers \$400...
 - POKE 6,0 : POKE 7,80 : POKE 8,0 : POKE 9,04 : CALL 24793.L'écran est affiché, mais il est toujours mémorisé en \$5000... où vous le retrouverez quand bon vous semblera.



1 FAIRE ÉCRAN

(VERSION 1-1)

```

10 LOMEM: 25650
20 ONERR GOTO 1100
30 TEXT : NORMAL : SPEED= 255: PRINT CHR$(21): HOME
40 DIM L$(41): GOSUB 960
50 :
60 REM *****
70 REM *** SAISIE D'UN ECRAN TEXTE ***
80 REM *****
90 :
100 VTAB 1: CALL 24751
110 GET H$:A% = ASC (H$): IF A% > 31 THEN 340
115 IF A% = 6 OR A% = 9 OR A% = 14 THEN 340
120 IF A% = 7 THEN GOSUB 810: GOSUB 880: CALL - 198:
POKE - 16368,0: WAIT - 16384,128,127: POKE - 1636
8,0: TEXT : GOTO 100
130 IF A% = 13 THEN GOSUB 610: GOTO 340: REM RETURN
140 IF A% = 16 THEN GOSUB 810: GOSUB 830: GOTO 100: RE
M CTRL-P (Page de fonctions)
150 IF A% = 3 THEN GOSUB 600: GOSUB 370: GOTO 340: REM
CTRL-C (Centrage)
160 IF A% = 20 THEN POKE 36,0: POKE 37,0: CALL 24696:
GOTO 100: REM CTRL-T (Cadre sans effacement)
170 IF A% = 30 THEN CALL 24693: GOTO 100: REM CTRL-FLE
X (Cadre avec effacement de page texte)
180 IF A% = 12 THEN 540: REM CTRL-L (Load éventuel écran)
190 IF A% = 5 THEN GOSUB 460: VTAB 23: HTAB 1: PRINT "
==== VOUS EFFACEZ VRAIMENT (O/N) ? "; GET R$: IF
R$ < > "O" THEN GOSUB 820: GOTO 90
200 IF A% = 5 THEN TEXT : HOME : GOTO 100: REM CTRL-E
(Effacement confirmé)
210 IF A% = 21 THEN CALL - 1036: GOTO 110: REM CTRL-U
(Déplacement -)
220 IF A% = 27 THEN GOSUB 810: GOTO 1030: REM Sortie d
u programme avec ESC
230 IF A% = 8 THEN CALL - 1008: GOTO 110: REM CTRL-H
(Déplacement <-)
240 IF A% = 17 THEN 1030: REM CTRL-Q (Quitter)
250 IF A% = 26 THEN VTAB 23: HTAB 1: GOTO 110: REM CTR
L-Z (Début de dernière ligne)
260 IF A% = 1 THEN CALL - 998: CALL - 926: GOTO 110:
REM CTRL-A (Début de ligne en cours)
270 IF A% = 2 THEN CALL - 998: CALL - 926: CALL - 1
008: CALL - 922: GOTO 110: REM CTRL-B (Fin de ligne)

```

← **LOMEM : 25650** protège les programmes, la table de caractères, les pages rangées en mémoire... L'APPLE peut encore travailler !

← **7 = CTRL-G.** On mémorise la page de texte à l'adresse 5000, puis on va vers la routine qui transfère la page 1 en HI-RES page 1 (GOSUB 880).

← Si l'on tape **CTRL-î** (Flexe) on efface la page avant de pouvoir tracer un nouveau cadre.

← La sortie avec **ESCAPE** est facile, mais peut être accidentelle. Heureusement, en 1030, tout peut s'arranger !

← **CTRL-Z** permet d'aller vite fait au début de la dernière ligne.

5 FAIRE.COM

Quand vous aurez terminé la frappe de votre programme, faites un "LOAD FAIRE-ECRAN", puis supprimez (provisoirement) la partie "BLOAD FAIRE.COM, A\$4000" de la ligne 960.

Apprenez ensuite à user au mieux de votre programme en composant cette page de **fonctions**, puis faites un CTRL-S... pour la sauver, sur disquette, sous le titre **FAIRE.COM** votre (petit) logiciel est maintenant complet.

CTRL-K : même effet que la flèche ↑ de l'Apple et **CTRL-J** même effet que la flèche ↓

Attention ! voyez les paramètres de votre imprimante si vous désirez imprimer quelque chose.

Routine de centrage appelée par **CTRL-C**, à la ligne 150. Ne fonctionne pas s'il y a un cadre, mais on peut tourner la difficulté en effaçant celui-ci par un **CTRL-T + ESPACE**, c'est instantané... de même que pour refaire un nouveau cadre, alors pourquoi s'en priver ?

FONCTIONS POSSIBLES AVEC CE PROGRAMME

CTRL-L = Load	CTRL-A = Début ligne
CTRL-S = Save	CTRL-B = Fin de ligne
CTRL-Q = Fin	
CTRL-E = Effacer	AUTRES DEPLACEMENTS
CTRL-C = Centrer	avec les flèches, sans
	"CTRL", bien entendu.
CTRL-I = Inverse	
CTRL-F = Flash	CTRL-G = TRANSF PAGE1
CTRL-N = Normal	CTRL-W = BSAVE PAGE1

----- COMMANDES: **CTRL-R** -----
(l'écran en cours est mémorisé en \$5000)

CTRL-X = IMPRESSION de la PAGE1 (simple)

CTRL-Y = IMPRESSION de la PAGE1 (double)

CADRE: CTRL-I (ou Flexe pour effacement)

```

280 IF A% = 11 THEN CALL - 998: GOTO 110: REM CTRL-K
(Ligne au-dessus)
290 IF A% = 10 THEN GOSUB 610: CALL - 922: GOTO 110:
REM CTRL-J (Ligne suivante)
300 IF A% = 19 THEN 520: REM CTRL-S (Save écran texte)
310 IF A% = 23 THEN 530: REM CTRL-W (Bsave écran graphi
que)
320 IF A% = 24 THEN GOSUB 810: GOSUB 710: GOTO 100: RE
M CTRL-X (Impression HGR simple)
330 IF A% = 25 THEN GOSUB 810: GOSUB 700: GOTO 100: RE
M CTRL-Y (Impression HGR double format)
335 IF A% < 32 THEN 110
340 GOSUB 600: IF P = 23 AND A = 39 THEN POKE 2039, AS
C (H$) + 128: GOTO 110
350 PRINT H$;: GOTO 110
360 :
370 FOR I = 0 TO 40: L%(I) = 160: NEXT
380 L = VAL ( MID$ ( E$, (P + 1) * 4, 4) ); FOR I = 0 TO 39
: L%(I) = PEEK ( I + L ): NEXT
390 FOR I = 0 TO 39: IF L%(I) < > 160 THEN D = I: GOTO
410
400 NEXT
410 FOR I = 39 TO 0 STEP - 1: IF L%(I) < > 160 THEN F
= I: GOTO 430
420 NEXT
430 B = F + 1 - D: C = (39 - B) / 2: IF D < > 1 THEN C =
C + 1 - D
440 POKE 37, P: POKE 36, 0: CALL - 868: FOR I = D TO F:
POKE I + C + L, L%(I): NEXT
450 RETURN
460 NORMAL : GOSUB 810: VTAB 21: HTAB 1: CALL - 958: P
RINT T$: RETURN

```

(Suite page 21)



2 FAIRE.LM

6000-	A6 22	LDX	\$22
6002-	8A	TXA	
6003-	85 1C	STA	\$1C
6005-	20 24 FC	JSR	\$FC24
6008-	A0 00	LDY	£\$00
600A-	A5 28	LDA	\$28
600C-	85 2A	STA	\$2A
600E-	A5 29	LDA	\$29
6010-	29 03	AND	£\$03
6012-	05 E6	ORA	\$E6
6014-	85 2B	STA	\$2B
6016-	B1 28	LDA	(\$28),Y
6018-	C9 20	CMP	£\$20
601A-	B0 05	BCS	\$6021
601C-	69 C0	ADC	£\$C0
601E-	CE 52 60	DEC	\$6052
6021-	C9 60	CMP	£\$60
6023-	B0 05	BCS	\$602A
6025-	69 80	ADC	£\$80
6027-	CE 52 60	DEC	\$6052
602A-	C9 80	CMP	£\$80
602C-	B0 05	BCS	\$6033
602E-	69 40	ADC	£\$40
6030-	CE 52 60	DEC	\$6052
6033-	C9 A0	CMP	£\$A0
6035-	90 31	BCC	\$6068
6037-	29 7F	AND	£\$7F
6039-	18	CLC	
603A-	2A	ROL	
603B-	2A	ROL	
603C-	2A	ROL	
603D-	48	PHA	
603E-	2A	ROL	
603F-	29 03	AND	£\$03
6041-	09 60	DRA	£\$60
6043-	8D 50 60	STA	\$6050
6046-	68	PLA	
6047-	29 F8	AND	£\$F8
6049-	8D 4F 60	STA	\$604F
604C-	A2 00	LDX	£\$00
604E-	BD F8 62	LDA	\$62F8,X
6051-	49 00	EOR	£\$00

6053-	EA	NOP	
6054-	EA	NOP	
6055-	91 2A	STA	(\$2A),Y
6057-	18	CLC	
6058-	A5 2B	LDA	\$2B
605A-	69 04	ADC	£\$04
605C-	85 2B	STA	\$2B
605E-	E8	INX	
605F-	E0 08	CPX	£\$08
6061-	90 EB	BCC	\$604E
6063-	A9 00	LDA	£\$00
6065-	8D 52 60	STA	\$6052
6068-	C8	INX	
6069-	C4 21	CPY	\$21
606B-	90 9D	BCC	\$600A
606D-	A6 1C	LDX	\$1C
606F-	E8	INX	
6070-	E4 23	CPX	\$23
6072-	90 8E	BCC	\$6002
6074-	60	RTS	
6075-	20 5B FC	JSR	\$FC5B
6078-	20 0C FD	JSR	\$FD0C
607B-	A2 00	LDX	£\$00
607D-	9D 00 04	STA	\$0400,X
6080-	9D D0 07	STA	\$07D0,X
6083-	E8	INX	
6084-	E0 28	CPX	£\$28
6086-	D0 F5	BNE	\$607D
6088-	A2 00	LDX	£\$00
608A-	99 00 04	STA	\$0400,Y
608D-	99 00 05	STA	\$0500,Y
6090-	99 00 06	STA	\$0600,Y
6093-	99 00 07	STA	\$0700,Y
6096-	E8	INX	
6097-	E0 0C	CPX	£\$0C
6099-	F0 08	BEQ	\$60A3
609B-	48	PHA	
609C-	BD A3 60	LDA	\$60A3,X
609F-	A8	TAY	
60A0-	68	PLA	
60A1-	D0 E7	BNE	\$608A
60A3-	60	RTS	
60A4-	27	???	
60A5-	28	PLP	
60A6-	4F	???	
60A7-	50 77	BVC	\$6120
60A9-	80	???	
60AA-	A7	???	
60AB-	A8	TAY	
60AC-	CF	???	
60AD-	D0 F7	BNE	\$60A6
60AF-	A9 BB	LDA	£\$BB
60B1-	85 36	STA	\$36
60B3-	A9 60	LDA	£\$60

60B5-	85 37	STA	\$37
60B7-	20 EA 03	JSR	\$03EA
60BA-	60	RTS	
60BB-	C9 86	CMP	£\$86
60BD-	F0 08	BEQ	\$60CA
60BF-	C9 8E	CMP	£\$8E
60C1-	F0 0C	BEQ	\$60CF
60C3-	C9 89	CMP	£\$89
60C5-	F0 0D	BEQ	\$60D4
60C7-	4C F0 FD	JMP	\$FDF0
60CA-	A9 7F	LDA	£\$7F
60CC-	85 32	STA	\$32
60CE-	60	RTS	
60CF-	A9 FF	LDA	£\$FF
60D1-	4C CC 60	JMP	\$60CC
60D4-	A9 3F	LDA	£\$3F
60D6-	4C CC 60	JMP	\$60CC
60D9-	A2 04	LDX	£\$04
60DB-	A0 00	LDY	£\$00
60DD-	B1 06	LDA	(\$06),Y
60DF-	91 08	STA	(\$08),Y
60E1-	88	DEY	
60E2-	D0 F9	BNE	\$60DD
60E4-	E6 07	INC	\$07
60E6-	E6 09	INC	\$09
60E8-	CA	DEX	
60E9-	D0 F0	BNE	\$60DB
60EB-	60	RTS	

BSAVE FAIRE.LM, A\$6000, L\$EC

3 FAIRE.MEVEC

Ce programme sauve, puis restaure les 8 octets de MEV réservés aux divers périphériques. Peu utile sur l'APPLE IIe, mais indispensable sur le IIc.

0340-	A2 00 BD 78 04 9D 00 03
0348-	BD F8 04 9D 08 03 BD 78
0350-	05 9D 10 03 BD F8 05 9D
0358-	18 03 BD 78 06 9D 20 03
0360-	BD F8 06 9D 28 03 BD 78
0368-	07 9D 30 03 BD F8 07 9D
0370-	38 03 E8 E0 08 D0 CB 60
0378-	A2 00 BD 00 03 9D 78 04
0380-	BD 08 03 9D F8 04 BD 10
0388-	03 9D 78 05 BD 18 03 9D
0390-	F8 05 BD 20 03 9D 78 06
0398-	BD 28 03 9D F8 06 BD 30
03A0-	03 9D 78 07 BD 38 03 9D
03A8-	F8 07 E8 E0 08 D0 CB 60

BSAVE FAIRE.MEVEC, A\$340, L\$70

Attention ! Une page
texte occupe 1024
octets (6 secteurs de la dis-
quette) et une page HGR
8192 octets (34 secteurs) !

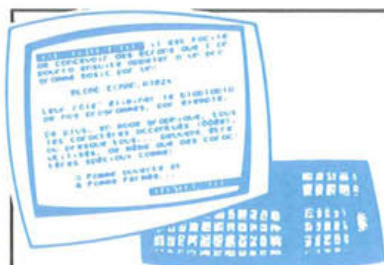
Si le titre n'existe pas,
un message d'erreur
s'affiche puis on vous
demande si vous désirez le
catalogue.

Si vous ne possédez pas
une telle carte, il vous
faudra incorporer une rou-
tine de copie d'écran dans
votre programme. On n'a
pas prévu la copie d'un
écran texte.

Les adresses des zones
intéressées par le trans-
fert sont Pokées en page 0.

```
470 :
480 REM *****
490 REM **** SAVE TEXTE OU BSAVE PAGE 1 **
500 REM *****
510 :
520 GOSUB 550: PRINT CHR$(4)"BSAVE"F$,A1024,L1024":
TEXT : VTAB 1: GOTO 110
530 GOSUB 550: PRINT CHR$(4)"BSAVE"F$,A8192,L8192":
TEXT : VTAB 1: GOTO 110
540 CALL 832: GOSUB 550: PRINT CHR$(4)"BLOAD"F$,A102
4": TEXT : NORMAL : VTAB 1: HTAB 1: CALL 888: GOTO 1
00
550 GOSUB 460
560 HTAB 1: VTAB 23: INPUT "TITRE DE L'ECRAN ";F$: IF
F$ = "" THEN GOSUB 820: GOTO 630
570 GOSUB 820
580 VTAB 1: PRINT "": POKE 34,23: POKE 50,128
590 RETURN
600 P = PEEK (37):A = PEEK (36): RETURN
610 P = PEEK (37): IF P = 23 THEN 630
620 RETURN
630 CALL - 198: POP : GOTO 100
640 :
650 REM *****
660 REM *** IMPRESSION DE LA PAGE GRAPH ***
670 REM *****
680 :
690 REM CES CODES SONT CEUX DE LA CARTE BUFFER DE BIP
700 PRINT : PRINT CHR$(4)"PR£1": PRINT CHR$(27)"L00
5": PRINT CHR$(9)"GLED": GOTO 720
710 PRINT : PRINT CHR$(4)"PR£1": PRINT CHR$(27)"L00
5": PRINT CHR$(9)"GLE"
720 PRINT CHR$(12): PRINT CHR$(4)"PR£0": RETURN
730 POKE 51,128
740 GOSUB 820: GOSUB 460: PRINT "TAPER <C> POUR LE CATA
LOGUE ";: GET R$: IF R$ = "C" THEN TEXT : PRINT :
HOME : PRINT D$"CATALOG": .GET R$: PRINT : GOTO 740
750 GOSUB 820: GOTO 100
760 :
770 REM *****
780 REM *** DEPLACEMENT DES ECRANS ***
790 REM *****
800 :
810 POKE 6,00: POKE 7,04: POKE 8,00: POKE 9,80: CALL 24
793: RETURN
820 POKE 6,00: POKE 7,80: POKE 8,00: POKE 9,04: CALL 24
793: RETURN
830 POKE 6,00: POKE 7,64: POKE 8,00: POKE 9,04: CALL 24
793: POKE - 16368,0: WAIT - 16384,128,127: POKE -
16368,0: GOTO 820
840 :
850 K = 1: REM HI-RES PAGE 1
860 HOME : HTAB 12: INPUT "FONT ";F$: IF F$ = "" THEN
RETURN : REM CARACTERES GRAPHIQUES
```

(Suite page 23)



4 FAIRE.F1

Je sais, c'est indigeste, mais il faut y passer... sans se tromper. Notez que vous pouvez utiliser les FONTES de l'utilitaire TOOLKIT APPLE (non accentuées).

CARACTERES DE LA FONTE FAIRE.F1													
1	2	3	4	5	6	7	8	9	0	á	-		
A	Z	E	R	T	Y	U	I	O	P	é	*		
Q	S	D	F	G	H	J	K	L	M	%	#		
»	W	X	C	V	B	N	?	.	/	+			
ó	é	"	'	(€	è	:	ç	à)	-		
a	z	e	r	t	y	u	i	o	p	â	\$		
q	s	d	f	g	h	j	k	l	m	ù	'		
«	w	x	c	v	b	n	,	;	:	=			

MEMORISATION : BSAVE FAIRE.F1, A\$6100, L\$300

6100-	00	00	00	00	00	00	00	00	00	6200-	04	0C	3E	23	23	33	2E	00	6300-	0C	0C	08	00	00	00	00	00	00
6108-	00	0C	0C	0C	00	0C	0C	00	00	6208-	00	1E	23	23	3F	23	23	00	6308-	00	00	3E	23	23	33	2E	00	00
6110-	80	36	12	00	00	00	00	00	00	6210-	00	1F	23	1F	23	23	1F	00	6310-	00	03	1B	27	23	23	1F	00	00
6118-	00	16	37	16	16	37	16	00	00	6218-	00	3C	02	03	03	02	3C	00	6318-	00	00	3E	03	03	03	3E	00	00
6120-	00	08	3E	81	1E	20	1F	08	00	6220-	00	0F	13	23	23	13	0F	00	6320-	00	20	2E	33	23	23	3E	00	00
6128-	00	27	17	08	04	3A	39	00	00	6228-	00	3F	03	1F	03	03	3F	00	6328-	00	00	3E	23	1F	03	3E	00	00
6130-	08	14	1E	23	23	23	1E	00	00	6230-	00	3F	03	1F	03	03	03	00	6330-	00	1E	03	1F	03	03	03	00	00
6138-	00	0C	04	00	00	00	00	00	00	6238-	00	3C	02	33	33	22	1C	00	6338-	00	00	1E	23	33	2E	20	1F	00
6140-	00	30	08	0C	0C	08	30	00	00	6240-	00	23	23	3F	23	23	23	00	6340-	00	03	1B	27	23	23	23	00	00
6148-	00	03	04	0C	0C	04	03	00	00	6248-	00	1E	0C	0C	0C	0C	1E	00	6348-	00	0C	00	0C	0C	0C	0C	00	00
6150-	00	2A	1C	36	1C	2A	00	00	00	6250-	00	3C	18	18	18	19	0E	00	6350-	00	18	00	18	18	18	18	0E	00
6158-	00	08	08	3E	08	08	00	00	00	6258-	00	13	0B	07	0B	13	23	00	6358-	00	03	23	13	0F	13	23	00	00
6160-	00	00	00	00	00	06	06	02	00	6260-	00	03	03	03	03	03	3F	00	6360-	00	0C	0C	0C	0C	0C	0C	00	00
6168-	00	00	00	1E	00	00	00	00	00	6268-	00	1F	2B	2B	2B	2B	2B	00	6368-	00	00	1F	2B	2B	2B	2B	00	00
6170-	00	00	00	00	00	06	06	00	00	6270-	00	1F	23	23	23	23	23	00	6370-	00	00	1F	23	23	23	23	00	00
6178-	00	20	10	08	04	02	01	00	00	6278-	00	1E	23	23	23	23	1E	00	6378-	00	00	1E	23	23	23	1E	00	00
6180-	00	3E	B3	2B	27	23	1E	00	00	6280-	00	1F	23	1F	03	03	03	00	6380-	00	00	1F	23	23	27	1B	03	00
6188-	00	18	1E	18	18	18	18	00	00	6288-	00	1E	23	23	23	23	12	3C	6388-	00	00	3E	23	23	33	2E	20	00
6190-	00	1F	30	98	0C	07	3F	00	00	6290-	00	1F	23	1F	23	23	23	00	6390-	00	00	1E	03	03	03	03	00	00
6198-	00	3F	18	1E	30	B0	1F	00	00	6298-	00	1E	07	1E	38	38	1F	00	6398-	00	00	3E	03	1E	30	1F	00	00
61A0-	00	0C	06	1B	1F	18	18	00	00	62A0-	00	3F	0C	0C	0C	0C	0C	00	63A0-	00	03	03	1F	03	03	1E	00	00
61A8-	00	3F	03	1F	30	30	9F	00	00	62A8-	00	23	23	23	23	23	3E	00	63A8-	00	00	23	23	23	23	1E	00	00
61B0-	00	1C	06	1F	23	23	1E	00	00	62B0-	00	23	23	23	26	1C	18	00	63B0-	00	00	23	23	23	16	0C	00	00
61B8-	80	3F	18	0C	06	06	06	00	00	62B8-	00	2B	2B	2B	26	1C	18	00	63B8-	00	00	2B	2B	2B	2B	36	00	00
61C0-	00	1E	A3	1E	23	23	1E	00	00	62C0-	00	23	16	0C	0C	16	23	00	63C0-	00	00	23	16	0C	16	23	00	00
61C8-	00	1E	A3	23	3E	18	0C	00	00	62C8-	00	23	23	1E	0C	0C	1E	00	63C8-	00	00	33	23	23	16	0C	07	00
61D0-	00	00	06	06	00	06	06	00	00	62D0-	00	3F	18	0C	06	03	3F	00	63D0-	00	00	3F	18	0C	06	3F	00	00
61D8-	00	00	06	06	00	06	06	02	00	62D8-	10	08	36	11	21	59	26	00	63D8-	08	0C	3E	23	1F	03	3E	00	00
61E0-	00	24	12	09	12	24	00	00	00	62E0-	00	00	3E	03	03	03	3E	18	63E0-	04	0C	23	23	23	23	1E	00	00
61E8-	00	00	1E	00	9E	00	00	00	00	62E8-	10	08	36	1F	3F	7F	36	00	63E8-	84	8C	BE	A3	9F	83	BE	80	00
61F0-	00	09	12	24	12	09	00	00	00	62F0-	08	14	3E	23	23	33	2E	00	63F0-	08	14	3E	23	1F	03	3E	00	00
61F8-	00	1E	A2	10	08	00	08	00	00	62F8-	7F	00	00	00	00	00	00	00	63F8-	14	00	3E	23	1F	03	3E	00	00

Installation de FAIRE.F1
ou autre fonte de caractères, si vous le désirez.

Installation des sous-programmes et de la page de fonctions.

Adresses décimales des 23 lignes d'écran.

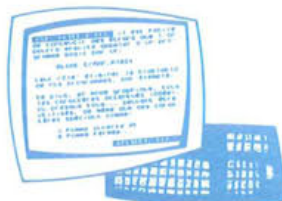
On ne sait jamais : il est toujours possible d'enfoncer la touche ESCAPE par mégarde !

Si vous vous plantez un jour dans un programme, après avoir saisi X données (évidemment importantes !) listez-le pour savoir où vous pouvez redémarrer puis tapez, sur une seule ligne :

POKE 51,0 : GOTO
(votre ligne)

```
870 D$ = CHR$(4) : PRINT D$"BLOAD"F$,A$6100" : CALL 247
      51 : RETURN
880 POKE - 16304,0 : POKE - 16302,0 : POKE - (16301 -
      K),0 : POKE - 16297,0 : POKE 230,K * 32 : CALL 24576
890 POKE 24659,234 : POKE 24660,234 : REM ON EFFACE (AVE
      C 17 ET 42,ON ECRIT SUR LES TEXTES OU TRAITES EXISTAN
      TS)
900 RETURN
910 :
920 REM *****
930 REM ***   DEBUT DE PROGRAMME   ***
940 REM *****
950 :
960 D$ = CHR$(13) + CHR$(4) : PRINT D$"BLOAD FAIRE.CO
      M,A$4000" : PRINT D$"BLOAD FAIRE.LM,A$6000" : PRINT D$
      "BLOAD FAIRE.MEVEC" : CALL 832
970 T$ = "" : FOR I = 1 TO 10 : T$ = T$ + "____" : NEXT
980 E$ = "0001024115212801408153616641792192010641192132
      0144815761704183219601104123213601488161617441872200
      0"
990 HOME : VTAB 22 : HTAB 5 : INVERSE : PRINT "1" :; NORMA
      L : PRINT " CLEAR PAGE1 " :; INVERSE : PRINT "2" :; N
      ORMAL : PRINT " SUITE NORMALE " :; GET H$:A% = VAL (
      H$)
1000 IF A% = 1 THEN POKE 230,32 : CALL 62450
1010 GOSUB 850 : HOME : RETURN
1020 :
1030 POKE 50,255 : TEXT : HOME : VTAB 22 : PRINT "RIEN N'E
      ST PERDU EN TAPANT GOTO 2000"
1040 END
1050 :
1060 REM *****
1070 REM ***   TRAITEMENT DES ERREURS   ***
1080 REM *****
1090 :
1100 NORMAL :ER = PEEK (222)
1110 GOSUB 460 : IF ER = 4 THEN PRINT "DISQUETTE PROTEGE
      E " :; GET R$ : PRINT "" : GOTO 730
1120 IF ER = 9 THEN PRINT "DISQUETTE COMPLETE " :; GET
      R$ : PRINT "" : GOTO 730
1130 IF ER = 10 THEN PRINT "FICHER VERROUILLE " :; GE
      T R$ : PRINT "" : GOTO 730
1140 IF ER = 6 THEN PRINT "PAS DE FICHER "F$" " :; GET
      R$ : PRINT "" : GOTO 730
1150 PRINT "PROBLEME IMPREVU " :; GET R$ : PRINT "" : GOTO
      730
2000 POKE 51,0 : HOME : GOSUB 820 : GOTO 100
```

Ce programme fonctionne convenablement sur les APPLE IIe et IIc. Il n'a pas été testé sur les autres modèles.



Utilisez donc le langage machine de votre Apple !

Lorsque nous déplaçons allégrement les 1024 octets représentant notre mémoire-écran, nous risquons fort de perdre le contrôle de nos périphériques.

En effet, dans ces 1024 octets, 8 * 8 sont réservés aux cartes d'interface desdits périphériques.

D'où l'utilité de mémoriser les valeurs (ô combien précieuses !) recelées par ces capricieux octets.

Deux solutions : le mini-programme en basic ci-après ou la minable routine en langage machine ci-contre (on sauve de \$340 à \$377 et on peut restituer de \$378 à \$3AF).

Un fait certain : la première solution a besoin de ses 2 petites secondes... l'autre est instantanée.

Quant à la mémoire : plus de 200 octets pour le basic (voyez ci-dessous !) et 112 pour le langage machine. Ça vaut la peine, non ?

0800- 00 6F 08 B8 08 53 41 28
0808- 31 2C 30 29 D0 31 31 34
0810- 34 3A 53 41 28 32 2C 30
0818- 29 D0 31 32 37 32 3A 53
0820- 41 28 30 2C 30 29 D0 31
0828- 34 30 30 3A 53 41 28 32
0830- 2C 30 29 D0 31 32 37 32
0838- 3A 53 41 28 35 2C 30
0840- D0 31 36 35 36 3A 53 41
0848- 28 36 2C 30 29 D0 31
0850- 38 34 3A 53 41 28 32 37
0858- 30 29 D0 31 39 31 38 3A
0860- 53 41 28 38 2C 31 38 3A
0868- 32 30 34 30 3A B1 00 9E
0870- 08 C2 0B B1 49 D0 31 C1
0878- 38 3A B1 4A D0 31 C1 38
0880- 3A 53 41 28 49 2C 4A 29
0888- D0 E2 28 53 41 28 49 2C
0890- 30 29 C8 4A C9 31 29 3A
0898- B2 3A 82 3A B1 00 CB 08
08A0- CC 0B B1 49 D0 31 C1 38
08A8- 3A B1 4A D0 31 C1 38 3A
08B0- B9 53 41 28 49 2C 30 29
08B8- C8 4A C9 31 2C 53 41 28
08C0- 49 2C 4A 29 3A 82 3A 82
08C8- 3A B1 00

SAUVE.MEVEC

OCCUPE 112 OCTETS EN MÉMOIRE

```

0340- A2 00 LDX £$00
0342- BD 78 04 LDA $0478,X
0345- 9D 00 03 STA $0300,X
0348- BD F8 04 LDA $04F8,X
034B- 9D 08 03 STA $0308,X
034E- BD 78 05 LDA $0578,X
0351- 9D 10 03 STA $0310,X
0354- BD F8 05 LDA $05F8,X
0357- 9D 18 03 STA $0318,X
035A- BD 78 06 LDA $0678,X
035D- 9D 20 03 STA $0320,X
0360- BD F8 06 LDA $06F8,X
0363- 9D 28 03 STA $0328,X
0366- BD 78 07 LDA $0778,X
0369- 9D 30 03 STA $0330,X
036C- BD F8 07 LDA $07F8,X
036F- 9D 38 03 STA $0338,X
0372- E8 INX
0373- E0 08 CPX £$08
0375- D0 CB BNE $0342
0377- 60 RTS
0378- A2 00 LDX £$00
037A- BD 00 03 LDA $0300,X
037D- 9D 78 04 STA $0478,X
0380- BD 08 03 LDA $0308,X
0383- 9D F8 04 STA $04F8,X
0386- BD 10 03 LDA $0310,X
0389- 9D 78 05 STA $0578,X
038C- BD 18 03 LDA $0318,X
038F- 9D F8 05 STA $05F8,X
0392- BD 20 03 LDA $0320,X
0395- 9D 78 06 STA $0678,X
0398- BD 28 03 LDA $0328,X
039B- 9D F8 06 STA $06F8,X
039E- BD 30 03 LDA $0330,X
03A1- 9D 78 07 STA $0778,X
03A4- BD 38 03 LDA $0338,X
03A7- 9D F8 07 STA $07F8,X
03AA- E8 INX
03AB- E0 08 CPX £$08
03AD- D0 CB BNE $037A
03AF- 60 RTS

```

```

3000 SA(1,0) = 1144:SA(2,0) = 1272:SA(3,0) = 1400:
      SA(4,0) = 1528:SA(5,0) = 1656:SA(6,0) = 1784:
      SA(7,0) = 1912:SA(8,0) = 2040: RETURN
3010 FOR I = 1 TO 8: FOR J = 1 TO 8:SA(I,J) = PE
      EK (SA(I,0) + J - 1): NEXT : NEXT : RETURN
3020 FOR I = 1 TO 8: FOR J = 1 TO 8: POKE SA(I,0)
      + J - 1,SA(I,J): NEXT : NEXT : RETURN

```

**PROGRAMME
EN BASIC**
avec utilisation
de variables.
(203 OCTETS)

Ecrire en page 2 (HGR2)

```

10 TEXT : NORMAL : SPEED= 255:
   PRINT CHR$(21): HOME
16 POKE - 16368,0: WAIT - 163
   84,128: PRINT
20 HGR2 : HCOLOR= 3
30 :
40 REM CADRE

50 HPLLOT 63,10 TO 199,10: HPLLOT
   TO 199,24: HPLLOT TO 63,24:
   HPLLOT TO 63,10
60 :
70 REM ABC DU BASIC

80 XO = 62:Y = 14
90 FOR I = 1 TO 12:X = XO + 10
   * I
100 ON I GOSUB 710,830,950,680,1
   070,1190,680,830,710,1310,143
   0,950
110 NEXT
120 :
130 REM APPLE & POMME

```

```

140 Y = 14:X = 10
150 GOSUB 210
160 X = 233: GOSUB 380
170 :
180 END
190 :
200 REM APPLE

```

```

210 A$ = ""
220 A$ = A$ + "*****"
230 A$ = A$ + "*"
240 A$ = A$ + "* *****"
250 A$ = A$ + "* *****"
260 A$ = A$ + "* *****"
270 A$ = A$ + "* *****"
280 A$ = A$ + "* *****"
290 A$ = A$ + "* *****"
300 A$ = A$ + "*****"
310 A$ = A$ + "* *****"
320 A$ = A$ + "* *****"
330 A$ = A$ + "*****"
340 :

```

Ce programme présente un avantage non négligeable (surtout à une époque aussi matérialiste que la nôtre) : il ne sert à rien.

Par contre, il montrera à celles et à ceux qui n'ont pas encore abordé cette question, comment il est possible d'écrire (lentement, mais sûrement), dans la page 2 — HGR2 — de l'Apple !

A partir de là, on peut tout imaginer, et même réinventer le graphisme de cette bonne vieille Pomme !

```

350 GOTO 540
360 :
370 REM POMME

380 A$ = ""
390 A$ = A$ + " ** "
400 A$ = A$ + " **** "
410 A$ = A$ + " **** "
420 A$ = A$ + " *** *** *** "
430 A$ = A$ + " *****"
440 A$ = A$ + "*****"
450 A$ = A$ + "*****"
460 A$ = A$ + "*****"
470 A$ = A$ + "*****"
480 A$ = A$ + "*****"
490 A$ = A$ + "*****"
500 A$ = A$ + "*****"
510 A$ = A$ + " **** **** "
520 :
530 REM ECRITURE APPLE & POMME

540 FOR H = 0 TO 12
550 FOR L = 1 TO 16
560 IF MID$(A$,L + H * 16,1) =
   "*" THEN HPLLOT X + L,Y + H
570 NEXT : NEXT : RETURN
580 :
590 REM ECRITURE DU TEXTE

600 S = FRE (0)
610 FOR H = 0 TO 7
620 FOR L = 1 TO 5
630 IF MID$(A$,L + H * 5,1) =
   "*" THEN HPLLOT X + L,Y + H
640 NEXT : NEXT
650 FOR BR = 1 TO 10:BZZZ = PEEK
   K (49200): NEXT : RETURN
660 :
670 REM ESPACE

680 FOR BR = 1 TO 10:BZZZ = PEEK
   K (49200): NEXT : RETURN
690 :

```

(Suite au verso)

POKEZ VOTRE PROGRAMME EN LANGAGE MACHINE

Parfois, il est intéressant de transformer en autant de Pokes les octets d'un court programme en langage machine.

Cela peut se faire en MODE DIRECT, très simplement, de la manière suivante (il faut évidemment connaître l'adresse décimale du programme et sa longueur) :

A = ADRESSE DÉBUT : B = FIN : P = NUMÉRO de la ligne de Pokes prévue.

```
] PRINT "]" ; P ; : FOR I = A TO B : PRINT "POKE" ; I ;
  "," PEEK(I) ; ":" ; : NEXT ... et RETURN
```

... ou, pour une ligne de DATA :

```
] PRINT "]" ; P ; "DATA" ; : FOR I = A TO B : PRINT
  PEEK(I) ; "," ; : NEXT ... et RETURN
```

Ecrire en page 2 (HGR2)



700 REM A

710 A\$ = ""

720 A\$ = A\$ + " * "

730 A\$ = A\$ + " * * "

740 A\$ = A\$ + " * * * "

750 A\$ = A\$ + " * * * * "

760 A\$ = A\$ + " * * * * * "

770 A\$ = A\$ + " * * * * * "

780 A\$ = A\$ + " * * * * * "

790 :

800 GOSUB 590: RETURN

810 :

820 REM B

830 A\$ = ""

840 A\$ = A\$ + " * * * * "

850 A\$ = A\$ + " * * * * * "

860 A\$ = A\$ + " * * * * * "

870 A\$ = A\$ + " * * * * * "

880 A\$ = A\$ + " * * * * * "

890 A\$ = A\$ + " * * * * * "

900 A\$ = A\$ + " * * * * * "

910 :

920 GOSUB 590: RETURN

930 :

940 REM C

950 A\$ = ""

960 A\$ = A\$ + " * * * * "

970 A\$ = A\$ + " * * * * * "

980 A\$ = A\$ + " * * * * * "

990 A\$ = A\$ + " * * * * * "

1000 A\$ = A\$ + " * * * * * "

1010 A\$ = A\$ + " * * * * * "

1020 A\$ = A\$ + " * * * * * "

1030 :

1040 GOSUB 590: RETURN

1050 :

1060 REM D

1070 A\$ = ""

1080 A\$ = A\$ + " * * * * * "

1090 A\$ = A\$ + " * * * * * "

1100 A\$ = A\$ + " * * * * * "

1110 A\$ = A\$ + " * * * * * "

1120 A\$ = A\$ + " * * * * * "

1130 A\$ = A\$ + " * * * * * "

1140 A\$ = A\$ + " * * * * * "

1150 :

1160 GOSUB 590: RETURN

1170 :

1180 REM U

1190 A\$ = ""

1200 A\$ = A\$ + " * * * * * "

1210 A\$ = A\$ + " * * * * * "

1220 A\$ = A\$ + " * * * * * "

1230 A\$ = A\$ + " * * * * * "

1240 A\$ = A\$ + " * * * * * "

1250 A\$ = A\$ + " * * * * * "

1260 A\$ = A\$ + " * * * * * "

1270 :

1280 GOSUB 590: RETURN

1290 :

1300 REM S

1310 A\$ = ""

1320 A\$ = A\$ + " * * * * * "

1330 A\$ = A\$ + " * * * * * "

1340 A\$ = A\$ + " * * * * * "

1350 A\$ = A\$ + " * * * * * "

1360 A\$ = A\$ + " * * * * * "

1370 A\$ = A\$ + " * * * * * "

1380 A\$ = A\$ + " * * * * * "

1390 :

1400 GOSUB 590: RETURN

1410 :

1420 REM I

1430 A\$ = ""

1440 A\$ = A\$ + " * * * * * "

1450 A\$ = A\$ + " * * * * * "

1460 A\$ = A\$ + " * * * * * "

1470 A\$ = A\$ + " * * * * * "

1480 A\$ = A\$ + " * * * * * "

1490 A\$ = A\$ + " * * * * * "

1500 A\$ = A\$ + " * * * * * "

1510 :

1520 GOSUB 590: RETURN

DES OCTETS EN DÉPLACEMENT



Ce petit programme autorise le transfert d'une zone mémoire dans une autre, pourvu que les deux ne se chevauchent pas... bien entendu !

Nous utilisons pour cela la routine FE2C (CALL — 468), du Moniteur Auto Start (en mémoire morte).

Elle déplace une zone préalablement délimitée par A1L—A1H et A2L—A2H, vers la zone d'accueil débutant en A4L—A4H.

Il faut savoir que ces adresses sont en PAGE-0, exactement en 3C 3D 3E 3F (60 61 62 63) et en 42 43 (66 67).

D'OÙ LE PROGRAMME SUIVANT :

SOIT À TRANSFÉRER UN ÉCRAN (1024-2000) À L'ADRESSE 16384 (\$4000)

10000 D = 1024 : Z = 60 : GOSUB 10020
: D = 2000 : Z = 62 : GOSUB
10020 : D = 16384 : Z = 66 :
GOSUB 10020

10010 POKE 768,160 : POKE 769,0 :
POKE 770,76 : POKE 771,44 :
POKE 772,254 : CALL 768 : END

10020 POKE Z, D-INT (D/256) * 256 :
POKE Z + 1, INT (D/256) :
RETURN

Pour rappeler la page, il suffira d'inverser les valeurs, comme ceci :

10000 D = 16384 : Z = 60 : GOSUB
10020 : D = 17360 : Z = 62 :
GOSUB 10020 : D = 1024 : Z = 66
: GOSUB 10020



ABC DU BASIC



Elémentaire, quand on le sait, mais terriblement compliqué quand on est complètement allergique au langage machine...

TRANSFERT D'ÉCRANS EN MÉMOIRE AUXILIAIRE

Quand on dispose d'une carte 80 COLONNES et de 64K de MÉMOIRE AUXILIAIRE, il est bon d'y stocker certaines pages d'informations.

Rien de plus facile, mais en passant non plus par la routine FE2C, mais bien par la routine C311 de l'Apple IIe.

ATTENTION ! LA RETENUE (CARRY) EST MISE A

- 1 pour ÉCRIRE en MEM AUXILIAIRE (SEC)
- 0 pour TRANSF en MEM PRINCIPALE (CLC)

Soit : 0300 — 38 (ou 18) SEC (ou CLC)

0301 — 20 11 C3 JSR \$C311

0304 — 60 RTS

VERS LA MÉMOIRE AUXILIAIRE

10000 POKE 769,32 : POKE 770,17 :
POKE 771,195 : POKE 772,96 :
REM ROUTINE \$C311

10010 POKE 60,00 : POKE 61,04 :
POKE 62,208 : POKE 63,07 :
POKE 66,00 : POKE 67,32 :
POKE 768,56 : REM CARRY = 1

VERS LA MÉMOIRE PRINCIPALE

10020 POKE 60,00 : POKE 61,32 :
POKE 62,208 : POKE 63,35 :
POKE 66,35 : POKE 67,04 :
POKE 768,24 : REM CARRY = 0

Vous pouvez constater que, comme dans la page précédente, on délimite les zones de transfert et d'accueil en pokant les adresses en A1L-A1H, A2L-A2H (60 à 63) et A4L-A4H (66-67).

JEU

vous avez emprunté, mais combien devez-vous payer ?

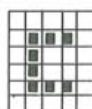
Ce court programme fonctionne avec un basic compris par la plupart des machines, même celles de poche... mais il comporte une erreur. A vous de la découvrir. (réponse page 62).

```

10 TEXT : PRINT CHR$(21): HOME
20 VTAB 12: GOSUB 230: HOME
30 INPUT "MONTANT DE L'EMPRUNT ";E
40 INPUT "NOMBRE DE VERSEMENTS ANNUELS
   ";V
50 INPUT "TAUX ANNUEL INTERET% ";I
60 PRINT
70 TR$ = "-----"
80 I = I * .01:C = E:AC = E:RA = E / V
90 PRINT "RANG INTERETS MONTANTS
   "
100 PRINT TR$
110 FOR K = 1 TO V
120 IN = AC * I:MO = RA + IN:C = C + IN:
   AC = AC + RA
130 V$ = " "; IF K > 9 THEN V$ = ""
140 IN = INT (IN * 100 + .5) / 100:US$
   = STR$(IN): GOSUB 210:H1 = U
150 MO = INT (MO * 100 + .5) / 100:US$
   = STR$(MO): GOSUB 210:H2 = U
160 PRINT V$;K;: HTAB 14 - H1: PRINT IN
   ;: HTAB 27 - H2: PRINT MO
170 NEXT K
180 PRINT TR$
190 PRINT "TOTAL REMBOURSEMENTS ";C
200 END
210 FOR U = 1 TO LEN (US$): IF MID$ (
   US$,U,1) = "." THEN RETURN
220 NEXT U: RETURN
230 PRINT "Un emprunt (E) est remboursé
   en (V) an- nuités."
240 PRINT "Le montant de chaque annuité
   est égal à:"
250 PRINT " - E/V + Montant des intérêt
   ts (I%) dus sur le capital non re
   mboursé."
260 VTAB 22: PRINT " PRESSER UNE TOUCHE
   POUR LA SUITE ";: GET H$: PRINT : R
   ETURN

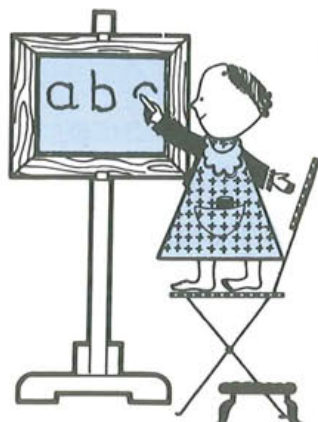
```

Une petite révision d'anglais



'EST vraiment la première leçon, mais rien ne vous interdit, en modifiant les lignes de DATA, tout en en respectant approximativement la forme, d'écrire plusieurs

suites. Un bon conseil : proposez cet exercice à l'une des personnes de votre entourage. Je vous promets des réactions surprenantes... et l'ordinateur ne se fatigue jamais !



```

10 TEXT : SPEED= 255: NORMAL : PRINT CHR$(21): HOME
20 HTAB 7: INVERSE : PRINT " PETITE REVISION D'ANGLAIS
  ": NORMAL
30 V = 4: Y = V: D$ = CHR$(7): TR$ = "": FOR I = 1 TO 10
  : TR$ = TR$ + "----": NEXT : PRINT TR$: POKE 34,2
40 GOTO 240
50 :
60 REM *****
70 REM ****  PETITES ROUTINES  ****
80 REM *****
90 :
100 FOR ZZ = 1 TO 5: BZZZ = PEEK (49200): NEXT ZZ: RETURN
110 FOR I = 1 TO 10: GOSUB 100: NEXT : RETURN
120 VTAB 23: HTAB 8: INVERSE : PRINT " PRESSEZ UNE TOUC
  HE SVP ": NORMAL
130 VTAB 22: CALL - 958: PRINT : HTAB 3: IF ER = 0 THE
  N 160
140 IF ER = 1 THEN PRINT "1 MAUVAISE REPONSE - "; GOT
  O 160
150 INVERSE : PRINT ER;: NORMAL : PRINT " MAUVAISES REP
  ONSES - ";
160 INVERSE : PRINT X;: NORMAL : PRINT " QUESTIONS": FO
  R I = 1 TO 1500: NEXT : RETURN
170 VTAB 23: HTAB 11: INVERSE : PRINT "PRESSEZ UNE TOUC
  HE": NORMAL
180 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127
  : POKE - 16368,0: RETURN
190 :
200 REM *****
210 REM ****  REVISION D'ANGLAIS  ****
220 REM *****
230 :
240 PRINT : PRINT : PRINT : PRINT "COMMENCONS PAR DES M
  OTS ANGLAIS DONT IL VOUS FAUT DONNER LA TRADUCTION..
  ."
250 GOSUB 110: GOSUB 110: HOME
260 READ A$: Y = Y + 1: X = X + 1
270 IF X = 18 OR X = 27 OR X = 33 THEN GOSUB 170: GOSU
  B 130: Y = V: HOME : Y = Y + 2: IF X = 33 THEN 420
  
```

← PRINT CHR\$(21) vous oblige à travailler en 40 colonnes. SPEED = 255 n'est utile que si un autre programme avait réduit la vitesse d'affichage.

← Pour un bruit bref : X = PEEK (49200) ... utilisé ici en boucle.

← Attente d'un caractère dont on peut connaître la valeur par un C = PEEK (-16384).

Attention ! les CTRL sont acceptés, et un CTRL-C provoquera donc un BREAK. Pour empêcher l'utilisation du CTRL-C (BREAK), il faudrait par exemple écrire :

```

180 CALL-198 : GET W$:
  IF ASC (W$) > 31
  THEN RETURN
185 GOTO 180
  
```

CALL-868 vide la ligne d'écran, et seulement elle.

On est en Anglais, pas vrai ?

Ce système d'affichage est rapide, mais le programme agaçant à écrire. Un conseil : tapez 4 fois ceci «-.....-», puis remplissez !

Il n'est pas interdit de compléter le programme par des appréciations (flatteuses ou non) sur le score du participant. On peut aussi lui demander son prénom (au début), puis lui lancer un «Bravo ! Toto» à la fin. Mieux : si vous mémorisez score et prénom, vous le surprendrez, s'il rejoue... en comparant son nouveau résultat à l'ancien.

```
280 IF X = 18 THEN PRINT "  DONNEZ MAINTENANT LE MOT
    ANGLAIS !"
290 IF X = 27 THEN PRINT "  DONNEZ MAINTENANT LE MOT
    FRANCAIS!"
300 HTAB 1: VTAB Y: INVERSE : PRINT X;: NORMAL
310 HTAB 4: PRINT A$: GOSUB 100
320 HTAB 17: VTAB Y: INPUT B$: IF B$ = "" THEN 320
330 READ C$: HTAB 17: VTAB Y: CALL - 868
340 IF B$ < > C$ THEN : INVERSE : PRINT "?": NORMAL : E
    R = ER + 1: GOTO 360
350 PRINT C$
360 HTAB 25: VTAB Y: PRINT C$
370 HTAB 37: VTAB Y
380 IF B$ = C$ THEN PRINT "YES": GOTO 400
390 PRINT "NO" + D$
400 IF X = 32 THEN X = 33: GOTO 270
410 GOTO 260
420 HOME :Y = V + 1:Z = Y: HTAB (1): VTAB (V)
430 VTAB Y + 1: HTAB 1:T$ = "                QUE          SIGNI
    FIE CETTE      PHRASE ?          ": FOR I = 1 TO L
    EN (T$) STEP 10: INVERSE : PRINT MID$ (T$,I,10): NE
    XT : NORMAL
440 POKE 32,12: HTAB 13: PRINT : POKE 33,27: POKE 35,22
450 READ A$:Z = Z + 2: HTAB 1: VTAB Z - 1: PRINT A$: GO
    SUB 100
460 Z = Z + 1: READ A$: GOSUB 110
470 HTAB 4: PRINT "--> ";A$
480 IF Z < 19 THEN 450
490 TEXT : GOSUB 120: HOME : VTAB 12
500 T$ = "C'EST TOUT POURCETTE FOIS!  A LA PROCHAINE!"
    : FOR I = 1 TO LEN (T$) STEP 15: HTAB 12: PRINT MI
    D$ (T$,I,15): PRINT : NEXT
510 HTAB 6: VTAB 23: INVERSE : PRINT " <RETURN> POUR RE
    COMMENCER"
520 GOSUB 180: IF PEEK ( - 16384) = 13 THEN RUN
530 NORMAL : HOME : END
540 :
550 REM *****
560 REM **** QUESTION/REPONSES ****
570 REM *****
580 :
590 DATA A/AN,UN/UNE,BOOK,LIVRE,CHAIR,CHAISE,DESK,BUREA
    U,GOOD,BON,HE,IL,IT,IL/ELLE,MAGAZINE,REVUE,MORNING,M
    ATIN,NEW,NOUVEAU,NEWSPAPER,JOURNAL,NOTEBOOK,CAHIER,O
    N,SUR,PEN,STYLO,PENCIL,CRAYON,TABLE,TABLE,TEACHER,PR
    OFESSEUR
600 DATA CE/CET/CELLE,THAT,CELUI-LA,THAT,LE/LE/LES,THE
    ,CE/CET/CELLE,THIS,CELUI-CI,THIS,QUEL?/QUELLE?,WHAT?
    ,QU'EST-CE QUE?,WHAT?,OU,WHERE,QUI?,WHO?
610 DATA TO BE,ETRE,TO DO,FAIRE,TO READ,LIRE,TO TALK,PA
    RLER,TI WALK,MARCHER,TO WRITE,ECRIRE
620 DATA WHAT'S THIS?,QU'EST-CE QUE C'EST?,WHAT'S HE
    DOING?,QUE FAIT-IL?,HE'S READING,IL LIT,WHERE'S THE
    PEN?,OU EST LE STYLO?,WHO'S HE?,QUI EST-CE?
```



De nouvelles perspectives pour vos histogrammes



IMPLE, mais efficace (et fort agréable à utiliser), ce programme vous aidera à donner un volume inhabituel à vos chiffres les plus rébarbatifs.

Vous l'adopterez sans aucune difficulté à vos propres besoins... à moins qu'il ne fasse votre bonheur tel quel !

```

10 LOMEM: 25600
20 HGR :C = 2: HCOLOR= 3: TEXT : HOME
30 D$ = CHR$ (13) + CHR$ (4):G$ = CHR$ (34)
40 GOSUB 1160
50 DIM A(12),V(12),I(12)
60 B = 10: REM  LARGEUR DES COLONNES
70 A = PEEK (49233)
80 GOTO 870
90 :
100 REM *****
110 REM *****  PERSPECTIVE  *****
120 REM *****
130 :
140 HPLLOT 1,1 TO 1,140 TO 279,140 TO 279,1 TO 1,1
150 FOR I = 1 TO N
160 C = ABS ( - 5 + C): HCOLOR= C
170 A = V(I)
180 X = B + (I - 1) * B
190 Y = 140 - B * A
200 FOR J = 0 TO B: HPLLOT X + J,140 TO X + J,Y: NEXT
    J
210 HCOLOR= 3: REM  CONTOURS POUR HCOLOR=2
220 HPLLOT X,Y TO X + B,Y TO X + B,140 TO X,140 TO X,
    Y
230 REM  ARRIERE
240 HPLLOT X,Y TO X + 5,Y - 5 TO X + B + 5,Y - 5 TO X
    + B + 5,140: HPLLOT X + B,Y TO X + B + 5,Y - 5
250 NEXT I
260 POKE 49232,0
270 CALL - 198: POKE - 16368,0: WAIT - 16384,128,
    127: POKE - 16368,0: PRINT :H = 1: IF 0 = 10 THE
    N 910
280 GOTO 870
  
```

← **LOMEM** protège votre table de caractères, le programme de transfert et la page graphique.

← **B=10** a été choisi parce que donnant le meilleur résultat. **A=PEEK (49233)** permet de retrouver la page TEXT.

← Formule donnant alternativement à **C** une valeur de 2 (vert) ou 3 (blanc).

← Quand **HCOLOR=VERT**, on retrace les contours de la figure.

← Perspective.

← **POKE 49232,0=HGR**.

On ne peut pas saisir des valeurs telles que la plus grande soit égale à plus de 10 fois la plus petite.

Avec cette formule, pas d'Extra Ignored à redouter !

Si A\$ = ZÉRO, terminé ! Pour corriger c'est l'astérisque !

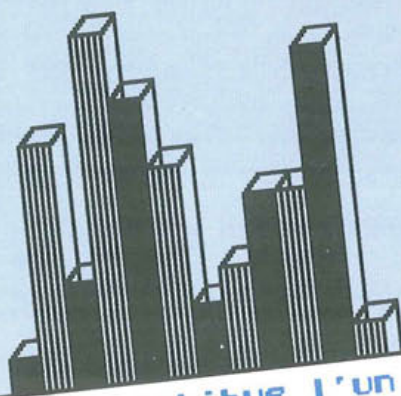
On va voir si les valeurs conviennent ou pas...

Si vous désirez que votre titre commence par un blanc, ou comporte une virgule, ouvrez d'abord les guillemets, puis tapez votre texte. Finissez par un espace si vous aviez commencé ainsi.

```
310 REM *****
320 REM ***** PARAMETRES *****
330 REM *****
340 :
350 TEXT : HOME : HTAB 11: INVERSE : PRINT " SAISIE
    DES VALEURS ": NORMAL
360 B$ = "": FOR I = 1 TO 10: B$ = B$ + "____": NEXT I
    PRINT B$
370 PRINT "Vous ne pouvez pas saisir plus de 12 va-
    leurs. Pour stopper, tapez <0> et un <*> pour corri-
    ger une valeur."
380 PRINT B$: IF N < > 0 THEN E = N: N = 0
390 N = N + 1
400 VTAB N + 8: HTAB 1: V$ = " ": IF N > 9 THEN V$ =
    ""
410 PRINT V$N". VALEUR -> "A(N)": HTAB 15: CALL -
    657: A$ = "": FOR F = 512 TO 767: IF PEEK (F) <
    > 141 THEN A$ = A$ + CHR$ ( PEEK (F) - 128): NEX
    T F
420 IF A$ = "" AND A(N) < > 0 THEN VTAB N + 8: HTA
    B 15: PRINT A(N): GOTO 490
430 IF A$ = "0" THEN N = N - 1: GOTO 500
440 IF A$ = "*" AND N > 1 THEN N = N - 1: GOTO 400
450 L = LEN (A$): IF L > 11 THEN 400
460 FOR F = 1 TO LEN (A$): G = ASC ( MID$ (A$, F, 1))
    : IF G < 48 OR G > 57 THEN 400
470 NEXT F
480 A(N) = VAL (A$)
490 IF N < 12 THEN 390
500 H = 0: GOSUB 760
510 :
520 REM *****
530 REM ***** TEXTE *****
540 REM *****
550 :
560 VTAB 22: HTAB 1: CALL - 958: PRINT "TITRE .....
    .....": IF E$ < > "" THEN
    HTAB 7: PRINT G$ + E$:
570 HTAB 7: INPUT "": H$: IF H$ = "" THEN 590
580 H = 0: E$ = H$
590 IF 0 = 5 THEN 910
600 D = 1: GOTO 870
610 TEXT : HOME
620 VTAB 2: HTAB (41 - LEN (E$)) / 2: INVERSE : PRI
    NT E$: NORMAL
630 A = LEN ( STR$ (A)) + 5
640 FOR I = 1 TO N: VTAB 5 + I: V$ = " ": IF I > 9 TH
    EN V$ = ""
650 HTAB 33 - A: PRINT V$I". ";
660 HTAB 31 - LEN ( STR$ (A(I))): PRINT A(I)" "": H$
    = STR$ (I(I)): K = LEN (H$): IF I(I) = INT (I(
    I)) THEN 690
```

(Suite page 32)

HISTOGRAMME INUTILE



1.	12345	1.75	%
2.	78961	11.21	%
3.	34567	4.91	%
4.	110000	15.62	%
5.	90000	12.78	%
6.	70000	9.94	%
7.	30000	4.26	%
8.	40000	5.68	%
9.	60000	8.52	%
10.	65000	9.23	%
11.	100000	14.2	%
12.	13000	1.84	%

Ceci constitue l'un des exemples géniaux d'HISTOGRAMME ne servant vraiment à rien

```

670 IF MID$(H$,K - 1,1) = "." THEN K = K + 1: GOTO
    700
680 IF MID$(H$,K - 2,1) = "." THEN 700
690 K = K + 3
700 HTAB 38 - K: PRINT I(I);: HTAB 39: PRINT "%": NE
    XT
710 VTAB 19: HTAB 1: CALL - 958: PRINT C$
720 VTAB 22: PRINT "Autre commentaire O/N ";: GET H
    $: IF H$ = "" OR H$ = "N" THEN 740
730 VTAB 19: HTAB 1: CALL - 198: CALL - 657: C$ = "
    ": FOR F = 512 TO 767: IF PEEK (F) < > 141 THEN
    C$ = C$ + CHR$( PEEK (F) - 128): NEXT F
740 VTAB 19: HTAB 1: CALL - 958: PRINT C$: H = 0: IF
    0 = 6 THEN 870
750 GOSUB 1180: GOTO 140
760 A = 0: S = 0: FOR I = 1 TO N: IF A(I) > (A) THEN A
    = A(I)
770 S = S + A(I): NEXT
780 M = A / 10
790 FOR I = 1 TO N: V(I) = INT (A(I) / M): IF V(I) <
    1 THEN VTAB 22: HTAB 1: PRINT "LA VALEUR NUMERO
    "I" EST TROP PETITE": N = I: GOTO 400
800 I(I) = INT (A(I) * 10000 / S) / 100
810 NEXT: RETURN
820:
830 REM *****
840 REM ***** MENU *****
850 REM *****
860:
870 T$ = " / HISTOGRAMME /
    ": HOME: INVERSE: FOR I = 1 TO 51 STEP
    17: HTAB 12: PRINT MID$(T$,I,17): NEXT
880 NORMAL
    
```

Vous obtiendrez des histogrammes tels que celui-ci, mais je vous conseille des titres et commentaires un peu plus sérieux !

← Tant pendant la correction des données que lors de la saisie du titre ou du commentaire, un simple RETURN confirme la donnée ou le texte déjà en mémoire.

← On profite de cette routine pour calculer le pourcentage.

← Chaque portion de T\$ comporte 17 caractères ou espaces. Qu'on se le dise !

La page zéro de l'Apple IIe[®]

QUELQUES ADRESSES UTILISABLES N'oubliez pas que certains logiciels font largement appel à ces adresses.	1	\$06 06	\$07 07	\$08 08	\$09 09				
	2	\$18 24	\$19 25	\$1A 26	\$1B 27	\$1C 28	\$1D 29	\$1E 30	\$1F 31
	3	\$4A 74	\$4B 75	\$4C 76	\$4D 77	<u>Avec prudence</u>			
	4	\$CE 206	\$CF 207	\$D6 214	\$D7 215				
	5	\$EB 235	\$EC 236	\$ED 237	\$EE 238	\$EF 239			
	6	\$F9 249	\$FA 250	\$FB 251	\$FC 252	\$FD 253	\$FE 254	\$FF 255	

LA FENÊTRE

\$20 32 POKE 32,0 = COLONNE 1
POKE 32,3 = COLONNE 4

\$21 33 POKE 33,10 = 10 colonnes DE TEXTE
Par exemple : POKE 32,5 : POKE 33,10 permet de limiter l'affichage aux colonnes 6 à 15 de l'écran.

Important : Le total ne doit jamais dépasser 40. Ainsi POKE 32,5 : POKE 33,36 provoquerait un désastre !

\$22 34 POKE 34,0 = LIGNE 1
POKE 34,5 = LIGNE 6, et on n'écrit plus sur les 5 lignes précédentes qu'un HOME n'efface pas. Retour à la normale par POKE 34,0 ou TEXT.

\$23 35 POKE 35,10 limite la hauteur de la fenêtre aux lignes 1 à 10. Si l'on fait un POKE 34,3, cette même fenêtre ne comporte plus que les lignes 4 à 10.

VALEURS MIN. et MAX.	32,0 — 32,39
	33,1 — 33,40
	34,0 — 34,23
	35,1 — 35,24

LE CURSEUR

\$24 36 POSITION HORIZONTALE DU CURSEUR
= (PEEK (36)) + 1

POKE 36,9 placera le curseur sur la colonne 10.
POKE 36,49 (avec une carte 80 COLONNES) placera le curseur sur la colonne 50.

Dans ce cas, préférer POKE 1403,49

\$25 37 POSITION VERTICALE DU CURSEUR
= (PEEK (37)) + 1

POKE 37,8 place le curseur sur la ligne 9.

Attention ! après un POKE 32,5 par exemple, POKE 36,10 place le curseur à la colonne 16, mais seulement s'il y a déjà eu un PRINT.

Exemples :

10	POKE 32,5 : POKE 33,35
20	POKE 36,10 : PRINT "T"

• "T" est affiché à la 11^e colonne.

10	POKE 32,5 : POKE 33,35
15	PRINT
20	POKE 36,10 : PRINT "T"

• "T" est affiché à la 16^e colonne.

FICHIERS PROGRAMMES

Ce sont les plus utilisés, et les mieux connus.

INSTRUCTIONS

SAVE
LOAD
RUN

MODE DIRECT

SAVE TOTO (,D1,S6)
LOAD TOTO (,D1,S6)
RUN TOTO (,D1,S6)

DANS UN PROGRAMME

PRINT CHR\$(4) ; "SAVE TOTO"
PRINT CHR\$(4) ; "LOAD TOTO"
RUN TOTO (,D1,S6)

Les mentions "D1,S6" (où 1 est le numéro du DRIVE et 6 le numéro du SLOT) ne sont indispensables que si l'on doit passer d'un DRIVE à un autre ou d'un SLOT à un autre. Sinon, l'instruction choisit, par défaut, les numéros en cours d'utilisation. On peut aussi indiquer le volume.

Exemple : SAVE TOTO,D1,S6,V1, ou PRINT CHR\$(4) ; "SAVE TOTO,D1,S6,V1".

FICHIERS EXEC

Fichiers de texte qu'APPLESOFT lit et tente d'exécuter.

INSTRUCTIONS

EXEC

MODE DIRECT

EXEC TOTO (,D1,S6)

DANS UN PROGRAMME

PRINT CHR\$(4) ; "EXEC TOTO"

EXEMPLE DE PROGRAMME EXEC PERMETTANT DE CONNAÎTRE L'ADRESSE ET LA LONGUEUR D'UN FICHIER BINAIRE QUI VIENT D'ÊTRE CHARGÉ.

```
10 D$ = CHR$(4) : G$ = CHR$(34)
20 PRINT D$ ; "OPEN ADR" : POKE 33,30 : PRINT D$ ; "WRITE ADR"
30 PRINT "HOME"
40 PRINT "PRINT : PRINT" ; G$ ; "A" ; G$ "PEEK(43634) + PEEK(43635) * 256" ; G$ ; "L" ; G$ "PEEK(43616) + PEEK(43617) * 256"
50 PRINT D$ ; "CLOSE ADR"
60 POKE 33,40 et RUN
```

EXÉCUTION :

EXEC ADR ...
... et le résultat apparaît : A768, L80, par exemple.

PROGRAMME EN EXEC

Vous pouvez mettre un programme (ou une partie de celui-ci) en fichier EXEC, puis le récupérer... ou l'insérer dans un autre.

CRÉATION

Une seule ligne de programme suffit :
60000 D\$ = CHR\$(4) : PRINT : PRINT D\$"OPEN LISTE" : POKE 33,30 : PRINT D\$"WRITE LISTE" : LIST 0,59999 : PRINT D\$"CLOSE LISTE" : POKE 33,40 : END

Taper RUN 60000. (Pour sauvegarder les lignes 30 à 70, mettre "LIST30,70").

RÉCUPÉRATION

EXEC LISTE (ou autre titre choisi par vous).

TRANSFORMER UN FICHIER EXEC EN X VARIABLES

```
10 D$ = CHR$(4) : HOME : DIM L$(250)
20 ONERR GOTO 80
30 PRINT D$"OPEN LISTE" : N = 0
40 PRINT D$"READ LISTE, R" : N : GET R$
50 IF ASC(R$) = 13 THEN L$(L) = L$ : L$ = "" : L = L + 1 : GOTO 70
60 L$ = L$ + R$
70 N = N + 1 : GOTO 40
80 POKE 216,0 : PRINT D$"CLOSE"
90 FOR I = 1 TO L : PRINT L$(I) : NEXT
```

Et maintenant, 9 fois 25 caractères... pour afficher le MENU que vous pouvez admirer en bas de la page.

Je sais : ce ne sont pas ceux de l'APPLE, mais ils appartiennent à FAIRE.F1... du programme FAIRE.ECRAN (page 22).

```

890 T$ = "1. Saisir des valeurs      2. Corriger des va
      leurs 3. Lire un fichier      4. Enregistrer va
      leurs 5. Corriger un titre    6. Autres commen
      taires 7. Voir l'histogramme  8. Imprimer l'h
      istogramme 9. Fin de travail  "
900 FOR I = 1 TO 225 STEP 25: HTAB 8: VTAB 6 + (I +
      24) / 25: PRINT MID$(T$,I,25): NEXT
910 TEXT : VTAB 17: HTAB 8: PRINT "
      _____": CALL - 958
920 VTAB 19: HTAB 8: PRINT "Option choisie ? "; GE
      T O$:O = ASC (O$) - 48: IF O < 1 OR O > 9 THEN 9
      20
930 IF H = 0 AND O = 8 THEN R$ = "PAS D'HISTOGRAMME
      PAGE 1": GOSUB 980: GOTO 920
940 IF D = 0 AND (O = 2 OR O = 4 OR O = 6 OR O = 7)
      THEN R$ = "AUCUNE VALEUR EN MEMOIRE": GOSUB 980:
      GOTO 920
950 IF H = 1 AND O = 7 THEN O = 10
960 ON O GOTO 350,350,1040,1040,560,710,610,1250,970
      ,260
970 TEXT : HOME : END
980 VTAB 22: HTAB 1: CALL - 958: HTAB 8: PRINT R$:
      RETURN
990 :
1000 REM *****
1010 REM ***** FICHER *****
1020 REM *****
1030 :
1040 POKE 34,21: HOME : VTAB 22: HTAB 8: INPUT "TITRE
      -> ";F$: IF F$ = "" THEN 910
1050 ONERR GOTO 1320
1060 VTAB 22: IF O = 4 THEN 1080
  
```

LE MINI-PROGRAMME

ci-après transfère votre page TEXT en HGR, avec utilisation de la table de caractères chargée à l'adresse \$6100.

*6000.6074

6000-	A6	22	8A	85	1C	20	24	FC
6008-	A0	00	A5	28	85	2A	A5	29
6010-	29	03	05	E6	85	2B	B1	28
6018-	C9	20	B0	05	69	C0	CE	52
6020-	60	C9	60	B0	05	69	80	CE
6028-	52	60	C9	80	B0	05	69	40
6030-	CE	52	60	C9	A0	90	31	29
6038-	7F	18	2A	2A	2A	48	2A	29
6040-	03	09	60	8D	50	60	68	29
6048-	F8	8D	4F	60	A2	00	BD	00
6050-	60	49	00	51	2A	91	2A	18
6058-	A5	2B	69	04	85	2B	E8	E0
6060-	08	90	EB	A9	00	8D	52	60
6068-	C8	C4	21	90	9D	A6	1C	E8
6070-	E4	23	90	8E	60			

HISTOGRAMME

1. Saisir des valeurs
2. Corriger des valeurs
3. Lire un fichier
4. Enregistrer valeurs
5. Corriger un titre
6. Autres commentaires
7. Voir l'histogramme
8. Imprimer l'histogramme
9. Fin de travail

Option choisie ?

Histogrammes (suite et fin)

```

1070 PRINT D$"OPEN" F$: PRINT D$"READ" F$: INPUT N: FOR
      I = 1 TO N: INPUT A(I): NEXT : INPUT E$: INPUT C
      $: PRINT D$"CLOSE" F$: D = 1: H = 0: GOSUB 760: GOTO
      1090
1080 PRINT D$"OPEN" F$: PRINT D$"WRITE" F$: PRINT N: FO
      R I = 1 TO N: PRINT A(I): NEXT : PRINT G$ + E$: P
      RINT G$ + C$: PRINT D$"CLOSE" F$: GOTO 1090
1090 ONERR GOTO 870
1100 GOTO 910
1110 :
1120 REM *****
1130 REM ***** TEXTE EN HGR *****
1140 REM *****
1150 :
1160 PRINT : PRINT D$"BLOAD HISTO.LM"
1170 PRINT : PRINT CHR$(13) + CHR$(4)"BLOAD FAIRE
      .F1,A$6100": RETURN
1180 POKE 24659,234: POKE 24660,234: CALL 24576
1190 RETURN
1200 :
1210 REM *****
1220 REM ***** EDITION GRAPH *****
1230 REM *****
1240 :
1250 POKE 49232,0
1260 PRINT D$"PR£1": PRINT CHR$(9)"G": PRINT D$"PR£
      0": GOTO 870
1270 :
1280 REM *****
1290 REM ***** TRAITEMENT ERREURS *****
1300 REM *****
1310 :
1320 P = PEEK(222)
1330 IF P = 5 THEN R$ = "PAS DE DONNEES " + F$
1340 IF P = 6 THEN R$ = "PAS DE FICHER " + F$
1350 POKE 216,0: GOSUB 980: GOTO 920

```

Le programme ne teste pas la présence, sur la disquette, d'un fichier portant le même nom. Il y a donc risque d'écraser des données déjà enregistrées. A vous de vous casser la tête sur ce problème !

← HISTO.LM, c'est la même chose que la partie \$6000 — \$6074 de FAIRE.LM (page 20). Ne vous amusez pas à le retaper...

← HGR (nullement obligatoire pour imprimer).

Veillez consulter la notice concernant votre imprimante et éventuellement celle de votre interface.

← On pourrait en rajouter, bien sûr, mais en l'occurrence ce serait superflu...

Un moyen de savoir si un fichier existe

Tenez ! il pourrait être inclus dans le programme «HISTOGRAMMES», par exemple.

```

10 HOME : D$ = CHR$(13) + CHR$(4)
15 INPUT "FICHER" : FI$: IF FI$ = "" THEN END
20 GOSUB 10000 : GOTO 15
10000 ONERR GOTO 10025
10005 PRINT D$"OPEN" FI$: PRINT D$"READ" FI$: INPUT RR : : PRINT
      D$"CLOSE" FI$
10010 PRINT "ATTENTION ! LE FICHER " FI$ " EXISTE !" : REM ETC.
10015 ONERR GOTO 15 : REM VOTRE LIGNE HABITUELLE
10020 RETURN : REM EVENTUELLEMENT
10025 P = PEEK(222) : POKE 216,0 : IF P = 5 THEN PRINT D$"DELETE" FI$
      : GOTO 10015
10030 PRINT "AUTRE PROBLÈME"

```

INVERSION DE HGR1 ou HGR2

Taper le programme suivant le processus habituel (CALL-151, puis 300 : A5 06, etc.). Sauver par un :

- 1 BSAVE INVER.1, A\$300, L\$50
- 2 BSAVE INVER.2, A\$300, L\$41 (celui-ci pourra être chargé à n'importe quelle adresse).

1

```

0300- A5 06 LDA $06
0302- C9 02 CMP £$02
0304- F0 14 BEQ $031A
0306- 48 PHA
0307- A9 00 LDA £$00
0309- 8D 54 C0 STA $C054
030C- A9 20 LDA £$20
030E- 8D 36 03 STA $0336
0311- 2A ROL
0312- 8D 4C 03 STA $034C
0315- 68 PLA
0316- C9 01 CMP £$01
0318- F0 02 BEQ $031C
031A- D0 33 BNE $034F
031C- A9 00 LDA £$00
031E- 8D 55 C0 STA $C055
0321- A9 40 LDA £$40
0323- 8D 36 03 STA $0336
0326- 2A ROL
0327- 8D 4C 03 STA $034C
032A- A9 00 LDA £$00
032C- 8D 50 C0 STA $C050
032F- 8D 52 C0 STA $C052
0332- 8D 57 C0 STA $C057
0335- A9 20 LDA £$20
0337- 85 1C STA $1C
0339- A9 00 LDA £$00
033B- 85 1B STA $1B
033D- A8 TAY
033E- B1 1B LDA ($1B),Y
0340- 49 FF EOR £$FF
0342- 91 1B STA ($1B),Y
0344- C8 INY
0345- D0 F7 BNE $033E
0347- E6 1C INC $1C
0349- A5 1C LDA $1C
034B- C9 40 CMP £$40
034D- D0 EF BNE $033E
034F- 60 RTS
    
```

2

```

0300- A5 06 LDA $06
0302- C9 00 CMP £$00
0304- D0 01 BNE $0307
0306- 60 RTS
0307- C9 03 CMP £$03
0309- 90 01 BCC $030C
030B- 60 RTS
030C- 0A ASL
030D- 0A ASL
030E- 0A ASL
030F- 0A ASL
0310- 0A ASL
0311- 85 07 STA $07
0313- 0A ASL
0314- 85 08 STA $08
0316- A6 06 LDX $06
0318- A9 00 LDA £$00
031A- 9D 53 C0 STA $C053,X
031D- 8D 50 C0 STA $C050
0320- 8D 52 C0 STA $C052
0323- 8D 57 C0 STA $C057
0326- A5 07 LDA $07
0328- 85 1C STA $1C
032A- A9 00 LDA £$00
032C- 85 1B STA $1B
032E- A8 TAY
032F- B1 1B LDA ($1B),Y
0331- 49 FF EOR £$FF
0333- 91 1B STA ($1B),Y
0335- C8 INY
0336- D0 F7 BNE $032F
0338- E6 1C INC $1C
033A- A5 1C LDA $1C
033C- C5 08 CMP $08
033E- D0 EF BNE $032F
0340- 60 RTS
    
```

(Adapté de Beagle Bros)

VERSION RELOGEABLE

TAPER

POKE 6,1 pour Page 1
POKE 6,2 pour Page 2

ENSUITE

CALL 768 INVERSE, un autre rétablit, et ainsi de suite.

NOTA

Le programme écrase les adresses 7 et 8.

VERSION NON RELOGEABLE

Fonctionnement identique, mais on n'utilise que l'adresse 6.

Amortissement

Quelle somme (AM) amortira, chaque année, la personne ayant contracté un emprunt de (ME) Francs, aux taux de (IN) pour cent, remboursable en (AN) annuités ? On sait que le montant des premières annuités est sérieusement entamé par le coût des intérêts, celui-ci diminuant au fur et à mesure que l'on rembourse le capital.

ME = Montant de l'Emprunt
IN = Taux de l'INtérêt annuel.
NA = Nombre d'Annuités.
AM = AMortissement.

L'affichage, tel qu'il a été programmé sur APPLE, ne permet pas d'aller au-delà de 18 annuités, mais le programme est évidemment modifiable.

```
10 TEXT : NORMAL : PRINT CHR$(21): HOME
15 FOR I = 1 TO 40 STEP 4: HTAB 1: VTAB 1: PR
INT "___": VTAB 21: HTAB 1: PRINT "___":
NEXT : INVERSE : FOR I = 2 TO 21: VTAB I: H
TAB 1: PRINT " "; HTAB 40: PRINT " ": NEXT
: NORMAL
20 :
25 REM *****
30 REM *** SAISIE DES DONNEES ***
35 REM *****
40 :
45 Q = 0
50 Q = Q + 1: GOSUB 135: GOSUB 130
55 VTAB 23: HTAB H: CALL - 198: INPUT "":R$
60 IF R$ = "" AND Q = 1 THEN 160
65 FOR I = 1 TO LEN (R$):AS = ASC ( MID$ (R
$,I,1)): IF AS > 47 AND AS < 58 THEN 75
70 IF Q < 3 OR (Q = 3 AND AS < ) 46) THEN C
ALL - 198: GOTO 55
75 NEXT : GOSUB 160: IF Q < 3 THEN 40
80 GOTO 205
85 :
90 VTAB 22: HTAB 1: CALL - 958: RETURN
95 POKE 32,2: POKE 33,37: POKE 34,10: POKE 35
,20: HTAB 3: PRINT : RETURN
100 HTAB 3: VTAB 1 + Q * 2: PRINT Q$ " "; INVE
RSE : PRINT R$: NORMAL : RETURN
105 A$ = "AMORT.= ":R$ = "RESTE=": RETURN
110 A$ = " = ":R$ = " = ": RETURN
115 Q$ = "FIN = <ESC> - AUTRE CALCUL = TOUCHE":
GOTO 130
120 Q$ = "<1> REVOIR LE DEBUT <2> SUITE": GOTO
130
125 Q$ = "ENFONCEZ UNE TOUCHE POUR LA SUITE"
130 VTAB 23: HTAB 1: CALL - 868: PRINT Q$;:H
= PEEK (36) + 2: RETURN
135 ON Q GOTO 140,145,150,155
140 Q$ = "MONTANT DE L'EMPRUNT": RETURN
145 Q$ = "NOMBRE DES ANNUITES ": RETURN
150 Q$ = "TAUX DE L'INTERET ": RETURN
155 Q$ = "MONTANT DE L'ANNUITE": RETURN
160 ON Q GOTO 165,170,175,100
165 ME = VAL (R$): GOTO 100
170 NA = VAL (R$): GOTO 100
175 IN = VAL (R$): GOTO 100
180 :
185 REM *****
190 REM *** CALCUL & AFFICHAGE ***
195 REM *****
200 :
205 GOSUB 90: GOSUB 95
210 IN = IN / 100:MS = ME
215 MA = ME * IN / (1 - (1 / (1 + IN) ^NA)):VA
= INT (MA * 100 + .05) / 100:Q = Q + 1:R$
= STR$ (VA): GOSUB 135: GOSUB 160
220 V = 10: GOSUB 105
225 FOR I = 1 TO NA:AM = MA - ME * IN:ME = ME
- AM
230 V = V + 1: VTAB V: PRINT I;: HTAB 3: PRINT
A$;
235 U$ = STR$ ( INT (AM * 100 + .5) / 100): GO
SUB 295: HTAB 17 - U: PRINT U$;: HTAB 21: P
RINT R$;:U$ = STR$ ( INT (ME * 100 + .5) /
100): GOSUB 295: HTAB 35 - U: PRINT U$
240 IF I = 1 OR I = 10 THEN GOSUB 110
245 IF I = 9 THEN TEXT : GOSUB 125: GET H$: G
OSUB 95:V =.10: GOSUB 105: HOME
250 NEXT I
255 IF NA > 9 THEN TEXT : GOSUB 120: GET H$:
PRINT "": IF H$ = "1" THEN GOSUB 95: HOME
:Q = 3:ME = MS: GOTO 215
260 TEXT : GOSUB 90: GOSUB 115: GET H$: IF AS
C (H$) = 27 THEN HOME : END
265 RUN
270 :
275 REM *****
280 REM *** PRINT USING ***
285 REM *****
290 :
295 L = LEN (U$):LP = L - 2: IF LP < = 0 THEN
310
300 FOR U = L TO LP STEP - 1: IF MID$ (U$,U,
1) = CHR$ (46) THEN 315
305 NEXT U
310 U = L + 1
315 RETURN
```

Programmer ?

Bien sûr que cela en vaut la peine !

Je ne suis pas informaticien de métier, mais j'utilise l'informatique — celle qui passe par des machines des types IBM 34 et 38 — depuis de nombreuses années. Disons, pour être plus précis, que je la SUBISSAIS, comme tout un chacun subit actuellement la télévision.

Premier contact

Mon premier contact avec un ordinateur personnel ne se révéla pas aussi concluant que l'eût souhaité mon fils (le talentueux rédacteur en chef de CHASSEUR D'IMAGES), possesseur d'un APPLE de la première génération.

Il crut en effet éblouir le damiste (et problémiste) que je suis en mettant en œuvre un logiciel qui, effectivement, jouait aux dames, mieux que moi certes, mais sans me révéler ses secrets.

Que l'on ne puisse pas savoir avec quoi on faisait quoi me parut à la fois étonnant, et prodigieusement agaçant. Aux Editions Jibena, les listages des programmes de l'IBM 38 ne sont pas confidentiels. Je n'y comprends rien, mais je jauge... des yeux, le génie de leur auteur.

Avec ce jeu de dames : le mystère. Je me promis alors d'aborder l'informatique autrement, de l'autre côté de la barrière. J'ignorais quand, mais je savais que je n'accepterais pas de me laisser passivement guider dans des labyrinthes — fussent-ils géniaux —, construits par d'autres.

D'abord, un PC 1210

Cela se passa un peu plus tard avec le tout nouveau PC 1210 de SHARP. Que de soirées se prolongèrent... tôt le matin ! Comme mes pauvres yeux souffrirent en essayant de déchiffrer l'unique ligne d'affichage de cet étonnant calculateur !

Mais quel plaisir, pour le ludographe, que je suis, ainsi placé devant des énigmes jamais rencontrées ! Tout dans la tête... tout le programme, s'entend. Lister restait évidemment possible, mais relati-

vement longuet... et coûteux.

J'ai réinventé, sur SHARP, les carés magiques et autres amusettes mathématiques, et j'ai même réussi, sans MID\$, LEFT\$ et RIGHT\$, à croiser des mots de trois lettres. Parfaitement !

Ce fut à ce moment-là que l'un de mes meilleurs collaborateurs — peu satisfait d'ailleurs de me voir plonger dans l'univers du BASIC (il a changé d'avis depuis) — essaya, lui aussi, sur son IBM 34, de créer des grilles de mots croisés.

Apple, enfin !

Mais entre temps j'avais acquis, sur les conseils judicieux de mon fils, un APPLE II+, flambant neuf.

Que croyez-vous qu'il arriva ?

Parole : je n'ai jamais essayé le logiciel de dames (un nouveau, plus performant, paraît-il) que l'on m'avait remis en prime.

Par contre, depuis trois ans, j'ai écrit plus de 300 programmes inédits dont les produits paraissent régulièrement dans la cinquantaine de revues de jeux et mots croisés Guy-Hachette.

Je ne raisonne plus comme autrefois, et je reste persuadé que je n'aurais jamais inventé, sans la micro-informatique... et le Basic,

le dixième des amusettes que j'ai programmées. D'ailleurs, où aurais-je trouvé le temps d'y penser ?

Les heures passées à aligner les lignes de Basic sont rapidement amorties quand le micro-logiciel fonctionne et produit, en quelques minutes, et sans erreur, ce que le penseur élucubrait en dix fois plus de temps !

Naturellement, il ne servirait à rien de vouloir imiter MAGICALC (excellent logiciel, efficace et pratique). Pourtant, à quoi bon utiliser un puissant tableur, quand on sait n'avoir à afficher que dix colonnes de chiffres ? Il vaut mieux alors écrire sa propre application, avec de beaux écrans — bien à soi —, agréables à regarder.

C'est ainsi que l'on ne subit pas l'informatique.

Tenez, je vous signale au passage que j'ai préféré écrire ce texte à la main... plutôt que sur mon APPLE IIC, avec EPISTOLE, ou sur LISA... ou encore sur MACINTOSH-LISA.

Je suis un original. Un original qui, de temps en temps, pour le plaisir, aime reprendre sa liberté et retrouver le chemin du papier... Un original qui reste un éditeur de revues de jeux, pas vrai ? G-H

MultiJeux

Guy-Hachette

156

PROBLÈMES DE LOGIQUE

conçus sur ordinateur personnel APPLE



Tous les jeux que vous aimez !

NUMÉRO HORS SÉRIE DE MULTIJEUX - 15 F

COPIE D'ÉCRAN APPLE IIe

avec carte 80 COLONNES

Quand on ne dispose pas d'une interface imprimante autorisant la copie d'écran en 80 colonnes, on s'offre l'un des petits programmes que voici :

Le n°1 vous obligera à patienter un peu plus de 30 secondes, plus l'impression.

Le n°2 utilise une routine en langage machine, et il est instantané.

*
*300.348

```

0300- AD 14 03 8D 1E 03 AD 15
0308- 03 8D 1F 03 A2 00 A0 00
0310- 8D 55 C0 BD 00 04 99 00
0318- 60 C8 8D 54 C0 BD 00 04
0320- 99 00 60 C8 E8 E0 28 D0
0328- E7 A5 69 18 69 02 8D 3E
0330- 03 A5 6A 69 00 8D 3F 03
0338- A2 00 BD 46 03 9D 00 00
0340- E8 E0 03 D0 F5 60 50 00
0348- 60
    
```

Attention !
M\$ doit être
la première
variable du
programme !

BSAVE IMP.LM, A\$300, L\$49

1 BASIC SEUL

```

10 PRINT CHR$(4)"PRÉ3": PRINT : REM
   ACTIVER LA CARTE 80 COLONNES
20 DIM M$(23):A = 39
30 VTAB 1: PRINT "IMPRESSION D'UN ECRAN
   80 COLONNES": VTAB 2: FOR I = 1
   TO 20: PRINT "____";: NEXT
40 POKE 34,3: LIST : POKE 34,23
50 :
60 REM *****
70 REM * LECTURE DE L'ECRAN 80 COL. *
80 REM *****
90 :
100 FOR E = 0 TO 23: READ L
110 FOR I = L TO L + A: POKE - 16299,
   0:M$(E) = M$(E) + CHR$( PEEK (I))
   : POKE - 16300,0:M$(E) = M$(E) +
   CHR$( PEEK (I)): NEXT : NEXT
120 PRINT CHR$(4)"PRÉ1": PRINT CHR$(
   27)"E";: PRINT CHR$(27)"L015":
   PRINT CHR$(9)"80N"
130 FOR I = 0 TO 23: PRINT M$(I): NEXT
   : PRINT
140 PRINT CHR$(4)"PRÉ0": PRINT CHR$(
   4)"PRÉ3": END
150 :
160 REM *****
170 REM * ADRESSES DES 24 LIGNES *
180 REM *****
190 :
200 DATA 1024,1152,1280,1408,1536,1664
   ,1792,1920,1064,1192,1320,1448,1576
   ,1704,1832,1960,1104,1232,1360,1488
   ,1616,1744,1872,2000
    
```

```

10 M$ = "X": REM VARIABLE LIGNE D'ECRAN
20 PRINT CHR$(4)"PRÉ3": PRINT : REM   ACTIVER LA CARTE 80 COLONNES
30 VTAB 1: PRINT "IMPRESSION D'UN ECRAN 80 COLONNES": VTAB 2: FOR I = 1 TO 20
   : PRINT "____";: NEXT
40 POKE 34,3: LIST : POKE 34,23: PRINT   CHR$(4)"BLOAD IMP.LM"
50 :
60 REM *****
70 REM * LECTURE DE L'ECRAN 80 COL. *
80 REM *****
90 :
100 PRINT CHR$(4)"PRÉ1": PRINT CHR$(27)"E";: PRINT CHR$(27)"L015": PRINT
   CHR$(9)"80N" (ou "BIN", éventuellement)
110 FOR E = 0 TO 23: READ L
120 POKE 788,L - INT (L / 256) * 256: POKE 789, INT (L / 256)
130 CALL 768
140 PRINT M$: NEXT
150 PRINT CHR$(4)"PRÉ0": PRINT CHR$(4)"PRÉ3": END
160 :
170 REM *****
180 REM * ADRESSES DES 24 LIGNES *
190 REM *****
200 :
210 DATA 1024,1152,1280,1408,1536,1664,1792,1920,1064,1192,1320,1448,1576,1704
   ,1832,1960,1104,1232,1360,1488,1616,1744,1872,2000
    
```

2 BASIC
& LM



- Parce que, en devenant riche, tu deviens fière, tu oublies que «santé passe richesse»...
- Mais aussi parce que nous t'aimons bien, pépins compris !

Ma chère vieille pomme,

NOUS sommes nombreux, un peu partout sur la planète, à passer de trop courtes soirées (ce n'est pourtant pas faute de les prolonger !) en ta compagnie.

Tu nous empoisonnes l'existence avec des défauts que la jeunesse excusait, mais que les années rendent proprement insupportables.

Ton clavier n'est pas détachable, et ne comporte ni pavé numérique ni touches programmables. Quant au modèle franco-américain, il est difficile de s'y habituer (à quand la possibilité d'afficher en AZERTY ou QUERTY par programme ?). Ton basic a laissé tomber le PRINT USING et quelques autres instructions en grattant ses puces.

Depuis le temps, au moins à l'intention de ceux qui n'usent pas (ou plus) du BASIC INTERGER, tu aurais pu créer une jolie petite disquette MASTER permettant de doper le basic APPLESOFT, mais passons : d'autres pallient ces oublis. Il faut bien que tout le monde vive, n'est-ce pas ?

Heureusement, ton image de marque est bonne. Excellente même. Chez tes amis du moins. A tel point que certains d'entre eux ont commandé, les yeux fermés, un Lisa puis un MAC. On ne peut pas dire que leur fidélité ait été récompensée. De Lisa, évitons de parler : attendre les nouveaux modèles, comme tes clients prudents, alléchés par les réclames que nous avons payées, eût été préférable. Pour MAC, les choses ont été un peu plus claires. Fermons cette parenthèse : elle n'a qu'un faible rapport avec toi, vieille Pomme.

Nous disposons maintenant de ta version portable. C'est vrai que, de nom, si l'on exclut les fils, l'alimentation et l'écran, il devient réellement portable, ce modèle IIc.

Je te signale au passage que, personnellement, je l'aurais pourvu de la version rapide du 6502, tu sais, celle que l'on trouve sur les accélérateurs, couramment diffusés au prix... d'une unité centrale ?

Et voilà que nous arrive, avec son PRODOS, son microprocesseur de 16 bits, et son basic complet, l'APPLE IIx*, un véritable ordinateur personnel *up to date* (pour combien de temps ?), avec — bravo ma vieille pomme ! — une compatibilité qui honore ton nom.

Permetts-moi de te rappeler que plus d'un million d'amateurs des fruits de la passion, cela compte. Même quand une pub budgétivore croit pouvoir convaincre les chefs d'entreprise et autres clients potentiels que, indiscutablement, c'est une Pomme qu'il leur faut (moi, je suis d'accord, note bien... et sans que cela te coûte un seul denier).

Ton image de marque était excellente, je le rappelle. Il nous déplairait que tu la ternisses en te compromettant avec des gens qui sont incapables de t'aimer... parce qu'ils ne feront jamais l'effort d'apprendre à te connaître.

Car ton avenir, quoi qu'en pensent tes conseillers, passe encore par la programmation... et ceux qui s'y adonnent.

Parfaitement, ma vieille Pomme ! En toute amitié

Guy-Hachette.

* Nous sommes bien renseignés, hein ?

--* LA MEMOIRE DE VOTRE APPLE *--

0801-	17	08	0A	00	89	3A	BA	E7	2049	23	8	10	0	137	58	186	231
0809-	28	34	29	22	50	52	23	33	2057	40	52	41	34	80	82	35	51PR..
0811-	22	3A	BA	3A	97	00	53	08	2065	34	58	186	58	151	0	93	8S.
0819-	14	00	54	24	D0	22	2D	2D	2073	20	0	84	36	208	34	45	45	..T.....
0821-	2A	20	4C	41	20	4D	45	4D	2081	42	32	76	65	32	77	69	77	..LA.MEM
0829-	4F	49	52	45	20	44	45	20	2089	79	73	82	69	32	68	69	32	OIRE.DE.
0831-	56	4F	54	52	45	20	41	50	2097	86	79	84	82	69	32	65	80	VOTRE.AP
0839-	50	4C	45	20	2A	2D	2D	22	2105	90	76	69	32	42	45	45	34	PLE.....
0841-	3A	B9	31	34	30	33	2C	32	2113	58	185	49	52	48	51	44	50T..
0849-	33	3A	9E	3A	BA	54	24	3A	2121	51	58	158	58	186	84	36	58I
0851-	9D	00	82	08	1E	00	81	49	2129	157	0	130	8	30	0	129	73TR
0859-	D0	31	C1	31	30	3A	54	52	2137	208	49	193	49	48	58	84	82TR...
0861-	24	D0	54	52	24	C8	22	5F	2145	36	208	84	82	36	200	34	95
0869-	5F	5F	5F	5F	5F	5F	5F	22	2153	95	95	95	95	95	95	95	34T
0871-	3A	82	3A	A2	32	3A	BA	54	2161	58	130	58	162	50	58	186	84R.....
0879-	52	24	3A	B9	33	34	2C	33	2169	82	36	58	185	51	52	44	51	

(1) SUITE (2) AUTRE ADRESSE DE DEPART (3) FIN DE PROGRAMME

Evidemment, qu'elle vous intéresse... la mémoire de votre Apple !

Vous est-il déjà arrivé de vouloir modifier, sans en changer la longueur, un programme, en basic... terminé par une routine (invisible) en langage machine ?

En examinant la mémoire de votre Apple, tout devient plus facile, et un simple POKE permet de remplacer un caractère ASCII par un autre.

Dans cette version, je ne fais apparaître que les lettres capitales (de A à Z), mais le programme est perfectible (de \$3AA à \$3CE).

A vous de jouer !

```
10 TEXT : PRINT CHR$(4)"PR£3": PRINT : HOME
20 T$ = "--* LA MEMOIRE DE VOTRE APPLE *--": POKE 1403,23:
  INVERSE : PRINT T$: NORMAL
30 FOR I = 1 TO 10:TR$ = TR$ + "_____": NEXT : VTAB 2:
  PRINT TR$: POKE 34,3
40 HOME : VTAB 21: PRINT TR$
50 VTAB 23: POKE 1403,24: CALL - 198: INPUT "ADRESSE DEC
  IMALE DE DEPART => ";AD$: IF AD$ = "" THEN 140
60 L = LEN (AD$): IF L > 5 THEN 50
70 FOR I = 1 TO L:AS = ASC ( MID$ (AD$,I,1)): IF AS < 48
  OR AS > 59 THEN 50
80 NEXT
90 AD = VAL (AD$)
100 POKE 6,AD - INT (AD / 256) * 256: POKE 7, INT (AD / 2
  56)
110 HOME : VTAB 21: PRINT TR$: VTAB 5: POKE 1403,0: CALL 7
  68
120 VTAB 5: CALL 874
130 VTAB 5: CALL 938
140 VTAB 23: POKE 1403,10: CALL - 198: PRINT "<1> SUITE
  <2> AUTRE ADRESSE DE DEPART <3> FIN DE PROGRAMME ";: G
  ET H$: PRINT ""
150 IF H$ = "1" THEN AD = AD + 128: GOTO 100
160 IF H$ = "2" THEN 40
170 TEXT : HOME : END
180 GOTO 140
```

VOUS POUVEZ AJOUTER UNE LIGNE 15 :
15 PRINT CHR\$(4)"BLOAD EXAMEM.LM"

0300-	A5 06	LDA	\$06
0302-	8D 3E 03	STA	\$033E
0305-	8D 75 03	STA	\$0375
0308-	8D B3 03	STA	\$03B3
030B-	A5 07	LDA	\$07
030D-	8D 3F 03	STA	\$033F
0310-	8D 76 03	STA	\$0376
0313-	8D B4 03	STA	\$03B4
0316-	A2 00	LDX	£\$00
0318-	8A	TXA	
0319-	48	PHA	
031A-	A9 00	LDA	£\$00
031C-	E0 08	CPX	£\$08
031E-	90 02	BCC	\$0322
0320-	A9 08	LDA	£\$08
0322-	18	CLC	
0323-	65 06	ADC	\$06
0325-	85 06	STA	\$06
0327-	AA	TAX	
0328-	A5 07	LDA	\$07
032A-	69 00	ADC	£\$00
032C-	85 07	STA	\$07
032E-	20 41 F9	JSR	\$F941
0331-	A9 AD	LDA	£\$AD
0333-	20 ED FD	JSR	\$FDED
0336-	20 57 DB	JSR	\$DB57
0339-	68	PLA	
033A-	AA	TAX	
033B-	A0 00	LDY	£\$00
033D-	BD 00 08	LDA	\$0800,X
0340-	20 DA FD	JSR	\$FDDA
0343-	20 57 DB	JSR	\$DB57
0346-	E8	INX	
0347-	C8	INY	
0348-	C0 08	CPY	£\$08
034A-	D0 F1	BNE	\$033D
034C-	20 57 DB	JSR	\$DB57
034F-	8A	TXA	
0350-	48	PHA	
0351-	A6 06	LDX	\$06
0353-	A5 07	LDA	\$07
0355-	20 80 FE	JSR	\$FE80
0358-	20 24 ED	JSR	\$ED24
035B-	20 84 FE	JSR	\$FE84
035E-	68	PLA	
035F-	AA	TAX	
0360-	E0 80	CPX	£\$80
0362-	F0 05	BEQ	\$0369
0364-	20 FB DA	JSR	\$DAFB
0367-	D0 AF	BNE	\$0318
0369-	60	RTS	
036A-	A2 00	LDX	£\$00
036C-	A9 25	LDA	£\$25
036E-	85 24	STA	\$24
0370-	A9 00	LDA	£\$00
0372-	85 0A	STA	\$0A
0374-	BD 00 08	LDA	\$0800,X
0377-	85 08	STA	\$08
0379-	C9 0A	CMP	£\$0A
037B-	90 06	BCC	\$0383
037D-	C9 64	CMP	£\$64
037F-	90 05	BCC	\$0386
0381-	B0 06	BCS	\$0389
0383-	20 57 DB	JSR	\$DB57
0386-	20 57 DB	JSR	\$DB57

0389-	20 57 DB	JSR	\$DB57
038C-	86 09	STX	\$09
038E-	A6 08	LDX	\$08
0390-	A9 00	LDA	£\$00
0392-	20 24 ED	JSR	\$ED24
0395-	A6 09	LDX	\$09
0397-	E8	INX	
0398-	E6 0A	INC	\$0A
039A-	A5 0A	LDA	\$0A
039C-	C9 08	CMP	£\$08
039E-	D0 D4	BNE	\$0374
03A0-	E0 80	CPX	£\$80
03A2-	F0 05	BEQ	\$03A9
03A4-	20 FB DA	JSR	\$DAFB
03A7-	D0 C3	BNE	\$036C
03A9-	60	RTS	
03AA-	A2 00	LDX	£\$00
03AC-	A9 47	LDA	£\$47
03AE-	85 24	STA	\$24
03B0-	A0 00	LDY	£\$00
03B2-	BD 00 08	LDA	\$0800,X
03B5-	20 7D E0	JSR	\$E07D
03B8-	B0 02	BCS	\$03BC
03BA-	A9 AE	LDA	£\$AE
03BC-	20 ED FD	JSR	\$FDED
03BF-	C8	INY	
03C0-	E8	INX	
03C1-	C0 08	CPY	£\$08
03C3-	D0 ED	BNE	\$03B2
03C5-	E0 80	CPX	£\$80
03C7-	B0 05	BCS	\$03CE
03C9-	20 62 FC	JSR	\$FC62
03CC-	90 DE	BCC	\$03AC
03CE-	60	RTS	

*300.3CE

0300-	A5 06 8D 3E 03 8D 75 03
0308-	8D B3 03 A5 07 8D 3F 03
0310-	8D 76 03 8D B4 03 A2 00
0318-	8A 48 A9 00 E0 08 90 02
0320-	A9 08 18 65 06 85 06 AA
0328-	A5 07 69 00 85 07 20 41
0330-	F9 A9 AD 20 ED FD 20 57
0338-	DB 68 AA A0 00 BD 00 08
0340-	20 DA FD 20 57 DB E8 C8
0348-	C0 08 D0 F1 20 57 DB 8A
0350-	48 A6 06 A5 07 20 80 FE
0358-	20 24 ED 20 84 FE 68 AA
0360-	E0 80 F0 05 20 FB DA D0
0368-	AF 60 A2 00 A9 25 85 24
0370-	A9 00 85 0A BD 00 08 85
0378-	08 C9 0A 90 06 C9 64 90
0380-	05 B0 06 20 57 DB 20 57
0388-	DB 20 57 DB 86 09 A6 08
0390-	A9 00 20 24 ED A6 09 E8
0398-	E6 0A A5 0A C9 08 D0 D4
03A0-	E0 80 F0 05 20 FB DA D0
03A8-	C3 60 A2 00 A9 47 85 24
03B0-	A0 00 BD 00 08 20 7D E0
03B8-	B0 02 A9 AE 20 ED FD C8
03C0-	E8 C0 08 D0 ED E0 80 B0
03C8-	05 20 62 FC 90 DE 60

BSAVE EXAMEM.LM, A\$300, L\$CF

Hexadécimal en décimal, et vice versa

- Il faut absolument connaître les nombres hexadécimaux jusqu'à... 255. Le calcul ne nécessite pas de programme, sinon dans la tête.
- **JUSQU'À \$15** (le dollar signale un nombre hexa), s'habituer à compter en hexa : 1, 2, ..., 8, 9, A, B, C, D, E, F.

Pour convertir rapidement une valeur DÉCIMALE en valeur HEXADÉCIMALE, il existe aussi un moyen simple, à partir du moniteur.

UTILISEZ LA FORMULATION SUIVANTE :

-] N = 2048 (par exemple) (RETURN)
-] POKE 6,N — INT(N/256) * 256 (RETURN)
-] POKE 7,INT(N/256) (RETURN)
-] CALL-151 (RETURN)
- *6.7 (RETURN)
- *0006 — 00 08 (EN INVERSANT = \$0800)

Pour transformer de l'HEXA en DÉCIMAL, le processus est inversé.

-] CALL-151 (RETURN)
- *6:00 08 (RETURN)
- *CTRL-C (retour au basic) (RETURN)
-] PRINT PEEK(6) + PEEK(7) * 256 (RETURN)
-] 2048... ET LE TOUR EST JOUÉ !

Ou encore, toujours à partir du moniteur :

-] CALL-151 (RETURN)
- *45:08 00 (dans le bon ordre) (RETURN)
- *ED24G (le G équivaut au RUN du BASIC) (RETURN)
- *2048 (Extra, hein ?)

EXPLICATION :

A l'adresse \$45, on sauvegarde l'ACCUMULATEUR, et en \$46 le REGISTRE X.

\$ED24 est une routine APPLESOFT qui affiche gentiment la valeur DÉCIMALE de A, X.

À PARTIR DE \$10

se livrer mentalement à la petite gymnastique suivante :

- Multiplier le premier chiffre du nombre hexa par 16, puis ajouter le chiffre des unités au total.

Exemples : \$10 = 1 × 16 + 0, soit 16. — \$11 = 1 × 16 + 1, soit 17. — \$1A = 1 × 16 + 10 (A = 10 en hexa), soit 26. — \$F1 = 15 × 16 + 1, soit 241.

- Pour convertir un nombre décimal en nombre hexa, adopter une démarche inverse.
- Le diviser par 16... pour obtenir le chiffre des unités, puis prendre le reste de la division pour faire le chiffre des dizaines.

Exemple : 241/16 donne 15 et un reste de 1. Naturellement, on commence par convertir 15 en \$F, puis on place le 1 à côté pour obtenir \$F1.

H	D	H	D	H	D
\$0 = 0		\$10 = 16		\$20 = 32	
\$1 = 1		\$11 = 17		\$21 = 33	
\$2 = 2		\$12 = 18		\$22 = 34	
\$3 = 3		\$13 = 19		\$23 = 35	
\$4 = 4		\$14 = 20		\$24 = 36	
\$5 = 5		\$15 = 21		\$25 = 37	
\$6 = 6		\$16 = 22		\$26 = 38	
\$7 = 7		\$17 = 23		\$27 = 39	
\$8 = 8		\$18 = 24		\$28 = 40	
\$9 = 9		\$19 = 25		\$29 = 41	
\$A = 10		\$1A = 26		\$2A = 42	
\$B = 11		\$1B = 27		\$2B = 43	
\$C = 12		\$1C = 28		\$2C = 44	
\$D = 13		\$1D = 29		\$2D = 45	
\$E = 14		\$1E = 30		\$2E = 46	
\$F = 15		\$1F = 31		\$2F = 47	

Ce programme comporte une erreur.

A vous de la découvrir !

(Réponse page 62).

Aiguillage vers la bonne destination. ➔

Erreur de saisie. ➔

Retour à INPUT si le caractère n'est pas un chiffre. ➔

```

10 TEXT : NORMAL : HOME
20 INVERSE : PRINT " CONVERSION DECIMALE-HEXA ET INV
   ERSE " : NORMAL : POKE 34,3
30 FOR I = 1 TO 10:T$ = T$ + "____": NEXT : VTAB 19:
   PRINT T$
40 VTAB 21: PRINT " FAIRE PRECEDER LA VALEUR PAR <H>
   OU <D>"
50 :
60 REM *****
70 REM ** INPUT VALEUR A CONVERTIR **
80 REM *****
90 :
100 POKE 35,18:VE = 2
110 VE = VE + 2: IF VE > 17 THEN VE = 4
120 VTAB VE: HTAB 1: CALL - 868: INPUT "VOTRE VALEUR
   ";V$: IF V$ = "" THEN 490
130 L = LEN (V$) - 1:VV$ = RIGHT$ (( " " + V$),L)
140 IF ASC (V$) = 72 THEN 220
150 IF ASC (V$) = 68 THEN VA = VAL (VV$): GOTO 340
160 GOTO 120
170 :
180 REM *****
190 REM *** HEXA VERS DECIMALE ***
200 REM *****
210 :
220 VA = 0: IF L > 4 THEN 120
230 GOSUB 470: PRINT "HEXA->DECI = ";
240 FOR I = 1 TO L:AS = ASC ( MID$ (VV$,I,1)) - 48
250 IF AS > = 0 AND AS < 10 THEN 280
260 IF AS > 16 AND AS < 23 THEN AS = AS - 7: GOTO 280
270 GOTO 120
280 VA = 16 * VA + AS: NEXT :V$ = STR$ (VA): GOTO 480
290 :
300 REM *****
310 REM *** DECIMALE VERS HEXA ***
320 REM *****
330 :
340 V$ = "": IF L > 5 THEN 120
350 FOR I = 1 TO L:AS = ASC ( MID$ (VV$,I,1))
360 IF AS < 48 OR AS > 57 THEN 120: NEXT
370 GOSUB 470: PRINT "DECI->HEXA = ";
380 VB = INT (VA / 16):CH = VA - 16 * VB + 48:VA = VB
390 IF CH > 57 THEN CH = CH + 7
400 V$ = CHR$ (CH) + V$: IF VB < > 0 THEN 380
410 GOTO 480
420 :
430 REM *****
440 REM *** AFFICHAGE MODE INVERSE ***
450 REM *****
460 :
470 VTAB VE: HTAB 14: INVERSE : PRINT V$;: NORMAL : P
   RINT " ";: RETURN
480 INVERSE : PRINT V$: NORMAL : GOTO 110
490 TEXT : HOME : END

```

Vous êtes certes au courant... mais cela vous coûte combien ?

Amusez-vous à enregistrer (ce sera vite fait !) vos factures d'électricité depuis 1980, par exemple, (consommation et montant).

Les moyennes mensuelles ne manqueront pas de vous surprendre.

Si vous ne possédez pas de carte MID pour votre imprimante, utilisez le petit programme de la page 40.

Le listage ci-après devrait maintenant vous... éclairer !

Pendant la saisie : En mode correction, il est inutile de retaper une donnée exacte. Un simple **(RETURN)** suffit.

```

10 TEXT : PRINT CHR$(4)"PRÉ3": PRINT
20 DIM C(10,6,1),T(4,1)
30 GOTO 1070
40 :
50 REM *****
60 REM ***** SAISIE *****
70 REM *****
80 :
90 POKE 32,0: POKE 33,49
100 IF CR THEN INVERSE
110 VTAB 4: POKE 1403,7: PRINT ND + AN
    - 1: IF CR THEN NORMAL: RETURN
120 FOR I = 1 TO 6: IF C(AN,I,0) = 0 THEN N = I - 1: GOTO 140
130 NEXT
140 INVERSE: FOR I = 1 TO 6: VTAB 4 + I * 2: PRINT I: NEXT: NORMAL: GOSUB 420
150 N = N + 1
160 VTAB 19: POKE 1403,0: INVERSE: PRINT N;: NORMAL: INPUT " CONSOMMATION -> ";KW$: IF KW$ = "" AND C(AN,N,0) < > 0 THEN KW$ = STR$(C(AN,N,0)):A$ = KW$:V = 19: GOSUB 290
170 IF KW$ = "?" THEN GOTO 650
180 IF KW$ = "*" AND N > 1 THEN N = N - 1: GOTO 160
190 IF KW$ = "*" AND N AND AN > 1 THEN N = 6:MA = AN:AN = AN - 1:CR = 1: GOSUB 100: GOSUB 420: GOTO 160
200 V$ = KW$: GOSUB 350: IF NOT B THEN 160
210 VTAB 21: POKE 1403,0: INPUT " MONTANT FACT.-> ";CN$: IF CN$ = "" AND C(AN,N,1) < > 0 THEN CN$ = STR$(C(AN,N,1)):A$ = CN$:V = 21: GOSUB 290
220 IF CN$ = "?" THEN C(AN,N,0) = VAL(KW$): GOSUB 420: GOTO 650
230 IF CN$ = "*" THEN 160
240 V$ = CN$: GOSUB 350: IF NOT B THEN 210
250 C(AN,N,0) = VAL(KW$):C(AN,N,1) = VAL(CN$): GOSUB 420
260 IF CR THEN AN = MA: GOSUB 420:N = 1:CR = 0: GOSUB 100: GOTO 470
270 IF N / 6 = INT(N / 6) THEN AN = AN + 1:N = 0:ND = ND + 1: CALL - 198: GOTO 110
280 GOTO 150
290 PRINT: VTAB V: POKE 1403,18: INVERSE: PRINT A$: NORMAL: RETURN
300 :
310 REM *****
320 REM ***** CONTROLE CHIFFRES *****
330 REM *****
340 :
350 B = 0: FOR I = 1 TO LEN(V$):AS = ASC(MID$(V$,I,1)): IF (AS < 48 OR AS > 57) AND AS < > 46 THEN RETURN

```

```

360 NEXT :B = 1: RETURN
370 :
380 REM *****
390 REM ***** AFFICHAGE DONNEES *****
400 REM *****
410 :
420 FOR I = 1 TO 6
430 VTAB 4 + I * 2: POKE 1403,3: CALL
    - 868
440 POKE 1403,7 - LEN ( STR$ ( C(AN,I,
    0))) : PRINT C(AN,I,0);
450 POKE 1403,14 - LEN ( STR$ ( INT (
    C(AN,I,1))) : PRINT C(AN,I,1)
460 NEXT : RETURN
470 :
480 REM *****
490 REM ***** AFFICHAGE DU MENU *****
500 REM *****
510 :
520 ONERR GOTO 650
530 FOR I = 1 TO 10:T$ = T$ + "
    _":TR$ = TR$ + "-----": NEXT
540 POKE 1403,0: VTAB 1: PRINT "CONSO
    MATION D'ELECTRICITE ET COUT
    ANNEES "ND;"
    -"AN + ND - 1: VTAB 2: PRINT T$
550 VTAB 4: POKE 1403,51: PRINT LEFT$
    (T$,29): VTAB 14: POKE 1403,51: PR
    INT LEFT$ (T$,29)
560 INVERSE : FOR I = 5 TO 14: VTAB I:
    POKE 1403,51: PRINT " "; POKE 140
    3,79: PRINT " ": NEXT : NORMAL
570 ME$ = "<1> ENREGISTRER FACTURE <2>
    SIMPLE CONSULTATION <3> EDITION D
    U TABLEAU <4> FIN DE PROGRAMME
    "
580 FOR I = 1 TO LEN (ME$) STEP 25: V
    TAB 5 + (I + 24) / 25 * 2: POKE 140
    3,54
590 PRINT MID$ (ME$,I,25): NEXT
600 :
610 REM *****
620 REM **** M E N U *****
630 REM *****
640 :
650 POKE 32,52: POKE 33,29: POKE 1403,
    53: VTAB 17: PRINT : PRINT "Six fac
    tures annuelles -----"
660 VTAB 21: PRINT : PRINT "* pour cor
    riger ? pour obtenir le
    MENU"
670 VTAB 15: PRINT : CALL - 868: PRIN
    T "-> "; GET R$:R = PEEK (49152)
    - 48: IF R < 1 OR R > 4 THEN 670
680 PRINT R$: ON R GOTO 90,750,750,690

```

```

690 FIN = 1: GOSUB 1110: TEXT : HOME :
    END
700 :
710 REM *****
720 REM ***** TABLEAU ANNEES *****
730 REM *****
740 :
750 TEXT : HOME
760 VTAB 20: POKE 1403,0: PRINT T$
770 VTAB 22: POKE 1403,0: PRINT "ANNEE
    S A AFFICHE "; INVERSE : PRINT ND
    -"ND + AN - 1"; NORMAL : INPUT " (
    X,Y) -> ";A1$,A2$
780 IF A1$ = "" OR A2$ = "" THEN 770
790 IF A1$ = "?" OR A2$ = "" THEN HOM
    E : GOTO 540
800 A1 = VAL (A1$):A2 = VAL (A2$): IF
    A1 < ND OR A2 > ND + AN - 1 THEN 7
    70
810 IF A2 - A1 > 4 THEN 770
820 HOME : PRINT : VTAB 2: POKE 1403,2
    7: PRINT "CONSOMMATION D'ELECTRICIT
    E": VTAB 3: PRINT TR$
830 NE = A2 - A1 + 1: INVERSE : FOR I =
    1 TO NE: VTAB 4: POKE 1403,I * 20
    - 10:: PRINT A1 + I - 1: NEXT : NOR
    MAL
840 VTAB 5: POKE 1403,0: PRINT TR$
850 Ai = A1 + 1 - ND:A2 = A2 + 1 - ND:
    FOR I = A1 TO A2:T(I,0) = 0:T(I,1)
    = 0: NEXT
860 FOR I = 1 TO 6: FOR J = A1 TO A2
870 H = J * 20 - 20: VTAB 4 + I * 2
880 POKE 1403,H + 8 - LEN ( STR$ ( IN
    T (C(J,I,0))) : PRINT C(J,I,0);
890 POKE 1403,H + 15 - LEN ( STR$ ( I
    NT (C(J,I,1))) : PRINT C(J,I,1);
900 T(J,0) = T(J,0) + C(J,I,0):T(J,1) =
    T(J,1) + C(J,I,1): IF I < 6 THEN 9
    60

```



ÉLECTRICITÉ (suite et fin)

```
910 PRINT : PRINT TR$: INVERSE
920 POKE 1403,H + 8 - LEN ( STR$ ( IN
T (T(J,0))) ): PRINT T(J,0);: POKE 1
403,H + 15 - LEN ( STR$ ( INT (T(J
,1))) ): PRINT T(J,1);: NORMAL
930 T(J,0) = INT (T(J,0) / 12):T(J,1)
= INT (T(J,1) / 12)
940 PRINT : PRINT : IF C(J,I,1) = 0 TH
EN 960
950 POKE 1403,H + 8 - LEN ( STR$ ( IN
T (T(J,0))) ): PRINT T(J,0);: POKE 1
403,H + 15 - LEN ( STR$ ( INT (T(J
,1))) ): PRINT T(J,1);
960 NEXT : PRINT : NEXT
970 :
980 REM **** ATTENTION! CARTE MID ****
990 :
1000 VTAB 22: IF R = 3 THEN POKE 34,23
: PRINT CHR$ (4)"PRÉ5": PRINT CHR
$ (27)"q": PRINT CHR$ (27)"L020":
PRINT CHR$ (9)"T": CALL 49230 + (2
56 * 5)
1010 CALL - 198: POKE - 16368,0: WAIT
- 16384,128,127: POKE - 16368,0:
HOME : GOTO 540
1020 :
1030 REM
1040 REM
1050 REM
1060 :
1070 ONERR GOTO 1090
1080 PRINT CHR$ (4)"OPEN ELEC": PRINT
CHR$ (4)"READ ELEC": INPUT AN: INP
UT ND: FOR I = 1 TO AN: FOR J = 1 T
O 6: FOR K = 0 TO 1: INPUT C(I,J,K)
: NEXT K,J,I: PRINT CHR$ (4)"CLOSE
ELEC": GOTO 1130
1090 VTAB 22: HTAB 1: INPUT "PREMIERE A
NNEE POUR INITIALISATION ";AN$: IF
AN$ = "" OR LEN (AN$) < > 4 THEN
1090
1100 AN = 1:ND = VAL (AN$)
1110 PRINT CHR$ (4)"OPEN ELEC": PRINT
CHR$ (4)"WRITE ELEC": PRINT AN: PR
INT ND: FOR I = 1 TO AN: FOR J = 1
TO 6: FOR K = 0 TO 1: PRINT C(I,J,K
): NEXT K,J,I: PRINT CHR$ (4)"CLOS
E ELEC"
1120 IF F THEN RETURN
1130 POKE 216,0: HOME : GOTO 520
```

Nota : En mode correction, il est possible de revenir sur l'année précédente en tapant autant de fois (*) que c'est nécessaire.

Tout le monde le sait, pourtant...

• PRINT USING

L'APPLESOFT ignore le PRINT USING (c'est d'ailleurs dommage !), mais il existe de nombreux moyens pour pallier cette lacune, en fonction des besoins, naturellement. *EN VOICI UN :*

```
] 100 US = 6
] 110 FOR I = 1 TO 6 : READ N
] 120 IF NOT INT (N) THEN HTAB US : PRINT N :
GOTO 140
] 130 HTAB US — LEN (STR$ (INT (N))) : PRINT N
] 140 NEXT
] 145 DATA 1234.23,243.7,789,5.56,10000.9,.96
```

```
1234.23 //
243.7 //
789 //
5.56 //
10000.9 //
.76 //
```

S'il n'y a pas de valeur inférieure à 1, la ligne 120 devient superflue.

• HTAB EN 80 COLONNES

Avec les cartes 80 COLONNES, il ne faut pas utiliser HTAB H et pas davantage POKE 36,H (qui, néanmoins, fonctionne), mais de préférence un POKE 1403,H.

• VALEUR ASCII APRÈS GET

A la suite d'un GET (suivi d'une variable alphanumérique) vous pouvez remplacer la formule :

```
] 100 GET A$: A = ASC (A$) par
] 100 GET A$: A = PEEK (49152)
```

• & (LIST) avec l'AMPERSAND

Quand on programme, on est souvent amené à taper LIST. Ce mot (déjà court, reconnaissons-le) peut être remplacé par l'AMPERSAND : &.

Comment ? En écrivant, aux adresses 1013 — 1015 (\$3F5-\$3F7) les paramètres de la commande LIST (\$D6A5) :

— POKE 1014,165 : POKE 1015,214 (inutile de modifier 1013, qui doit être à 76).

(Valeurs normales : 1014,88 — 1015,255)

Saisie avec **TRI** immédiat

Jusqu'à 50 mots (mais on peut aller au-delà), il est bon d'effectuer le TRI au fur et à mesure de la saisie, comme le prouve ce court programme de démonstration.

```
100 TEXT : HOME : DIM M$(50)
110 FOR I = 1 TO 10:T$ = T$ + "----": NEXT
120 PRINT "SAISIE DE MOTS, TRIES AU FUR ET A MESURE";
130 PRINT T$: POKE 34,2: VTAB 20: PRINT T$: POKE 35,19
140 VTAB 22: HTAB 1: CALL - 868: PRINT " <1> SAISIE
DES MOTS <2> MOTS TRIES "; GET CHOIX$: PRINT "":
IF ASC (CH$) < 49 OR ASC (CH$) > 50 THEN 140
150 IF CH$ = "1" THEN 220
160 GOTO 440
170 :
180 REM *****
190 REM *** SAISIE DES MOTS ***
200 REM *****
210 :
220 VTAB 22: PRINT "50 MOTS MAXIMUM (PAS PLUS DE 16 LE
TTRES"
230 VTAB 3: HOME : HTAB 13: INVERSE : PRINT "FIN DE SA
ISIE="/: NORMAL
240 S = 0:V = 5
250 S = S + 1:V = V + 1:H = 2: IF V = 19 THEN V = 6
260 IF S > 9 THEN H = 1
270 VTAB V: HTAB H: CALL - 868: PRINT S;" -> .....
....."
280 VTAB V: HTAB 7: INPUT " ";M$(0): IF M$(0) = "" THEN
270
290 IF LEN (M$(0)) > 16 THEN 270
300 IF M$(0) = "/" THEN S = S - 1: GOTO 140
310 VTAB V: HTAB 23: PRINT "1) OK 2) REFAIRE "; GET
CH$:A = ASC (CH$): IF A < 49 OR A > 50 THEN 310
320 IF A = 50 THEN 270
330 VTAB V: HTAB 23: PRINT " "
340 FOR I = S TO 1 STEP - 1
350 IF M$(0) > M$(I - 1) THEN M$(I) = M$(0): GOTO 370
360 M$(I) = M$(I - 1): NEXT
370 IF S < 50 THEN 250
380 GOTO 140
390 :
400 REM *****
410 REM *** LISTE TRIEE ***
420 REM *****
430 :
440 VTAB 3: HOME :V = 3:H = 2:SA = 0:H = 1
450 SA = SA + 1:V = V + 1: IF V = 19 AND H = 1 THEN V =
4:H = 20
460 IF V = 19 AND H = 20 THEN V = 4:H = 1: VTAB 22: HT
AB 1: CALL - 868: PRINT "APPUYER SUR UNE TOUCHE SV
P "; GET CH$: PRINT : VTAB 3: HOME
470 VTAB V: HTAB H: PRINT M$(SA): IF SA < S THEN 450
480 IF SA < S THEN 450
490 VTAB 22: HTAB 1: CALL - 868: PRINT "<1> REVOIR LE
S MOTS <2> SAISIE <3> FIN "; GET CHOIX$: PRINT "":
IF ASC (CH$) < 49 OR ASC (CH$) > 51 THEN 490
500 ON VAL (CH$) GOTO 440,220,510
510 TEXT : HOME
```

NOTA : Corriger la liste en fin de saisie ferait évidemment gagner du temps.

Boucles simples en langage machine

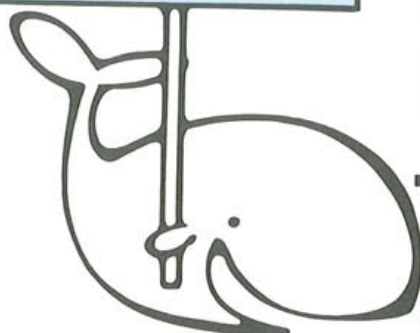
Ne vous laissez pas décourager par l'apparente complexité du langage machine... et essayez de comprendre le fonctionnement de chacune de ces boucles.

EXEMPLE DE BOUCLE (255 TO 1)

```

BASIC
10 HOME
20 X = 255
30 PRINT CHR$(7)
40 X = X - 1
50 IF X <> 0 THEN 30
60 END
    
```

LANGAGE MACHINE		SENS
0300 — 20 58 FC	JSR \$FC58	(HOME)
0303 — A2 FF	LDX £\$FF	(BOUCLE)
0305 — 20 DD FB	JSR \$FBDD	(BIP)
0308 — CA	DEX	(MOINS 1)
0309 — DO FA	BNE \$0305	(NEXT)
030B — 60	RTS	(BASIC)



Noter l'effacement de l'écran par appel de la routine \$FC58, puis les 255 (\$FF) (bip) obtenus en appelant la routine \$FBDD

POURQUOI \$F5

Si l'on voulait repartir de l'adresse \$30C, on écrirait, à partir de \$30B : DO FF.

En faisant le compte à rebours, on trouve la vraie valeur : FE, FD, FC, FB, FA, F9, F8, F7, F6, F5... qui correspond bien à l'adresse \$302.(CQFD).

Autre moyen : On désire reculer de 10 cases.

La soustraction \$F - \$A (15 - 10) donne bien 5.

```

0300 — A2 00
0302 — BD 00 20
0305 — 9D 00 08
0308 — E8
0309 — E0 32
030B — D0 F5
030D — 60
030E —
    
```

```

1 *****
2 * BOUCLE AVEC BNE *
3 *****
4 *
5 * 50 données sont stockées *
6 * de $2000 à $2031 *
7 *
8 * On les réécrit à partir de *
9 * l'adresse $800 *
10 *
11 *****
12 *
13 ORG $300
14 *
15 LDX £$00 ; L'index X est mis à zéro
16 BOUCLE LDA $2000,X ; On charge la donnée de l'adresse-origine
17 STA $0800,X ; et on l'écrit à la nouvelle adresse
18 INX ; Le registre X augmenté de 1
19 CPX £$32 ; Si égalité, indicateur Z mis à 1
20 BNE BOUCLE ; Si Z=0, encore un tour
21 RTS ; sinon, c'est terminé.
22 END
    
```

AVEC UN
ASSEMBLEUR,
TOUT
DEVIENT
PLUS CLAIR !

Vous dites addition ?

Tout d'abord, il faut savoir que, pour votre 6502, — \$FF + \$01 = \$00, mais la retenue $c = 1$

Utilisons les adresses \$06 et \$07 (libres dans la Page 0 de l'Apple) pour stocker le nombre d'origine : \$00FE (254, si vous préférez). Sa partie haute est placée en \$07 et sa partie basse en \$06 (à l'envers, comme pour une adresse, c'est-à-dire \$ FE 00).

Et maintenant, suivez simplement l'évolution de ce petit programme, lequel se contente d'ajouter 1 au contenu de la case-mémoire \$06, et 1 à celui de la case-mémoire \$07, s'il y a retenue.

0300	—	A5 06	LDA	\$06
0302	—	A4 07	LDY	\$07
0304	—	18	CLC	
0305	—	69 01	ADC	£\$01
0307	—	90 03	BCC	\$030C
0309	—	C8	INY	
030A	—	84 07	STY	\$07
030C	—	85 06	STA	\$06
030E	—	60	RTS	

*6:FE 00 \$ FE 00, soit \$00FE (254)

*300G G correspond au RUN du Basic

*6.7 Pour afficher le nouveau contenu

0006 — FF 00 On a maintenant \$00FF (255)

*300G Et ça repart !

*6.7

0006 — 00 01 Ça y est ! \$0100 (256), ce qui justifie les 2 octets.

*300G Attention ! quand les 2 valeurs seront à \$FF, il faudra un troisième octet.

0006 — 01 01

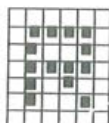
Cela vous paraît compliqué ? placez-vous devant votre clavier et votre écran... et ne vous découragez surtout pas !

INSTRUCTIONS UTILISÉES DANS CES DEUX PAGES

- ADC** Le contenu de la mémoire spécifiée (ou la donnée) est ajouté à l'accumulateur A + la retenue éventuelle. c est mis à 1 si le résultat dépasse \$FF.
- BCC** Branchement si la retenue $c = 0$
- BEQ** Branchement si $z = 1$ (or, $z = 1$ si la dernière comparaison a donné l'égalité, ou si le résultat était 0).
- BNE** Branchement si $z = 0$ (le contraire de BEQ)
- CLC** Annule la retenue (met c à zéro)
- CPX** Comparaison avec le registre X (une donnée est comparée à X. L'indicateur z est mis à 1 s'il y a égalité. c (carry) est mis à 1 si X est plus grand que la donnée, ou la mémoire indiquée).
- CMP** et **CPY** jouent respectivement le même rôle avec l'accumulateur A et le registre Y.
- DEX** Le contenu du registre index X est diminué de 1.
- DEY** Fait de même avec Y.
- INX** On ajoute 1 au contenu de X.
- INC** (case-mémoire) et **INY** (registre Y) n'agissent pas autrement que INX.
- JMP** Saut sans retour à l'adresse indiquée.
- JSR** Saut à l'adresse où attend un sous-programme, puis retour à l'instruction suivante.
- LDA** Copie dans l'accumulateur A le contenu de la mémoire spécifiée, ou une donnée (en mode immédiat).
- LDX** Même fonction que LDA, mais pour le registre X.
- LDY** Idem, pour le registre Y.
- RTS** Retour de sous-programme (c'est le \$60 du langage machine)
- STA** Le contenu de l'accumulateur est transféré dans la case-mémoire indiquée.
- STX** et **STY** font rigoureusement la même chose, mais à partir des registres X et Y.

Avec ProCODE, la programmation en assembleur devient réellement un plaisir !

LOGICIEL FRANÇAIS
utilisable sur
APPLE IIe et APPLE IIc,
sous ProDOS



RICHARD THIBERT, 17 ans, un surdoué assurément, est l'auteur de ProCODE, l'un des nouveaux logiciels de VERSION SOFT* (dont on connaît déjà les excellents EPISTOLE et VERSION CALC).

Qu'on se le dise : Richard n'a pas perdu son temps en nous livrant, en français, et sur 80 colonnes, le premier assembleur-éditeur pleine page, pour APPLE IIe et APPLE IIc.

Une précision : ProCODE m'avait séduit dès la lecture de son annonce. Ma connaissance de l'anglais étant non seulement insuffisante, mais aléatoire, j'attendais ce logiciel. Mais oui !

Et je n'ai pas été déçu, croyez-le bien. Comment pourrait-il en être autrement, quand on constate avec quelle facilité il est possible de se promener à travers un programme et d'en modifier, ajouter... ou supprimer une ligne ?

De plus, ProCODE exploite toutes les ressources de l'APPLE, 64 K ou 128 K, selon la configuration (un menu spécial permet de figer celle-ci et de démarrer avec la bonne version).

On pourrait certes lui reprocher de ne pas utiliser la souris, mais combien d'Apple IIe disposent actuellement de cette interface ? Richard Thibert a certainement eu raison...

PREMIER CONTACT

LA lecture de la documentation est indispensable. Elle demande une petite heure d'attention soutenue, et une connaissance élémentaire du langage machine.

Dans un premier temps, l'essentiel est de savoir où pouvoir, par la suite, trouver tel ou tel renseignement, au fur et à mesure de l'apprentissage.

Personnellement, j'estime qu'elle ne fournit pas suffisamment d'exemples au néophyte, mais elle n'affirme pas la prétention d'initier au langage machine, alors...

Par contre, si elle est succincte, elle se révèle d'une grande clarté, tant dans sa présentation agréable et aérée, que dans ses explications..

Les questions que l'on se pose n'y restent pas sans réponses.

Bravo !

À DÉCOUVRIR

POUR le reste, c'est l'affaire du programmeur — en herbe... ou chevronné — que vous êtes... ou allez devenir.

Découvrir les *macro-instructions* et les *instructions de chaînage*, ainsi que tous les autres points forts de ProCODE constituera, pour vous, un jeu souvent passionnant.

Un jeu qui, je n'en doute pas, va de nouveau stimuler votre intellect... et prolonger vos soirées tard dans la nuit !

G.H.

* VERSION SOFT,
19 rue Ganneron — 75018 PARIS

ProCODE 1.22 128K (C) VERSION SOFT Auteur: Richard Thibert			
Source	Objet	Label	General
\$0D01 - \$0D03 Type TXT	\$D000 - \$D000 Type BIN	\$7800 - \$7800 Type BIN	Date: Sans date
- Précédent + Suivant	O Sauver	L Sauver	• 1 Prefix 2 Date 3 Catalogue 4 Cat. imp. 5 Source typ 6 Systeme typ 7 Suppression 8 Creation 9 En ligne <input checked="" type="checkbox"/> Quitter
C Charger S Sauver J Ajouter D Dupliquer <input checked="" type="checkbox"/> Vider			
E Editer A Assembler I Imprimer			
Commande			

Un seul menu pour ProCODE, mais dont bien des auteurs devraient s'inspirer.

Assemblage par ProCODE

```

0 *****
1 ***** COPIE D'ECRAN EN 80 COLONNES *****
2 *****
3 *
4          ORG    $300
5 *
6 ORG      EQU    $300
7 PAGE2    EQU    $C055      ;Lecture 80 colonnes
8 PAGE1    EQU    $C054      ;Lecture 40 colonnes
9 LIRE80   EQU    ORG+$14    ;Adresse ligne écran (poquée)
10 LIRE40  EQU    ORG+$1E    ;Réécrite ici
11 LOMEM   EQU    $69       ;Zone des variables
12 ADRVAR  EQU    ORG+$3E
13 LECT    EQU    $400      ;Première ligne d'écran
14 HEXA    EQU    ORG+$46   ;Longueur et adr. ligne mémorisée
15 *
0300: AD 14 03 16          LDA  LIRE80      ;Valeur poquée depuis le basic
0303: 8D 1E 03 17          STA  LIRE40      ;Recopiée ici
0306: AD 15 03 18          LDA  LIRE80+1   ;Partie haute
0309: 8D 1F 03 19          STA  LIRE40+1
030C: A2 00      20          LDX  £$00      ;Mise à zéro
030E: A0 00      21          LDY  £$00
0310: 8D 55 C0 22          BOUCLE1 STA  PAGE2    ;Page2 commutée
0313: BD 00 04 23          LDA  LECT,X    ;Lecture 80 colonnes
0316: 99 00 60 24          STA  $6000,Y  ;Mémo ligne provisoire
0319: C8          25          INY          ;Plus 1 pour mémo
031A: 8D 54 C0 26          STA  PAGE1    ;Page 2 commutée
031D: BD 00 04 27          LDA  LECT,X    ;Lecture 40 colonnes
0320: 99 00 60 28          STA  $6000,Y  ;Mémo
0323: C8          29          INY          ;Plus 1 pour mémo
0324: E8          30          INX          ;Plus 1 pour la boucle1
0325: E0 28      31          CPX  £$28      ;A-t-on lu 2 fois 40 caractères ?
0327: D0 E7      32          BNE  BOUCLE1  ;Encore un tour!
0329: A5 69      33          LDA  LOMEM    ;Pour adresse de la variable
032B: 18          34          CLC          ;Annulation de la retenue
032C: 69 02      35          ADC  £$02      ;Pour obtenir l'adresse
032E: 8D 3E 03 36          STA  ADRVAR   ;Préparation de la boucle2
0331: A5 6A      37          LDA  LOMEM+1  ;Partie haute
0333: 69 00      38          ADC  £$00      ;Plus retenue éventuelle
0335: 8D 3F 03 39          STA  ADRVAR+1 ;Partie haute pour boucle2
0338: A2 00      40          LDX  £$00      ;Mise à zéro
033A: BD 46 03 41          BOUCLE2 LDA  HEXA,X    ;Adresse bidon (Voir ADRVAR)
033D: 9D 00 10 42          STA  $1000,X
0340: E8          43          INX
0341: E0 03      44          CPX  £$03      ;Retour au basic si 3 tours
0343: D0 F5      45          BNE  BOUCLE2
0345: 60          46          RTS
0346: 50 00 60 47          HEX  500060   ;Longueur variable et adresse
48

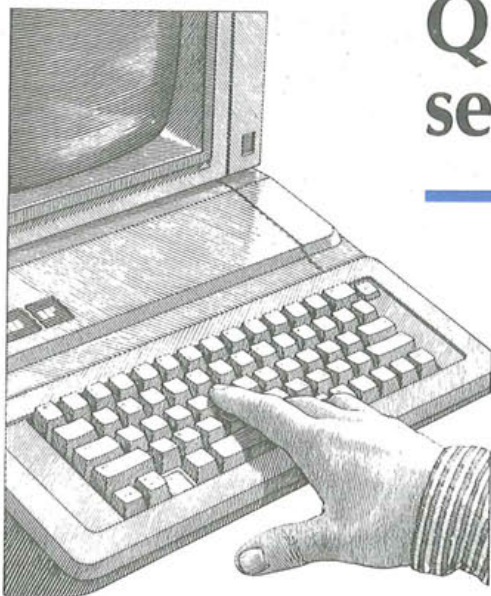
```

Table des symboles ordre alphabétique

-ADRVAR....\$033E	-BOUCLE1...\$0310	-BOUCLE2...\$033A	-HEXA.....\$0346
-LECT.....\$0400	-LIRE40....\$031E	-LIRE80....\$0314	-LOMEM.....\$0069
-ORG.....\$0300	-PAGE1.....\$C054	-PAGE2.....\$C055	

table des symboles ordre numerique

-LOMEM.....\$0069	-ORG.....\$0300	-BOUCLE1...\$0310	-LIRE80....\$0314
-LIRE40....\$031E	-BOUCLE2...\$033A	-ADRVAR....\$033E	-HEXA.....\$0346
-LECT.....\$0400	-PAGE1.....\$C054	-PAGE2.....\$C055	



Que va-t-il se passer ?

PETIT ESSAI
D'UNE INITIATION
AU FONCTIONNEMENT
DE L'APPLE II e.

En général, intéresser les jeunes à la micro-informatique ne présente aucune difficulté... si l'initiation est facultative. Il en va autrement lorsque les néophytes ne se sentent pas motivés, ce qui est parfois le cas dans les établissements d'enseignement.

On peut alors leur présenter les choses autrement, un peu comme le fait ce programme, lequel n'évoque que 4 CALL remarquables de l'APPLE, mais pas forcément connus de la majorité.

```

10 TEXT : PRINT CHR$ (21): HOME
15 VITESSE = 10: VTAB 22: HTAB 1: PRINT "VITESSE D'AFF
  ICHAGE (1/9) ";: GET VI$: IF ASC (VI$) > 48 AND
  ASC (VI$) < 50 THEN VI = VAL (VI$)
20 PRINT : HOME : HTAB 8: INVERSE : PRINT " QUE VA-T-
  IL SE PASSER ? ": NORMAL : FOR I = 1 TO 10:TR$ = TR
  $ + " _": NEXT : VTAB 2: PRINT TR$: POKE 34,2
25 :
30 REM *****
35 REM **      PREMIERE PAGE      **
40 REM *****
45 :
50 INVERSE : PRINT "1": NORMAL
55 GOSUB 360
60 PRINT "$100 A$ = " + CHR$ (34) + "A" + CHR$ (34)
65 PRINT "$105 HTAB 12"
70 PRINT "$110 FOR I = 1 TO 12: PRINT A$;:          CAL
  L - 922: CALL - 1008: NEXT"
75 GOSUB 355: VTAB 15: PRINT TR$: GOSUB 330
80 A$ = "A": HOME
85 HTAB 12: VTAB 4
90 FOR I = 1 TO 12: PRINT A$;: CALL - 922: CALL - 1
  008: NEXT
95 PRINT : PRINT TR$
100 PRINT : GOSUB 335: POKE 34,2: HOME : PRINT : PRINT

105 PRINT "LA LIGNE 110 COMPORTE DEUX INSTRUCTIONS INT
  ERESSANTES :
  - CALL-922 et                               - CALL-100
  8"
110 GOSUB 360
115 PRINT "$110 FOR I = 1 TO 12: PRINT A$;: C      ALL
  - 922: CALL - 1008 : NEXT"
  
```

← La variation de la vitesse d'affichage est obtenue par le nombre de X = PEEK (49200) émis entre chaque caractère.

← On aurait pu remplacer les CHR\$ (34) par une variable GUS = CHR\$ (34).

← CALL-922 positionne le curseur sur la ligne suivante, mais sur la même colonne. CALL-1008 le fait reculer d'une position.

← Attention aux espaces !

Utilisation du mode **inverse** pour attirer l'attention... ➔

Il n'empêche que le néo-phyte va se poser une question élémentaire : «Mais pourquoi reculer... puisque **CALL-922** permet de rester sur la même colonne ?» ➔

Réponse : Après l'affichage d'un caractère, le curseur avance bien d'une case... or, il vient d'afficher un «A»... et se trouve donc une position plus loin... CQFD.

Avouez qu'il est facile d'afficher une partie de la mémoire ! ➔

Mais il est peut-être sage de rappeler comment poker une adresse ? ➔

```
120 PRINT :T$ = " La première": GOSUB 350: VTAB 7: HTAB 13: INVERSE :T$ = "CALL-922": GOSUB 350: NORMAL : VTAB 13: HTAB 13:T$ = " fait passer le curseur à la ligne suivante sans changement de colonne (htab + 1).": GOSUB 350
125 PRINT : PRINT
130 T$ = " La seconde": GOSUB 350: VTAB 7: HTAB 13: PRINT "CALL-922": VTAB 8: HTAB 13:T$ = "CALL-1022": INVERSE : GOSUB 350: NORMAL : VTAB 17: HTAB 12:T$ = " déplace le curseur vers la gauche d'une division... d'où le résultat."
135 GOSUB 350: PRINT "": GOSUB 355: VTAB 20: PRINT TR$
140 FOR I = 13 TO 19: VTAB I: HTAB 1: PRINT "": NEXT I: GOSUB 340
145 HOME : PRINT
150 :
155 REM *****
160 REM ** DEUXIEME PAGE **
165 REM *****
170 :
175 INVERSE : PRINT "2": NORMAL : PRINT : GOSUB 360: CALL - 998: CALL - 998
180 PRINT "ET MAINTENANT ?"
185 PRINT "$100 HOME"
190 PRINT "$110 POKE 60,0: POKE 61,8: POKE 62 ,55 : POKE 63,8: CALL 64947"
195 GOSUB 355: VTAB 11: PRINT TR$: GOSUB 330
200 GOSUB 355
205 VTAB 12: CALL - 958: POKE 60,0: POKE 61,8: POKE 62,55: POKE 63,8: CALL 64947: PRINT : PRINT TR$: GOSUB 335
210 VTAB 12: CALL - 958: PRINT : PRINT
215 T$ = "On a poké en 60-61 ($3C-$3D) et en 62-63($3E-$3F) l'adresse de départ du DUMP en HEXA que l'on désirait effectuer. Un CALL 64947 (routine de l'Apple) a en-suite fait le reste de la besogne...": PRINT : GOSUB 350: PRINT
220 :
225 REM *****
230 REM ** POKER UNE ADRESSE HEXA ? **
235 REM *****
240 :
245 GOSUB 340: VTAB 12: CALL - 958: PRINT :T$ = "Pour poker une adresse HEXA en DECIMAL: ": GOSUB 350: FOR I = 1 TO 10: PRINT "----";: NEXT I:T$ = "EXEMPLES ": INVERSE : GOSUB 350: NORMAL : PRINT : PRINT
250 PRINT " - $0800 = $08 (8 DECI) et $00 (0 DECI) - $21FF = $21 (33 DEC) et $FF (ou 255)"
255 PRINT : GOSUB 355
260 :
```

(Suite page 56)


```

265 REM *****
270 REM **      LE CHOIX      **
275 REM *****
280 :
285 VTAB 22: PRINT "<1> REVOIR LE TOUT <2> MENU DE DIS
QUETTE": GOSUB 345: R = PEEK ( - 16384): IF R = 49
THEN HOME : PRINT : GOTO 50
290 IF R = 50 THEN PRINT : PRINT CHR$ (4)"RUN MENU"
295 IF R = 27 THEN TEXT : HOME : END
300 GOTO 285
305 :
310 REM *****
315 REM **      MINI SOUS-PROGRAMMES      **
320 REM *****
325 :
330 VTAB 22: PRINT "ENFONCEZ UNE TOUCHE POUR LA REPONS
E SVP": GOTO 345
335 VTAB 22: PRINT "ET UNE AUTRE TOUCHE POUR TOUT SAVO
IR...": GOTO 345
340 VTAB 22: PRINT "          UNE TOUCHE POUR LA SUITE
"
345 CALL - 198: POKE - 16368,0: WAIT - 16384,128,12
7: POKE - 16368,0: RETURN
350 FOR I = 1 TO LEN (T$): PRINT MID$ (T$,I,1);: FOR
J = 1 TO VI: X = PEEK (49200): NEXT : NEXT : RETUR
N
355 TEXT : POKE 34,2: RETURN
360 POKE 32,5: POKE 33,35: PRINT : RETURN

```



Suite de
la page
précédente.

Que va-t-il se passer ?

← Cette partie du programme ne se justifie vraiment que si l'on doit faire appel à ces routines à plusieurs reprises.

Ce sera le cas quand vous aurez triplé... mais oui !... la longueur de cette monture de base.

Bonne programmation !

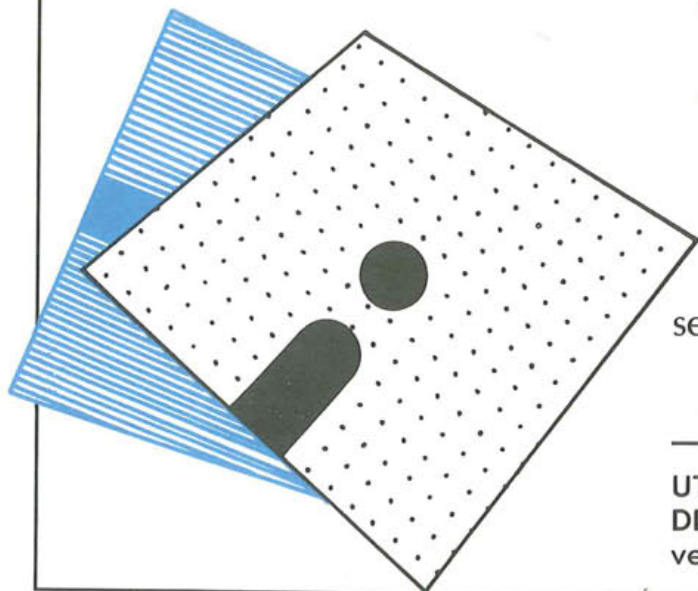
Si vous possédez un Apple *Ilc* ou
un Apple *Ile*... offrez-vous notre
disquette n° 1

Tous les programmes de

TREMPLIN MICRO

seront chez vous dans quelques jours.

UTILISEZ OBLIGATOIREMENT LE BULLETIN
DE COMMANDE DE LA PAGE 66 (vous pou-
vez le recopier ou le photocopier, bien sûr).



Un jeu graphique :

Logic-grilles

JE vous laisse le soin de découvrir à quelle logique (mais oui, elle existe... vous allez la rencontrer !) obéit ce petit jeu graphique.

Notez qu'il paraît parfois dans certaines revues de la collection «guy-hachette».

La version basic est relativement lente (au moins 30 secondes... suivant les aléas), mais une optimisation est possible.

La disquette d'accompagnement fournit une version compilée beaucoup plus rapide.

Quelle grille du haut peut logiquement occuper l'emplacement vide ?

Si vous disposez d'une imprimante graphique, voilà ce que vous obtiendrez...

```
10 LOMEM: 24577
15 G = 5300
20 TEXT : PRINT CHR$(21): HOME : D$
  = CHR$(13) + CHR$(4)
25 DIM C(4,12),B(4,12)
30 GOSUB 540
35 HGR : HGR2 : HCOLOR= 3
40 GOSUB 335
45 :
50 REM *****
55 REM TRACE DES GRILLES *****
60 REM *****
65 :
70 POKE 230,32: POKE - 16299,0
75 X = 0:D = 87
80 FOR I = X TO 72 STEP 6:E = D + I
85 HPLOT X,I TO 53,I: HPLOT 64,I TO
  117,I: HPLOT 128,I TO 181,I: HPLOT
  192,I TO 245,I
90 HPLOT 0,E TO 53,E: HPLOT 64,E TO
  117,E: HPLOT 128,E TO 181,E
95 IF E < 88 OR E > 153 THEN HPLOT
  192,E TO 245,E
100 NEXT
105 :
110 Y = 72:F = Y + D:E = D
115 FOR I = X TO 54 STEP 6
120 HPLOT I,X TO I,Y: HPLOT I + 64,X
```

Pour obtenir G, tapez PEEK (105) + PEEK (106)*256 et ajoutez 1 au résultat (avant de lancer votre programme).

```
TO I + 64,Y: HPLOT I + 128,X TO I
+ 128,Y: HPLOT I + 192,X TO I + 19
2,Y
125 HPLOT I,E TO I,F: HPLOT I + 64,E
TO I + 64,F: HPLOT I + 128,E TO I
+ 128,F: IF I > 0 AND I < 53 THEN
135
130 HPLOT I + 192,E TO I + 192,F
135 NEXT :Z = 74
140 DRAW 18 AT 0,Z: DRAW 19 AT 63,Z:
DRAW 20 AT 128,Z: DRAW 21 AT 192,Z
145 :
150 REM *****
155 REM CASES NOIRES *****
160 REM *****
165 :
170 FOR I = 1 TO 12: FOR J = 1 TO 3:
FOR K = 1 TO 9:N = C(J,I): IF N =
0 THEN 180
175 N = (N * 6) - 5:N = N + (64 * J) -
64:M = 81 + (I * 6): DRAW 95 AT N
,M
180 NEXT K,J,I
185 FOR I = 1 TO 12: FOR J = 1 TO 4:
FOR K = 1 TO 9:N = B(J,I): IF N =
0 THEN 195
190 N = (N * 6) - 5:N = N + (64 * J)
```

(Suite page 59)

FONTE CARAC

A\$6000, L\$633

Si vous tapez cette longue suite de nombres hexadécimaux, faites une sauvegarde toutes les 50 lignes... par exemple, et sous des noms différents.

6000- 64 01 D0 00 D9 00 E0 00
 6008- E8 00 EF 00 F8 00 08 01
 6010- 1A 01 20 01 28 01 30 01
 6018- 41 01 4C 01 53 01 5C 01
 6020- 62 01 6D 01 7F 01 89 01
 6028- 98 01 A7 01 B6 01 C6 01
 6030- D7 01 E1 01 F2 01 02 02
 6038- 08 02 0F 02 23 02 2E 02
 6040- 42 02 4D 02 5E 02 70 02
 6048- 81 02 90 02 A0 02 AE 02
 6050- BC 02 CE 02 E0 02 EC 02
 6058- FA 02 0A 03 12 03 27 03
 6060- 38 03 49 03 57 03 69 03
 6068- 79 03 8A 03 94 03 A5 03
 6070- B5 03 CA 03 DA 03 E7 03
 6078- F3 03 04 04 10 04 20 04
 6080- 31 04 3C 04 42 04 51 04
 6088- 60 04 68 04 7C 04 8B 04
 6090- 96 04 A7 04 B5 04 C1 04
 6098- CC 04 D9 04 E2 04 F2 04
 60A0- 00 05 0E 05 1D 05 2C 05
 60A8- 36 05 45 05 50 05 5E 05
 60B0- 68 05 7C 05 89 05 9A 05
 60B8- A7 05 B8 05 C9 05 DA 05
 60C0- EE 05 06 06 06 06 06 06
 60C8- 08 06 73 01 00 00 00 00
 60D0- 09 89 92 92 00 00 92 0A
 60D8- 00 36 36 16 4E 00 00 00
 60E0- 0D FE 6E 96 52 01 00 3E
 60E8- 36 36 36 6E 00 00 1F 49
 60F0- 49 36 36 36 6E 00 00 4E
 60F8- 6D 11 DF 77 09 1E 17 4D
 6100- 3A DF 4E 29 4E 00 00 2D
 6108- 09 8D 3F BF 0D 8D 3F 3F
 6110- 4E 0D 3E 3F 77 71 49 00
 6118- 00 36 36 96 52 01 00 00
 6120- F5 36 36 2E 4E 00 00 00
 6128- 35 36 36 17 75 01 00 00
 6130- 09 8D 1F 1F 0E 2D 1E 17
 6138- 2D 15 1F 1F 4E 4E 09 00
 6140- FF 89 6E 3A 3F 77 31 56
 6148- 49 00 00 92 92 8A F6 4D
 6150- 00 00 37 52 89 3F 87 52
 6158- 49 01 00 00 92 92 4E 00
 6160- 00 00 49 11 17 1E 17 1E
 6168- 56 49 09 00 35 29 AD DF
 6170- 33 4D 31 DF 33 4D 31 DF
 6178- 33 2D 2D 4E 00 00 6D 2D
 6180- 36 36 FE 2A 2D 75 01 00
 6188- 29 29 AD DF 73 49 1E 17
 6190- 1E 17 2D 2D 4E 00 00 01
 6198- 29 AD DF 73 49 1E 77 31
 61A0- DF 73 2D 4E 01 00 3E 4D
 61A8- 31 DF 33 4D 31 3F 3F 4E
 61B0- 09 36 4E 00 00 2D 2D 2D
 61B8- DE 1B 6E 11 3F 77 49 FE
 61C0- 9B 2D 75 09 00 00 29 2D
 61C8- DE 1B 6E 11 3F 37 4D 31

61D0- DF 73 2D 4E 01 00 00 2D
 61D8- 2D 36 1E 17 36 4E 09 00
 61E0- 2D 29 AD DF 33 4D F1 3F
 61E8- 17 4D 31 DF 73 2D 4E 01
 61F0- 00 00 29 AD DF 33 4D 31
 61F8- 3F 77 09 FE 9B 2D 75 09
 6200- 00 00 12 16 96 09 00 3E
 6208- 52 B2 F6 4D 00 00 B7 49
 6210- 3E 17 2D 2D 35 3F 3F 3F
 6218- 0E 2D 2D F5 1B 77 4E 49
 6220- 00 00 09 12 2D 2D DE 9B
 6228- 2D 2D 96 09 00 2A 49 15
 6230- FF 13 2D 2D AD 3F 3F 3F
 6238- 2E 2D 2D 1E 37 4E 49 00
 6240- 00 09 29 AD DF 73 09 1E
 6248- B6 4E 09 00 09 09 15 9F
 6250- 2D 35 DF 33 4D 31 FF 33
 6258- 2D 0D 4E 00 00 29 2D
 6260- FE 1B 6E 09 FE 1B 2E 2D
 6268- 35 DF 33 4D 71 01 00 09
 6270- 2D 15 DF 6E 31 3F 37 4D
 6278- 31 DF 33 2D 2D 4E 00 00
 6280- 4A 29 AD DF 33 36 4D 11
 6288- DF 33 2D 2D 4E 00 00 2D
 6290- 2D AD DF 6E 31 DF 6E 31
 6298- DF 17 2D 2D 4E 00 00 2D
 62A0- 2D F5 1B 6E 11 3F 37 36
 62A8- 2D 2D 4E 00 00 2D 2D
 62B0- DE 1B 6E 11 3F 37 36 4E
 62B8- 49 01 00 49 29 2D DE 1B
 62C0- 6E 89 FF 33 4D 31 DF 33
 62C8- 2D 2D 4E 00 00 33 4D 31
 62D0- DF 33 4D 31 3F 3F 6E 09
 62D8- FE 1B 6E 09 4E 00 00 09
 62E0- 2D 2D DE 36 36 9F 2D 2D
 62E8- 4E 00 00 31 09 2D 1E 36
 62F0- DF 6A 31 DF 2E 2D 4E 01
 62F8- 00 DF 4D 31 DF 33 4D 1E
 6300- 3F 6E A9 DF 33 4D 71 01
 6308- 00 36 36 36 36 2D 75 01
 6310- 00 00 29 0D AD DF DF 6E
 6318- 69 31 DF DF 6E 69 31 DF
 6320- DF 6E 69 71 01 00 00 2D
 6328- AD DF 33 4D 31 DF 33 4D
 6330- 31 DF 33 4D 71 01 00 00
 6338- 29 AD DF 33 4D 31 DF 33
 6340- 4D 31 DF 33 2D 75 09 00
 6348- 00 2D AD DF 33 4D 31 3F
 6350- 3F 36 76 49 09 00 00 29
 6358- AD DF 33 4D 31 DF 33 0D
 6360- 0D 1E DF 2E 6D 4E 00 00
 6368- 09 2D 15 DF 6E 31 3F 37
 6370- 4D 31 DF 33 4D 71 01 00
 6378- 00 29 2D DE 1B 6E 11 3F
 6380- 4E 31 DF 13 2D 2D 4E 00
 6388- 00 00 2D 2D DE 36 36 76
 6390- 49 00 00 4D 4D 31 DF 33
 6398- 4D 31 DF 33 4D 31 DF 33

! " | | % \$ ' [] * +
 , - . / 0 1 2 3 4 5 6
 7 8 9 : ; < = > ? à Á
 B C D E F G H I J K L
 M N O P Q R S T U V W
 X Y Z é ç è à _ ' a b
 c d e f g h i j k l m
 n o p q r s t u v w x
 y z é ù è ■ ■

63A0- 2D 75 09 00 00 4D 31 DF
 63A8- 33 4D 31 DF 33 4D F1 1F
 63B0- 0E 4E 09 00 49 4D 49 FE
 63B8- FB 33 4D 4D FE FB 33 4D
 63C0- 4D 1E 1F 1F 0E 0D 4E 09
 63C8- 00 00 4D 31 DF 73 0D 1E
 63D0- 17 0D 15 DF 33 4D 71 01
 63D8- 00 4A 4D 31 DF 33 4D F1
 63E0- 3F 0E 36 4E 09 00 00 2D
 63E8- 2D BE 1E 17 1E 2E 2D 75
 63F0- 01 00 00 09 15 1F 2E AD
 63F8- DF 33 2D 2D DE 1B 2E 2D
 6400- 75 01 00 00 52 2D F5 DB
 6408- 36 2E 2D F5 68 09 00 69
 6410- 69 DE 2A AD DF 33 2D 2D
 6418- DE 1B 2E 2D 75 01 00 00
 6420- 09 15 1F 2E 2D FE 1B 6E
 6428- 09 3E DF 2E 6D 4E 00 00
 6430- 00 92 92 49 11 3F 3F 4D
 6438- 49 00 00 09 76 96 52 01
 6440- 00 00 52 2D 35 DF 33 4D
 6448- 31 FF 33 2D 0D 4E 00 00

(Suite et fin page 60)

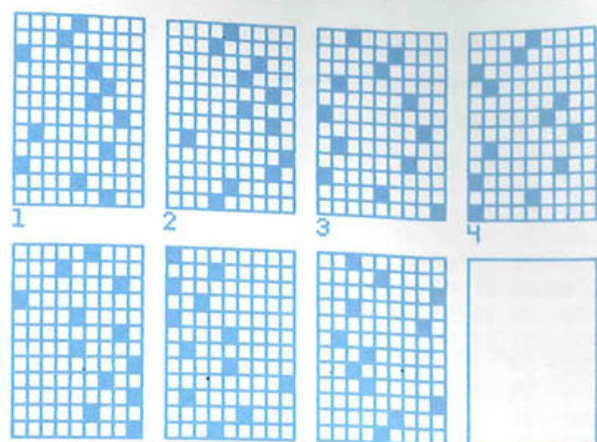
Notez que tout cela est déjà écrit dans la disquette n°1 de **TREPLIN MICRO**



```

- 64:M = (I * 6) - 6: DRAW 95 A
T N,M
195 NEXT K,J,I
200 :
205 REM *****
210 REM PAGE 1 VISIBLE *****
215 REM *****
220 :
225 POKE - 16300,0
230 :
235 REM *****
240 REM TEXTE ET REPONSE *****
245 REM *****
250 :
255 T$ = "Quelle grille du haut peut
logiquement occuper l'emplacement
t vide ?"
260 T = 163:S = - 7: FOR I = 1 TO
LEN (T$):S = S + 7:A = ASC ( MI
D$ (T$,I,1)): IF A = 32 THEN S =
S - 2: GOTO 285
265 IF A = 105 THEN S = S - 1
270 IF I = 39 THEN T = 173:S = 0
275 DRAW A - 31 AT S,T: IF A = 108
THEN S = S - 2
280 IF A = 77 OR A = 87 OR A = 109
OR A = 119 THEN S = S + 2
285 NEXT
290 PRINT " ": GET H$:O = ASC (H$)
: IF O < 49 OR O > 52 THEN 290
295 GOSUB 525: VTAB 22: HTAB 7: INV
ERSE : IF O - 48 = H THEN PRINT
" BRAVO! VOUS AVEZ GAGNE! ": G
OTO 305
300 PRINT " HELAS! VOUS AVEZ PERDU!
"
305 FOR I = 1 TO 50: FOR J = 1 TO 8
:X = PEEK (49200): NEXT :X = X
* X: NEXT : GOTO 445
310 :
315 REM *****
320 REM GRILLES ALEATOIRES *****
325 REM *****
330 :
335 FOR J = 1 TO 4 STEP 2: FOR I =
1 TO 12
340 P = 1 + INT ( RND (1) * 9): IF
I = 1 THEN 350
345 IF P = C(J,I - 1) THEN 340
350 C(J,I) = P:C(J + 1,13 - I) = 10
- P
355 NEXT I,J
360 H = 1 + INT ( RND (1) * 4): IF

```



Quelle grille du haut peut logiquement occuper l'emplacement vide ?

```

H = 3 THEN 360: REM BONNE GRILLE
365 FOR J = 1 TO 4: FOR I = 1 TO 12
:L = 0: IF J = (H) THEN B(J,I) =
C(4,I): GOTO 400
370 IF 1 + INT ( RND (1) * 4) < >
1 THEN 390: REM MODIF DE CERTAINES
LIGNES
375 B(J,I) = 1 + INT ( RND (1) * 9)
: IF B(J,I) < > C(4,I) AND B(J,
I) = 1 THEN L = 1: GOTO 395
380 IF B(J,I) < > C(4,I) AND B(J,I
) < > B(J,I - 1) THEN L = 1: GO
TO 395
385 GOTO 375
390 B(J,I) = C(4,I)
395 IF I = 12 AND L = 0 THEN 375
400 NEXT I,J: IF INT ( RND (1) * 2
) < > 1 THEN 410
405 FOR I = 1 TO 12:Z = C(2,I):C(2,
I) = C(3,I):C(3,I) = Z: NEXT
410 RETURN
415 :
420 REM *****
425 REM MINIMENU *****
430 REM *****
435 :
440 GOSUB 525
445 HOME : VTAB 22: HTAB 4: INVERSE
: PRINT "1";: NORMAL : PRINT "
EDITION ";: INVERSE : PRINT "2";
: NORMAL : PRINT " AUTRE JEU ";:
INVERSE : PRINT "3";: NORMAL :
PRINT " FIN"
450 VTAB 22: HTAB 32: PRINT " ";:
GET H$: IF VAL (H$) < 1 OR VA
L (H$) > 3 THEN 450

```

FONTI CARAC

(Suite de la page 58)

6450-	36	36	2D	AD	DF	33	4D	31	6588-	2D	12	4D	31	DF	33	4D	31
6458-	DF	33	2D	2D	4E	00	00	52	6590-	3F	77	09	3E	3F	6F	49	01
6460-	52	2D	F5	DB	36	2E	2D	75	6598-	00	00	12	2D	2D	1E	17	1E
6468-	01	00	49	49	31	DF	2A	2D	65A0-	17	2D	2D	4E	00	00	01	49
6470-	FE	1B	6E	09	FE	1B	2E	2D	65A8-	1E	17	2D	15	DF	33	2D	2D
6478-	75	01	00	2D	52	2D	15	DF	65B0-	DE	1B	2E	2D	75	01	00	00
6480-	33	2D	2D	DE	1B	2E	2D	75	65B8-	09	15	DF	6A	09	FE	1B	6E
6488-	91	60	DF	09	F5	33	8D	3F	65C0-	09	3E	DF	0E	6D	4E	00	00
6490-	77	36	4E	09	00	11	52	2D	65C8-	2D	09	15	9F	2D	15	DF	33
6498-	35	DF	33	4D	31	3F	77	09	65D0-	2D	2D	DE	1B	2E	2D	75	01
64A0-	3E	3F	6F	49	01	00	00	36	65D8-	00	00	49	11	3F	3F	2E	2D
64A8-	2D	AD	DF	33	4D	31	DF	33	65E0-	35	3F	3F	2E	2D	35	3F	3F
64B0-	4D	71	01	00	00	09	DE	2A	65E8-	2E	2D	75	01	00	09	2D	2D
64B8-	35	36	9F	2D	2D	4E	00	00	65F0-	35	3F	3F	37	2D	2D	35	3F
64C0-	F6	09	DE	2A	35	36	36	3F	65F8-	3F	37	2D	2D	35	3F	3F	37
64C8-	4D	01	00	4D	36	4D	1E	1F	6600-	2D	2D	75	01	00	40	40	C0
64D0-	2E	2D	FE	33	4D	4E	00	00	6608-	40	43	00	19	00	00	00	00
									6610-	00	00	00	00	00	00	DB	1B
									6618-	3F	2E	2D	0D	AD	3F	1F	3F
									6620-	4E	09	2D	35	3F	DF	DB	2A
									6628-	6D	09	2D	3E	3F	1F	3F	77
									6630-	2D	0D	2D					
64D8-	35	35	36	36	17	2D	4E	00									
64E0-	00	35	52	6D	AD	DF	DF	6E									
64E8-	69	31	DF	DF	6E	69	71	01									
64F0-	00	00	12	2D	AD	DF	33	4D									
64F8-	31	DF	33	4D	71	01	00	2D									
6500-	52	2D	15	DF	33	4D	31	DF									
6508-	33	2D	75	09	00	00	12	2D									
6510-	AD	DF	33	4D	31	3F	3F	36									
6518-	4D	49	00	00	52	52	2D	35									
6520-	DF	33	4D	31	3F	3F	4E	09									
6528-	6E	00	00	12	12	0D	F5	3B									
6530-	36	76	49	01	00	00	52	2D									
6538-	F5	DB	2E	2D	35	DF	13	2D									
6540-	2D	4E	00	00	00	31	17	2D									
6548-	F5	33	36	2D	4E	00	00	12									
6550-	12	4D	31	DF	33	4D	31	FF									
6558-	73	6D	4E	00	00	00	12	4D									
6560-	31	DF	33	4D	F1	1F	0E	4E									
6568-	09	00	35	12	4D	49	FE	FB									
6570-	33	4D	4D	1E	1F	1F	0E	0D									
6578-	4E	09	00	00	12	4D	F1	1F									
6580-	0E	15	1F	17	4D	71	01	00									

LOGIC-GRILLES I (FIN)

```

455 GOSUB 530
460 ON VAL (H$) GOTO 505,465,475
465 GOSUB 525: HOME : VTB 22: PRIN
T "CONFIRMER EFFACEMENT DONNEES
PAR <O> ";: GET H$: IF H$ < >
"O" THEN 440
470 FOR I = 1 TO 4: FOR J = 1 TO 12
:C(I,J) = 0:B(I,J) = 0:Q = 0: NE
XT J,I: PRINT CHR$(32): GOTO 3
5
475 TEXT : HOME : PRINT : PRINT D$ "
RUN MENU"
480 :
485 REM *****
490 REM EDITION *****
495 REM *****
500 :
505 PR$ 1
510 PRINT CHR$(9)"GD": PRINT : PR

```

```

INT : PRINT : PRINT : PRINT "
GRILLE-REPOSE = "H: PRINT C
HR$(12): PR$ 0
515 GOTO 440
520 :
525 POKE 49236,0: POKE 49235,0: POK
E 49239,0: POKE 49232,0: HOME :
RETURN
530 POKE 49236,0: POKE 49234,0: POK
E 49239,0: POKE 49232,0: RETURN
535 :
540 REM *****
545 REM FONTE L$633 *****
550 REM *****
555 :
560 ROT= 0: SCALE= 1
565 PRINT D$"BLOAD CARAC,A"G
570 POKE 232,G - INT (G / 256) * 2
56
575 POKE 233, INT (G / 256)
580 RETURN

```

RIEN + PAS
GRAND-CHOSE...
ET IL LEUR FAUT
UN ORDINATEUR...



La grande timidité des hommes (d'un certain âge) devant l'informatique personnelle

Avez-vous remarqué l'étrange comportement des hommes mariés, visitant (en compagnie de leur épouse) un salon consacré à la micro-informatique ? Ils regardent beaucoup. Écoutent davantage encore, mais se gardent bien de poser des questions.

Heureusement, les dames connaissent les secrets désirs de leurs chers époux, et interrogent vendeurs ou démonstrateurs à leur place.

Si elles savaient quelles longues soirées de solitude les attendent, elles se garderaient bien d'intervenir... à moins qu'elles n'y prennent goût elles aussi !

De bons livres, et en français !

Pendant plusieurs années, les EDITIONS DU P.S.I furent les seules à vraiment s'intéresser à l'informatique personnelle en général... et aux différentes versions de l'Apple en particulier. Rares sont ceux qui, mordus de la pomme, n'ont pas inté-rieurement remercié les Nicole Bréaud-Pouliquen et autres Jacques Boisgontier ou Frédéric Lévy pour leur excellente documentation... en français.

Aujourd'hui, les grands éditeurs entourent leur collection MICRO de soins attentifs. Les publications sont nombreuses, et il arrive même que la machine n'existe déjà plus quand paraît son mode d'emploi.

Avec Apple — valeur sûre, comme l'on sait — pas de problème !

L'ÉDITION ET LE LANGAGE MACHINE

Mon propos n'était pas de vous parler des ouvrages généraux, mais seulement de ceux capables, à mes yeux du moins, de vous inciter à programmer en langage machine.

On ne compte plus les livres traitant de l'Apple. Certains ne sont que la traduction d'ouvrages américains. D'autres ont été conçus en français, par des auteurs plus ou moins prolifiques... et parfois — hélas ! — prolifiques.

Ils ne sont pas légion. Le sujet est d'un abord difficile. Il y a des risques. Aux Editions du P.S.I, pas de problème, on a couru le risque depuis longtemps. On y a l'habitude... et on aime probablement cela.

D'où le tome 3 de la collection «APPLE II», **Langage machine et assembleur du 6502**, par Nicole Bréaud-Pouliquen et Daniel-Jean David.

Une bonne initiation, un peu aride parfois (les auteurs savent, c'est certain... mais le lecteur est ignare par définition... alors !). Il n'empêche que je vous le recommande, et plutôt deux fois qu'une.

L'Assembleur facile du 6502 (Eyrolles) n'est pas aussi facile qu'il le laisse entendre. François Monteil a adopté une autre démarche. Personnellement, je souhaite, de la même plume, une œuvre uniquement consacrée à l'Apple.

La Programmation du 6502 (A.P. Stephenson, Dunod) est une traduction de *6502 : MACHINE CODE FOR BEGINNERS*, mais alors là, chapeau ! Si vous n'avez pas compris la structure des courtes routines qui y sont clairement exposées, c'est uniquement par manque d'attention. Vous deviez penser à vos amours... ou à la fin du week-end, pas au 6502 !

Un bon conseil : offrez-vous sans tarder cette **Programmation du 6502**... et potassez-la. C'est un régal !

G-H.



Les deux
meilleurs →

EFFACEMENT D'ÉCRAN PAR LA GAUCHE

Programmation étrange.
Heureusement, le langage machine
est performant !

ASSEMBLAGE par ProCODE

```

0 *****
1 *   EFFACEMENT DE L'ECRAN   *
2 *   DE GAUCHE A DROITE     *
3 *                               *
4 *   guy-hachette   fév 85   *
5 *****
6 *
7 *           ORG   $300
8 *
9 VTAB      EQU   $FC22      ;A la ligne spécifiée dans CV $25
10 STORADV  EQU   $FBF0      ;Affichage plus déplac. curseur
11 LF       EQU   $FC66      ;A la ligne suivante
12 BS       EQU   $FC10      ;Une position à gauche
13 *
0300: A9 00      14          LDA   £$00      ;Initialisation CH
0302: 85 24      15          STA   $24      ;Horizontalement
0304: A2 00      16 BOUCLE1 LDX   £$00      ;Pour partir de la ligne 0
0306: 86 25      17          STX   $25      ;CV
0308: 20 22 FC   18 BOUCLE2 JSR   VTAB
030B: A9 20      19          LDA   £$20      ;Poke 780 à partir du basic
030D: 20 F0 FB   20          JSR   STORADV
0310: A6 25      21          LDX   $25      ;Où en sommes-nous verticalement?
0312: E0 17      22          CPX   £$17      ;23ème ligne ou pas ?
0314: F0 09      23          BEQ   POINTH    ;Si oui, on avance d'une colonne
0316: 20 66 FC   24          JSR   LF       ;Sinon, ligne suivante...
0319: 20 10 FC   25          JSR   BS       ;En reculant d'une case
031C: 4C 08 03   26          JMP   BOUCLE2  ;Et ça repart!
031F: A6 24      27 POINTH  LDX   $24      ;Où en sont les horizontaux ?
0321: E0 27      28          CPX   £$27      ;39ème position ou pas ?
0323: F0 03      29          BEQ   FIN       ;Si égalité, on a terminé
0325: 4C 04 03   30          JMP   BOUCLE1  ;Retour à la ligne 0
0328: 60         31 FIN      RTS
                                32

```

Table des symboles ordre alphabétique

-BOUCLE1...\$0304	-BOUCLE2...\$0308	-BS.....\$FC10	-FIN.....\$0328
-LF.....\$FC66	-POINTH....\$031F	-STORADV...\$FBF0	-VTAB.....\$FC22

Page 5.— CATFREE

Si vous utilisez un autre SLOT que le SLOT 6, procédez de la manière suivante pour modifier votre programme définitivement :

- BLOAD CATFREE
- POKE 785, SLOT * 16 (vous modifiez ainsi le contenu de l'adresse \$311).
- BSAVE CATFREE, A\$300, L\$84.

JEU (page 27).— Voici la ligne 120 corrigée :

120 IN = AC * I : MO = RA + IN : C = C + IN : AC = AC - RA

JEU (page 45).— Ligne 360 : Par pitié, accordez une ligne spéciale (365 par exemple) à ce NEXT, maladroitemment placé après un IF...

Affichage accéléré en langage machine

Pour quelques octets de plus, offrez un cadre à vos menus.

A partir du Basic :

BLOAD CADRE, puis CALL 768 ou, directement, BRUN CADRE.

PAGE SUIVANTE, version avec saisie d'un caractère.

Si vous ne disposez pas d'un assembleur, tapez seulement cette colonne puis sauvez votre routine par un BSAVE CADRE 1, A\$300, L\$2F



ASSEMBLAGE
par ProCODE

0	*****			
1	*			*
2	*	TRACER UN CADRE		*
3	*	EN MODE INVERSE		*
4	*			*
5	*	Guy Hachette - Février 85		*
6	*****			
7	*			
8		ORG \$300		
9	*			
10	ORG	EQU \$300		;Il suffira de modifier ORG...
11	LIGNE	EQU ORG+\$0D		;Tracé de la ligne complète
12	EXTREM	EQU ORG+\$1A		;Seulement les extrémités
13	HOME	EQU \$FC58		
14	AFFIC	EQU \$FBF0		;Affichage + déplacement curseur
15	*			
16		JSR HOME		
17		JSR LIGNE		
18		JSR EXTREM		
19		JSR LIGNE		
20		RTS		;Retour au bercail
21	*			
22		LDX £\$00		;Préparation de la boucle
23	BOUCLE1	LDA £\$20		;Espace en mode inverse
24		JSR AFFIC		
25		INX		
26		CPX £\$28		;Longueur de la ligne
27		BNE BOUCLE1		
28		RTS		
29	*			
30	BOUCLE2	LDA £\$20		;Toujours l'espace
31		JSR AFFIC		
32		LDX £\$27		;Une position avant fin de ligne
33		STX \$24		;CH: posit. hor. du curseur fixée
34		LDA £\$20		;Chargement de l'espace
35		JSR AFFIC		
36		LDA \$25		;CV: position vert. du curseur
37		CMP £\$16		;En sommes-nous à la ligne 22 ?
38		BNE BOUCLE2		;Si non on recommence
39		RTS		;LE CADRE EST TERMINE (47 OCTETS)

Table des symboles ordre alphanétique

-AFFIC.....\$FBF0	-BOUCLE1...\$030F	-BOUCLE2...\$031A	-EXTREM....\$031A
-HOME.....\$FC58	-LIGNE.....\$030D	-ORG.....\$0300	

table des symboles ordre numerique

-ORG.....\$0300	-LIGNE.....\$030D	-BOUCLE1...\$030F	-EXTREM....\$031A
-BOUCLE2...\$031A	-AFFIC.....\$FBF0	-HOME.....\$FC58	

ASSEMBLAGE PAR ProCODE

	0	*****		
	1	*		*
	2	* TRACER UN CADRE *		
	3	* CARACTERE SAISI AU CLAVIER *		
	4	* APRES APPEL DE LA ROUTINE *		
	5	*		*
	6	* Guy Hachette - Février 85 *		
	7	*****		
	8	*		
	9	ORG \$300		
	10	*		
	11	ORG EQU \$300		;Il suffira de modifier ORG...
	12	LIGNE EQU ORG+\$12		;Tracé de la ligne complète
	13	EXTREM EQU ORG+\$1F		;Seulement les extrémités
	14	SAISIE EQU \$FD0C		;Lecture d'une touche
	15	HOME EQU \$FC58		
	16	AFFIC EQU \$FBF0		;Affichage + déplacement curseur
	17	*		
	18	JSR HOME		
	19	JSR SAISIE		;Pas de contrôle de prévu
	20	LDA \$06		;Stockage en page 0
	21	JSR LIGNE		
	22	JSR EXTREM		
	23	JSR LIGNE		
	24	RTS		;Retour au bercail
	25	*		
	26	LDX \$00		;Préparation de la boucle
	27	BOUCLE1 LDA \$06		;Récupération caractère
	28	JSR AFFIC		
	29	INX		
	30	CPX \$28		;Longueur de la ligne
	31	BNE BOUCLE1		
	32	RTS		
	33	*		
	34	BOUCLE2 LDA \$06		;Toujours le caractère
	35	JSR AFFIC		
	36	LDX \$27		;Une position avant fin de ligne
	37	STX \$24		;CH: posit. hor. du curseur fixée
	38	LDA \$06		;Encore le caractère (terminé)!
	39	JSR AFFIC		
	40	LDA \$25		;CV: position vert. du curseur
	41	CMP \$16		;En sommes-nous à la ligne 22 ?
	42	BNE BOUCLE2		;Si non on recommence
	43	RTS		;LE CADRE EST TERMINE (52 OCTETS)

Table des symboles ordre alphabetique

-AFFIC.....\$FBF0	-BOUCLE1...\$0314	-BOUCLE2...\$031F	-EXTREM....\$031F
-HOME.....\$FC58	-LIGNE.....\$0312	-ORG.....\$0300	-SAISIE....\$FD0C

table des symboles ordre numerique

-ORG.....\$0300	-LIGNE.....\$0312	-BOUCLE1...\$0314	-EXTREM....\$031F
-BOUCLE2...\$031F	-AFFIC.....\$FBF0	-HOME.....\$FC58	-SAISIE....\$FD0C

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

QUI ÊTES-VOUS ?

* Facultatif

Nom* Prénom*
Année de naissance Sexe
Profession
Adresse*
Code postal [][][][][] Ville

VOTRE MATÉRIEL

Marquer d'une X les casus qui vous concernent.

Apple II Apple II + Apple IIe Apple IIc
 Apple III Macintosh Lisa

Autres :

VOS OUTILS

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

1. | 3.
2. | 4.

ANCIENNETÉ

Cocher les casus de votre choix.

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

Concevez-vous des programmes ? OUI NON

Langage BASIC PASCAL LOGO

AUTRE Précisez :

Programmez-vous en LANGAGE MACHINE ? OUI NON

Utilisez-vous un assembleur ? OUI NON

LISA BIG MAC MERLIN TOOLKIT ProCODE

Des magazines de jeux extra

En vente
chez votre
marchand
de journaux

Guy-Hachette

SUPER RÉCRÉ

ISSN 0247 - 1418

POISSONS VOIS !

AVEC PASSION

DU MOUTON NOTE

CHAPITRE DU CORAN HABILLER LES MURS

GH

Guy-Hachette

Force 2

intellect

ISSN 0181 - 9445

en supplément GRATUIT : **LE BASIC EN JOUANT**

Le Dénoué

Dans l'attente Occasionnel

Activité Dénouée

Félicité Différence

Informations Élément Personnel

PE
EAR
NSMN
EOAEE
IRMURG
UBEUVTN
AVALAEEO
REDITSURU
DEMOLIRP

GH

Bimestriel - N° 42 - 2 Fév - 2 Avr 1985 - 67 FB - 2,90 FS - 8,70 F

Guy-Hachette

Biforce 1-2

pile ou face

ISSN 0184 - 8062

SUPPLÉMENT GRATUIT (NOS PAGES VERTES)

INITIEZ-VOUS AU BASIC, EN JOUANT !

DRDC E B O F F H

LENH Z P L A

ELAEAO

IT T U R O

FSSR

ME

DRDC E B O F F H

LENH Z P L A

ELAEAO

IT T U R O

FSSR

ME

Pillages

Eclimé

Misanthropie Désert

Il relie Avec de l'opium

Vieux Entre-lacer

GH

Bimestriel - N° 40 - 9 Mars - 7 Mai 1985 - 67 FB - 2,90 FS - 8,70 F

TREMPIN MICRO — Bimestriel — C'est une publication des Editions JIBENA, 4, rue de la Cour-des-Noeues, 75020 PARIS — S.A. au capital de 3600 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.