

tremplin micro

**Des fromages
sur votre Apple !**



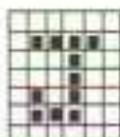
**Du basic
à l'assembleur**

**Test : un drive
pour 1 500 F !**

**Dominez
votre Apple !**

**Les pièces
qui manquent
à vos
programmes**

Programmer, c'est aussi vivre avec son temps



Je connais (bien) une cinquantaine de personnes qui utilisent quotidiennement, pour leur travail, des ordinateurs professionnels et ignorent à peu près tout de la manière dont ils sont programmés. On se contente de critiquer — et aussi de conseiller — celles et ceux chargés de les rendre plus intelligents et conviviaux.

Tant il est vrai que, dans ce pays (peut-être est-ce partout ainsi ?), les critiques les plus intolérants se rencontrent parmi les gens les plus ignorants des sujets sur lesquels ils portent des jugements.

Je connais également des amateurs — possesseurs d'ordinateurs personnels —, et amenés à se servir d'IBM 34 ou 36. Indéniablement, leur comportement se révèle différent de celui des premiers. Ils admettent volontiers que la machine de leur société a des limites... qui sont aussi celles des programmeurs. Leurs exigences — car ils en émettent ! — vont de pair avec les possibilités du matériel. Par contre — car il y a un revers à la médaille — ils ne s'en laissent jamais conter par les personnes chargées de la programmation.

Jouant eux-mêmes avec le Basic, langage que l'on considère comme simpliste et peu performant, ils ne prêtent cependant pas au GAP des pouvoirs illimités, encore qu'il puisse leur arriver d'envier à leurs aînés de l'informatique professionnelle, les impressionnants outils mis à leur disposition.

Demain, sans doute, il suffira de penser à un problème pour qu'un cerveau artificiel — captant ce désir — tente aussitôt de nous livrer la solution... directement, de microprocesseur à matière grise.

Mac permet de lever le pied pour atteindre la première marche d'un escalier dont on ne voit pas la fin, mais qui conduira au cerveau artificiel. Bravo ! Hélas ! (ou heureusement !) nous n'en sommes pas là ! En attendant, contentons-nous de programmer — sur nos APPLE IIe ou IIc, pour vivre avec notre temps. Il vaut mieux jouer avec le Basic, puis avec l'Assembleur, que d'attendre les hypothétiques (mais probables) solutions promises par l'intelligence artificielle.

Et puis, l'un n'empêche pas l'autre !

GUY-HACHETTE.

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

BIMESTRIEL

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

EDITIONS JIBENA

Direction-Rédaction :

Editions JIBENA

Guy-HACHETTE

La Petite Motte — Senillé
86100 CHÂTELLERAULT.

Téléphone :

(49) 93-66-66

PUBLICITÉ :

Joëlle (même numéro)

Commission paritaire :

Demande en cours.

Sommaire 2

Jouer sur un Apple

- Additions binaires 19
- Le jeu de l'homme libre ... 32

Si cela vous amuse !

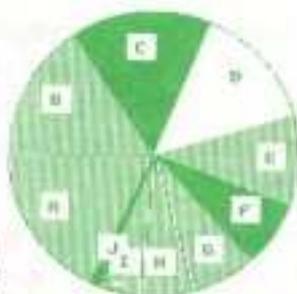
A vous d'écrire le petit programme qui va décrypter ce charabia (la solution paraîtra dans le prochain numéro) 10

Initiation

- Les fichiers aléatoires 15
- Langage machine, soustraction 43

Un drive pour 1500 F 46

Vos chiffres valent bien un fromage !



C'est la suite logique des HISTOGRAMMES de notre précédent numéro... avec une certaine compatibilité d'ailleurs 49

PROGRAMMES UTILITAIRES

- Vos menus à la carte 5
- Double listage en 80 colonnes 11
- Listage-contrôle d'un fichier "T" 44

Imprimez des listages formatés de vos programmes 27

ÉTUDE POUR LE PLAISIR

- PAIR ou IMPAIR avec quelques octets 35

Les caractères "souris" de l'Apple IIc 34

LISEZ AUSSI :

- Gribouille, le plus que parfait 3
- Plus loin, et plus sûrement, avec l'assembleur 39
- Imprimantes et graphisme 56

TRUCS ET ASTUCES

- Un accessoire nullement inutile : le magnétophone 31
- Disséquer un nombre 38
- Onerr goto 38
- Deux trucs pour votre Apple 40

- Votre fiche n°2 41
- Le courrier des lecteurs 59
- Bulletin de commande et d'abonnement 62

Le numéro 3 de **TREMPIN MICRO** paraîtra le 3 juillet

Gribouille

LE PLUS QUE PARFAIT

Vous tirez la disquette de traitement de texte Gribouille de sa pochette et vous la placez délicatement dans le lecteur 1 de votre Apple IIe ou IIc. Gribouille se présente. Très simplement, sans chercher à donner des complexes aux programmeurs en herbe avec des écrans extraordinaires. Ce n'est pas un reproche. Il vous prescrit de ne point copier sa disquette, même si c'est pour votre usage personnel. C'est gentil, mais ferme. Il en prend d'ailleurs grand soin, puisqu'il vous invite finalement à la remplacer par la vôtre. Vous réfléchissez... et vous risquez un support vierge dans le slot 1.

L'option Y du menu général vous renvoie alors au menu de disquette dont l'option I (comme initialiser) va produire son petit effet, après vous avoir demandé confirmation. Bonne précaution, indispensable quand on s'adresse à des étourdis... comme moi.

Retour au menu général, par le truchement de la touche ESCAPE (souvent sollicitée), et sélection de la première option : «C. Créer lire ou modifier».

Super machine à écrire

Si vous possédez une carte 80 colonnes — ce qui est mon cas — vous disposez alors d'une machine à écrire heureusement dotée d'un bel écran de visualisation,



Plus que parfait, c'est la société Berlingot qui le prétend.

Excellent et performant, c'est ce que j'ai moi-même constaté

Guy-HACHETTE.

lisation, et d'une mémoire confortable (ce logiciel fonctionne également en 40 colonnes).

Vous allez enregistrer votre texte au kilomètre, sans vous soucier des coupures de mots, mais en allant évidemment à la ligne quand bon vous chante. Affectionnez-vous les flèches de votre Apple ? Si oui, rassurez-vous : elles jouent leur rôle habituel, mais avec quelque chose en plus : remonter, descendre, insérer, supprimer... tout est facile, et relativement rapide.

A ce niveau, le traitement de texte Gribouille vous présente votre texte sans aucune mise en page. Par contre, les accents circonflexes et les trémas sont suivis par un "/" en mode inverse.

L'effacement, facile avec la touche «Del», est rapide, même quand votre écran est pratiquement plein.

La première impression est satisfaisante. Par contre, n'ayant volontairement pas lu le mode d'emploi de Gribouille, et pas davantage visualisé la disquette de démonstration, j'avoue me poser des questions quant aux manœuvres à exécuter pour sauver mon texte, puis pour le mettre en page... et l'éditer...

Sauvegardez de temps en temps

Ca y est : il est sauvé, enfin, partiellement. Je l'ai titré «Premier contact» et il occupe 9 secteurs de ma disquette (un peu plus d'un écran, au stade actuel). Un regret : une sauvegarde automatique après la frappe de 2000 caractères (par exemple) me paraîtrait une bonne précaution. Toutes les œuvres ne se révèlent certes pas impérissables, mais on sait combien les micro-coupures entament le moral des secrétaires et autres épistoliers !

Tiens ! comme cela, rien que pour voir, j'ai tapé un «Ctrl-X», et je me suis retrouvé au début de mon bablabla. Moi, j'aurais choisi un «Ctrl-D»... mais Madeleine Hodé, la mère de Gribouille, doit avoir ses raisons, pas vrai ? Je me suis dit que, logiquement, «Ctrl-Y» provoquerait l'effet contraire, mais je me suis trompé. C'est plus simple que cela : un autre «Ctrl-X» renvoie illico à la fin du texte.

Nouvelle exploration des possibilités, apparemment nombreuses, de *Gribouille* : j'ai vu la mise en page de mon texte, coupures de mots comprises. Extra. Et — oh ! surprise ! — le programme se comporte parfaitement avec ma carte-buffer de 64 K. Visiblement, il se moque totalement de sa présence. Moi pas : le premier jet s'imprime tandis que je tente de rédiger ces lignes.

Autre surprise agréable : les accents circonflexes ne perturbent pas la justification. «Evident», me direz-vous ? Peut-être... dans les traitements de texte français, mais pas forcément dans ceux qui sont seulement francisés. J'ai bien envie d'essayer le *soulignement*. Pomme ouverte doit bien servir à quelque chose... Ou Pomme fermée... ou bien le point d'interrogation... à moins que ce ne soit une combinaison des deux ? Vieux, hein ?

C'était bien «? + Pomme ouverte», mais j'aurais pu obtenir la même chose avec un simple «Ctrl-A».

LES BONS POINTS

- Coupure des mots en fin de ligne (conformément aux règles typographiques).
- Conception cent pour cent française.
- Documentation complète et claire.
- Facturation facile.
- Calcul intégré.
- Edition de graphique possible.
- Possibilité d'utiliser les textes écrits avec *Gribouille* dans des programmes personnels.
- Substitution instantanée de caractères (bein devient bien) et autres petites facilités du même ordre.
- Impression sans problème avec les imprimantes Apple.

DEUX GRIEFS

- La sauvegarde des textes est laissée à l'initiative de l'utilisateur, ce qui comporte des dangers.
- Ecrasement ou effacement relativement rapide (en mémoire vive) si ce même utilisateur se révèle un peu trop nerveux.

Bon ménage avec l'imprimante

En tout cas, je sais maintenant comment incorporer des codes pour mon imprimante : «Ctrl-F + CODE + FLEXE». Je vais donc aller souligner «soulignement», un peu plus haut. A tout à l'heure !

O.K. pour le soulignement, mais *Gribouille* a prévu un processus moins compliqué que ma commande d'imprimante : «Ctrl-WSL» et «Ctrl-WNS». C'est parlant et rapidement tapé. De plus, c'est parfaitement assimilé par le programme de visualisation... qui présente la partie soulignée en mode inverse.

Il y a le glossaire

Il convient de remarquer qu'une démarche consistant à découvrir empiriquement le fonctionnement de *Gribouille* n'est probablement pas la plus judicieuse. D'accord, j'aurais dû parcourir le manuel avant de m'aventurer dans les méandres d'un logiciel qui, en l'absence de touches de fonction sur l'Apple, n'a pas d'autres issues que l'emploi des différents CTRL.

Je dois pourtant constater, et non sans plaisir, que l'aventure s'est plutôt bien passée, jusqu'à présent en tout cas. Grâce aux divers menus, il est facile d'accéder à toutes les fonctions et il n'est pas indispensable d'être doté d'une mémoire d'éléphant pour utiliser *Gribouille* d'une manière satisfaisante.

Les codes ennuyeux peuvent par exemple prendre place dans un glossaire où les repêcher n'exige plus que la commande «Ctrl-G» suivie d'un code facile à retenir, comme «G» pour «gras», «M» pour «maigre», etc.

Et maintenant, à vous de jouer !

Mais je ne vais pas vous dévoiler, un à un, tous les trucs qui font de *Gribouille* un bon traitement de texte. Vous trouverez d'ailleurs, dans ces pages, un tableau évoquant ses points forts.

Disons que rien n'est surfait. Facilité d'emploi, rapidité, clarté de la documentation, service après-vente réellement décidé à rester à la disposition des utilisateurs (par téléphone, par le réseau Calvados, et par l'intermédiaire d'un bulletin) : de bonnes raisons pour l'essayer... et peut-être — s'il correspond à vos critères personnels — pour l'adopter.

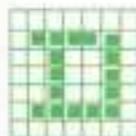
Cocorico !

Guy-Hachette

MENU DU JOUR

- 1 EMPLOI DU TEMPS
- 2 MISE À JOUR
- 3 NOUVELLES RUBRIQUES
- 4 COMPTE BANCAIRE
- 5 LES CALORIES ET MOI
- 6 FIN DE CONSULTATION

Vos menus à la carte !



ES menus courts, affichés instantanément,... et modifiables à volonté : voilà ce que vous propose ce programme ! Notez qu'il comporte une partie *DÉMO* dont vous n'aurez vraiment rien à faire par la suite (à moins que ce ne soit pour le simple plaisir de promener votre «Menu-du-jour-démo» sur votre écran !).

La routine en langage machine vous est fournie en deux parties :

- Programme proprement dit, indispensable, puisque c'est lui qui affiche le menu (\$2000 à \$205B) ;
- Texte (ASCII), qui commence à l'adresse \$205C, à la suite du programme, et dont la longueur dépendra du nombre de lignes constituant votre menu personnel, et aussi de leur longueur.

La routine finale, sauvée sous le nom de *VMENU*, regroupera programme et texte.

ES = HTAB origine
VT = VTAB origine

```

10 TEXT : PRINT CHR$ (21) : HOME
15 ES = 8:VT = 10: REM VTAB & HTAB DE BASE DANS "MENU
"
20 FOR I = 1 TO 10:TR$ = TR$ + "____": NEXT
25 T$ = "BASE MODIFIABLE D'UN MENU": HTAB 8: INVERSE :
GOSUB 155: NORMAL : PRINT
30 PRINT :T$ = "POUR FACILITER L'ADAPTATION DU PROGRA
MMELA PARTIE (ASCII) EST MEMORISEE A PART, MAIS ILS
VA Y AVOIR REGROUPEMENT DES QUEVOTRE MENU DEFINITI
F SERA MIS AU POINT.": GOSUB 155
35 PRINT : PRINT :T$ = "ATTENTION!": INVERSE : GOSUB
155: NORMAL :T$ = " CHAQUE LIGNE DU MENU DOIT SE TE
RMINER PAR UN * ($AA) ET LA DERNIERE LIGNE PAR É (
$A3), MAIS LE PROGRAMME SE CHARGE DE TOUT.": GOSUB
155: PRINT : PRINT
40 INVERSE : PRINT "POKE 8239,X";: NORMAL : PRINT " p
our modifier VTAB ORIGINE"
45 : PRINT : INVERSE : PRINT "POKE 8277,Y";: NORMAL :
PRINT " pour modifier HTAB ORIGINE"
50 PRINT : PRINT TR$
55 PRINT : PRINT CHR$ (4)"BLOAD MENUASC": PRINT CHR
$ (4)"BLOAD MENU"

```

Les lignes 30 à 90 peuvent être supprimées dans le programme définitif. A vous de décider !

```

60 GOSUB 140
65 CALL 8226
70 GOSUB 145: HOME
75 X = X + 1: ON X GOTO 80,80,85,90
80 GOTO 120
85 A = 8:B = 10: GOTO 125
90 X = 1: GOTO 120
95 ;
100 REM *****
105 REM *** SOUS-PROGRAMMES ***
110 REM *****
115 ;
120 A = 2 + INT ( RND (1) * 8):B = 2 + INT ( RND (1)
    * 13)
125 POKE 8239,A: POKE 8277,B: PRINT "-> POKE 8239,";
    INVERSE : PRINT A;: NORMAL : PRINT " : POKE 8277,";
    : INVERSE : PRINT B: NORMAL
130 VTAB 22: HTAB 7: PRINT "TOUCHE <ESC> POUR TERMINER
    "
135 VTAB 12: HTAB 1: PRINT "$GOTO 60";: FOR I = 1 TO 7
    : HTAB I + 1: FLASH : PRINT MID$ ("GOTO 60",I,1);:
    NORMAL : GOSUB 145: CALL - 1008: PRINT MID$ ("60
    TO 60",I,1);: NEXT : GOTO 65
140 VTAB 22: HTAB 5: PRINT "ENFONCEZ N'IMPORTE QUELLE
    TOUCHE"
145 CALL - 198: POKE - 16368,0: WAIT - 16384,128,12
    7: POKE - 16368,0:R = PEEK ( - 16384): IF R = 27
    THEN 185
150 RETURN
155 FOR J = 1 TO LEN (T$): PRINT MID$ (T$,J,1);: FOR
    I = 1 TO 3:BUZZ = PEEK (49200): NEXT : NEXT : RET
    URN
160 ;
165 REM *****
170 REM *** VOTRE MENU PERSONNEL ***
175 REM *****
180 ;
185 HOME :T$ = " FAITES VOTRE MENU PERSONNEL " : INVERS
    E : HTAB 7: GOSUB 155: NORMAL : PRINT
190 PRINT :T$ = "VOUS AVEZ DROIT A " : GOSUB 155: INVER
    SE : PRINT B;: NORMAL :T$ = " LIGNES MAXIMUM.": GOS
    UB 155
195 PRINT : PRINT TR$
200 POKE 34,6:LIGNE = 0:DE = 8283: HOME
205 VTAB 22: HTAB 5: PRINT "/ POUR ARRETER AVANT 8 LIG
    NES"
210 LI = LI + 1
215 VTAB 6 + LI: INVERSE : PRINT LI;: NORMAL : PRINT "
    .....": IF LI$(LI) < >
    "" THEN HTAB 2: PRINT LEFT$ (LI$(LI), LEN (LI$(LI)
    )) - 1);
220 HTAB 3: INPUT " ";LI$: IF LI$ = "" AND LI$(LI) < >
    "" THEN VTAB 6 + LI: HTAB 2: PRINT LEFT$ (LI$(LI)

```

(Suite page 9)



← CALL 8226 correspond à l'adresse \$2022 du programme en langage machine.

← HTAB et VTAB aléatoires, pour la partie Démo du programme.

← Attente d'un caractère. CALL - 198 = BEEP

← BUZZ = PEEK (49200) émet un bruit.

← Le nombre de lignes pourrait être augmenté si l'on ne sautait pas une ligne sur deux.

← DE = 8283 correspond à l'adresse \$205C - 1, départ de la partie texte du programme.

ASSEMBLAGE PAR ProCODE

Si vous n'avez pas d'assembleur, contentez-vous de taper **MENU**, et **MENUASC**, page 58.

```

0 *****
1 *
2 *           EXEMPLE DE MENU           *
3 *           EN LANGAGE MACHINE       *
4 *
5 * Guy Hachette - Février 85 *
6 *****
7 *
8           ORG $2000
9 *
10 ORG      EQU $2000           ;Il suffira de modifier ORG...
11 LIGNE    EQU ORG            ;Tracé de la ligne complète
12 EXTREM   EQU ORG+$0D        ;Seulement les extrémités
13 HOME     EQU $FC58
14 AFFIC    EQU $FBF0          ;Affichage + déplacement curseur
15 VTAB     EQU $FC22          ;A la ligne spécifiée dans CV$25
16 ASCI     EQU ORG+$5C
17 *
18         LDX $00              ;Préparation de la boucle
19 BOUCLE1  LDA $20              ;Espace en mode inverse
20         JSR AFFIC
21         INX
22         CPX $28              ;Longueur de la ligne
23         BNE BOUCLE1
24         RTS
25 *
26 BOUCLE2  LDA $20              ;Toujours l'espace
27         JSR AFFIC
28         LDX $27              ;Une position avant fin de ligne
29         STX $24              ;CH: posit. hor. du curseur fixée
30         LDA $20              ;Chargement de l'espace
31         JSR AFFIC
32         LDA $25              ;CV: position vert. du curseur
33         CMP $16              ;En sommes-nous à la ligne 22 ?
34         BNE BOUCLE2         ;Si non on recommence
35         RTS                  ;LE CADRE EST TERMINE
36 *
37         JSR HOME              ;Effacement
38         JSR LIGNE
39         JSR EXTREM
40         JSR LIGNE            ;Et on passe au text...
41 *
42         LDA $08              ;Vtab départ
43         STA $25              ;Ecrit dans CV
44         JSR INITH
45         LDX $FF              ;Pour démarrer à zéro
46 TEXT     INX
47         LDA ASCI,X           ;Lecture texte menu
48         CMP $AA              ;Est-ce un astérisque ?
49         BNE CTRLFIN         ;Si non égal
50         INC $25              ;Deux lignes...
51         INC $25              ;...plus bas
52         JSR INITH           ;Repartir de la bonne position
53         JMP TEXT
54 CTRLFIN  CMP $A3            ;Si c'est un £, FIN

```

(Suite au verso)

```

204B: F0 06      55      BEQ  TERMINE
204D: 20 F0 FB   56      JSR  AFFIC
2050: 4C 37 20   57      JMP  TEXT
2053: 60         58      TERMINE RTS
                59      *
2054: A9 0A     60      INITH  LDA  £$0A      ;Htab pour toutes les lignes
2056: 85 24     61      STA  $24      ;Ecriture dans CH
2058: 20 22 FC   62      JSR  VTAB
205B: 60         63      RTS
                64      ASCII  DSK  HEXMEN
                65

```

Table des symboles ordre alphanumérique

```

-AFFIC.....$FBF0      -ASCII.....$205C      -BOUCLE1...$2002      -BOUCLE2...$2000
-CTRLFIN...$2049      -EXTREM....$2000      -HOME.....$FC58      -INITH.....$2054
-LIGNE.....$2000      -ORG.....$2000      -TERMINE...$2053      -TEXT.....$2037
-VTAB.....$FC22

```

table des symboles ordre numerique

```

-ORG.....$2000      -LIGNE.....$2000      -BOUCLE1...$2002      -EXTREM....$2000
-BOUCLE2...$2000      -TEXT.....$2037      -CTRLFIN...$2049      -TERMINE...$2053
-INITH.....$2054      -ASCII.....$205C      -AFFIC.....$FBF0      -VTAB.....$FC22
-HOME.....$FC58

```

PARTIE «TEXT» DE VOTRE MENU

Sa longueur variera suivant le nombre de caractères, mais le programme se charge de le mémoriser, et vous n'avez donc pas à en tenir compte.

Naturellement, rien ne vous interdit de modifier les mentions «Menu du jour» et «Réponse : ».

Revoyez pour cela les lignes 310 et 330 du programme basic.

N'oubliez pas qu'un retour ligne s'indique par un astérisque et la fin de la partie «text» par une £ (ou un dièse, si vous utilisez le clavier QWERTY).

```

205C: A0 A0 A0
      A0          8 ASC * *
2060: 00 05 0E
      15 20 04
      15 20 0A
      0F 15 12   9 INV *MENU DU JOUR*
206C: AA        10 ASC **
206D: 31        11 INV *1*
206E: A0 D0 D2
      CF C7 D2
      C1 CD CD
      C5 A0 CE
      D5 CD C5
      D2 CF A0
      B1 AA      12 ASC * PROGRAMME NUMERO 1*
2082: 32        13 INV *2*
2083: A0 D0 D2
      CF C7 D2
      C1 CD CD
      C5 A0 CE
      D5 CD C5
      D2 CF A0
      B2 AA      14 ASC * PROGRAMME NUMERO 2**
2097: 33        15 INV *3*
2098: A0 D0 D2
      CF C7 D2
      C1 CD CD
      C5 A0 CE
      D5 CD C5
      D2 CF A0
      B3 AA      16 ASC * PROGRAMME NUMERO 3**
20AC: 34        17 INV *4*
20AD: A0 D0 D2
      CF C7 D2
      C1 CD CD
      C5 A0 CE
      D5 CD C5
      D2 CF A0
      B4 AA      18 ASC * PROGRAMME NUMERO 4**
20C1: A0 A0 A0
      A0 D2 C5
      00 CF CE
      03 C5 A0
      BA A0      19 ASC * REponse : *
20CF: 20        20 INV * *
20D0: AA A3     21 ASC **£

```

SAISIE

Nombre maximum de lignes : 8, mais vous stoppez la saisie en tapant un "/" au début de la ligne que vous ne désirez pas utiliser.

Ne tenez aucun compte des astérisques ou de la livre pendant la saisie : le programme les rajoute.

DE est le numéro de la case-mémoire dans laquelle vous écrivez. →

VMENU est en un seul bloc. Rien ne vous interdit de le rebaptiser Toto ou Zozo. →

MENUS À LA CARTE

(Suite de la page 7)

```
), LEN (LI$(LI)) - 1): GOTO 240
225 IF LI$ = "/" AND LI > 1 THEN VTAB 6 + LI: HTAB 1:
    CALL - 958:LI = LI - 1: GOTO 250
230 L = LEN (LI$): IF L > 30 OR L < 1 THEN 215
235 LI$(LI) = " " + LI$ + "*"
240 IF C < > 0 THEN C = 0:LI = LM: GOTO 250
245 IF LI < 8 THEN 210
250 VTAB 22: HTAB 1: PRINT "<1> LA SUITE <2> CORRIGER
    <3> ANNULER": GOSUB 145: PRINT : IF R < 49 OR R > 5
    1 THEN 250
255 ON R - 48 GOTO 295,260,200
260 VTAB 22: HTAB 1: CALL - 868: PRINT "NUMERO DE LIG
    NE A CORRIGER ? "; GET C$: PRINT C$:C = ASC (C$)
    - 48: IF C < 1 OR C > LI THEN 260
265 LM = LI:LI = C: PRINT "": GOTO 215
270 :
275 REM *****
280 REM *** ECRITURE EN MEMOIRE ***
285 REM *****
290 :
295 LM = 0: FOR I = 1 TO LI: IF LEN (LI$(I)) > LM THEN
    LM = LEN (LI$(I))
300 NEXT
305 ESPACE = (41 - LM) / 2:T$ = "": FOR I = ES TO 13:T$
    = T$ + CHR$ (32): NEXT : GOSUB 375
310 FOR I = 1 TO 12:DE = DE + 1:AS = ASC ( MID$ ("MEN
    U DU JOUR",I,1)): IF AS = 32 THEN POKE DE,AS: GOTO
    320
315 POKE DE,AS - 64
320 NEXT :DE = DE + 1: POKE DE,170
325 FOR I = 1 TO LI:DE = DE + 1: POKE DE, ASC ( STR$ (
    I)): FOR J = 1 TO LEN (LI$(I)):DE = DE + 1: POKE D
    E, ASC ( MID$ (LI$(I),J,1)) + 128: NEXT : NEXT
330 T$ = T$ + "REPONSE : ": GOSUB 375:DE = DE + 1: POKE
    DE,32:T$ = "£": GOSUB 375
335 VT = 10 - LI: POKE 8239,VT: POKE 8277,ES: TEXT
340 :
345 REM *****
350 REM *** VOTRE MENU EST SAUVE ***
355 REM *****
360 :
365 VTAB 22: HTAB 1: CALL - 958: PRINT "BSAVE VMENU,A
    8192,L"DE - 8191: VTAB 22: PRINT CHR$ (4)"BSAVE VM
    ENU,A8192,L"DE - 8191: VTAB 21: CALL - 958
370 GOSUB 140: CALL 8226: GOSUB 145: HOME : GOTO 380
375 FOR I = 1 TO LEN (T$):DE = DE + 1: POKE DE, ASC (
    MID$ (T$,I,1)) + 128: NEXT : RETURN
380 VTAB 22: HTAB 2: PRINT "<1> FIN <2> MODIFIER <3> R
    EFAIRE "; GET C$: IF C$ = "3" THEN CLEAR : GOTO
    10
385 IF C$ = "2" THEN 185
390 :
```

(Suite page 10)

```

395 REM *****
400 REM *** VOTRE PROGRAMME BASIC ***
405 REM *****
410 :
415 HOME :G$ = CHR$(34)
420 T$ = " VOTRE PROGRAMME EN BASIC "; INVERSE : HTAB 7
    ; GOSUB 155: NORMAL : PRINT : PRINT
425 T$ = "Il ressemblera à ces lignes:"; GOSUB 155: PRI
    NT : PRINT "-----"; PRINT :
    PRINT
430 T$ = "$10 TEXT: PRINT CHR$(21) ; D$ = CHR$(13) +
    CHR$(4)": GOSUB 155: PRINT
435 T$ = "$20 PRINT D$" + G$ + "BLOAD MENU" + G$: GOSUB
    155: PRINT
440 T$ = "$30 CALL 8226": GOSUB 155: PRINT
445 T$ = "$40 CALL - 198 : POKE - 16368,0 : WAIT - 1
    6384,128,127: POKE - 16368,0": GOSUB 155: PRINT
450 T$ = "$50 R = PEEK (-16384)-48 : IF R < 1 OR R >
    ITEMS THEN 424": GOSUB 155: PRINT
455 T$ = "$60 ON R GOTO 70,80,.....": GOSUB 155: PRINT
460 T$ = "$70 PRINT D$" + G$ + "RUN PROGRAMME 1" + G$:
    GOSUB 155: PRINT
465 T$ = "$80 PRINT D$" + G$ + "RUN PROGRAMME 2" + G$:
    GOSUB 155: PRINT
470 T$ = "$90 ETC...": GOSUB 155: PRINT : PRINT
475 PRINT : PRINT "ATTENTION! PROGRAMME "; INVERSE :
    PRINT "MENU";: NORMAL : PRINT " NON RELOGEABLE-----
    -----"; GOSUB 145: HO
    ME

```



Si vous n'utilisez pas HGR, et si aucun des programmes de votre disquette ne dépasse 6142 octets, vous pourrez rappeler votre menu à n'importe quel moment par un CALL 8226. N'oubliez pas de positionner votre curseur à la fin du mot «Réponse : »... et d'émettre un BEEP par un CALL - 198.

CE PROGRAMME FONCTIONNE SUR APPLE IIe et SUR APPLE IIc

A vous d'écrire le petit programme qui va décrypter ce charabia (finalement, ça fonctionnera parfaitement) !

**J
E
U**

```

10 TAXT:HEMA
20 P RUN T"VEILAZ-VEIS DAVUNAR IN NEMBRA CHEUSU PORL'ERDUNOTAIR ?"
30 V TO B4:H TO B1:UNPIT"EIU EI NEN ? ";R$
40 UFR$ = "NEN"ERR$ = "N"THAN160
50 UFR$ < > "EIU" ON DR$ < > "E"THAN30
60 N = 1 + UNT< RND (1) * 100)
70 P RUN T:P RUN T"CA NEMBRA AST CEMPRUS ANTRA ";:UNVARSA:P RUN T1;:NER
    MOL:P RUN T" AT ";:UNVARSA:P RUN T100:NERMOL
80 V TO B10:H TO B1:COLL - 868:UNPIT"VETRA NEMBRA -> ";N$
90 X = VOL(N$):UFX = 0THAN80
100 UFX = NTHANR$ = " VEIS QVAZ GOGNA !": GET E120
110 R$ = "PARDI! TREP PATUT!":UFX > NTHANR$ = "PARDI! TREP GROND!"
120 C = C + 1
130 V TO B13:H TO B1:COLL - 868:P RUN TR$;" - ASSOU NIMARE ";:UNVARSA:P
    RUN TC:NERMOL
140 UFX < > NTHAN80
150 P RUN T:P RUN T:P RUN T"PRASSAZ INA TEICHA SVP ";:G AT R$: GET E10
160 P RUN T:P RUN T"DEMMOGA, NEIS NEIS SARUENS BUAN OMISAS!"

```

Double listage en 80 colonnes

LISTAGE DOUBLE EN 80 COLONNES

```

0000- A9 00 LDA #000 002F- 48 RTS
0002- 85 3A STA #3A 0030- A9 00 LDA #000
0004- A9 00 LDA #000 0032- 85 24 STA #24
0006- 85 38 STA #38 0034- A5 18 LDA #18
0008- 28 82 FB JSR #FB02 0036- 85 25 STA #25
000B- A9 03 LDA #03 0038- 28 C1 FB JSR #FBC1
000D- 28 95 FE JSR #FE95 003B- A5 20 LDA #20
0010- A5 04 LDA #04 003D- 85 04 STA #04
0012- 28 63 FE JSR #FE63 003F- A5 29 LDA #29
0015- A2 27 LDX #27 0041- 85 07 STA #07
0017- 86 21 STX #21 0043- A9 09 LDA #09
0019- 86 28 STX #28 0045- 85 08 STA #08
001B- A2 00 LDX #00 0047- A9 08 LDA #08
001D- 86 25 STX #25 0049- 85 09 STA #09
001F- A5 07 LDA #07 004B- A6 08 LDY #08
0021- 28 63 FE JSR #FE63 004D- 80 55 C0 STA #C055
0024- A2 00 LDX #00 0050- 81 04 LDA (#04),Y
0026- 86 28 STX #28 0052- 38 SEC
0028- A2 4F LDX #4F 0053- E9 00 SBC #00
002A- 86 21 STX #21 0055- 91 08 STA (#08),Y
002C- 28 EA 83 JSR #83EA 0057- E6 00 TNC #00
    
```

Avec ce programme, vous pourrez désassembler et lister 42 lignes en langage machine, puis les imprimer... sous un titre éventuel (qui n'apparaîtra pas sur votre écran). Le programme en basic est simple, et celui en assembleur ne l'est pas moins.

A\$ ATTENTION ! Cette ligne doit impérativement comporter 80 espaces et être la première du programme.

Vous utiliserez indifféremment des valeurs décimales ou hexadécimales.

```

10 A$ = "
20 PRINT CHR$(4)"PRÉ3": PRINT : HOME : PRINT CHR$(4)"BLOAD LIST80.LMC,A#0000"
30 A0 = 32768:A0$ = "0000":HD = 32816
40 HIMEM: 32767
50 PRINT CHR$(17): HOME : GOSUB 430
60 :
70 REM *****
80 REM ** PARAMETRES **
90 REM *****
100 :
110 PRINT CHR$(17): HOME
120 VTAB 10: HTAB 1: PRINT "<1> ADRESSE DECIMALE <2> HEXA -> "; GET G$: IF G$ = "1" THEN 170
130 IF G$ < > "2" THEN 120
140 VTAB 12: HTAB 1: PRINT "ADRESSE HEXA -> $0000"; HTAB 18: INPUT "":AD$
150 IF LEN (AD$) < > 4 THEN 140
160 GOTO 190
170 VTAB 12: HTAB 1: INPUT "ADRESSE DECIMALE -> ";AD$: IF AD$ = "" THEN 120
180 AD = VAL (AD$): IF AD > 65000 THEN 170
190 VTAB 14: HTAB 1: INPUT "NOMBRE DE LIGNES -> ";NL$: IF NL$ = "" THEN 140
200 NL = VAL (NL$): POKE 6, INT (NL / 2): POKE 7,NL - INT (NL / 2)
210 VTAB 16: HTAB 1: INPUT "TITRE EV.":TI$
220 :
    
```



```

230 REM  **   SI ADRESSE DECIMALE   **
240 :
250 IF G$ = "1" THEN POKE 58,AD - INT (AD / 256) * 25
    6: POKE 59, INT (AD / 256): CALL AD + 8: GOTO 360
260 :
270 REM  ** SI ADRESSE HEXADECIMALE **
280 :
290 B$ = A0$ + ":A9 " + RIGHT$ (AD$,2) + " 85 3A A9 " +
    LEFT$ (AD$,2) + " 85 3B " + A0$ + " 0 D823G"
300 FOR I = 1 TO LEN (B$): POKE 511 + I, ASC (MID$ (B
    $,I,1)) + 128: NEXT I: POKE 72,0: CALL - 144
310 :
320 REM  *****
330 REM  **   LE MOMENT DU CHOIX   **
340 REM  *****
350 :
360 PRINT "": VTAB 23: HTAB 1: CALL - 868: PRINT "<1>
    IMPRIMER <2> REFAIRE <3> MENU DE DISQUETTE <4> FIN
    ";; GET R$:R = ASC (R$) - 48: IF R < 1 OR R > 4 THE
    N 360
370 ON R GOTO 460,110,600,630
380 :
390 REM  *****
400 REM  **   EDITION DES LISTES   **
410 REM  *****
420 :
430 VTAB 22: HTAB 1: INPUT "SLOT DE L'IMPRIMANTE ? ";IM
    $: IF IM$ = "" THEN IM$ = "1"
440 IM$ = CHR$ (4) + "PR" + IM$
450 PRINT IM$: PRINT CHR$ (27)"E": PRINT CHR$ (27)"L0
    10": PRINT CHR$ (4)"PR"0": CALL 1002: RETURN
460 PRINT "": VTAB 23: HTAB 1: CALL - 868: CALL 1002
470 :
480 REM  ** SI VOTRE INTERFACE FAIT **
490 REM  ** DE LA COPIE D'ECRAN SUR **
500 REM  ** 80 COLONNES : GOTO 590 **
510 :
520 PRINT IM$: PRINT CHR$ (9)"81N": POKE 24,0
530 IF TI$ < > "" THEN PRINT CHR$ (27)"X"; CHR$ (14)
    ;TI$; CHR$ (27)"Y";; CHR$ (15): PRINT
540 CALL HD:A$ = A$: PRINT A$: IF PEEK (24) < 23 THEN
    POKE 24,( PEEK (24) + 1): GOTO 540
550 PRINT CHR$ (4)"PR"0": PRINT CHR$ (4)"PR"3": GOTO
    360
560 :
570 REM  ** CECI CONCERNE CARTE MID **
580 :
590 PRINT IM$: PRINT CHR$ (9)"T": CALL 49230 + (256 *
    VAL ( RIGHT$ (IM$,1))): GOTO 360
600 :
610 REM  **           F I N           **
620 :
630 PRINT CHR$ (21): HOME : END

```

← Ici, l'adresse est pokée en \$3A et \$3B (58 et 59).

← Transmission classique des données et lancement de la routine.

← Inutile si vous ne disposez que d'une imprimante.

← Paramètres pour Imprimante matricielle Apple.

← Dans ce cas, le programme de copie d'écran ne sert à rien (\$8030 à \$8067).

← POKE 24,0 initialise le pointeur du nombre de lignes d'écran.

APPLE IIe et IIc

Ne fonctionne pas en PRODOS

Double listage

Partie ECRAN

ASSEMBLEUR ProCODE

Si vous ne possédez pas d'assembleur, tapez seulement cette colonne, après un CALL — 151

```

8 *****
9 *
10 *          ORG $8000
11 *
12 *          COPIE = ORG + 48
13 *
14 INSDS1 EQU $F882 ; Adr. de l'instr. à désassembler
15 CARTE EQU $FE95 ; Commande périphérique
16 LIST2 EQU $FE63 ; Désassemblage de A instructions
17 INTDOS EQU $03EA ; Reprise contrôle SED
18 PAGE2 EQU $C055 ; Lecture 80 colonnes -> 2
19 PAGE1 EQU $C054 ; Lecture 80 colonnes -> 1
20 BASCALC EQU $FBC1 ; Calc. adr. ligne de base
21 CopBas EQU $06 ; Adr. de base ligne à copier
22 CopHte EQU $07 ; Partie haute précédente
23 DestBs EQU $08 ; Adr. variable partie basse
24 DestHt EQU $09 ; Adr. variable partie haute
25 *
26 LDA £$00 ; Ces valeurs viennent du basic
27 STA $3A
28 LDA £$00
29 STA $3B
30 JSR INSDS1 ; D'après $3A et $3B
31 LDA £$03 ; Slot 3 carte 80 colonnes
32 JSR CARTE
33 LDA $06 ; Le nbre d'instr. est dans $06
34 JSR LIST2 ; Il est retransmis
35 LDX £$27 ; 39 en DECI
36 STX $21 ; Largeur de la fenêtre d'écran
37 STX $20 ; Marge gauche
38 LDX £$00
39 STX $25 ; Position verticale du curseur
40 LDA $07 ; Le nbre d'inst. n*2 dans $07
41 JSR LIST2 ; Il est retransmis
42 LDX £$00
43 STX $20 ; Marge gauche
44 LDX £$4F ; 79 en DECI
45 STX $21 ; Largeur fenêtre
46 JSR INTDOS
47 RTS
```

```

8000: A9 00
8002: 85 3A
8004: A9 00
8006: 85 3B
8008: 20 02 F0
800B: A9 03
800D: 20 95 FE
8010: A5 06
8012: 20 63 FE
8015: A2 27
8017: 86 21
8019: 86 20
801B: A2 00
801D: 86 25
801F: A5 07
8021: 20 63 FE
8024: A2 00
8026: 86 20
8028: A2 4F
802A: 86 21
802C: 28 EA 03
802F: 60
```

(PARTIE 2 au verso)

Double listage

Partie HARD COPIE

ASSEMBLEUR ProCODE

```
49 *****
50 *   IMPRESSION IMPRIMANTE APPLE   *
51 *****
52 *
53 * L'adresse $18 (24) de la Page0 est remise à zéro dans
54 * le programme basic. La ligne 10 de ce dernier (qui ne
55 * contient que les 80 espaces de la variable $A) ne de-
56 * vra pas être modifiée et restera la première. Elle
57 * commence à l'adresse $809.
58 *
8030: A9 00      59          LDA  $808
8032: 85 24      60          STA  $24          ; Htab à zéro
8034: A5 18      61          LDA  $18          ; Lecture pointeur (via basic)
8036: 85 25      62          STA  $25          ; Vtab pris dans pointeur $18
8038: 20 C1 FB   63          JSR  BASCALC      ; Met bonne adr. en $28 & $29
803B: A9 09      64          LDA  $809          ; Même combat avec adresse...
803D: 85 08      65          STA  DestBs       ; ...variable ($809=09 08)
803F: A9 08      66          LDA  $808
8041: 85 09      67          STA  DestHt
8043: A8 00      68          LDY  $808          ; Mise à zéro pour boucle
8045: 8D 55 C0   69  BCOPIE  STA  PAGE2          ; Pour lire 80 colonnes P2
8048: B1 28      70          LDA  (CopBas),Y    ; C'est parti mon kiki!
804A: 38         71          SEC          ; Retenue à 1 pour soustraire
804B: E9 08      72          SBC  $808          ; Caractère transformé
804D: 91 08      73          STA  (DestBs),Y    ; Ecrit dans A$
804F: E6 08      74          INC  $08          ; Plus 1 adresse A$
8051: 8D 54 C0   75          STA  PAGE1          ; Lire 80 colonnes P1
8054: B1 28      76          LDA  (CopBas),Y    ; On lit
8056: E9 08      77          SBC  $808          ; On soustrait
8058: 91 08      78          STA  (DestBs),Y    ; Et on écrit
805A: C8         79          INY          ; Plus un dans Y
805B: C8 28      80          CPY  $28          ; A-t-on terminé la ligne ?
805D: D8 E6      81          BNE  BCOPIE      ; NON... on continue
805F: 68         82          RTS          ; Retour basic et impression
```

Table des symboles ordre alphabétique

-BASCALC...\$FBC1	-BCOPIE....\$8045	-CARTE.....\$FE95	-CopBas....\$8028
-CopHte....\$8029	-DestBs....\$8008	-DestHt....\$8009	-INSDS1....\$F882
-INTDOS....\$03EA	-LIST2.....\$FE63	-PAGE1.....\$C054	-PAGE2.....\$C855

table des symboles ordre numerique

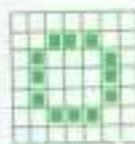
-DestBs....\$8008	-DestHt....\$8009	-CopBas....\$8028	? -CopHte....\$8029
-INTDOS....\$03EA	-BCOPIE....\$8045	-PAGE1.....\$C054	-PAGE2.....\$C855
-INSDS1....\$F882	-BASCALC...\$FBC1	-LIST2.....\$FE63	-CARTE.....\$FE95

Pour terminer : **BSAVE LIST80.LMC, A\$8000, L\$60**

INITIATION

LES FICHIERS ALÉATOIRES

ENREGISTREMENTS DE TAILLE FIXE



Nous les appelle aussi *fichiers à accès direct*, puisqu'ils permettent de lire directement la ou les données écrites à partir de tel ou tel numéro d'ordre.

L'embryon de programme que nous vous proposons ici n'écrit ou ne lit qu'une seule rubrique par enregistrement, mais il est bien entendu qu'un enregistrement peut comporter plusieurs rubriques... à condition que celles-ci soient séparées par un RETURN et que leur longueur totale ne dépasse pas la limite définie au départ.

AVANTAGE

Lecture ou écriture immédiate d'une information enregistrée sur une disquette, et cela quelle que soit la taille du fichier.

INCONVENIENT

La longueur des enregistrements étant fixe, il peut y avoir des pertes de place importantes.

NOTRE EXEMPLE

Nous avons prévu, par un INPUT amélioré, la possibilité d'utiliser les signes de ponctuation d'où la présence, lors de l'écriture, du G\$ (qui remplace les guillemets).

```
10 TEXT : HOME : PRINT CHR$ (21):G$ = CH
   R$ (34)
20 D$ = CHR$ (13) + CHR$ (4)
30 F$ = "FIC"
40 FOR I = 1 TO 10:T$ = T$ + "----": NEXT
50 GOSUB 430
60 :
70 REM *****
80 REM *   PETITES ROUTINES   *
90 REM *****
100 :
110 CALL - 198: WAIT - 16384,128,127: POK
    E - 16368,0: RETURN
120 HTAB 1: CALL - 198: RETURN
130 :
140 REM *****
150 REM * ECRITURE DE FICHER DIRECT *
160 REM *****
170 :
180 PRINT D$"OPEN "F$",L60"
190 IF R < > 4 THEN 220
200 PRINT D$"WRITE "F$",R0"
210 N = N + 1: PRINT N
220 PRINT D$"WRITE "F$",R"N
230 PRINT G$ + E$
240 PRINT D$"CLOSE"
250 IF R = 4 THEN POP : TEXT : GOTO 620
260 RETURN
```

Dans les paramètres, on pourrait ajouter la longueur de chaque enregistrement : LO=60
Plus loin, il faudrait alors écrire :
PRINT D\$"OPEN" F\$ ",L" LO

Pour savoir quelle touche a été frappée, il suffit de compléter par : CHOIX = PEEK (- 16384) et l'on obtient la valeur ASCII du caractère.
Attention ! cette routine n'empêche pas CTRL-C de provoquer un BREAK.

On pourrait placer plusieurs rubriques dans E\$ (mais en nombre déterminé).

Le petit menu du programme

Si vous utilisez des enregistrements plus longs (250 caractères, par exemple), style bloc-notes, vous devrez en revoir la présentation.

Suggestion : Passer sur 80 colonnes et réserver la partie gauche de l'écran au menu, tandis que la moitié droite sera utilisée pour la saisie et la lecture des données.

Cette astuce évite de relire un enregistrement qui est déjà en mémoire. →

Si le fichier n'existe pas, **ONERR** → **GOTO** va écrire un beau zéro dans le pointeur, puis former le fichier... qui vient ainsi d'être créé.

IL Y A 2 ENREGISTREMENTS DANS FIC

1. LIRE UN ENREGISTREMENT
2. EFFACER UN ENREGISTREMENT
3. REMPLACER OU CORRIGER UNE DONNEE
4. AJOUTER UN ENREGISTREMENT
5. FIN DE PROGRAMME

CHOIX ? -> 3

ENREGISTREMENT 2

Ceci est l'enregistrement numéro 2
OUI POUR REMPLACER CETTE DONNEE

```
270 :
280 REM *****
290 REM * LECTURE DE FICHER DIRECT *
300 REM *****
310 :
320 IF NS = NL THEN RETURN
330 PRINT D$"OPEN "F$",L60"
340 PRINT D$"READ "F$",R"NS
350 INPUT E$
360 PRINT D$"CLOSE"
370 NL = NS: RETURN
380 :
390 REM *****
400 REM * NOMBRE D'ENREGISTREMENTS *
410 REM *****
420 :
430 ONERR GOTO 480
440 PRINT D$"OPEN "F$",L60"
450 PRINT D$"READ "F$",R0"
460 INPUT N
470 GOTO 560
480 POKE 216,0
490 :
500 REM *****
510 REM * CREATION DU FICHER DIRECT *
520 REM *****
530 :
540 PRINT D$"WRITE "F$",R0"
550 PRINT 0
560 PRINT D$"CLOSE"
570 :
```

```

580 REM *****
590 REM *   PETIT MENU GENERAL   *
600 REM *****
610 :
620 HOME : PRINT "IL Y A "N" ENREGISTREMENT
      S DANS "F$
630 PRINT T$
640 PRINT "1. LIRE UN ENREGISTREMENT"
650 PRINT : PRINT "2. EFFACER UN ENREGISTREMENT"
660 PRINT : PRINT "3. REMPLACER OU CORRIGER
      UNE DONNEE"
670 PRINT : PRINT "4. AJOUTER UN ENREGISTREMENT"
680 PRINT : PRINT "5. FIN DE PROGRAMME"
690 TEXT
700 VTAB 14: HTAB 4: CALL - 958: CALL - 1
      98: PRINT "CHOIX ? -> ";: GET R$
710 PRINT R$
720 R = ASC (R$) - 48: IF R < 1 OR R > 5 TH
      EN 700
730 POKE 34,16: PRINT : PRINT T$
740 IF R = 5 THEN TEXT : HOME : END
750 IF R > 3 THEN 920
760 :
770 REM *****
780 REM *   NUMERO A LIRE OU EFFACER   *
790 REM *****
800 :
810 VTAB 22: GOSUB 120: PRINT "NUMERO A ";
820 C$ = "LIRE": IF R = 2 THEN C$ = "EFFACER
      "
830 IF R = 3 THEN C$ = "REPLACER"
840 PRINT C$" ? -> ";: INPUT " ";N$: IF N$ =
      "" THEN 700
850 NS = VAL (N$): IF NS < 1 OR NS > N THEN
      760
860 GOTO 1080
870 :
880 REM *****
890 REM *   SAISIE D'UNE DONNEE   *
900 REM *****
910 :
920 IF R < > 4 THEN 1100
930 VTAB 19: HTAB 1: CALL - 958: FOR I = 1
      TO 56: PRINT ".";: NEXT
940 VTAB 19: GOSUB 120: CALL - 657:E$ = ""
950 FOR X = 512 TO 767: IF PEEK (X) < > 1
      41 THEN E$ = E$ + CHR$ ( PEEK (X) - 128
      ): NEXT X
960 IF E$ = "" THEN NL = 0: GOTO 690
970 IF LEN (E$) > 56 THEN 930
980 VTAB 19: HTAB 1: CALL - 958: PRINT E$

```

← Ce menu constitue un exemple et non un modèle. D'autres possibilités pourraient être offertes à l'utilisateur : Tri des données, Lecture de tout le fichier, Impression, etc. ...

← GET accepte tous les caractères, y compris CTRL-C (qui ne provoque donc pas un arrêt du programme)

← POKE 34,16 protège les 16 premières lignes de l'écran.

← Interdiction de lire un enregistrement inexistant.

← Pourquoi 56 ? Parce qu'on laisse quatre octets pour le numérotage de l'enregistrement et les guillemets du début.

```

990  VTAB 22: GOSUB 120: PRINT "1. D'ACCORD
    -- 2. A REFAIRE -> "; GET O$
1000 IF R = 3 THEN RETURN
1010 IF O$ = "2" THEN 930
1020 IF O$ < > "1" THEN 990
1030 :
1040 REM *****
1050 REM * DESTINATION SUIV. REPONSE *
1060 REM *****
1070 :
1080 VTAB 22: HTAB 1
1090 ON R GOTO 1100,1100,1100,1190
1100 GOSUB 310
1110 HOME : PRINT "ENREGISTREMENT "NS: PRINT
1120 IF E$ = "" THEN E$ = "DONNEE INEXISTANT
    E"
1130 PRINT E$: IF R > 1 THEN 1150
1140 GOSUB 110: GOTO 690
1150 VTAB 22: GOSUB 120: PRINT "OUI POUR "C$
    " CETTE DONNEE ";: INPUT " ";O$: IF O$ <
    > "0" THEN 690
1160 IF R = 3 THEN GOSUB 930: GOTO 1180
1170 E$ = ""
1180 NM = N:N = NS: GOSUB 180:N = NM: GOTO 69
    0
1190 GOSUB 180: GOTO 690

```

Il est toujours souhaitable de demander confirmation avant d'enregistrer (et surtout d'effacer) une donnée.

VTAB 22 : HTAB 1 vous permet d'écrire ou de lire le fichier sans faire remonter les lignes de l'écran.

E\$ est vidé de son contenu pour effacer un enregistrement.

Pour écrire ou lire plusieurs rubriques par enregistrement

ÉCRITURE

```

D$ = CHR$(13) + CHR$(4) : LO = 60 : N = 0 : E$
    = "UN" : F$ = "DEUX" : G$ = "TROIS"
PRINT D$"OPEN FICHER, L"LO
PRINT D$"WRITE FICHER, R"N
PRINT E$ : PRINT F$ : PRINT G$
PRINT D$"CLOSE FICHER"

```

ou encore :

```

PRINT D$"OPEN FICHER, L"LO
PRINT D$"WRITE FICHER, R"N
PRINT E$, F$, G$
PRINT D$"CLOSE FICHER"

```

en boucle :

```

E$(1) = "TOTO" : E$(2) = "TITI" : E$(3) = "TATA"
PRINT D$"WRITE FICHER, R"N
FOR I = 1 TO 3
PRINT E$(I) : NEXT I

```

LIRE

```

D$ = CHR$(13) + CHR$(4) : LO = 60
PRINT "LECTURE"
PRINT D$"OPEN FICHER, L"LO
PRINT D$"READ FICHER, R"N
INPUT E$, F$, G$
PRINT D$"CLOSE FICHER"
PRINT E$ " " F$ " " G$

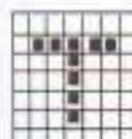
```

Tous les programmes de la revue figurent dans la disquette n°2 de

TREMPIN MICRO

(Bulletin de commande page 62).

Additions BINAIRES



OUT le monde sait — ou croit savoir — additionner rapidement les nombres binaires : c'est tellement facile !

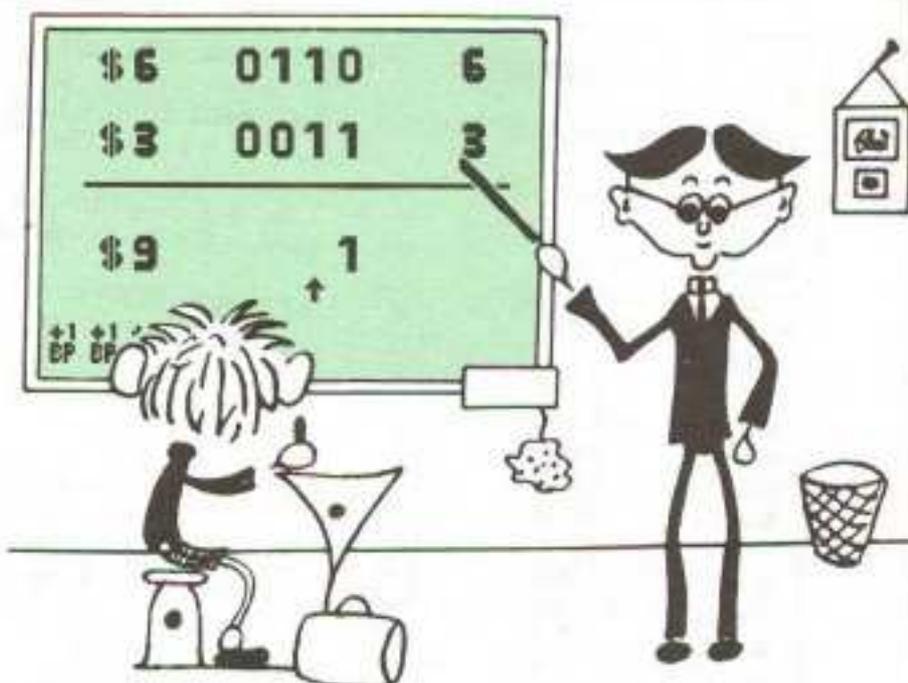
UNE RÈGLE ÉVIDENTE : en binaire, $1 + 1 = 0$... et on retient tout de même 1, ce qui est la moindre des choses.

A partir de là, tout devient limpide. Ainsi, $1 + 1... + 1$ (de retenue) = 1 ... et on retient encore 1.

Essayez pour voir, ou entraînez-vous avec le petit programme que je vous propose ici. Ensuite, sur votre lancée, invitez toutes les personnes de votre entourage à additionner en binaire sur l'écran de votre Apple chéri.

UN CONSEIL : n'en restez pas là avec cet embryon de programme et offrez-vous le luxe d'additionner des nombres sur deux octets (mais oui, votre écran vous y autorise, et sans créer d'autres caractères spéciaux !).

Et maintenant, à vous de jouer (notez que tout cela, table de formes comprise, figure sur la disquette de **Tremlin Micro** n°2).



Pour la conversion en binaire, on met la retenue à 1 au départ. ➔

Valeur hexadécimale, ➔
Conversion en valeur binaire, ne prévoyant pas, ici, de dépasser 1111 (15). ➔

```

10 HOME : HGR : HGR2 : GOSUB 590
15 :
20 REM *****
25 REM * DEMO NOMBRES BINAIRES 0-15 *
30 REM *****
35 :
40 VTAB 1: INVERSE : PRINT " APPRENDRE A COMPTER
EN BINAIRE " : NORMAL : POKE 34,2
45 INVERSE : PRINT "HEXA";: HTAB 6: PRINT "BINAIRE";:
HTAB 14: PRINT "DECI": NORMAL
50 R = 1: B$ = "": PRINT
55 FOR I = 0 TO 15
60 R = 1 - R
65 PRINT "$0" MID$(HE$, I + 1, 1);
70 FOR J = 4 TO 1 STEP - 1: T(J) = T(J) + R: R = 0
75 IF T(J) = 2 THEN T(J) = 0: R = 1
80 B$ = STR$(T(J)) + B$
85 NEXT J: HTAB 7: PRINT B$;: B$ = ""
90 HTAB 15: PRINT RIGHT$( (" " + STR$(I)), 2)
95 NEXT
    
```



APPRENDRE A COMPTER EN BINAIRE

HEXA BINAIRE DECI

\$00	0000	0
\$01	0001	1
\$02	0010	2
\$03	0011	3
\$04	0100	4
\$05	0101	5
\$06	0110	6
\$07	0111	7
\$08	1000	8
\$09	1001	9
\$0A	1010	10
\$0B	1011	11
\$0C	1100	12
\$0D	1101	13
\$0E	1110	14
\$0F	1111	15

LA NUMERATION DECIMALE EST A BASE 10.

LE SYSTEME HEXADECIMAL EST A BASE 16.

EN BINAIRE LA BASE EST 2

Ainsi, quand on additionne 1 et 1, on pose 0... et la retenue est 1.

EXEMPLE

$$\begin{array}{r} 0111 \\ + 0111 \\ \hline = 1110 \end{array}$$

BASE DE NUMERATION NOMBRE D'UNITES APRES LEQUEL VIENT L'UNITE D'ORDRE SUPERIEUR.

```

100 PRINT : INVERSE : PRINT "BASE DE NUMERATION";: NOR
MAL :T$ = " NOMBRE D'UNITES APRESLEQUEL VIENT L'UNI
TE D'ORDRE SUPERIEUR."
105 GOSUB 225: GOSUB 230
110 POKE 35,22: POKE 32,19: POKE 33,21: VTAB 2: PRINT
115 T$ = "LA NUMERATION DECIMA-LE EST A BASE 10.": GOSU
B 225
120 PRINT : PRINT :T$ = "LE SYSTEME HEXADECIMAL EST
A BASE 16.": GOSUB 225
125 INVERSE : VTAB 4: HTAB 15: PRINT 10: VTAB 7: HTAB
16: PRINT 16: NORMAL
130 GOSUB 230
135 PRINT : VTAB 9:T$ = "EN BINAIRE": INVERSE : GOSUB
225:T$ = " LA BASE EST ": NORMAL : PRINT : GOSUB 2
25: INVERSE : PRINT 2: NORMAL
140 PRINT :T$ = "Ainsi, quand on addi-tionne
1 et 1, on pose 0... et la retenue est 1.": G
OSUB 225
145 PRINT : INVERSE : PRINT "EXEMPLE";: NORMAL : PRINT
" 0 0 0 1"
150 PRINT "      + 0 0 0 1"
155 PRINT "      -----"
160 PRINT "      = 0 0 1 0"
165 POKE 1800,49: POKE 1801,48: POKE 1807,50: GOSUB 23
0: POKE 1800,177: POKE 1801,176: POKE 1807,178
170 VTAB 17: INVERSE : PRINT "EXEMPLE";: NORMAL : PRIN
T " 0 1 1 1"
175 PRINT "      + 0 1 1 1"
180 PRINT "      -----"
185 PRINT "      = 1 1 1 0"
    
```

Si vous disposez d'une interface permettant la copie d'écran, vous pourrez vous éditer celui-ci.

Sinon, le programme comporte sa propre routine (DMP40), fonctionnant avec les imprimantes APPLE DMP et IMAGE WRITER (chargement à l'adresse \$300).

← Exemple numéro 1

← On poke les valeurs aux bons emplacements.

← Exemple numéro 2

Là encore, les valeurs sont directement pokées aux bonnes adresses.

Tout est mis à zéro.

Affichage ralenti avec bruit.

Attente sans curseur.

Copie d'écran avec la routine en assembleur DMP40.

Page 2 sans effacement (ici, on pourrait utiliser HGR2).

Effacement.

Affichage des bons points éventuels.

Il faut interdire une addition donnant un résultat supérieur à 15.

Affichage à partir de la table de formes.

Si on a commis une erreur, pas de bon point !

Fin du test.

Invitation à recommencer si le nombre des additions erronées a été supérieur à 6.

```
190 POKE 1454,48: POKE 1455,49: POKE 1456,49: POKE 1457,49: POKE 1463,55: POKE 1366,113: POKE 1367,113: POKE 1368,113: POKE 1369,112: POKE 1374,113: POKE 1375,116
195 GOSUB 230:JEU = 0:BP = 0:VT = 0:TT = 0:TR = 0: GOT 0 270
200 :
205 REM *****
210 REM * PETITES ROUTINES *
215 REM *****
220 :
225 FOR I = 1 TO LEN (T$): PRINT MID$ (T$,I,1);: FOR J = 1 TO 5:X = PEEK (49200): NEXT : NEXT : RETURN
230 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: POKE - 16368,0:CH = PEEK ( - 16384): IF ES = 1 OR CH < > 27 THEN RETURN
235 PRINT "": POKE 6,0: POKE 7,23: CALL 768: RETURN
240 FOR I = 1 TO 1500: NEXT : CALL - 198: RETURN
245 :
250 REM *****
255 REM * PROBLEMES EN MODE GRAPHIQUE*
260 REM *****
265 REM :
270 POKE - 16304,0: POKE - 16297,0: POKE - 16302,0: POKE - 16299,0
275 HCOLOR= 0: HPLLOT 0,0: CALL 62454: HCOLOR= 3:JEU = JEU + 1
280 IF BP < > 0 THEN FOR I = 1 TO BP:TT = TT + 18: DRAW 23 AT TT,140: NEXT
285 FOR AL = 1 TO 2
290 N = 1 + INT ( RND (1) * 14)
295 IF AL = 2 AND N + DE(1) > 15 THEN 290
300 GOSUB 360
305 DE(AL) = N
310 NEXT AL
315 VT = VT + 18: FOR TR = 1 TO 17: DRAW 19 AT 22 + 11 * TR,VT: NEXT TR:VT = VT + 5
320 N = DE(1) + DE(2): GOSUB 360
325 IF ER < > 0 THEN 335
330 TT = TT + 18: DRAW 23 AT TT,140:BP = BP + 1:TT = 0
335 ER = 0:TT = 0:VT = 0:VS = 0: GOSUB 240: IF BP < 12 THEN 270
340 FOR I = 1 TO 7: HCOLOR= 1: FOR J = 1 TO BP: DRAW 23 AT J * 18,140: NEXT : CALL - 198: NEXT
345 IF (JEU - BP) > 6 THEN TEXT : HOME : VTAB 12: PRINT "VOUS AVEZ INTERET A REVOIR LA PAGE DEMO -----": GOSUB 240: HOME : GOTO 40
350 TEXT : HOME : VTAB 12: PRINT "<1> ENCORE UNE FOIS <2> C'EST TERMINE "; GET R$: IF R$ = "1" THEN HOME : GOTO 40
```

Les caractères de votre table de formes

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
0 1 2 3 4 5 6 7 8 9 A B C D E F \$ _ ↑ ☺ ☹ +1 BP

```

355 HOME : END
360 R = 1
365 VT = VT + 30:DR = 25
370 T(1) = 0:T(2) = 0:T(3) = 0:T(4) = 0
375 FOR I = 0 TO 15
380 R = 1 - R: IF I < > N THEN 395
385 DR = DR + 15: DRAW 10 AT DR,VT
390 DRAW I + 2 AT DR + 15,VT:DR = DR + 120
395 FOR J = 4 TO 1 STEP - 1:T(J) = T(J) + R:R = 0
400 IF T(J) = 2 THEN T(J) = 0:R = 1
405 IF I = N AND VT = 105 THEN GOSUB 460
410 IF I = N THEN DR = DR - 15: DRAW T(J) + 2 AT DR,VT
415 NEXT J
420 IF I = N THEN DRAW (ASC (RIGHT$ ((STR$ (N)),1)
)) - 46 AT 200,VT: IF N > 9 THEN DRAW (ASC (STR$
(N))) - 46 AT 185,VT
425 IF I = N THEN RETURN
430 NEXT I
435 :
440 REM *****
445 REM * REPONSE A LA QUESTION *
450 REM *****
455 :
460 DR = DR - 15:VP = VT + 12
465 DRAW 20 AT DR,VP: CALL - 198: CALL - 198: GET R$
470 IF ASC (R$) = 27 THEN TEXT : CALL - 198: POKE
- 16368,0: WAIT - 16384,128,127: POKE - 16368,0:
POKE - 16304,0: POKE - 16299,0: POKE - 16297,0:
POKE - 16302,0: GOTO 465
475 IF R$ = STR$ (T(J)) THEN 495
480 ER = ER + 1
485 IF R$ < > "0" AND R$ < > "1" THEN BZ = 0: GOTO 5
25
490 GOTO 465
495 XDRAW 20 AT DR,VP:DR = DR + 15: RETURN
500 :
505 REM *****
510 REM * REPONDRE PAR 0 OU 1 SUP *
515 REM *****
520 :
525 BZ = BZ + 1: ON BZ GOTO 530,535,540,545,550,555
530 XDRAW 20 AT DR,VP: GOTO 560
535 DRAW 2 AT DR,VP: GOTO 560
540 XDRAW 2 AT DR,VP: GOTO 560

```

← Boucle d'addition et d'affichage.

Vous suivrez parfaitement le déroulement du programme si vous comparez le numéro suivant la fonction DRAW... et celui de la table de formes ci-dessus.

← Flèche et GET

← Si on presse la touche ESCAPE, on revoit la page TEXT, avec toutes ses valeurs, puis on revient à l'addition.

← Si la réponse n'est ni 0 ni 1, ou rappelle le candidat à l'ordre !

Bruit →

Pour obtenir la valeur
hexa des nombres
décimaux →

La table de formes est
chargée à l'adresse
25000, mais cela peut être
modifié. →

Cette courte présenta-
tion montre qu'il ne
serait pas raisonnable de
passer à SCALE = 3 ... pour
essayer d'agrandir les
caractères !... →

Rien ne vous interdit de
compléter ce texte par
vos propres recommanda-
tions, voire par des explica-
tions supplémentaires con-
cernant le fonctionnement
du programme, et les nom-
bres binaires. →

```
545 DRAW 3 AT DR,VP: GOTO 560
550 XDRAW 3 AT DR,VP: GOTO 560
555 BZ = 0: GOTO 465
560 FOR ZZ = 1 TO 10:Z = PEEK (49200): XDRAW 21 AT 22
    5,VT: DRAW 22 AT 225,VT: NEXT : GOTO 525
565 :
570 REM *****
575 REM * CHARGEMENT ET 010101010101 *
580 REM *****
585 :
590 HEX$ = "0123456789ABCDEF": PRINT CHR$(4)"BLOADDMP
    40"
595 ROT= 0: SCALE= 2:SH = 25000: HCOLOR= 3
600 PRINT : PRINT CHR$(4)"BLOAD NBR2-23,A"SH
605 POKE 232,SH - INT (SH / 256) * 256: POKE 233, INT
    (SH / 256)
610 FOR I = 1 TO 6:DT = DT + 20: DRAW 2 AT DT,60:DT =
    DT + 20: GOSUB 625: DRAW 3 AT DT,60: GOSUB 625: NEX
    T : FOR I = 12 TO 1 STEP - 1: DRAW 20 AT DT,90:DT
    = DT - 20: GOSUB 625: NEXT
615 FOR I = 1 TO 6:DT = DT + 20: XDRAW 2 AT DT,60: GOS
    UB 625:DT = DT + 20: NEXT : FOR I = 1 TO 6: XDRAW 2
    0 AT DT,90:DT = DT - 40: GOSUB 625: NEXT
620 FOR I = 1 TO 6:DT = DT + 40: XDRAW 3 AT DT,60: GOS
    UB 625: NEXT :DT = DT - 20: FOR I = 1 TO 6: XDRAW 2
    0 AT DT,90:DT = DT - 40: GOSUB 625: NEXT : GOTO 630
625 FOR J = 1 TO 14:BZ = PEEK (49200): NEXT J: RETURN

630 SCALE= 1: TEXT : HOME
635 :
640 REM *****
645 REM * COURTE PRESENTATION *
650 REM *****
655 :
660 VTAB 7:T$ = "VOUS POURREZ OBTENIR LA COPIE DE L'EC
    RAN": GOSUB 225:T$ = "TEXTE": INVERSE : GOSUB 225:T
    $ = " SUR VOTRE IMPRIMANTE (SLOT 1) MAISPAS CELLE D
    ES ECRANS ": NORMAL : GOSUB 225
665 T$ = "GRAPHIQUES": INVERSE : GOSUB 225: NORMAL :: P
    RINT ".
670 PRINT :T$ = "IL VOUS SUFFIRA POUR CELA DE PRESSER
    LA TOUCHE (ESCAPE) APRES UN TINTEMENT DE LASONNETTE
    , COMME CELUI QUE VOUS ALLEZ SU-REMENT ENTENDRE.":
    GOSUB 225: CALL - 190: PRINT
675 PRINT :T$ = "PENDANT LA RESOLUTION DES PETITES OPE
    RATIONS BINAIRES, VOUS POURREZ REVOIR LA PAGE DE
    DEMONSTRATION EN UTILISANT CETTEMEME TOUCHE (ESCAPE
    ).": GOSUB 225: PRINT
680 VTAB 22: PRINT "VOUS ETES INVITE A PRESSER UNE TOU
    CHE":ES = 1: GOSUB 230:ES = 0: HOME : RETURN
```

TABLE DE FORMES AU VERSO

ADDITIONS BINAIRES

BSAVE NBR2-23, A\$6000, L\$615

*6880,6615

6880-	64	82	08	08	09	08	19	81	6200-	20	20	20	20	F5	DB	1B	3F	6410-	2E	6D	89	2D	35	3F	FF	1B	
6888-	42	01	81	81	C3	81	F6	81	6208-	37	2D	6D	89	3F	3F	3F	3F	6418-	3F	2E	6D	89	2D	35	3F	FF	
6810-	39	02	78	82	AE	82	F1	82	6218-	2D	2D	2D	2D	15	3F	3F	3F	6428-	1B	3F	2E	6D	89	2D	35	3F	
6818-	36	03	7F	83	C4	83	FE	83	6218-	3F	77	49	89	2D	35	3F	FF	6428-	FF	1B	3F	2E	2D	2D	2D	2D	
6820-	44	04	81	84	83	84	E2	84	6220-	0B	13	2D	2D	2D	2D	35	3F	6438-	3E	3F	3F	3F	3F	2E	2D	2D	
6828-	F9	04	18	85	58	85	98	85	6228-	3F	3F	3F	77	2D	2D	2D	F5	6438-	2D	F5	3F	3F	3F	B7	49	49	
6830-	D8	88	08	08	D8	88	D8	88	6238-	3F	3F	B7	49	49	49	88	88	6448-	49	89	88	3F	2D	2D	2D	2D	
6838-	D8	88	08	08	D8	88	D8	88	6238-	B7	89	2D	2D	2D	3E	3F	3F	6448-	3E	3F	3F	3F	37	2D	2D	2D	
6848-	D8	88	08	08	D8	88	D8	88	6248-	3F	17	2D	2D	2D	2D	3E	3F	6458-	2D	3E	3F	3F	3F	37	2D	35	
6848-	D8	88	08	08	D8	88	D8	88	6248-	3F	3F	37	2D	35	3F	37	2D	6458-	3F	37	2D	2D	2D	35	3F	3F	
6848-	D8	88	08	08	D8	88	D8	88	6258-	2D	2D	AD	3F	3F	3F	3F	2E	6468-	3F	37	2D	2D	2D	F5	DB	3F	
6848-	D8	88	08	08	D8	88	D8	88	6258-	2D	2D	2D	2D	3E	FF	1B	3F	6468-	37	2D	2D	2D	2D	3E	3F	3F	
6850-	D8	88	08	08	D8	88	D8	88	6268-	37	2D	2D	2D	2D	35	3F	3F	6478-	3F	37	2D	2D	2D	2D	3E	3F	
6850-	D8	88	08	08	D8	88	D8	88	6268-	3F	3F	77	2D	2D	2D	F5	3F	6478-	3F	3F	B7	49	49	49	81	88	
6868-	D8	88	08	08	D8	88	D8	88	6278-	3F	B7	49	49	49	88	88	2D	6488-	2D	2D	2D	2D	2D	3E	3F	3F	
6878-	D8	88	08	08	D8	88	D8	88	6278-	2D	2D	2D	2D	35	3F	3F	3F	6488-	3F	37	2D	2D	2D	2D	3E	3F	
6878-	D8	88	08	08	D8	88	D8	88	6288-	3F	37	2D	2D	2D	2D	35	3F	6498-	3F	3F	37	2D	35	3F	37	2D	
6888-	D8	88	08	08	D8	88	D8	88	6288-	3F	3F	3F	77	49	49	2D	3E	6498-	2D	2D	3E	3F	3F	37	2D	2D	
6888-	D8	88	08	08	D8	88	D8	88	6298-	3F	17	2D	2D	1E	3F	3F	17	64A8-	2D	DE	3B	3F	2E	2D	3E	3F	
6898-	D8	88	08	08	D8	88	D8	88	6298-	2D	2D	1E	3F	3F	17	2D	2D	64A8-	2E	2D	3E	3F	56	49	49	49	
6898-	D8	88	08	08	D8	88	D8	88	62A8-	3E	3F	37	2D	2D	3E	3F	B7	64B8-	88	88	3E	49	29	8D	1F	FF	
68A8-	D8	88	08	08	D8	88	D8	88	62A8-	49	49	49	81	88	F5	89	2D	64B8-	17	6D	6D	35	FF	FF	3F	2E	
68A8-	D8	88	08	08	D8	88	D8	88	62B8-	2D	AD	3F	3F	3F	8F	2D	2D	64C8-	6D	6D	DE	FF	3F	8E	6D	6D	
68B8-	D8	88	08	08	D8	88	D8	88	62B8-	2D	2D	35	3F	FF	3B	3F	2E	64C8-	15	FF	FF	4E	6D	2D	3E	FF	
68B8-	D8	88	08	08	D8	88	D8	88	62C8-	2D	4D	2D	35	3F	FF	3B	3F	64D8-	FF	17	6D	6D	2D	1E	FF	FF	
68B8-	D8	88	08	08	D8	88	D8	88	62C8-	8E	2D	2D	2D	35	3F	3F	3F	64D8-	77	8D	6D	DE	87	49	49	81	
68C8-	D8	88	E8	85	E8	85	E8	85	62D8-	8F	2D	6D	29	2D	3E	3F	DF	64E8-	88	2D	92	92	92	92	12	2D	
68C8-	E9	85	52	88	88	88	88	88	62D8-	3F	37	2D	4D	29	2D	3E	3F	64E8-	2D	2D	2D	2D	35	3F	3F	3F	
68D8-	49	89	92	92	92	92	92	12	88	62E8-	3F	3F	3F	8E	2D	2D	2D	F5	64F8-	3F	3F	6F	49	49	49	89	88
68D8-	88	89	2D	2D	AD	3F	3F	3F	62E8-	3F	3F	B7	49	49	49	88	88	64F8-	29	92	4A	89	15	3F	17	2D	
68E8-	8F	2D	2D	2D	2D	35	3F	DF	62F8-	2D	89	2D	2D	AD	3F	3F	3F	6588-	2D	15	3F	3F	3F	17	2D	2D	
68E8-	1B	3F	2E	6D	49	2D	3E	FF	62F8-	8F	2D	2D	2D	2D	35	3F	FF	6588-	2D	2D	DE	3B	37	2D	3E	3F	
68F8-	DB	3F	2E	6D	49	2D	3E	FF	6388-	1B	3F	2E	6D	89	2D	35	3F	6518-	2D	3E	87	49	49	81	88	3F	
68F8-	DB	3F	2E	6D	49	2D	3E	FF	6388-	3F	3F	3F	37	2D	2D	2D	2D	6518-	49	49	89	3F	3F	8F	2D	2D	
6188-	DB	3F	2E	6D	49	2D	3E	3F	6318-	35	3F	3F	3F	3F	4E	49	2D	6528-	2D	AD	FF	DB	1B	8F	6D	69	
6188-	3F	3F	3F	8E	2D	2D	2D	F5	6318-	35	3F	FF	DB	13	2D	2D	2D	6528-	69	29	3E	1F	FF	3B	1F	37	
6118-	3F	3F	B7	49	49	49	88	88	6328-	2D	35	3F	3F	3F	3F	37	2D	6538-	6D	49	49	29	1E	DF	FB	DB	
6118-	35	89	2D	2D	35	3F	3F	3F	6328-	2D	2D	2D	1E	3F	3F	3F	87	6538-	6E	89	6D	89	FE	1B	DF	1B	
6128-	17	2D	2D	2D	35	3F	FF	3F	6338-	49	49	49	89	88	3F	49	29	6548-	6E	4D	89	8D	FE	FB	FF	73	
6128-	4E	89	2D	35	3F	37	2D	35	6338-	AD	3F	3F	17	2D	2D	2D	15	6548-	4D	6D	31	FF	DB	77	2D	2D	
6138-	3F	37	2D	35	3F	37	2D	35	6348-	3F	3F	3F	3F	17	2D	6D	89	6558-	F5	3F	6F	49	49	81	88	2D	
6138-	3F	37	2D	35	3F	B7	49	89	6348-	2D	35	3F	FF	1B	3F	37	2D	6558-	49	49	89	3F	3F	8F	2D	2D	
6148-	88	3F	89	2D	2D	AD	3F	3F	6358-	6D	89	2D	35	3F	3F	3F	3F	6568-	2D	AD	FF	DB	1B	8F	6D	69	
6148-	3F	8F	2D	2D	2D	2D	35	3F	6358-	3F	2E	2D	2D	2D	2D	35	3F	6568-	69	29	3E	1F	1F	FF	1F	37	
6158-	3F	3F	3F	37	2D	6D	29	2D	6368-	3F	3F	3F	3F	2E	2D	4D	29	6578-	6D	49	49	29	1E	DF	FB	DB	
6158-	3E	3F	8F	2D	2D	1E	3F	3F	6368-	2D	3E	3F	DF	3B	3F	2E	2D	6578-	6E	89	6D	89	FE	DB	DB	33	
6168-	17	2D	2D	1E	3F	3F	17	2D	6378-	4D	29	2D	3E	3F	DF	3B	3F	6588-	4D	29	4D	31	DF	DF	DF	8E	
6168-	2D	2D	2D	35	3F	3F	3F	3F	6378-	56	49	49	49	89	88	6F	2D	6588-	4D	6D	31	FF	DB	77	2D	2D	
6178-	37	2D	2D	2D	35	3F	3F	3F	6388-	2D	2D	AD	3F	3F	3F	3F	2E	6598-	F5	3F	6F	49	49	81	88	3F	
6178-	3F	3F	B7	49	49	49	89	88	6388-	2D	2D	2D	2D	3E	3F	3F	3F	6598-	49	49	49	35	3F	DF	1B	37	
6188-	49	29	2D	2D	AD	3F	3F	3F	6398-	3F	2E	6D	89	2D	F5	3F	FF	65A8-	6D	49	29	3E	DF	3B	3F	3F	
6188-	3F	2E	2D	2D	2D	2D	3E	3F	6398-	3B	37	2D	2D	2D	AD	3F	3F	65A8-	2E	2D	2D	4D	29	3E	DF	DB	
6198-	3F	3F	3F	4E	49	29	2D	1E	63A8-	3F	3F	2E	6D	29	2D	35	3F	65B8-	37	6D	49	29	DE	DB	DB	9B	
6198-	3F	3F	3F	37	2D	2D	2D	3E	63A8-	FF	1B	3F	2E	2D	2D	2D	2D	65B8-	2D	6D	29	2D	15	3F	3F	1F	
61A8-	3F	3F	37	2D	2D	2D	AD	3F	63B8-	3E	3F	3F	3F	3F	2E	2D	2D	65C8-	3F	3F	2E	4D	8D	6D	31	3F	
61A8-	FF	DB	13	2D	2D	2D	2D	35	63B8-	2D	F5	3F	3F	3F	B7	49	49	65C8-	3F	DF	3F	37	6D	69	2D	F5	
6188-	3F	3F	3F	3F	37	2D	2D	2D	63C8-	49	89	88	3F	89	2D	2D	2D	65D8-	38	1F	DF	37	2D	2D	8D	35	
6188-	2D	1E	3F	3F	B7	49	49	49	63C8-	15	3F	3F	3F	3F	17	2D	2D	65D8-	FF	3B	3F	4D	49	49	49	88	
61C8-	88	88	8E	49	49	AD	3F	8F	63D8-	2D	2D	35	3F	3F	3F	3F	37	65E8-	48	C8	48	C8	48	C8	48	88	
61C8-	2D	2D	1E	3F	3F	17	2D	2D	63D8-	2D	2D	1E	3F	37	2D	35	3F	65E8-	88	19	88	88	88	88	88	88	
61D8-	1E	3F	3F	17	2D	2D	2D	2D	63E8-	37	2D	AD	3F	3F	2E	2D	2D	65F8-	88	88	88	88	82	8D	8D	8D	
61D8-	35	3F	3F	3F	3F	37	2D	2D	63E8-	2D	2D	3E	3F	3F	3F	3F	8E	65F8-	8D	4D	45	4D	4F	52			

ADDITIONS BINAIRES

(Suite)

Ce programme a été assemblé avec Pro-CODE, mais peut être repris avec n'importe quel assembleur. Vous pouvez aussi taper directement, après un CALL—151



0300: A9 01
0302: 20 95 FE
0305: A2 08
0307: BD 63 03
030A: 20 5C DB
030D: CA
030E: D8 F7

0310: A5 24
0312: 85 08
0314: A5 25
0316: 85 09

0318: A9 08
031A: 85 24
031C: A5 06
031E: 85 25
0320: 20 C1 FB

```

0 *****
1 *
2 * HARD COPIE D'UN ECRAN DE 40 COLONNES *
3 * AVEC IMPRIMANTE DMP OU IMAGEWRITER *
4 *-----*
5 * Deux pokes obligatoires, avant un *
6 * CALL 768 qui ne l'est pas moins: *
7 * POKE 6,VTAB-1 - POKE 7,LIGNE-FIN *
8 * (LIMITES: POKE 6,0 : POKE 7,23) *
9 *
10 *****
11 ORG $300
12 *-----*
13 IMP EQU $FE95 ; Sortie IMP
14 OUTDO EQU $DB5C ; Imprime A
15 CRD0 EQU $DAFB ; Retour chariot
16 BASCALC EQU $FBC1 ; Calcule adr-base
17 INTDOS EQU $03EA ; Reprise par SED
18 LDEP EQU $06 ; Ligne de départ
19 LFIN EQU $07 ; Dernière ligne
20 HCURS EQU $24 ; Htab curseur
21 VCURS EQU $25 ; Vtab curseur
22 MHCURS EQU $08 ; Mémo htab curseur
23 MVCURS EQU $09 ; Mémo vtab curseur
24 BASL EQU $28 ; Adresse ligne
25 *-----*
26 * INITIALISATION DE L'IMPRIMANTE PR#1 *
27 * Caract. étroits expansés (ESC"q"-CTRL-N) *
28 * 80 COLONNES (CTRL-I"80N") *
29 *-----*
30 LDA #1 ; Pr#1
31 JSR IMP
32 LDX #8 ; 8 caractères
33 BLEC1 LDA PARAM-1,X ; à lire et
34 JSR OUTDO ; à transmettre
35 DEX ; à
36 BNE BLEC1 ; l'imprimante.
37 *-----*
38 * Sauvegarde de la position du curseur *
39 *-----*
40 LDA HCURS
41 STA MHCURS
42 LDA VCURS
43 STA MVCURS
44 *-----*
45 * Une ligne après l'autre, sup! *
46 *-----*
47 INIT LDA #0
48 STA HCURS ; Début de ligne
49 LDA LDEP
50 STA VCURS ; Vtab positionné
51 JSR BASCALC
  
```

```

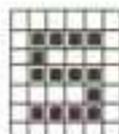
52 *-----*
53 *   L'adresse de départ, grâce à VARCALC, *
54 *   est maintenant dans $28-$29 *
55 *-----*
0323: A8 00      56          LDY  £$0
0325: B1 28      57 BLEC2  LDA  (BASL),Y
58 *-----*
59 * Pas de mode inverse, de flash ou de "ù" *
60 * mis à la place de la jolie case $FF... *
61 *-----*
0327: C9 E0      62          CMP  £$E0
0329: 30 06      63          BMI  CAP
032B: 09 00      64          ORA  £$80
032D: C9 E0      65          CMP  £$E0
032F: 10 0B      66          BPL  BDC
0331: 89 C0      67 CAP    ORA  £$C0
0333: C9 E0      68          CMP  £$E0
0335: 30 09      69          BMI  PAPIER
0337: 49 40      70 SUBST  EOR  £$40
0339: 4C 40 03   71          JMP  PAPIER
033C: C9 FF      72 BDC    CMP  £$FF
033E: F0 F7      73          BEQ  SUBST
74 *-----*
75 * Tout cela pour imprimer une seule lettre! *
76 *-----*
0340: 20 5C DB   77 PAPIER JSR  OUTDO
0343: C8         78          INY
0344: C8 20      79          CPY  £$20      ; Il y a 28 cases
0346: D0 0D      80          BNE  BLEC2
0348: 20 FB DA   81          JSR  CRD0      ; Sinon retour char
034B: E6 06      82          INC  LDEP     ; Une ligne de plus
034D: A5 06      83          LDA  LDEP     ; et on compare
034F: C5 07      84          CMP  LFIN     ; à ligne-fin
0351: D0 C5      85          BNE  INIT     ; Encore une, sinon
86 *-----*
87 *   Pr£0 et retour au programme basic *
88 *-----*
0353: A9 00      89          LDA  £$0
0355: 20 95 FE   90          JSR  IMP
0358: 20 EA 03   91          JSR  INTDOS
035B: A5 00      92          LDA  MHCURS
035D: 85 24      93          STA  HCURS
035F: A5 09      94          LDA  MVCURS
0361: 85 25      95          STA  VCURS
0363: 60         96          RTS
97 *-----*
98 *   PARAMETRES POUR L'IMPRIMANTE *
99 *-----*
0364: 0E 71 1B
        18 4E 30
        38 89
100 PARAM  HEX  0E711B184E303889

```

Avec ProDOS, remplacer
 358 : 20 EA 03 par
 35B : 20 6F F2

POUR TERMINER : BSAVE DMP40, A\$300, L\$6C

Imprimez des listages formatés de vos programmes



Si vous disposez d'une imprimante, vous avez certainement la bonne habitude de conserver une trace imprimée de vos principaux programmes, ne serait-ce que pour pallier le désagrément d'une intempestive initialisation de votre disquette (hélas ! cela arrive).

Parions qu'il ne vous déplairait pas d'obtenir des listages formatés (c'est-à-dire comportant une marge, un titre, un certain nombre de lignes par page... et un numéro de page) de vos chères élucubrations ! Exact ?

Ce sera désormais, pour vous en tout cas, chose facile, et en suivant un processus relativement simple :

FICHER T

Créez d'abord, à partir du programme à éditer, un fichier T. La fiche n°1 de *Tremplin Micro* vous indique la marche à suivre, mais il n'est pas inutile de revenir là-dessus. La routine FAITLIST est donc destinée, dans un premier temps, à créer le fichier EXEC LIST. Tapez-la, puis sauvegardez-la, une fois pour toutes, par un "SAVE FAITLIST", suivi d'un utile "LOCK FAITLIST".

Ensuite, un simple RUN va créer "EXEC LIST". Vérifiez, par un "CATALOG" que "LIST" existe réellement sur votre disquette, précédé de la mention "T" et du chiffre "2". Si le "2"... est malheureusement "1", FAITLIST comporte une erreur. Si tout va bien, tapez un "LOCK LIST".

Vous disposez maintenant du petit artifice qui va vous autoriser à transformer n'importe quel programme en fichier "T".

FAITLIST

```
10 D$ = CHR$(4) : G$ = CHR$(34)
15 PRINT D$"OPENLIST" : PRINT D$"WRITELIST" : POKE 33,30
20 R$ = "60000 D$ = CHR$(4) : PRINT :
PRINT D$" + G$ + "OPENP" + G$ +
": POKE 33,30 : PRINT D$" + G$ +
"WRITEP" + G$ + ": LISTO,59999 :
PRINT D$" + G$ + "CLOSEP" + G$ +
": POKE33,40 : END"
25 PRINT R$ : PRINT D$"CLOSELIST" :
END
```

Trois commandes sont nécessaires :

1. LOAD votre programme.
2. EXEC LIST.
3. RUN 60000.

Observation : Si le programme à lister comporte une ligne supérieure à 59999, prévoyez un autre LIST dont le numéro de ligne pourra par exemple être 0.

LISTAGE

Cet utilitaire va peut-être vous paraître longuet, mais vous constaterez, à l'usage, qu'il est réellement pratique.

L'un de ses avantages est de permettre la lecture, sur écran, de n'importe quel fichier "T", ce qui se révèle fort intéressant, dans bien des cas.

Notez qu'il n'accepte pas les "monstres", c'est-à-dire les lignes de plus de 255 caractères (méfiez-vous de DAKIN, notamment, qui élimine les REM, et compresse votre programme... mais en fabriquant précisément de tels monstres, lesquels interdisent d'ailleurs le renumérotage).

D'autre part, le nombre de lignes par page est dimensionné dans le programme. Il est rare, en effet, d'utiliser du papier dont la hauteur varie d'une fois à l'autre.

Et maintenant, à vous de jouer, sans oublier qu'un utilitaire ne devient réellement performant que si son utilisateur y place, ici et là, des instructions correspondant à ses besoins personnels.

Un conseil : Consacrez une disquette spéciale à *Faitlist*, *LIST* et *LISTAGE*. Vous pourrez la compléter par le *LISTAGE-LM* que vous a proposé le numéro 1 de *Tremplin Micro*.

LISTAGE «ECRAN-IMPRIMANTE»

```

10 TEXT : HOME : PRINT CHR$(21): GOSUB 605: REM SO
  US-PROGRAMME DE TRAITEMENT DES ERREURS (PAUL MERRY)
15 ONERR GOTO 620
20 DIM L$(70)
25 POKE 44452,11: POKE 44605,10: REM 10 LIGNES DE
  CATALOG
30 B$ = CHR$(1):D$ = CHR$(13) + CHR$(4)
35 T$ = " ": FOR I = 1 TO 10:T$ = T$ + "_____": NEXT
40 FOR I = 1 TO 25:E$ = E$ + "-----": NEXT
45 VTAB 1: PRINT " "; INVERSE
50 PRINT " FICHER (T) --- LECTURE & IMPRESSION ": NO
  RMAL
55 VTAB 2: PRINT T$: POKE 34,3: VTAB 20: PRINT T$
60 VTAB 22: PRINT "CTRL-A = CATALOG D1 CTRL-B = CATA
  LOG D2"
65 VTAB 23: PRINT "CTRL-C = CORRECTION CTRL-F = FIN-
  ERREUR"
70 VTAB 3: POKE 35,10: GOTO 135
75 :
80 REM *****
85 REM * SAISIE DES PARAMETRES *
90 REM *****
95 :
100 R = ASC (R$): IF R > 31 THEN 125
105 IF R = 6 THEN 690
110 IF R < > 0 AND R < 3 THEN PRINT : HOME : PRINT D
  $"CATALOG,D"R: GET R$: PRINT : HOME : GOTO 145
115 IF R < > 3 OR S < 2 THEN 145
120 S = S - 1: GOTO 130
125 G$ = G$ + R$: GOTO 145
130 G$ = " ": GOTO 145
135 S = 0:S$ = "E"
140 S = S + 1
145 VTAB 2 + S * 2: HTAB 1: CALL - 868
150 ON S GOTO 175,195,215
155 INVERSE : PRINT G$;: NORMAL : GET R$
160 IF ASC (R$) < > 13 THEN 100
165 IF G$ = " " THEN 145
170 ON S GOTO 180,200,220
175 PRINT "TITRE DU FICHER (T) ";: GOTO 155
180 IF ASC (G$) < 65 OR ASC (G$) > 90 THEN G$ = " ":
  GOTO 145
185 GOSUB 610
190 J$ = G$:G$ = " ": GOTO 140
195 PRINT "<E>CRAN <I>MPRIMANTE ";: GOTO 155
200 IF G$ = "I" THEN S$ = "0": GOTO 210
205 S$ = "E":C% = 80: IF G$ < > "E" THEN G$ = " ": GOTO
  145
210 G$ = " ": GOTO 140
215 PRINT "<I> AVEC minuscules <0> SANS ";: GOTO 155
220 IF G$ < > "I" AND G$ < > "0" THEN G$ = " ": GOTO
  145

```

← Pour plus de 70 lignes par page, modifier cette valeur.

← Ces indications ne sont valables que pendant la saisie des premiers paramètres.

← Si R=6 (CTRL-F), fin de programme.

← On ne peut corriger (CTRL-C) que dans la mesure où l'on a déjà rentré un paramètre (S plus grand que 1).

← En fait, avec la routine LIST, le fichier est toujours titré "P", mais on peut le renommer...

← Un essai sur écran constitue une bonne précaution avant de passer à l'édition (pour laquelle on repart à zéro).

← Le listage sans minuscules permet néanmoins de conserver les lettres accentuées.

Si l'on choisit l'option "Ecran", on peut stopper à tout moment en pressant la touche "POMME FERMÉE"

Ne pas oublier que le titre va être imprimé en caractères expansés, ce qui doublera sa longueur.

Quand V=13 (RETURN), une ligne de programme vient d'être lue.

Le programme se plante méchamment si la ligne compte plus de 255 caractères.

Si POMME FERMÉE est pressée, retour à la ligne 10.

P% = Nombre de lignes (modifiable). Ne pas oublier de corriger partiellement la ligne 365.

P=Page. Si Page=0, on va d'abord imprimer le titre.

Numéro de page, au milieu d'une ligne de tirets, de même longueur que la ligne de texte. C'est joli... et ça ne coûte pas cher !

```

225 S% = 0: IF G$ = "1" THEN S% = 1
230 IF S$ = "E" THEN POKE 35,24: HOME : VTAB 21: PRINT
    T T$: PRINT "          POMME FERMEE POUR ARRETER": PO
    KE 35,20: VTAB 19: GOTO 280
235 IF S$ = "0" THEN GOSUB 460
240 VTAB 15: HTAB 1: PRINT "ENTETE DU LISTAGE A EFFECT
    UER": POKE 35,24
245 HTAB 1: VTAB 17: INPUT "=) "; N$: IF N$ = "" OR L
    EN (N$) > 38 THEN PRINT "": GOTO 245
250 POKE 35,24: VTAB 15: HTAB 1: CALL - 958: PRINT N
    $: PRINT T$: POKE 34,17: PRINT
255 :
260 REM *****
265 REM *   LECTURE DU FICHIER (T)   *
270 REM *****
275 :
280 PRINT D$"OPEN"J$
285 PRINT D$"READ"J$:L$ = ""
290 GET C$:V = ASC (C$): IF V = 13 THEN 315
295 IF S% = 1 THEN 305
300 IF V > 96 AND V < 123 THEN C$ = CHR$ (V - 32)
305 L$ = L$ + C$: IF LEN (L$) = 255 THEN 315
310 GOTO 290
315 PRINT D$: PRINT B$L$:V$ = "          ": IF L$ = "" THE
    N 285
320 IF PEEK (49250) > 127 THEN PRINT D$"CLOSE": CLEA
    R : GOTO 10
325 FOR K = 1 TO 5: IF MID$ (L$,K,1) < > " " THEN 33
    5
330 IF K < 5 THEN L = 5 - K:L$ = LEFT$ (V$,L) + L$: G
    OTO 345
335 NEXT K
340 P% = 64: IF P = 0 THEN P% = 58
345 L = LEN (L$): IF L < = C% THEN A% = A% + 1:L$(A%)
    = L$: GOTO 365
350 A% = A% + 1:L$(A%) = LEFT$ (L$,C%):L$ = RIGHT$ (L
    $,L - C%)
355 L = LEN (L$): IF L < = (C% - 5) THEN A% = A% + 1:
    L$(A%) = V$ + L$: GOTO 365
360 A% = A% + 1:L$(A%) = V$ + LEFT$ (L$,C% - 5):L$ =
    RIGHT$ (L$,L - (C% - 5)): GOTO 355
365 P% = 64: IF P = 0 THEN P% = 61
370 IF A% < P% AND B < > 53 AND B < > 5 THEN 285
375 IF S$ = "E" THEN GOSUB 415: GOTO 285
380 IF P = 0 THEN GOSUB 410
385 P = P + 1: PRINT D$"PRÉ" D%: FOR I = 1 TO P%: PRINT
    L$(I): NEXT : PRINT
390 M$ = " PAGE " + STR$ (P) + " ":A = (C% - LEN (M$)
    ) / 2:M$ = " " + LEFT$ (E$,A) + M$ + LEFT$ (E$,A)
    : PRINT M$: PRINT CHR$ (12): PRINT D$"PRÉ0"
395 FOR I = 1 TO P%:L$(I) = "": NEXT
400 IF FIN = 1 THEN 675
405 GOSUB 415: GOTO 285

```

```

410 PRINT D$*PRE"D%: PRINT CHR$(14)" *I$*X"N$I$*Y"
    CHR$(15): PRINT : PRINT : RETURN
415 B% = 0: IF A% > P% THEN FOR I = P% + 1 TO A%: B% =
    B% + 1: L$(B%) = L$(I): L$(I) = "": NEXT
420 A% = B%
425 IF FIN = 1 AND S$ = "E" THEN POP : GOTO 675
430 RETURN
435 :
440 REM *****
445 REM * PARAMETRES DE L'IMPRIMANTE *
450 REM *****
455 :
460 I$ = CHR$(27): K$ = "n072N080E096q120Q136P160p144"
465 PRINT : POKE 35,24: VTAB 21: PRINT "PARAMETRES PAR
    DEFAULT = L80 M010 E SLOT1 POUR REVENIR EN ARRIERE
    , TAPER UN (?) ": POKE 35,18
470 V = 10: VTAB V: HTAB 1: INPUT "LONGUEUR DE LIGNE "
    ;H$: IF H$ = "" THEN H$ = "80"
475 IF H$ = "?" THEN POP : G$ = "": GOTO 145
480 C% = VAL (H%): H = 19: GOSUB 575
485 V = 11: VTAB V: HTAB 1: INPUT "LARGEUR DE MARGE "
    ;H$: IF H$ = "" THEN H$ = "10"
490 IF H$ = "?" THEN 470
495 D$ = I$ + "L0" + H$: M% = VAL (H%): H = 19: GOSUB 57
    5
500 V = 12: VTAB V: HTAB 1: INPUT "CARACTERE IMPRIMANTE
    ";H$: IF H$ = "" THEN H$ = "E"
505 IF H$ = "?" THEN 485
510 P$ = I$ + H$: C = 0: H = 22: GOSUB 575
515 FOR I = 1 TO 28 STEP 4: C$ = MID$(K$,I,1): L% = V
    AL (MID$(K$,I + 1,3)): IF C$ < > H$ THEN 535
520 IF L% < C% + M% THEN VTAB 15: HTAB 1: CALL - 950
    : PRINT "LONGUEUR MAXI AVEC "H$="": INVERSE : PRIN
    T L%: NORMAL : GOTO 470
525 IF H$ = "?" THEN 500
530 C = 1
535 NEXT I: IF C = 0 THEN VTAB 15: HTAB 1: INVERSE :
    PRINT "CODES IMPRIMANTE:": NORMAL : PRINT " " : F
    OR I = 1 TO 28 STEP 4: PRINT MID$(K$,I,1)"-" VAL
    (MID$(K$,I + 1,3))" " : NEXT : PRINT "": GOTO 470
540 V = 13: VTAB 13: HTAB 1: CALL - 950: PRINT "SLOT D
    E L'IMPRIMANTE " : GET H$: IF ASC (H%) = 13 THEN
    H$ = "1"
545 IF H$ = "?" THEN 520
550 H = 22: GOSUB 575
555 D% = VAL (H%): IF D% = 3 OR D% = 6 THEN 540
560 VTAB V + 2: HTAB 1: PRINT "ALLUMEZ L'IMPRIMANTE
    " : CALL - 198: POKE - 16368,0: WAIT - 16384,128:
    PRINT
565 PRINT D$*PRE"D%: PRINT CHR$(9)"160N": PRINT P$:
    : PRINT D$: PRINT D$*PRE0"
570 RETURN
575 VTAB V: HTAB H: INVERSE : PRINT H$: NORMAL : RETURN

```

← Impression du titre.

← On récupère les lignes qui n'ont pas été imprimées... si lignes il y a, bien sûr.

← Concerne l'imprimante Apple (carte parallèle ou série).

Cette partie du programme devra être modifiée si vous n'utilisez pas une imprimante APPLE. Reportez-vous alors à votre documentation.

Avant d'allumer l'imprimante, placer un pli du papier face à la tête d'impression.

← Initialisation en fonction des paramètres rentrés.

Cette routine est préconisée par Paul MERRY dans son excellent ouvrage : "BASIC APPLESOFT", la Bible des utilisateurs d'Apple.

Si on ne peut pas verrouiller un fichier, c'est bien que celui-ci n'existe pas.

On pourrait aussi renvoyer au début du programme.

```

580 :
585 REM *****
590 REM *   TRAITEMENT DES ERREURS   *
595 REM *****
600 :
605 POKE 768,104: POKE 769,168: POKE 770,104: POKE 771
,166: POKE 772,223: POKE 773,154: POKE 774,72: POKE
775,152: POKE 776,72: POKE 777,96: RETURN
610 PRINT : PRINT D$"LOCK"G$
615 PRINT D$"UNLOCK"G$: RETURN
620 CALL 768:E = PEEK (222): IF E = 53 THEN 305
625 IF E = 5 THEN FIN = 1: GOTO 375
630 IF E = 6 THEN VTAB 6: HTAB 1: PRINT "FICHIER INEX
ISTANT": POP :G$ = "": GOTO 145
635 VTAB 17: HTAB 1: CALL - 868: PRINT "ERREUR "E" LI
GNE " PEEK (218) + PEEK (219) * 256
640 FOR I = 1 TO 2000: NEXT I
645 TEXT : END
650 :
655 REM *****
660 REM *       FIN DE TRAVAIL       *
665 REM *****
670 :
675 PRINT "FICHIER (T) LU"
680 PRINT D$"CLOSE"J$: PRINT D$"PRE0"
685 FOR I = 1 TO 2000: NEXT
690 TEXT : HOME : END

```



Un accessoire nullement inutile : le magnétophone

Q UAND on dispose d'un lecteur de disquette, on a tendance à négliger l'utilisation d'un accessoire qui, dans certains cas, permet de sauver un programme : le magnétophone.

Il n'est pas rare, notamment quand on use abondamment des POKES, ou lorsqu'on lance un CALL X... n'aboutissant pas à une bonne adresse (ne s'oubliez pas : cela vous arrivera !), de perdre le DOS.

Et alors, bonjour les dégâts ! les commandes SAVE, LOAD, etc. restent sans effet... et le chef-d'œuvre que l'on a en mémoire semble bel et bien perdu.

Ultime ressource : le magnétophone (à piles de préférence) qui, à mon avis, devrait être branché en permanence.

Deux cas de figure : 1. La touche RESET n'agit plus (ou CTRL-RESET). C'est irrémédiable. L'Apple boucle... mais on ne sait pas où.

2. Le RESET vous redonne la main. LIST prouve que le programme est toujours là. Tout va bien !

- Rembobinez éventuellement la bande du magnéto.
- Enfoncez la touche "ENREGISTREMENT" (parfois 2 touches).
- Tapez tranquillement SAVE, suivi de RETURN.

Si la sauvegarde se passe bien, vous entendez un premier signal sonore après quelques secondes d'attente, un autre vous indiquera la fin de l'enregistrement. Laissez alors la bande se dérouler (quelques millimètres), puis stoppez le magnéto et rebobinez. Eteignez l'Apple. Remettez le contact, puis RESET. — Tapez LOAD.

— Enfoncez la touche LECTURE du magnéto, puis revenez à l'Apple par un RETURN indispensable.

Normalement, vous entendez de nouveau les deux signaux sonores et vous retrouverez votre cher programme, mais avec le DOS en mémoire et la possibilité d'effectuer une sauvegarde sur disquette.

Un conseil : Essayez tout cela à froid... pour vérifier que le magnéto est parfaitement connecté à votre Apple, et qu'il fonctionne.

En mode TEXT,
sur APPLE IIe et IIc



LE JEU DE L'HOMME LIBRE

Vous connaissez certainement le JEU DU PENDU. On ne peut pas dire que ce soit folichon de pendre les gens, alors que leur seule faute est de ne pas trouver le MOT CACHE!

C'est pourquoi je vous propose une autre version, plus pacifique, sinon plus amusante: le JEU DE L'HOMME LIBRE.

D'accord, pour commencer, le bonhomme se trouve... en cage, mais chaque BONNE REPONSE lui entrouvre la porte et, finalement, pour lui, c'est la LIBERTE!

PRESSEZ UNE TOUCHE POUR LE PREMIER MOT

Pour oublier le jeu du pendu

AUCUNE difficulté dans la programmation de ce petit jeu visuel, n'utilisant que les caractères normaux de l'Apple, et ne passant en mode graphique que pour dissimuler momentanément l'affichage de la «cage» (quelques secondes, et uniquement au début du jeu).

Attention ! le trait vertical n'existe pas sur tous les APPLE. Sur l'Apple IIc, par exemple, il faut passer en mode QWERTY... ou se servir des caractères "souris", (la routine de la page 34 montre comment les obtenir).

CHOIX DES MOTS Evitez les mots de plus de 15 lettres, mais n'hésitez pas à compléter la petite liste qui vous est fournie ici. Si vous désirez corser la difficulté, choisissez des mots comportant le plus possible de lettres différentes.

SUGGESTIONS Ecrire un petit programme permettant de rechercher de tels mots. Prévoir des appréciations suivant le score réalisé... Créer le même jeu en mode graphique, avec, dans la cage, un personnage évidemment sympa (puisqu'il mérite d'être libéré)...

```
10 LOMEM: 17400
20 TEXT : PRINT CHR$(21): HOME : HGR
30 DIM L$(15)
40 N = 12: REM NOMBRE DE MOTS
50 PRINT TAB(9)"LE JEU DE L'HOMME LIBRE"
60 PRINT TAB(9)"-----"
70 PRINT
80 PRINT " Vous connaissez certainement le JEU DU PE
NDU. On ne peut pas dire que ce soit folichon de pe
ndre les gens, alors que leur seule faute est de ne
pas trouver le MOT CACHE!"
90 INVERSE : FOR I = 4 TO 8: VTAB I: PRINT " ": NEXT I
NORMAL
```

← LOMEM interdit l'effacement de l'écran texte, mémorisé à partir de l'adresse 16384.

Ne pas oublier de modifier la valeur de N, selon le nombre de mots mis en DATA.

Fait subitement apparaître la page de texte...

Toujours placer un PRINT après un GET. On passe en mode graphique pour laisser à l'Apple le temps de tracer la "cage".

Mémorisation de l'écran à partir de l'adresse 16384.

Tel qu'il se présente ici, le programme peut proposer le même mot à plusieurs reprises. C'est le facteur chance qui joue. Il serait possible de mémoriser les mots déjà trouvés, pour éviter cette éventualité.

Effacement du barreau de la cage correspondant à la lettre trouvée.

```
100 PRINT : PRINT "C'est pourquoi Je vous propose une autre version, plus pacifique, sinon plus amusante: le JEU DE L'HOMME LIBRE."
110 PRINT : PRINT "D'accord, pour commencer, le bonhomme se trouve... en cage, mais chaque BONNE RE-PONSE lui entrouvre la porte et, finalement, pour lui, c'est la LIBERTE!"
120 TEXT
130 VTAB 22: PRINT "PRESSEZ UNE TOUCHE POUR LE PREMIER MOT "; GET R$
140 :
150 REM *****
160 REM * TRACE DE LA "CAGE" *
170 REM *****
180 :
190 PRINT : HOME
200 HOME : VTAB 22: HTAB 10: PRINT "-*- PATIENCE SVP -*-"; HGR
210 FOR I = 2 TO 17: FOR J = 2 TO 24: VTAB I: HTAB J: PRINT "à": NEXT : NEXT
220 VTAB 1: INVERSE
230 FOR I = 1 TO 25: PRINT " "; NEXT : PRINT
240 FOR I = 2 TO 17: VTAB I: HTAB 1: PRINT " "; HTAB 25: PRINT " "; NEXT
250 FOR I = 1 TO 25: PRINT " "; NEXT : NORMAL : PRINT
260 GOSUB 640: TEXT
270 :
280 REM *****
290 REM * CHOIX DU MOT *
300 REM *****
310 :
320 M = 1 + INT ( RND (1) * N); RESTORE
330 FOR I = 1 TO M: READ M$: NEXT I
340 L = LEN (M$): FOR I = 1 TO L: L$(I) = MID$(M$,I,1) : NEXT
350 VTAB 16: HTAB 28: PRINT "ENCORE": VTAB 10: HTAB 28: PRINT "A TROUVER"
360 NL = L
370 VTAB 20: PRINT "LE MOT A SECRET COMPTE "L" LETTRES"
380 I = 0
390 I = I + 1
400 VTAB 17: HTAB 28: CALL - 868: INVERSE : PRINT NL; : NORMAL : PRINT " LETTRE"; : IF NL > 1 THEN PRINT "S"
410 IF NL = 0 THEN I = I - 1: VTAB 28: HTAB 1: CALL - 868: PRINT "CAGE OUVERTE EN "; INVERSE : PRINT I; : NORMAL : PRINT " ESSAIS": GOTO 490
420 PRINT : VTAB 22: HTAB 1: CALL - 868: PRINT "LETTRE PROPOSEE -> "; GET R$: PRINT " "
430 IF ASC (R$) < 65 OR ASC (R$) > 90 THEN 420
440 FOR J = 1 TO L: IF R$ < > L$(J) THEN 470
450 FOR K = 8 TO 17: VTAB K: HTAB J + 2: PRINT " "; NEXT K: INVERSE : VTAB 18: HTAB J + 2: PRINT R$: NORMAL
```

```

460 L$(J) = "":NL = NL - 1
470 NEXT J
480 GOTO 390
490 VTAB 22: CALL - 868: FOR I = 1 TO 1500: NEXT
500 :
510 REM *****
520 REM * SUITE OU TERMINE *
530 REM *****
540 :
550 CALL - 198: PRINT " (1) UNE AUTRE PARTIE (2) TERM
INE ? ";; GET R$
560 IF R$ = "1" THEN GOSUB 650: GOTO 320
570 IF R$ < > "2" THEN 550
580 TEXT : HOME : END
590 :
600 REM *****
610 REM * DEPLACEMENT DE LA "CAGE" *
620 REM *****
630 :
640 D = 1824:Z = 68: GOSUB 670:D = 2000:Z = 62: GOSUB 67
0:D = 16384:Z = 66: GOSUB 670: GOTO 660
650 D = 16384:Z = 60: GOSUB 670:D = 17360:Z = 62: GOSUB
670:D = 1824:Z = 66: GOSUB 670
660 POKE 768,160: POKE 769,0: POKE 770,76: POKE 771,44:
POKE 772,254: CALL 768: RETURN
670 POKE Z,D - INT (D / 256) * 256: POKE Z + 1, INT (D
/ 256): RETURN
680 :
690 DATA ACCELERATEUR,ALIMENTATION,AUTOCHENILLE,BARAGO
UINEUR,CHAMAILLERIE,DISTRACTION,INFORMATIQUE,RECONST
RUCTION,INVITATION,COURAGEUSEMENT,EXCENTRICITE,NEBUL
OSITE

```

← La lettre trouvée est supprimée, NL fournit le nombre de lettres à identifier.

← GOSUB 650 permet de récupérer la cage (intacte) et de l'afficher instantanément.

← Cette routine est la copie de celle donnée par la fiche de TREPLIN MICRO N°1.

← A vous de donner libre cours à votre imagination... et pas de mots laids. S.V.P. !

LES CARACTÈRES "SOURIS" DE L'APPLE IIc

```

10 PRINT CHR$(4)"PRÉ3": PRINT CHR$(17):
30 FOR I = 64 TO 95 STEP 4
35 V = V + 2:H = - 9: VTAB V
45 FOR J = I TO I + 3:H = H + 10
50 HTAB H: PRINT CHR$(24);J;"...."; INVERSE
: PRINT CHR$(27); CHR$(J);: NORMAL
55 NEXT J: PRINT : NEXT I
60 PRINT CHR$(24)

```

Cette petite routine va vous permettre de visualiser les caractères "souris" de votre APPLE IIc, et de connaître la valeur ASCII de la touche correspondante.

JEU DE L'HOMME LIBRE

Pour un APPLE IIc, vous modifierez ainsi votre programme. Notez que vous pouvez regrouper les lignes 205, 210 et 215 sur une seule ligne.

```

20 PRINT CHR$(4)"PRÉ3": PRINT
CHR$(17): HOME : HGR
205 PRINT CHR$(27): INVERSE
210 FOR I = 2 TO 17: FOR J = 2 TO
24: VTAB I: HTAB J: PRINT " _
": NEXT : NEXT
215 PRINT CHR$(24)

```

PAIR ou IMPAIR avec quelques octets

Basic
et langage machine

```
10 FOR I = 1 TO 2500
20 N = 1 + INT ( RND (1) * 255)
30 IF INT (N / 2) = N / 2 THEN A =
  A + 1
40 NEXT I
50 PRINT A
```

Ce court programme en Basic contrôle si les nombres générés par la fonction RND sont pairs ou impairs. Il compte ceux qui sont exactement divisibles par deux et affiche le total.

LA BOUCLE... EST BOUCLÉE EN 60 SECONDES

Une mini-routine en langage machine permet de gagner un peu plus de 10 secondes. Une économie de 20 % ... ce n'est pas négligeable. Nous verrons plus tard qu'il est possible de faire beaucoup mieux, mais n'anticipons pas !

On transmet la valeur de A au sous-programme machine par la fonction CALL. Mais celle-ci aboutit à un JMP \$E6F5 qui place cette valeur dans l'accumulateur, la transforme en entier et la transfère dans le registre X... où notre routine la récupère. Lisez plutôt.

0300	— 20 F5 E6	JSR	\$E6F5	:	On aurait aussi pu paker A en 8 et l'y récupérer par un LDA \$08
0303	— 8A	TXA		:	On passe X dans A (accumulateur).
0304	— 18	CLC		:	Annulation d'une retenue antérieure éventuelle.
0305	— 6A	ROR		:	Rotation à droite (divise par 2). Si impair, le bit qui sort à droite est 1 (C = 1 ... retenue).
0306	— B0 0A	BCS	\$0312	:	Si C = 1, retour au basic.
0308	— E6 06	INC	\$06	:	Plus un dans le pointeur 06 (page zéro)...
030A	— A5 06	LDA	\$06	:	... chargé dans l'accumulateur...
030C	— C9 00	CMP	£\$00	:	... pour le comparer à \$00. S'il vaut zéro, c'est qu'on vient de faire SFF + \$01 = \$00.
030E	— D0 02	BNE	\$0312	:	S'il est différent, retour au basic.
0310	— E6 07	INC	\$07	:	... sinon plus un pour les centaines (partie haute).
0312	— 60	RTS		:	Basic.

EXPLICATION

Au départ, on a mis \$6 et \$7 à \$0. Il faut en effet 2 octets pour stocker le nombre de numéros pairs, puisque celui-ci va dépasser 255.

PROGRAMME EN BASIC

```
10 POKE 6,0: POKE 7,0: FOR I = 1 TO 2500
20 A = 1 + INT ( RND (1) * 254) : CALL
  768,A
30 NEXT I: PRINT PEEK (6) + PEEK (7)
  * 256: CALL - 198
```

Si vous êtes intéressé par les caprices du hasard, allez plus loin avec les nombres aléatoires !...

```

10 LOMEM: 4010
15 GOSUB 160
20 PRINT "LE PROGRAMME GENERE 3000 NOMBRES ALEA- TOIRES, COMPRIS E
    N TRE 1 ET 255, ET COMP-TE CEUX QUI SE REVELENT PAIRS":
25 PRINT : PRINT "TAPER <I> POUR EDITER LES RESULTATS <IM-PRIMANTE
    DANS LE SLOT 1)                -> "; GET IM$: PRINT IM$
30 FOR I = 1 TO 3000
35 N = 1 + INT ( RND (1) * 255)
40 POKE P + N,1 + PEEK (P + N)
45 IF N / 2 = INT (N / 2) THEN IM = IM + 1
50 NEXT
55 IF IM$ = "1" THEN PRINT CHR$ (4)"PRÉ1": PRINT CHR$ (27)"E" ;;
    PRINT CHR$ (27)"L010": GOTO 65
60 PRINT CHR$ (4)"PRÉ3"
65 FOR I = 1 TO 51: FOR J = 1 TO 65 STEP 13
70 POKE 36,J: PRINT RIGHT$ (<" " + STR$ (I + IN)),3) + " = " PEE
    K (P + I + IN);;IN = IN + 51
75 NEXT J: PRINT :IN = 0: NEXT I
80 PRINT " "IM" NUMEROS PAIRS"
85 IF IM$ = "1" THEN PRINT CHR$ (4)"PRÉ0"
90 PRINT : GOSUB 165: GOSUB 160: IF T = 1 THEN 125
95 IF T = 2 THEN 155
100 PRINT "VOICI LE MEME PROGRAMME, MAIS AVEC LE NOMBRE ALEATOIRE
    GENERE PAR UNE ROUTINE EN LANGAGE MACHINE, ET LES NOMBRES PAIRSCO
    MPTES PAR CETTE MEME ROUTINE.": PRINT : PRINT
105 FOR I = 768 TO 802: READ R: POKE I,R: NEXT
110 GOSUB 165: CALL - 198
115 FOR I = 1 TO 3000
120 CALL 768:N = PEEK (10): NEXT I
121 PRINT :T = T + 1:IM = PEEK (6) + PEEK (7) * 256: GOTO 55
125 PRINT : INVERSE : PRINT 93;: NORMAL : PRINT " SECONDES EN BASIC"
    : PRINT
130 PRINT : INVERSE : PRINT 43;: NORMAL : PRINT " SECONDES AVEC LM "
    : PRINT
135 FOR I = 768 TO 823: READ R: POKE I,R: NEXT
140 GOSUB 165: GOSUB 160
145 PRINT "POUR TERMINER, VOICI LE MEME PROGRAMME, ENTIEREMENT EN LA
    NGAGE MACHINE (ON NE RENVOIE PLUS LES NOMBRES ALEATOIRES AU BA
    SIC, MAIS ON FAIT TOUJOURS LES TOTALI-SATIONS.": PRINT
150 GOSUB 165: CALL 768: GOTO 121
155 PRINT : PRINT PRINT "16 SECONDES... C'EST ENCORE MIEUX!": GOSUB
    165: HOME : END
160 TEXT : PRINT CHR$ (21): HOME :P = 3744: FOR I = 3744 TO 3999: P
    OKE I,0: NEXT : FOR I = 6 TO 10: POKE I,0: NEXT : RETURN
165 PRINT : PRINT : CALL - 198: VTAB 23: HTAB 1: PRINT "ENFONCEZ UN
    E TOUCHE ": POKE - 16368,0: WAIT - 16384,128: PRINT : RETURN
170 DATA 32,174,239,160,14,169,0,32,127,233,230,161,165,161,133,10,1
    70,254,160,14,24,106,176,10,230,6,165,6,201,0,208,2,230,7,96
175 DATA 32,174,239,160,14,169,0,32,127,233,230,161,165,161,170,254,
    160,14,24,106,176,10,230,6,165,6,201,0,208,2,230,7,24,165,0,105,1
    ,133,8,165,9,105,0,133,9,201,11,208,207,165,8,201,184,208,201,96

```

ET POUR TERMINER, 16 SECONDES pour tout faire !

Générer 3000 nombres aléatoires, compris entre 1 & 255, totaliser la quantité de tirages, par numéro et le nombre de numéros, tout cela en 16 secondes !

Les numéros pairs sont totalisés aux adresses \$6 et \$7 (page zéro) et le compteur utilise les adresses \$8 et \$9 de cette même page. Le nombre de tirages par numéro est écrit sur 255 octets, à partir de \$EA0 (3744), à la suite du programme «démon» ci-contre. Ces 255 octets sont protégés par LOMEM.

On pourrait évidemment transmettre chaque tirage au basic en l'écrivant en page zéro, à l'adresse \$0A par exemple.

Il serait aussi possible de sacrifier 3000 octets... en page graphique notamment, ou sous un HIMEM à déterminer.

Ce programme ne générant que des nombres égaux ou inférieurs à 255, chacun d'eux ne nécessite qu'un octet.

1 = 17	52 = 19	88 = 11	124 = 11	160 = 17
2 = 13	53 = 14	89 = 17	125 = 11	161 = 2
3 = 6	54 = 19	90 = 12	126 = 7	162 = 13
4 = 12	55 = 14	91 = 17	127 = 9	163 = 13
5 = 13	56 = 12	92 = 12	128 = 6	164 = 19
6 = 13	57 = 4	93 = 19	129 = 14	165 = 4
7 = 13	58 = 7	94 = 12	130 = 7	166 = 12
8 = 13	59 = 13	95 = 11	131 = 19	167 = 11
9 = 14	60 = 13	96 = 12	132 = 15	168 = 19
10 = 14	61 = 14	97 = 15	133 = 6	169 = 19
11 = 14	62 = 14	98 = 15	134 = 29	170 = 7
12 = 14	63 = 7	99 = 12	135 = 17	171 = 13
13 = 7	64 = 6	100 = 12	136 = 19	172 = 13
14 = 7	65 = 20	101 = 19	137 = 19	173 = 9
15 = 11	66 = 6	102 = 14	138 = 9	174 = 9
16 = 7	67 = 14	103 = 9	139 = 14	175 = 11
17 = 18	68 = 14	104 = 17	140 = 12	176 = 7
18 = 18	69 = 9	105 = 12	141 = 6	177 = 13
19 = 8	70 = 11	106 = 12	142 = 13	178 = 14
20 = 11	71 = 7	107 = 29	143 = 14	179 = 14
21 = 7	72 = 13	108 = 9	144 = 21	180 = 13
22 = 9	73 = 14	109 = 11	145 = 18	181 = 13
23 = 18	74 = 13	110 = 12	146 = 18	182 = 7
24 = 18	75 = 13	111 = 19	147 = 6	183 = 13
25 = 5	76 = 19	112 = 16	148 = 13	184 = 14
26 = 17	77 = 4	113 = 14	149 = 15	185 = 13
27 = 7	78 = 14	114 = 19	150 = 7	186 = 19
28 = 11	79 = 19	115 = 12	151 = 6	187 = 6
29 = 14	80 = 3	116 = 12	152 = 7	188 = 14
30 = 18	81 = 8	117 = 11	153 = 19	189 = 9
31 = 9	82 = 15	118 = 13	154 = 14	190 = 12
32 = 14	83 = 7	119 = 7	155 = 12	191 = 13
33 = 18	84 = 19	120 = 11	156 = 12	192 = 6
34 = 9	85 = 17	121 = 11	157 = 13	193 = 13
35 = 6	86 = 7	122 = 15	158 = 11	194 = 14
36 = 7	87 = 7	123 = 9	159 = 7	195 = 14
37 = 15	88 = 9	124 = 12	160 = 13	196 = 19
38 = 12	89 = 14	125 = 11	161 = 15	197 = 6
39 = 9	90 = 17	126 = 6	162 = 14	198 = 19
40 = 4	91 = 12	127 = 8	163 = 6	199 = 19
41 = 10	92 = 7	128 = 13	164 = 9	200 = 21
42 = 13	93 = 6	129 = 6	165 = 13	201 = 14
43 = 6	94 = 12	130 = 19	166 = 14	202 = 12
44 = 4	95 = 12	131 = 7	167 = 18	203 = 18
45 = 11	96 = 11	132 = 11	168 = 13	204 = 18
46 = 11	97 = 9	133 = 17	169 = 7	205 = 12
47 = 4	98 = 15	134 = 19	170 = 17	206 = 7
48 = 4	99 = 18	135 = 12	171 = 18	207 = 11
49 = 15	100 = 14	136 = 19	172 = 14	208 = 15
50 = 17	101 = 12	137 = 18	173 = 14	209 = 15
51 = 14	102 = 28	138 = 9	174 = 8	210 = 19

1401 10/08/82 PAL/80

0300	—	20	AE	EF
0303	—	A0	0E	
0305	—	A9	00	
0307	—	20	7F	E9
030A	—	E6	A1	
030C	—	A5	A1	
030E	—	AA		
030F	—	FE	A0	0E

0312	—	18		
0313	—	6A		
0314	—	B0	0A	
0316	—	E6	06	
0318	—	A5	06	
031A	—	C9	00	
031C	—	D0	02	
031E	—	E6	07	

0320	—	18		
0321	—	A5	08	
0323	—	69	01	
0325	—	85	08	
0327	—	A5	09	
0329	—	69	00	
032B	—	85	09	
032D	—	C9	0B	
032F	—	D0	CF	
0331	—	A5	08	
0333	—	C9	B8	
0335	—	D0	C9	
0337	—	60		

ORG		\$300	
RND	EQU	\$EFAE	
FMULT	EQU	\$E97F	: ARG ← (Y,A) et appel FMULT
FACLO	EQU	\$A1	: Où récupérer RND sur un octet
ECRIT	EQU	\$EA0	: 255 octets sous LOMEM
PAIR	EQU	\$6	: Partie basse du nombre
B1	EQU	\$8	: Compteur boucle partie basse
B2	EQU	B1 + 1	

BOUCLE			
JSR	RND		: On va le faire 3000 fois !
LDY	£\$E		: Pour ARG partie basse
LDA	£\$0		: ARG partie haute
JSR	FMULT		: FAC / par ARG
INC	FACLO		: Plus 1 pour éviter le zéro
LDA	FACLO		: Nombre dans l'accumulateur
TAX			: On le passe dans le registre X
INC	ECRIT,X		: Plus un de cette valeur

LIMITE			
CLC			: Annulation de la retenue
ROR			: Rotation pour diviser par 2
BBC	LIMITE		: Si retenue, impair et saut
INC	PAIR		: Sinon un pair de plus
LDA	PAIR		: Lu pour voir si ancien \$FF
CMP	£\$0		: Si \$0, c'est le cas...
BNE	LIMITE		: ...ou alors, saut
INC	PAIR + 1		: Partie incrémentée

LIMITE			
CLC			: Précaution utile
LDA	B1		: Où on est la boucle ?
ADC	£\$01		: Addition d'abord
STA	B1		: Écriture ensuite
LDA	B2		: Lecture partie haute
ADC	£\$0		: Addition de la retenue si...
STA	B2		: Puis réécriture immédiate
CMP	£\$0B		: On compare à \$0B-B8 (3000)
BNE	BOUCLE		: Si pas égal, encore un tour
LDA	B1		: Voir partie basse
CMP	£\$B8		: Est-ce égal ou non ?
BNE	BOUCLE		: On repart si inégalité
RTS			: Retour au basic

DISSÉQUER un nombre

Il y a deux méthodes pour disséquer un nombre. La première est mathématique... et lente. La seconde utilise la fonction MID\$ de votre APPLESOFT, et se révèle plus rapide. Ce court programme vous en fournit visuellement la preuve.

```

10 TEXT : PRINT CHR$(21): HOME :
  INVERSE
20 PRINT " DEUX METHODES POUR DISSE
  QUER UN NOMBRE "
30 POKE 34,5: PRINT
40 HTAB 5: PRINT "1";: HTAB 24: PR
  INT "2";: HTAB 7: NORMAL : PRINT "
  MATHEMATIQUEMENT";: HTAB 26: PRIN
  T "AVEC " CHR$(34)"MID$" CHR$(3
  4)
50 PRINT
60 PRINT "          ESSAI AVEC LA MET
  HODE 1          -----
  -----"
70 CALL - 198: VTAB 9: HTAB 1: CAL
  L - 868: INPUT "NOMBRE ";N$: IF
  N$ = "" THEN 258
80 :
90 L = LEN (N$): IF L > 9 THEN 70
100 FOR I = 1 TO L:A = ASC ( MID$ (
  N$,I,1)): IF A > 47 AND A < 58 TH
  EN NEXT : GOTO 120
110 GOTO 70
120 S = VAL (N$):N$ = STR$ (S):L =
  LEN (N$): PRINT
130 FOR I = 1 TO L:D = L - I:N(I) =
  INT (S / 10 ^ D):S = S - INT (N
  (I) * 10 ^ D)
140 PRINT N(I)" "S
150 NEXT : CALL - 198
160 VTAB 6: HTAB 31: INVERSE : PRINT
  2: NORMAL
170 POKE 32,20: POKE 33,20: PRINT :
  PRINT
180 :
190 PRINT "VARIABLE$="N$: PRINT
200 FOR I = 1 TO L:N(I) = VAL ( MID
  $ (N$,I,1)):S = 0: IF I < L THEN
  S = VAL ( RIGHT$ (N$,L - I))
210 PRINT N(I)" "S
220 NEXT : CALL - 198
230 :
240 TEXT
250 VTAB 22: HTAB 1: CALL - 198: PR
  INT "<1> AUTRE ESSAI <2> MENU DIS
  QUE <3> FIN";: GET R$
260 PRINT : IF R$ = "1" THEN VTAB 6
  : CALL - 958: GOTO 60
270 IF R$ = "2" THEN PRINT CHR$(4
  )"RUN MENU"
280 IF R$ < > "3" THEN 250
290 TEXT : HOME : END
  
```

ONERR GOTO

Routine en assembleur

La ligne de POKES (10020) installe ce petit programme à partir de l'adresse \$300 (soit 768 en valeur décimale).

300 —	68	PLA	10020	POKE 768,104 : POKE 769,168 : POKE 770,104 : POKE
301 —	A8	TAY		771,166 : POKE 772,223 : POKE 773,154 : POKE
302 —	68	PLA		774,72 : POKE 775,152 : POKE 776,72 : POKE 777,96
303 —	A6 DF	LDX \$DF	10030	CALL 768 : ER = PEEK (222)
305 —	9A	TXS		
306 —	48	PHA	10040	PRINT "ERREUR "ER" LIGNE " PEEK (218) + PEEK
307 —	98	TYA		(219) * 256
308 —	48	PHA	10050	PRINT "ADRESSE " PEEK (220) + PEEK (221) * 256
309 —	60	RTS		

Un bon programme ne doit pas comporter d'erreurs, mais il est parfois utile (sinon indispensable) de pallier les fautes de manipulation par une routine de récupération et de traitement des erreurs.

L'exemple ci-dessous ne s'intéresse qu'à la récupération. Chaque cas doit ensuite être traité séparément.

Naturellement, une ligne de votre programme devra renvoyer à cette routine. Par exemple : 10 ONERR GOTO 10020. Notez que POKE 216,0 annule l'instruction et que POKE 216,128 la rétablit.

Plus loin, et plus sûrement, avec l'assembleur

A. Andrieux et G. Creuzet ont bien travaillé. Il est même probable que ce deuxième volume de la *PRATIQUE DES APPLE* deviendra une sorte de bible des utilisateurs.

Non contents d'expliquer, pas à pas, comment le langage Basic est implanté dans un ordinateur, les auteurs s'attachent à montrer le fonctionnement d'utilitaires assembleur indispensables à tout programmeur désirant exploiter les diverses ressources de sa machine.

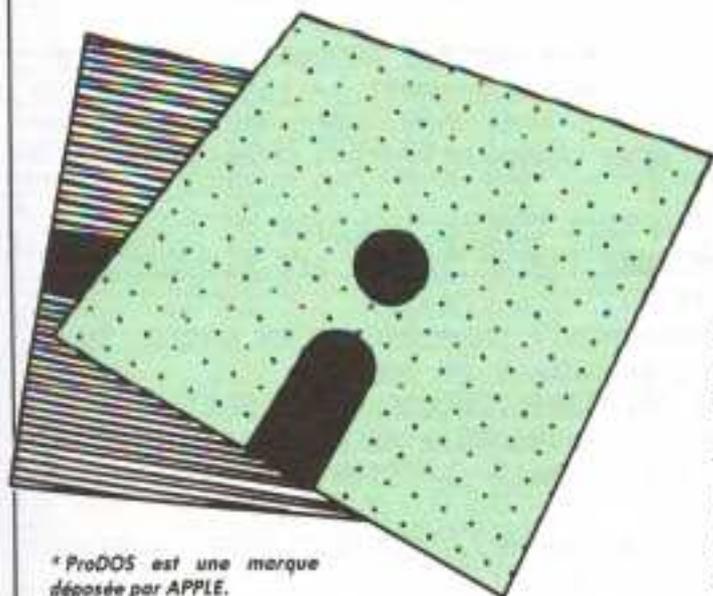
Il n'est jamais évident d'appeler une routine en assembleur à partir d'un programme écrit en Basic. Avec les commentaires de G. Creuzet et de A. Andrieux, de nombreux points deviennent plus clairs.

Il faut noter que l'ouvrage se termine par un Ampersoft, utilitaire inédit qui fera le bonheur des Lectrices et des Lecteurs, tout en les aidant à mieux comprendre ce difficile, mais passionnant langage, qu'est l'assembleur du 6502 et du 65C02.

Clément Ronard.



Si vous possédez un Apple IIc ou un Apple IIe... offrez-vous notre disquette n° 2
VERSION DOS 3.3 ou ProDOS*



* ProDOS est une marque déposée par APPLE.

Tous les programmes de

TREMPLIN MICRO

seront chez vous dans quelques jours.
(La version ProDOS n'est utilisable que si vous disposez déjà de ce système d'exploitation)

UTILISEZ NOTRE BULLETIN DE COMMANDE,
À LA FIN DE LA REVUE (vous pouvez, bien sûr, le recopier ou le photocopier).

Deux trucs pour votre Apple

1. INPUT EN MODE DIRECT

Immanquablement, si vous tapez un **INPUT A\$... RETURN** en mode direct, votre Apple préféré vous répondra par un **? ILLEGAL DIRECT ERROR ...** agaçant.

Il est possible de mettre fin à cette évidente mauvaise volonté par un judicieux

POKE 118,0 : INPUT A\$... RETURN

avec lequel le sympathique point d'interrogation de l'INPUT consent à apparaître.

Répondez **VIVE TOTO** puis interrogez votre Apple chéri par un **PRINT A\$... RETURN**, auquel il répondra : **VIVE TOTO**.

Pas plus difficile que cela !

2. RUN GÉNÉRALISÉ

Le truc est on ne peut plus connu, mais il étonnera tout de même vos amis non initiés.

Si vous dotez votre programme d'une ligne 0 ainsi rédigée (quel style !) :

0 POKE 214,255, nul ne pourra le lister après un premier RUN, toutes les commandes APPLESOFT se transformant précisément en un RUN désespérant.

Notez que le programme ci-dessous n'est qu'une démonstration, sur écran, pour le plaisir, du blablabla ci-contre et ci-dessus. A l'école, ou dans un club, ça peut tout de même servir. **NESTOR.**

DÉMONSTRATION SUR ÉCRAN

```
10 TEXT : PRINT CHR$(12); CHR$(21); HOME
20 T$ = " ": FOR I = 1 TO 10: T$ = T$ + " ____ ": NEXT I
30 HTAB 10: PRINT "DES TRUCS POUR APPLE": PRINT T$
40 VTAB 21: PRINT T$: GOSUB 205
50 POKE 34,3
60 GOSUB 130: READ Q$: PRINT Q$: PRINT : GOSUB 160
70 GOSUB 140: READ R$: GOSUB 190: READ R$: PRINT : PRINT : GOSUB 190: GOSUB
  B 200
80 GOSUB 130: READ Q$: PRINT Q$: PRINT : GOSUB 160
90 GOSUB 140: READ R$: GOSUB 190: READ R$: PRINT : GOSUB 190: READ R$: PRI
  NT : GOSUB 190: GOSUB 200: PRINT
100 PRINT : READ R$: GOSUB 190: GOSUB 200: PRINT
110 PRINT : READ R$: GOSUB 190: GOSUB 200
120 TEXT : HOME : END
130 HOME : PRINT : INVERSE : PRINT "QUESTION": NORMAL : PRINT : RETURN
140 INVERSE : PRINT "REPOSE": NORMAL : PRINT : RETURN
160 FOR I = 1 TO 10: FOR J = 1 TO 10: X = PEEK(49200): NEXT J
170 FOR K = 1 TO 100: IF ( PEEK(49249) ) > 127 THEN RETURN
180 NEXT K: NEXT I: RETURN
190 FOR I = 1 TO LEN(R$): FOR J = 1 TO 5: X = PEEK(49200): NEXT : PRINT
  MID$(R$,I,1);: NEXT : PRINT " ": RETURN
200 POKE 35,24: VTAB 23: PRINT " APPUYEZ SUR N'IMPORTE QUELLE TOUCHE ": VT
  AB 19: POKE 35,20: CALL - 198: POKE - 16368,0: WAIT - 16384,128,127:
  POKE - 16368,0: POKE 35,24
205 VTAB 23: PRINT " POMME OUVERTE ABREGERA VOTRE ATTENTE": VTAB 19: POKE
  35,20: RETURN
210 DATA COMMENT EXECUTER UN INPUT (OU UN GET) EN MODE IMMEDIAT ?
220 DATA "EN LA FAISANT PRECEDER PAR POKE 118,0", "$POKE 118,0:INPUT A$
  $POKE 118,0:GET A$"
230 DATA "COMMENT PROTEGER (ASSEZ SOMMAIREMENT) UN PROGRAMME EN BASIC, EN T
  RANSFORMANT LES COMMANDES DE L'APPLE EN AUTANT DE RUN!"
240 DATA "TOUT SIMPLEMENT EN TAPANT SA PREMIERE LIGNE COMME CECI:", "$0 P
  OKE 214,255", "APRES QUOI VOUS SEREZ VOUS-MEME PRIS AU PIEGE!"
250 DATA "UN CALL-151, SUIVI DE *D6:00 RETABLIRAIT LA SITUATION, MAIS ON NE
  PEUT JUSTEMENT PAS FAIRE DE CALL..."
260 DATA "MORALITE: SI VOUS UTILISEZ CE STRATAGEME... SAUVEZ VOTRE PROGRAMM
  E SUR DISQUETTE AVANT DE FAIRE UN PREMIER RUN!"
```

TRANSFERT D'UN ÉCRAN DE 80 COLONNES EN MÉMOIRE AUXILIAIRE

```

300 — 38          SEC
301 — 8D 55 C0   STA $C055
304 — 20 1C 03   JSR $031C
307 — A2 44      LDX $544
309 — 86 43      STX $43
30B — 20 11 C3   JSR $C311
30E — 8D 54 C0   STA $C054
311 — 20 1C 03   JSR $031C
314 — A2 40      LDX $540
316 — 86 43      STX $43
318 — 20 11 C3   JSR $C311
31B — 60         RTS
31C — A9 00      LDA $500
31E — 85 3C      STA $3C
320 — A2 04      LDX $504
322 — 86 3D      STX $3D
324 — A2 D0      LDX $5D0
326 — 86 3E      STX $3E
328 — A2 07      LDX $507
32A — 86 3F      STX $3F
32C — 85 42      STA $42
32E — 60         RTS
32F — 18         CLC
330 — 8D 55 C0   STA $C055
333 — 20 54 03   JSR $0354
336 — A2 44      LDX $544
338 — 86 3D      STX $3D
33A — A2 47      LDX $547
33C — 86 3F      STX $3F
33E — 20 11 C3   JSR $C311
341 — 18         CLC
342 — 8D 54 C0   STA $C054
345 — 20 54 03   JSR $0354
348 — A2 40      LDX $540
34A — 86 3D      STX $3D
34C — A2 43      LDX $543
34E — 86 3F      STX $3F
350 — 20 11 C3   JSR $C311
353 — 60         RTS
354 — A9 00      LDA $500
356 — 85 42      STA $42
358 — 85 3C      STA $3C
35A — A2 04      LDX $504
35C — 86 43      STX $43
35E — A2 D0      LDX $5D0
360 — 86 3E      STX $3E
362 — 60         RTS
    
```

```

300 : 38 8D 55 C0 20 1C 03 A2 44 86 43 20 11 C3 8D 54
C0 20 1C 03 A2 40 86 43 20 11 C3 60 A9 00 85 3C A2 04
86 3D A2 D0 86 3E A2 07 86 3F 85 42 60 18
330 : 8D 55 C0 20 54 03 A2 44 86 3D A2 47 86 3F 20 11
C3 18 8D 54 C0 20 54 03 A2 40 86 3D A2 43 86 3F 20 11
C3 60 A9 00 85 42 85 3C A2 04 86 43 A2 D0 86 3E 60
    
```

Le programme en langage machine transfère un écran de 80 colonnes (\$400-\$7D0) en mémoire auxiliaire (\$4000-\$43D0 et \$4400-\$47D0). Il travaille de la même manière que le programme en Basic que vous trouverez au verso de cette fiche.

UTILISATION

1. Activer la carte 80 colonnes par un PRINT CHR\$(4) ; "PRE1" (notez que le point-virgule n'est pas indispensable).
2. PRINT CHR\$(4) "BLOAD TRANS 80.LM"
3. CALL 768 (mémorisation)
4. CALL 815 (récupération)

PETIT PROGRAMME DÉMO POUR TRANSFERT DE 2 ÉCRANS

```

10 PRINT CHR$(4) "PRE3" : HOME : E = 8
20 PRINT CHR$(4) "BLOAD TRANS80.LM"
30 FOR I = 1 TO 110 : PRINT "AAAAAAAAAAAAAAAAAAAA" : :
NEXT
40 CALL 768 : HOME : REM MEMO ECRAN A
50 FOR I = 1 TO 110 : PRINT "BBBBBBBBBBBBBBBBBB" : : NEXT
: PRINT : VTAB 23 : INVERSE : PRINT "CTRL-C POUR ARRE-
TER" : : NORMAL
60 GET RS
70 GOSUB 150 : CALL 768 : GOSUB 170 : REM MEMO ECRAN
B
80 HOME : PRINT "RETOUR ECRAN A" : : GET RS : PRINT :
VTAB 23
90 CALL 815 : REM AFFICHE ECRAN A
100 GET RS : PRINT : HOME : PRINT "RETOUR ECRAN B" : :
GET RS : PRINT : VTAB 23
110 GOSUB 150 : CALL 815 : GOSUB 170 : REM AFFICHE
ECRAN B
120 GET RS : IF ASC (RS) < > 3 THEN 80
130 HOME : END
140 REM VALEURS POUR ECRAN B
150 POKE 776, PEEK (776) + E : POKE 789, PEEK (789) + E :
POKE 823, PEEK (823) + E : POKE 827, PEEK (827) + E :
POKE 841, PEEK (841) + E : POKE 845, PEEK (845) + E :
RETURN
160 REM RETABLISSEMENT DES VALEURS
170 POKE 776, PEEK (776) - E : POKE 789, PEEK (789) - E :
POKE 823, PEEK (823) - E : POKE 827, PEEK (827) - E :
POKE 841, PEEK (841) - E : POKE 845, PEEK (845) - E :
RETURN
    
```

EXEMPLE DE TRANSFERT D'UN ÉCRAN DE 80 COLONNES EN MÉMOIRE AUXILIAIRE

```

10 REM
20 REM
30 REM
40 REM
50 :
60 PRINT CHR$(4) "PRÉ3" : PRINT CHR$(12)
70 :
80 REM UTILISATION DE LA ROUTINE $C311 (SUR APPLE IIe et IIc)
90 :
100 POKE 769,32 : POKE 770,17 : POKE 771,195 : POKE 772,96
110 GOTO 360
120
130 REM RETENUE (CARRY) MISE À 1 OU À 0
140 :
150 POKE 768,56 : RETURN : REM CARRY = 1 (ÉCRITURE EN MÉMOIRE AUXILIAIRE)
160 POKE 768,24 : RETURN : REM CARRY = 0 (MÉMOIRE AUXILIAIRE VERS MÉMOIRE PRINCIPALE)
170
180 REM COPIE DE L'ÉCRAN (MÉMOIRE PRINCIPALE) À L'ADRESSE $4000 (MÉMOIRE AUXILIAIRE)
190 :
200 POKE 60,0 : POKE 61,4 : POKE 62,208 : POKE 63,7 : POKE 66,0 : POKE 67,64 : CALL 768
210
220 REM COPIE DE L'ÉCRAN (MÉMOIRE AUXILIAIRE) À L'ADRESSE $4400 (MÉMOIRE AUXILIAIRE)
230 :
240 POKE — 16299,0 : POKE 60,0 : POKE 61,4 : POKE 62,208 : POKE 63,7 : POKE 66,0 : POKE 67,68 :
CALL 768 : POKE — 16300,0 : RETURN
250
260 REM MÉMOIRE AUXILIAIRE ($4000) VERS MÉMOIRE PRINCIPALE ($400), PREMIÈRE PARTIE
270 :
280 POKE 60,0 : POKE 61,64 : POKE 62,208 : POKE 63,67 : POKE 66,0 : POKE 67,4 : CALL 768
290
300 REM MÉMOIRE AUXILIAIRE ($4400) VERS MÉMOIRE AUXILIAIRE ($400), SECONDE PARTIE
310 :
320 POKE — 16299,0 : POKE 60,0 : POKE 61,68 : POKE 62,208 : POKE 63,71 : POKE 66,0 : POKE 67,4 :
CALL 768 : POKE — 16300,0 : RETURN
330
340 REM ÉCRAN DÉMO REMPLI DE CHIFFRES POUR UN ESSAI DE TRANSFERT
350 :
360 FOR N = 1 TO 8 : A$ = A$ + "0123456789" : NEXT : B$ = "    32 espaces    " : FOR I = 1 TO
23 : PRINT A$ : : NEXT
370 :
380 REM COPIE ÉCRAN EN MÉMOIRE AUXILIAIRE
390 :
400 GOSUB 150 : GOSUB 200
410 :
420 REM FENÊTRE AU CENTRE DE LA PAGE
430 :
440 VTAB 8 : FOR N = 1 TO 8 : HTAB 25 : PRINT B$ : NEXT
450 :
460 REM ATTENTE D'UNE TOUCHE
470 :
480 GET C$
490 :
500 REM MÉMOIRE AUXILIAIRE VERS ÉCRAN
510 :
520 GOSUB 160 : GOSUB 280
530 END

```

ÉCRAN 80 COLONNES

Il ne faut pas oublier que, en mode 80 colonnes, un caractère sur deux est en mémoire écran (principale), tandis que l'autre est en mémoire auxiliaire, à la même adresse (de \$400 à \$7F0).

POKE — 16299,0 permet de lire en mémoire auxiliaire.

POKE — 16300,0 fait revenir en mémoire principale.

A = Accumulateur
M = Mémoire
C = Retenue (Carry)

POURQUOI ?

Pour soustraire, on utilise l'instruction **SBC** du 6502, alors que l'on fait appel à l'instruction **ADC** pour additionner. Mais là n'est pas la seule différence. Avant d'exécuter l'instruction **ADC**, le bit **C** de la retenue doit être mis à zéro par un **CLC**. Au contraire, dans le cas de la soustraction, **C** est forcé à 1 par l'instruction **SEC**.

Le microprocesseur utilise l'addition du complément à deux... pour soustraire. En fait, il se contente d'additionner et, pour lui, soustraire 6 de 9 n'existe pas. Il préfère totaliser $(-6) + 9$... ce qui, pour nous, revient au même.

Si, à l'issue d'une soustraction, on a $C = 1$, cela signifie que **M** (nombre à soustraire) était plus petit que **A** (nombre mis dans l'accumulateur). Si $C = 0$, cela prouve que le résultat est négatif... ou encore que l'on a fait un emprunt au rang suivant. Dans cette page, le caractère \square signifie RETURN.

Le but de notre mini-programme est de vous montrer comment se fait la soustraction, et de vous éclairer sur les mystères de la retenue. Commencez par un **CALL-151**, **RETURN**, puis tapez :

*300 : A5 06 38 E5 08 B0 03 C6 07 38 E5 09 B0 02 C6 07 85 06 A9 01 B0 02 A9 00 85 18 60 \square

Si maintenant vous tapez 300L, vous devez obtenir le listing ci-dessous.

PETIT PROGRAMME DE DÉMONSTRATION

300	—	A5 06	LDA \$06	La valeur de la Mémoire \$06 est mise dans A
302	—	38	SEC	Retenue forcée à 1
303	—	E5 08	SBC \$08	La Mémoire \$08 est soustraite de A.
305	—	B0 03	BCS \$030A	Si C = 1 on saute à \$309
307	—	C6 07	DEC \$07	C = 0 et on décrémente la Mémoire \$07 de 1
309	—	38	SEC	C remis à 1
30A	—	E5 09	SBC \$09	Le contenu de la Mémoire \$09 est soustrait de A
30C	—	B0 02	BCS \$0310	Si C = 1, saut à \$310
30E	—	C6 07	DEC \$07	C = 0 et on décrémente donc la Mémoire \$07 de 1
310	—	85 06	STA \$06	Le résultat des 2 soustractions est mis dans \$06
312	—	A9 01	LDA \$01	On place \$1 dans A
314	—	B0 02	BCS \$0318	Si la retenue est à 1, saut à \$318
316	—	A9 00	LDA \$00	On place \$0 dans A
318	—	85 18	STA \$18	Le contenu de A est inscrit dans la mémoire \$18
31A	—	60	RTS	Retour au Basic

DEMO

- *06 : 05 20 06 10 \square
*300G \square
*6.7 \square
0006 — EF 1F
*18 \square
- *06 : 00 0F FF FF \square
*6.7 \square
0006 — 00 0F
*300G \square
*6.7 \square
0006 — 02 0D
*18 \square
- *06 : FF 0F FF FF \square
*300G \square
*6.7 \square
0006 — 01 0E
*18 \square
- *06 : FF 0F EF 00 \square
*300G \square
*6.7 \square
0006 — 10 0F
*18 \square
0018 — 01

Le programme place le contenu de la mémoire \$06 dans A, puis lui soustrait successivement les contenus des mémoires \$08 et \$09. Quand C est à 0, il décrémente la mémoire \$07 de 1. Pour terminer, il inscrit la valeur de C à l'adresse \$18. Si vous exécutez exactement les opérations indiquées, vous constaterez que :

- Vous alimentez vous-même 6.7.8.9 en tapant *06 : 05 20 06 10 (notez que nous avons écrit la valeur \$2005 à l'envers, comme il est normal de le faire, partie basse d'abord). *300G correspond au RUN du Basic. Le résultat \$1F EF est correct ($\$2005 - 06 - 16$) = \$1F EF. En valeurs décimales, cela donne $8197 - 22 = 8175$. La retenue est à 1 puisque, lors de la seconde soustraction, il n'y a pas eu débordement.
- Nouvelles valeurs, mais cette fois la retenue est à 0 car il y a eu débordement (normal avec \$FF = 255). \$0F 00 — \$FF — \$FF = \$D02 : exact ($3840 - 510 = 3330$)
- La retenue est toujours à 1. La première soustraction ne provoquait pas de débordement, mais la seconde oui.
- Cette fois, pas de débordement. On retrouve bien la retenue à 1.

LISTAGE-CONTRÔLE D'UN FICHER "T"

SUR ÉCRAN OU
SUR IMPRIMANTE

Si vous désirez consulter ou éditer rapidement un fichier "T" (EXEC, par exemple), sans tenir compte de la longueur des lignes et de la présentation, inutile de vous compliquer l'existence : la routine ci-après se révélera rapide et efficace.

Naturellement, vous pourriez la compléter en prévoyant la saisie des paramètres de votre imprimante, mais elle ne tarderait pas à faire double emploi avec le programme, beaucoup plus complet, de la page 27... ce qui n'est pas le but recherché.

```
10 A$ = "" : REM LA VARIABLE A$ DOIT ÊTRE ABSOLUMENT LA PREMIÈRE DU PROGRAMME.
20 D$ = CHR$(4)
30 TEXT : PRINT CHR$(12) : CHR$(21) : HOME
40 INVERSE : PRINT "-* LECTURE-CONTROLÉ D'UN FICHER (T) *-": NORMAL : POKE 34,2
50 FOR I = 768 TO 789 : READ P : POKE I, P : NEXT
60 VTAB 22 : PRINT "----- (?) POUR CATALOGUE -----"
70 PRINT : VTAB 3 : INPUT "TITRE A LISTER " : F$ : IF F$ = "" THEN 250
80 IF F$ = "?" THEN HOME : PRINT : PRINT D$ "CATALOG" : GET F$ : PRINT : HOME : GOTO 70
90 IF ASC (F$) < 65 OR ASC (F$) > 90 THEN 70
100 ONERR GOTO 220
110 PRINT : PRINT D$ "VERIFY" F$
120 PRINT : VTAB 5 : CALL-958 : INPUT "(I) POUR IMPRIMANTE " : I$
130 PRINT : HOME
140 IF I$ = "I" THEN PRINT D$ "PR&1" : PRINT
150 ONERR GOTO 220
160 PRINT D$ "OPEN" F$ ".D1"
170 PRINT D$ "READ" F$
180 CALL 768 : T$ = MID$(A$,1) : PRINT T$
190 GOTO 170
200 PRINT D$ "CLOSE"
210 PRINT : PRINT D$ "PR&0" : GET A$ : GOTO 250
220 ER = PEEK (222)
230 IF ER = 6 THEN PRINT : PRINT "FICHER INEXISTANT" : GOTO 70
240 IF ER = 5 THEN POKE 216,0 : GOTO 200
250 TEXT : HOME : END
260 DATA 162,0,32,111,253,160,2,138,145,105,200,169,0,145,105,200,169,2,145,105,138,96
```

*No pas utiliser cette
mention sous ProDOS*

L'APPLESOFT accepte indifféremment ces deux syntaxes :
PRINT D\$, "OPEN" F\$, ".D1" et
PRINT D\$ "OPEN" F\$ ".D1"
mais cette tolérance doit être utilisée avec circonspection.

POURQUOI ?

Pourquoi la variable A\$ doit-elle être la première du programme ? Réponse qui n'explique pas tout, mais justifie cet impératif : parce qu'il est facile de la retrouver, sans passer par un inutile programme de recherche (on sait parfaitement comment Applesoft range ses variables).

Le pointeur \$69 — \$6A indique l'adresse du début d'une zone où le descriptif de chaque variable occupe 7 octets, comme les 7 Mercenaires.

Les deux premiers renferment le code ASCII de chacun de ses deux premiers caractères... et le troisième sa longueur. Ainsi, si la variable A\$ = "TOTO" on lira quelque part ces 7 octets : 41 00 04 DD 87 00 00 ou \$41 = code ASCII de "A", \$04 indique une chaîne de caractères, \$04 sa longueur et DD 87 (87 DD) son adresse (bidon ici).

(suite page 45)

**FAISONS UN ESSAI, EN MODE DIRECT.
TAPEZ AVEC MOI :**

- 1** NEW pour tout annuler
(sauvez d'abord votre programme !)
- 2** AA\$ = "ESSAI" : RETURN
- 3** CALL-151 : RETURN
(l'astérisque va apparaître)
- 4** * 69 - 6A ... et RETURN
(la réponse est 0069 - 04 08)
- 5** * 804 - 80A ... RETURN
(0804, c'est 04 08 mais inversés et 80A, c'est \$804 + \$7).

VOUS ALLEZ LIRE :

0804 — 41 C1 05 DC
0808 — 87 00 00

- \$41 = valeur ASCII de "A" (en hexa)
- \$C1 = valeur ASCII de "A" (toujours en hexa, mais dans un autre registre... si je puis dire)
- \$05 = longueur de "A\$" (Essai)
- \$DC\$87 = adresse (à l'envers) où est mémorisé le mot "ESSAI"

TAPEZ :

- 6** 87DC . 87E2 ... RETURN et votre grand ami des longues soirées, votre très cher Apple vous répond par :

87DC = 45 53 53 41
87E0 = 49 4C E8

Contrôlez : ce sont bien les codes ASCII de votre "ESSAI"

J'espère avoir été clair... mais je n'en suis pas sûr. NESTOR.

Si la variable n'a qu'un caractère, le second octet est figuré par \$80.

- // Dans le cas de notre programme (voir ci-dessous), l'adresse où se trouve la variable est celle du buffer : \$200. Vous pensez sans doute que l'on joue ainsi un bon tour à l'Apple, et vous avez raison. Le pire, c'est que cela marche à tous les coups !

POUR EN SAVOIR PLUS

Notre petite routine en langage machine (\$300 - \$315)* fait appel au sous-programme GETLN1 (\$FD6F) de l'Apple. A vrai dire, il s'agit là d'une entrée particulière de GETLN (\$FD6A), permettant tout simplement de ne pas afficher le prompt.

Vous savez peut-être que GETLN gère une entrée-ligne écrite à partir de l'adresse \$200. Dans ce cas, le registre X contient la longueur de la ligne (le code du prompt est quant à lui inscrit à l'adresse \$33).

Examinons en détail les 22 octets de notre micro-programme.

300 — A2 00	LDX E\$00	Le registre X est mis à zéro.
302 — 20 6F FD	JSR \$FD6F	Saut vers GETLN1 (qui va effectuer les 4/5 ^e du travail).
305 — A0 02	LDY E\$02	Le registre Y est mis à \$2
307 — 8A	TXA	Le contenu de X (longueur de la ligne obtenue par GETLN1) est mis dans l'accumulateur A.
308 — 91 69	STA (\$69), Y	Cette longueur est écrite à l'adresse contenue dans 69-6A + Y (adressage indirect).
30A — C8	INY	Y = Y + 1, soit 3
30B — A9 00	LDA E\$00	Accumulateur A mis à 0...
30D — 91 69	STA (\$69), Y	... et cette valeur est écrite à l'adresse contenue dans 69-6A + Y, c'est-à-dire : 3
30F — C8	INY	Y = Y + 1 soit 4
310 — A9 02	LDA E\$02	Accumulateur mis à \$2... (\$02 = partie haute de l'adresse \$200)
312 — 91 69	STA (\$69), Y	... puis écriture à l'adresse 69-6A + Y (ou + 4, si vous préférez)
314 — 8A	TXA	On recharge A avec le contenu du registre X
315 — 60	RTS	Retour au Basic.

De cette manière votre Apple sait que la variable A\$ (première du programme) a une longueur de, qu'il doit la lire à partir de l'adresse \$200, ce qui le comble évidemment de joie !

* Pour lire cette routine, taper RUN, RESET, CALL-151, RETURN et 300L. Ensuite, pour sortir du moniteur, taper GTRL-C et RETURN, ou bien RESET.



Un drive pour 1 500 F ? Banco !

Un drive pour Apple-II e à 1 500 F... de quoi rester rêveur ! Quand, en plus, l'objet est compact et fort joli, on se pose des questions. Pourtant, un prix aussi bas ne cache-t-il pas quelques mauvaises surprises ? Nous avons voulu en avoir le cœur net et, pour vous, nous avons testé les drives Distar. Bien nous en a pris car, tout compte fait, le bilan est très positif. Jugez plutôt...

Peut-être ne l'avez-vous pas encore remarqué mais, à la rédaction de *Tremplin Micro* on serait plutôt du genre publiphobes. Non par conviction, mais simplement parce que, à force de se faire échauder, on finit par redouter l'eau froide.

GUY-HACHETTE

Ainsi, quand nous avons vu apparaître les premières annonces pour les drives compatibles Apple, avons-nous adopté une attitude plutôt réservée. Un sursaut de conscience professionnelle étant intervenu au moment où des Lecteurs nous question-

naient pour savoir ce qu'il fallait penser des offres alléchantes parues partout dans la presse spécialisée, nous nous sommes rués chez l'importateur, bien décidés à tester ces drives à 1 500 F.

Le marché de la micro-informatique est encore tout jeune mais très prometteur : il attire donc pas mal de gens dont la vocation principale est de «faire des sous». Résultat, certains importent et distribuent des ordinateurs comme ils auraient importé et distribué des chars à bœufs il y a quelques siècles. C'est pourquoi nous n'hésitons pas à recommander à nos Lecteurs la plus grande

méfiance face à des propositions mirobolantes.

PGM-Informatique, qui assure la distribution en France des drives Distar, se démarque immédiatement de ses concurrents par un savoir-faire certain. Nous y avons trouvé des gens compétents, tant côté technique que commercial, sûrs de leurs produits. L'expérience commençait donc plutôt bien, et c'est sur ce premier contact rassurant que nous nous sommes retrouvés face à trois petites boîtes contenant respectivement un drive pour Apple-II, un contrôleur compatible et un drive spécialement étudié pour le II c.

Premier avantage : la demi-hauteur

Les lecteurs de disquettes commercialisés par Apple reposent sur une base Shugart, dont la réputation n'est plus à faire. La technologie mise en œuvre commence toutefois à accuser son âge car depuis dix ans, bien des choses ont changé. Les lecteurs Distar appartiennent à la *Half Height Family*, ce qui, en clair, signifie qu'ils sont moitié moins hauts que les autres. Cette réduction de taille peut se faire sans aucune concession à la qualité et résulte simplement d'une meilleure intégration des différents composants et de l'adoption d'un entraînement direct plutôt que par courroie. C'est là une solution à laquelle tout le monde vient progressivement, Apple en tête, puisque le drive installé à

l'intérieur même du II c n'est rien d'autre qu'un *Half Height..* miniaturisé.

Livré dans un coffret métallique beige, le drive Distar tient donc moitié moins de place qu'un modèle classique. Fini, donc, les problèmes d'encombrement et les moniteurs trop haut perchés ! Quant aux bricoleurs et adeptes de claviers détachables, ils pourront envisager de regrouper le «board» Apple, son alimentation et ses deux drives dans un unique coffret, plus fonctionnel que la configuration actuelle, avec ses nombreux câbles.

L'unique commande accessible à l'utilisateur concerne la traditionnelle trappe de verrouillage qui est, ici, remplacée par un court levier. Celui-ci assure exactement les mêmes fonctions :

- positionnement correct de la disquette sur le galet d'entraînement ;

- verrouillage empêchant tout retrait intempestif du disque, le simple fait d'actionner ce levier provoquant immédiatement le retrait des têtes de la surface du disque.

Inutile de dire que nous n'avons pas résisté à la tentation d'une petite visite à l'intérieur de la boîte. Le démontage, simple, nous a permis de découvrir un circuit principal très propre, complété par un circuit de contrôle du moteur très accessible. Les circuits «critiques» (ceux qui, statistiquement, rendent l'âme les premiers en cas de panne ou de fausse manœuvre) ne sont pas

Carte contrôleur : compatible, mais...

La société PGM-Informatique distribue également des cartes contrôleurs compatibles, vendues 500 F environ. Nous avons testé l'une d'entre elles.

Un premier examen ne laisse aucun doute sur l'origine de la carte : identique au contrôleur Apple, du tracé des pistes jusqu'au choix des composants, elle arrive en droite ligne de l'usine d'un sous-traitant qui, produisant pour Cupertino, en profite pour arrondir ses fins de mois avec sa production personnelle, sans marque (la sérigraphie Apple est absente).

Nous avons testé cette carte avec des drives Apple et avec le drive Distar ; l'initialisation d'un disque et les opérations de lecture/écriture prennent le même temps qu'avec le contrôleur Apple. Par contre, dès que l'on travaille sur des fichiers Texte, on remarque des différences significatives, voire spectaculaires. La lecture de 1000 variables numériques en fichier T (35 secteurs) nécessite 1,12 minute sur carte Apple et 1,27 sur carte compatible. En écriture, Apple creuse l'écart : 38 secondes contre 1,10 minute pour le compatible ! Explication : les temps d'arrêt entre chaque secteur sont plus longs.

Au départ, nous pensions que ce phénomène était dû à un problème logiciel sur les EPROM 5 et 6 ; un examen plus approfondi nous a montré que deux résistances avaient des valeurs différentes (1000 K d'écart !).

Si vous devez écrire souvent des fichiers de données, le contrôleur Apple se révèle donc plus performant et justifiera peut-être les quelques centaines de francs supplémentaires.

Dos 3.3	Sur contrôleur Apple		Sur contrôleur compatible	
	Drive Apple	Drive Distar	Drive Apple	Drive Distar
Initialisation d'un disque (Hello, 2 sect.)	30"	31"	30"	31"
LQAD programme Applesoft (32 secteurs)	9"	9"	10"	9"
SAVE programme Applesoft (32 secteurs)	13"	13"	13"	13"
Lecture fichier T 1000 variables (35 secteurs)	1'12"	1'12"	1'24"	1'27"
Ecriture fichier T 1000 variables (35 secteurs)	38"	38"	1'10"	1'10"

Exclusif : les 3,5 pouces Mac & Apple-IIe arrivent !

Les utilisateurs de Macintosh déplorent souvent le côté «grille-pain» de la machine qui, utilisée en solo, a la fâcheuse habitude d'éjecter ses disques dès qu'elle n'y trouve plus la place suffisante pour stocker ses innombrables «passes». Très vite, un drive externe apparaît donc indispensable. Et pour l'instant, le drive Apple reste la seule alternative.

Au moment où nous terminons ce dossier, PGM-Informatique teste des drives 3,5 pouces, compatibles Mac. L'un d'eux permet de stocker, sur une simple disquette Mac, 1,2 megaoctet ! Le côté grille-pain du Mac serait enfin résolu, de façon plus élégante et moins ruineuse qu'avec un disque dur. Et on pourrait peut-être alors espérer pouvoir enfin travailler sous Mac-Write et Mac-Paint, en version complète, toutes typographies possibles, avec plus de 900 K disponibles pour les fichiers ! Mac deviendrait capable de tenir ses promesses et d'imprimer, par exemple, plus de 20 pages à la suite...

La famille Apple-II n'est pas oubliée non plus et on trouvera bientôt de nombreuses unités de disquettes compatibles, en standard 3,5 pouces. La plupart seront capables de stocker jusqu'à 500 K par face, ce qui redonnera du tonus au bon vieux II e.

Vous le voyez, ça bouge beaucoup dans ce domaine, les tests sont en cours chez plusieurs importateurs...

Souhaitons qu'ils soient positifs et que ce matériel arrive bientôt dans les vitrines !

soudés, mais montés sur support : les éventuelles interventions techniques seront ainsi quasi instantanées, évitant à l'utilisateur une longue immobilisation de son matériel.

Côté poids, les drives Distar n'ont rien à envier aux autres et pèsent près de 2 kg. C'est très bien ainsi, car il s'agit d'un gage de fiabilité. Le lourd volant d'inertie en métal injecté est le garant d'une vitesse de rotation parfaitement stable, condition indispensable pour un transfert fiable des données. Soumis à un contrôle logiciel de la vitesse de rotation, le drive Distar s'est révélé meilleur que des drives Apple, aucune variation sensible n'étant décelable. En outre, l'ajustement de la vitesse peut, en cas de besoin, être effectué par un minuscule potentiomètre. Par contre, si vous êtes bricoleur, ne vous fiez pas à la piste stroboscopique imprimée sur le volant d'inertie : ces drives compatibles peuvent être utilisés sur diverses machines, à des vitesses de rotation différentes ; pour Apple, le standard est fixé à 200 tours/mminute, norme facilement contrôlable par certains utilitaires du marché (Copy II Plus et Dos Diversi, entre autres).

La vitesse de rotation est une chose... le transfert des data en est une autre. Nous avons soumis le drive Distar à différents tests

spécialement concoctés et nous avons trouvé sensiblement les mêmes mesures que sur les drives Apple. Certains revendeurs affirment que les Half Size sont plus rapides... c'est faux : pourquoi utiliser de tels arguments quand le prix, la taille et les performances réelles constituent autant d'arguments de vente... véridiques ?

Un petit coup d'œil sur le tableau publié sur cette page vous permettra de vous faire une idée plus précise de l'efficacité du drive Distar.

On adopte !

Parfaitement compatible Apple, le drive Distar peut être monté à tout moment sur n'importe quelle carte contrôleur Apple... en plus ou à la place d'un modèle de la marque. Moitié moins cher, moitié moins haut, moins bruyant mais aussi performant que les modèles *designed in Cupertino*, le drive Distar devrait intéresser tous les utilisateurs d'Apple-II et IIe, car il constitue un excellent choix. Le tarif alléchant n'implique aucune concession sur la qualité ou les performances et c'est là un point essentiel à nos yeux. La pub était prometteuse, les promesses sont tenues...

G.-M. C.

Un drive compatible pour l'Apple-II c

— Un drive pour II c, s'il vous plaît !

— Oui monsieur. A 1 900 ou à 2 900 F ?

— Pourquoi, quelle est donc la différence ?

...Telle est, justement, la question à laquelle nous avons voulu répondre en testant le drive Distar, proposé en version II c ! Pour ce qui est de la technique, relisez le test ci-contre. Tout ce qui est dit sur le drive du II e est valable pour celui du II c. La seule différence matérielle concerne le

câble de raccordement, qui est ici au standard II c (blindé et non en nappe) et qui se monte sur la prise DB 19 du portable Apple, sans qu'il soit besoin d'utiliser le moindre adaptateur.

Un peu plus petit que le drive Apple, le Distar en version II c a exactement les mêmes performances. Il est, en revanche, plus bruyant et moins esthétique... mais coûte 1 000 F de moins. La sécurité des données étant assurée, à vous de voir si le plaisir de vos yeux justifie la différence...

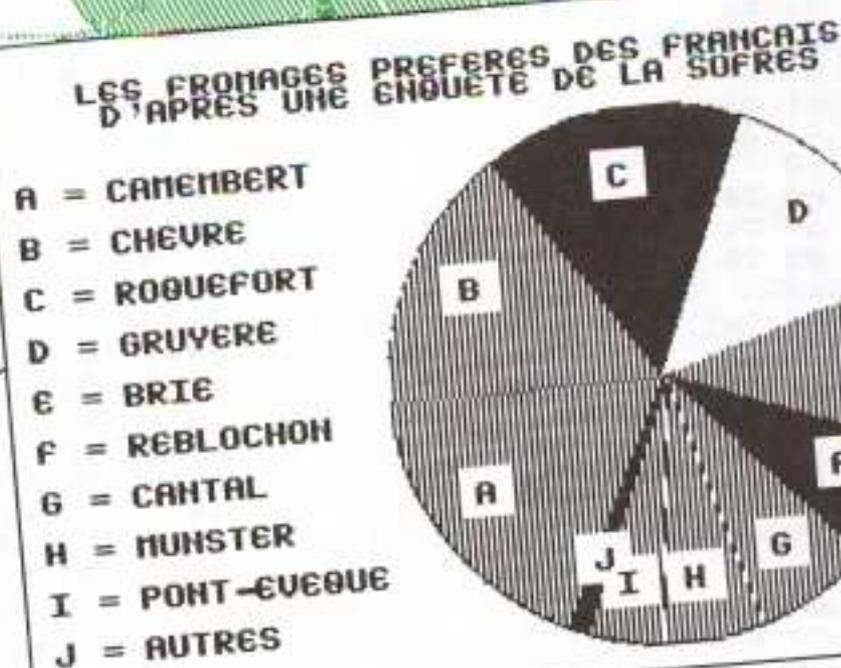
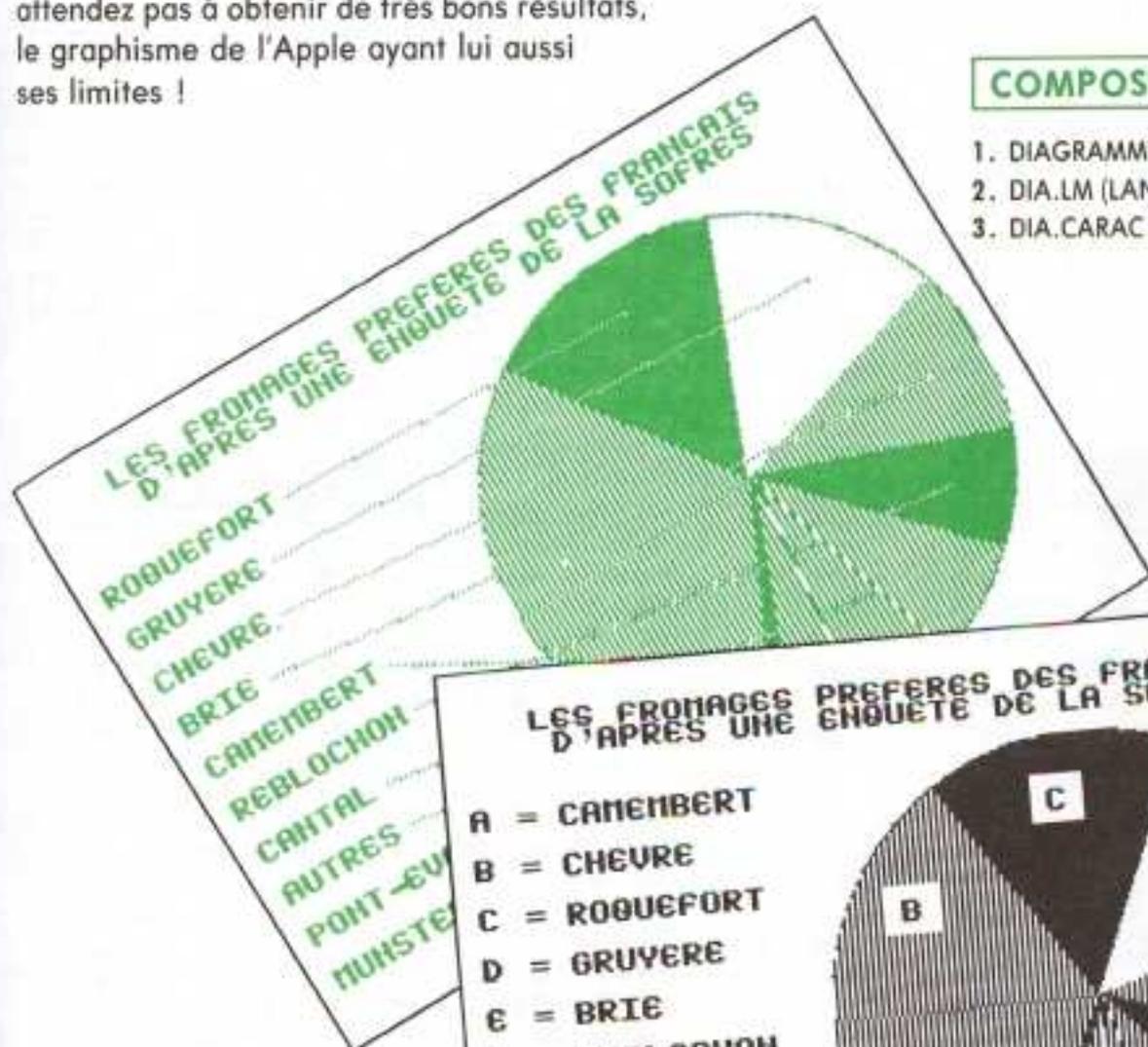
C E programme forme un tout, mais il est compatible avec le programme *HISTOGRAMME* paru dans le numéro 1 de *Tremplin Micro*, dont il peut reprendre les fichiers, à condition de modifier leurs titres en les faisant suivre de la mention «H» (taper, par exemple, *RENAME HISTO.1, HISTO.1H*, pour effectuer cette modification).

A noter que *DIAGRAMME* (ou *CHIFFRES EN ROND...* vous avez le choix des mots !) n'accepte que 10 valeurs, contre 12 pour *HISTOGRAMME*. Quand cette limite est dépassée, le programme ne retient que les 10 meilleures valeurs. Ceci dit, vous êtes libre de modifier les paramètres, mais ne vous attendez pas à obtenir de très bons résultats, le graphisme de l'Apple ayant lui aussi ses limites !

Vos chiffres valent bien un fromage !

COMPOSITION

1. DIAGRAMME (BASIC)
2. DIA.LM (LANGAGE MACHINE)
3. DIA.CARAC (LANGAGE MACHINE)



Exemples obtenus sur un Apple IIe, avec une imprimante Apple DMP et une interface graphique MID

* ProDOS est une marque déposée par APPLE.

6000-	A6 22	LDX	\$22	6075-	60	PLA	
6002-	8A	TXA		6076-	A8	TAY	
6003-	85 1C	STA	\$1C	6077-	60	PLA	
6005-	20 24 FC	JSR	\$FC24	6078-	A2 DF	LDX	£\$DF
6008-	A0 80	LDY	£\$80	607A-	BA	TSX	
600A-	A5 28	LDA	\$28	607B-	48	PHA	
600C-	85 2A	STA	\$2A	607C-	98	TYA	
600E-	A5 29	LDA	\$29	607D-	48	PHA	
6010-	29 83	AND	£\$83	607E-	60	RTS	
6012-	85 E6	ORA	\$E6	607F-	EA	NOP	
6014-	85 2B	STA	\$2B	6080-	EA	NOP	
6016-	B1 28	LDA	(£28), Y	6081-	EA	NOP	
6018-	C9 20	CMP	£\$20	6082-	EA	NOP	
601A-	80 85	BCS	\$6021	6083-	EA	NOP	
601C-	69 C0	ADC	£\$C0	6084-	EA	NOP	
601E-	CE 52 60	DEC	\$6052	6085-	AE 7F 60	LDX	\$607F
6021-	C9 60	CMP	£\$60	6088-	AC 80 60	LDY	\$6080
6023-	B0 85	BCS	\$602A	608B-	AD 81 60	LDA	\$6081
6025-	69 80	ADC	£\$80	608E-	20 57 F4	JSR	\$F457
6027-	CE 52 60	DEC	\$6052	6091-	AD 82 60	LDA	\$6082
602A-	C9 80	CMP	£\$80	6094-	AE 83 60	LDX	\$6083
602C-	B0 85	BCS	\$6033	6097-	AC 84 60	LDY	\$6084
602E-	69 40	ADC	£\$40	609A-	20 3A F5	JSR	\$F53A
6030-	CE 52 60	DEC	\$6052	609D-	EA	NOP	
6033-	C9 A0	CMP	£\$A0	609E-	A2 83	LDX	£\$83
6035-	90 31	BCC	\$6068	60A0-	20 EC F6	JSR	\$F6EC
6037-	29 7F	AND	£\$7F	60A3-	AE 82 60	LDX	\$6082
6039-	18	CLC		60A6-	AC 83 60	LDY	\$6083
603A-	2A	ROL		60A9-	AD 84 60	LDA	\$6084
603B-	2A	ROL		60AC-	20 57 F4	JSR	\$F457
603C-	2A	ROL		60AF-	60	RTS	
603D-	48	PHA					
603E-	2A	ROL					
603F-	29 83	AND	£\$83				
6041-	89 60	ORA	£\$60				
6043-	8D 50 60	STA	\$6050				
6046-	68	PLA					
6047-	29 FB	AND	£\$FB				
6049-	8D 4F 60	STA	\$604F				
604C-	A2 80	LDX	£\$80				
604E-	BD 80 60	LDA	\$6080, X				
6051-	49 80	EOR	£\$80				
6053-	11 2A	ORA	(£2A), Y				
6055-	91 2A	STA	(£2A), Y				
6057-	18	CLC					
6058-	A5 2B	LDA	\$2B				
605A-	69 84	ADC	£\$84				
605C-	85 2B	STA	\$2B				
605E-	E8	INX					
605F-	E8 88	CPX	£\$88				
6061-	90 EB	BCC	\$604E				
6063-	A9 80	LDA	£\$80				
6065-	8D 52 60	STA	\$6052				
6068-	C8	INY					
6069-	C4 21	CPY	\$21				
606B-	90 9D	BCC	\$608A				
606D-	A6 1C	LDX	\$1C				
606F-	E8	INX					
6070-	E4 23	CPX	\$23				
6072-	90 8E	BCC	\$6082				
6074-	60	RTS					

SCALL-151

DIA.LM

*6000.6080

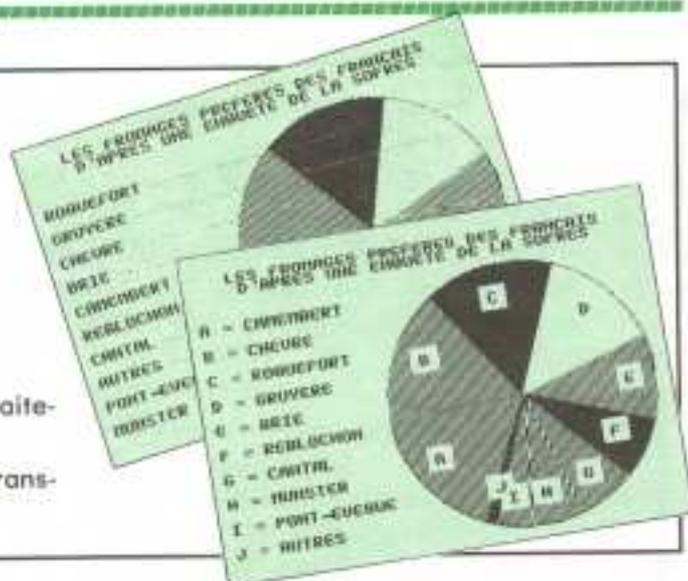
6000-	A6 22	8A 85 1C 20 24 FC
6008-	A0 80	A5 28 85 2A A5 29
6010-	29 83	85 E6 85 2B B1 28
6018-	C9 20	B0 85 69 C0 CE 52
6020-	60 C9 60	B0 85 69 80 CE
6028-	52 60 C9 80	B0 85 69 40
6030-	CE 52 60 C9 A0	90 31 29
6038-	7F 18 2A 2A 2A	48 2A 29
6040-	83 89 60 8D 50 60 68	29
6048-	F8 8D 4F 60 A2 80	BD 80
6050-	60 49 80 11 2A 91 2A	18
6058-	A5 2B 69 84 85 2B E8	E0
6060-	80 90 EB A9 80 8D 52 60	
6068-	C8 C4 21 90 9D A6 1C E8	
6070-	E4 23 90 8E 60 68 A8 68	
6078-	A2 DF BA 48 98 48 60 EA	
6080-	EA EA EA EA EA AE 7F 60	
6088-	AC 80 60 AD 81 60 20 57	
6090-	F4 AD 82 60 AE 83 60 AC	
6098-	84 60 20 3A F5 EA A2 83	
60A0-	20 EC F6 AE 82 60 AC 83	
60A8-	60 AD 84 60 20 57 F4 60	

BSAVE DIA.LM, A\$6000, L\$80

DIA.LM

TROIS ROUTINES :

- 1 **6000.6074** (qui permet d'écrire en page graphique, en utilisant la fonte de caractères DIA.CARAC). Cette routine doit tout à BEAGLE BROS (qui diffuse l'excellent *APPLE MECHANIC* dans le monde entier). Vous la trouverez dans *FAIRE.LM* de Tremplin Micro n°1.
- 2 **6075.607E**, petite routine recommandée avant le traitement des erreurs.
- 3 **607F.60AF** qui trace le diagramme, mais utilise une transmission de données trop lente pour être efficace.



LOMEM protège à la fois la page graphique HGR2, et les deux routines en langage machine.

```

10 LOMEM: 25600
20 TEXT : PRINT CHR$(12); CHR$(21); HOME
30 DIM Z(12),TE$(12),C(12),X2(12),Y2(12),A(12),T1$(2)
40 HGR2 : TEXT :D$ = CHR$(4)
50 R = 85:PI = 3.141593:M = 1:R0 = 1:X1 = 279 - R:Y1 =
    192 - R:CA = 24576:AD = 24783:AE = 24789: GOSUB 600
60 :
70 REM *****
80 REM ***      MENU      ***
90 REM *****
100 :
110 HOME :T$ = "          ": INVERSE : RESTORE
120 FOR I = 3 TO 5: VTAB I: HTAB 12: PRINT T$: NEXT : V
    TAB 4: HTAB 13: PRINT "CHIFFRES EN ROND": NORMAL
130 FOR I = 1 TO 10:RE = I: IF I > 9 THEN RE = 0
140 VTAB 7 + I: HTAB 9: READ T$: PRINT RE;" . ";T$: NEXT
    : TEXT
150 VTAB 19: CALL - 958: CALL - 198: HTAB 13: PRINT "
    VOTRE CHOIX ";: POKE 2043,223: GET R$
160 RE = ASC (R$): IF RE = ASC ("?") THEN PRINT : HOM
    E : PRINT D$"CATALOG": GET R$: PRINT : GOTO 110
170 IF RE = 48 THEN 2100
180 RE = RE - 48: IF RE < 1 OR RE > 9 THEN PRINT : GOTO
    150
190 IF (RE = 6 OR RE = 8) AND GH = 0 THEN VTAB 22: HTA
    B 10: CALL - 198: PRINT "PAGE GRAPHIQUE VIERGE": FO
    R I = 1 TO 1500: NEXT : GOTO 150
200 IF N < > 0 AND Z(1) < > 0 THEN 220
210 IF RE = 4 OR RE = 7 THEN VTAB 22: HTAB 3: CALL -
    198: PRINT "IL N'Y A PAS DE DONNEES EN MEMOIRE": FOR
    I = 1 TO 1500: NEXT : GOTO 150
220 ON RE GOTO 1510,1700,1920,340,240,1330,1420,230,270
230 X = 137: GOTO 1260
240 VTAB 21: HTAB 1: INVERSE : PRINT "L";: NORMAL : PRI
    NT "IGNES TRACANTES OU ";: INVERSE : PRINT "C";: NOR
    
```

(Suite page 52)

POKE 2043,223 trans-
forme le curseur de
l'Apple IIc en SOULIGNE
(sans effet sur IIe).

Il faut bien prévoir l'ab-
sence de données en
mémoire.

Deux catégories de
diagrammes.

```

MAL : PRINT "ODE ALPHABETIQUE ";; GET R$: IF R$ = "L
  " THEN M = 1
250 IF R$ = "C" THEN M = 2
260 GOTO 150
270 VTABLE 21: HTABLE 13: INVERSE : PRINT "ROTATION";: NORM
AL : PRINT " 1-360 > ";: INPUT "R$:"; R$: IF VAL (R$)
  > 0 AND VAL (R$) < 361 THEN RO = VAL (R$):RM = 0
280 GOTO 150
290 :
300 REM *****
310 REM ***     PREPARATION     ***
320 REM *****
330 :
340 IF GH < > 1 THEN 380
350 VTABLE 22: HTABLE 1: PRINT "<0> POUR EFFACER LE GRAPHIQ
UE ACTUEL ";; GET R$: IF R$ = "0" THEN 410
360 IF R$ < > "N" THEN 150
370 POKE 49237,0: POKE 49234,0: POKE 49239,0: POKE 4923
2,0: CALL - 198: GET R$: PRINT : TEXT : GOTO 150
380 TR = 0: FOR I = 1 TO N - 1: IF Z(I) > = Z(I + 1) TH
EN 400
390 T = Z(I):Z(I) = Z(I + 1):Z(I + 1) = T:T = C(I):C(I)
  = C(I + 1):C(I + 1) = T:T = 0:L$ = TE$(I):TE$(I) = T
  E$(I + 1):TE$(I + 1) = L$:L$ = "":TR = 1
400 NEXT : IF TR = 1 THEN 380
410 IF RM = 0 THEN RO = PI / 4 - RO / 360 * PI:RM = 1
420 T = 0: FOR I = 1 TO N:C(I) = VAL ( MID$ ("123457612
3",I,1)):T = T + Z(I): NEXT
430 FOR I = 1 TO N:A(I) = Z(I) / T * PI: NEXT
440 :
450 REM *****
460 REM ***     TRACE DU DIAGRAMME     ***
470 REM *****
480 :
490 HCOLOR= 0: HPLLOT 0,0: CALL - 3082: HGR2
500 FOR I = 1 TO N
510 A = B
520 B = B + A(I) * 2: POKE AD,X1: POKE AD + 1,0: POKE AD
  + 2,Y1
530 FOR J = A TO B STEP 0.01
540 HCOLOR= C(I)
550 X = X1 + R * COS (J - RO)
560 Y = Y1 + R * SIN (J - RO)
570 POKE AD + 3,X - INT (X / 256) * 256: POKE AD + 4,
  INT (X / 256): POKE AD + 5,Y: CALL 24709
580 NEXT
590 SI = (A + B) / 2 - RO
600 X2(I) = X1 + R / 1.3 * COS (SI)
610 Y2(I) = Y1 + R / 1.3 * SIN (SI)
620 NEXT : REM FACULT.:GOTO 760
630 :
640 REM *****
650 REM ***     TABLE DE CARACTERES     ***

```

◀ Jouez avec cette valeur pour éviter certains effets de moirage. Nos exemples utilisaient 320.

◀ S'il n'y a pas de diagramme dans la page graphique, il est inutile de l'effacer.

◀ Affiche la page HGR2, mais sans effacement.

◀ Tri des valeurs (la plus grande en tête).

Le tri pourrait être facultatif. Il suffirait d'ajouter une ligne 375 posant la question sur le modèle de la ligne 350 !

◀ Mélange (discutable) des couleurs.

◀ Calcul de l'angle.

◀ Effacement préalable de la page (CALL-3082), puis apparition de la page vierge avec HGR2.

Le tracé utilise la routine \$607F-\$60AF de DIA.LM (page 50).

AD = 24703

AE = 24709

Les données sont poquées aux adresses \$607F-\$6084

Quand BL = 1, les sous programmes sont déjà en place. On fait l'économie de cette variable (prévue pour un autre développement du programme) avec un GOTO 760 à la suite du NEXT I de la ligne 620.

Lignes traçantes ou lettres-codes.

Lignes de texte.
Lignes traçantes.

Les textes sont écrits normalement sur la page (invisible, puisque l'on est en mode graphique) Text de l'Apple, puis transférés en HGR2 par la routine DIA.LM, chaque caractère étant remplacé par son homologue de la fonte DIA.CARAC.

Dégagement carré pour la lettre-code.

```

660 REM *****
670 :
680 IF BL = 1 THEN 760
690 PRINT : PRINT D$"BLOAD DIA.LM,A"CA
700 PRINT D$"BLOAD DIA.CARAC,A$6100":BL = 1: RETURN
710 :
720 REM *****
730 REM *** AFFICHAGE DES TITRES ***
740 REM *****
750 :
760 HOME : FOR I = 1 TO 2
770 VTAB I: HTAB (42 - LEN (TI$(I))) / 2
780 PRINT TI$(I): NEXT : CALL CA: HOME : PRINT : PRINT
790 :
800 HCOLOR= 6: ON M GOTO 860,1010
810 :
820 REM *****
830 REM *** LIGNES TRACANTES ***
840 REM *****
850 :
860 VTAB 5:YC = 20
870 FOR I = 1 TO N:K = 200
880 FOR J = 1 TO N
890 IF Y2(J) > = K THEN 920
900 IF Y2(J) = 0 THEN 920
910 K = Y2(J):L = J
920 NEXT J: PRINT
930 Y2(L) = 0: PRINT TE$(L):YC = YC + 16
940 HPLOT 3 + LEN (TE$(L)) * 7,YC TO X2(L),K
950 NEXT I: CALL 24576: GOTO 1240
960 :
970 REM *****
980 REM *** LETTRES-CODES ***
990 REM *****
1000 :
1010 J = ASC ("A") - 1: PRINT
1020 FOR I = 1 TO N: PRINT
1030 PRINT CHR$( J + I) = "TE$(I)
1040 CV = PEEK (37) + 1
1050 VTAB Y2(I) / 8 + 1: HTAB X2(I) / 7 + 1
1060 Y2(I) = INT (Y2(I) / 8) * 8 + 2
1070 X2(I) = INT (X2(I) / 7) * 7 + 3
1080 :
1090 REM *****
1100 REM *** EMPLACEMENT DU CODE ***
1110 REM *****
1120 :
1130 HCOLOR= 8: IF C(I) = 5 THEN HCOLOR= 4
1140 FOR K = 1 TO 16
1150 HPLOT X2(I) - 8,Y2(I) - 8 + K TO X2(I) + 8,Y2(I) -
8 + K
1160 NEXT K
1170 PRINT CHR$( J + I): HTAB I: VTAB CV

```

(Suite page 54)

```

1180 NEXT I: CALL CA
1190 :
1200 REM *****
1210 REM ****      EDITION      ***
1220 REM *****
1230 :
1240 GH = 1: X = PEEK ( - 16384): IF X < 128 THEN 1240
1250 POKE - 16368,0
1260 IF X = 137 THEN PR# 1: PRINT CHR$(9)"G2": PRINT
      : PR# 0
1270 PR# 0: GOTO 110
1280 :
1290 REM *****
1300 REM ***  MEMO FICHER PAGE2  ***
1310 REM *****
1320 :
1330 VTAB 22: HTAB 13: INPUT "TITRE ? ";FG$: IF FG$ = "
      " THEN 150
1340 IF FG$ = "?" THEN R$ =
      FG$: GOTO 160
1350 IF ASC (FG$) < 65 OR ASC (FG$) > 90 THEN 1330
1360 FG$ = FG$ + ".P2": VTAB 21: PRINT : PRINT D$"BSAVE"F
      G$",A16384,L8192": GOTO 150
1370 :
1380 REM *****
1390 REM ***  ECRITURE DE FICHER  ***
1400 REM *****
1410 :
1420 PRINT : VTAB 22: HTAB 9: INPUT "TITRE DU FICHER "
      ;F$: IF F$ = "" THEN 150
1430 IF F$ = "?" THEN R$ = F$ :
      GOTO 160
1440 IF ASC (F$) < 65 OR ASC (F$) > 90 THEN 1400
1450 VTAB 22: PRINT : VTAB 22: PRINT D$"OPEN"F$: PRINT D
      $"WRITE"F$: PRINT N: FOR I = 1 TO N: PRINT Z(I): PRI
      NT T$(I): PRINT C(I): NEXT : PRINT TI$(1): PRINT TI
      $(2): PRINT D$"CLOSE"F$: GOTO 150
1460 :
1470 REM *****
1480 REM ***  LECTURE D'UN FICHER  ***
1490 REM *****
1500 :
1510 PRINT : VTAB 22: HTAB 9: INPUT "FICHER A LIRE ? "
      ;F$: IF F$ = "" THEN 150
1520 IF RIGHT$(F$,1) = "H" THEN F = H
1530 IF F$ = "?" THEN R$ = F$ :
      GOTO 160
1540 IF ASC (F$) < 65 OR ASC (F$) > 90 THEN 1510
1550 ONERR GOTO 2040
1560 VTAB 22: PRINT D$"VERIFY"F$
1570 IF F < > H THEN 1590
1580 VTAB 22: PRINT : VTAB 22: PRINT D$"OPEN"F$: PRINT D
      $"READ"F$: INPUT N: FOR I = 1 TO N: INPUT Z(I): INPU

```

← Appel de la routine de transfert (\$6000, ou 24576).

← On a prévu un Contrôle I pour imprimer (quand la page graphique est affichée).

ATTENTION ! À PROPOS DE L'IMPRIMANTE, LISEZ LE TEXTE DE LA PAGE 56.

← Le ? permet d'obtenir le catalogue de la disquette.

← En mémorisant le diagramme, il est facile, ensuite de le récupérer, puis de l'éditer à partir d'un autre programme.

← Ce fichier ne pourra pas être lu par HISTOGRAMME (Tremplin Micro n°1), mais des adaptations restent possibles.

← Si c'est un fichier H (HISTO...), on le lit autrement.

← Ce fichier existe-t-il ?

← Lecture d'un fichier de données DIAGRAMME.

Lecture d'un fichier de données provenant du programme HISTOGRAMME.

Elimination des données supplémentaires (plus de 10).

Saisie simple, mais évitant les erreurs. Cette routine permet aussi de compléter, ou de modifier les données existantes. Il suffit de taper le numéro de ligne, puis de rentrer le texte, et la valeur.

```
T TE$(1): INPUT C(1): NEXT : INPUT TI$(1): INPUT TI$(2): PRINT D$"CLOSE"F$: GOTO 1600
1590 VTAB 22: PRINT : VTAB 22: PRINT D$"OPEN"F$: PRINT D$"READ"F$: INPUT N: FOR I = 1 TO N: INPUT Z(I): NEXT : INPUT TI$(1): INPUT TI$(2): PRINT D$"CLOSE"F$
1600 IF N < 11 THEN 1630
1610 PP = 1: FOR I = 2 TO N: IF Z(I) < Z(PP) THEN PP = I
1620 NEXT I: FOR I = PP TO N - 1: Z(I) = Z(I + 1): C(I) = C(I + 1): TE$(I) = TE$(I + 1): NEXT : N = N - 1: IF N > 10 THEN 1610
1630 IF TE$(1) < > "*" THEN 150
1640 VTAB 21: PRINT : CALL - 958: PRINT "COMPLETER LES DONNEES (O/N) ? "; GET R$: IF R$ < > "0" THEN 150
1650 :
1660 REM *****
1670 REM ***      DONNEES      ***
1680 REM *****
1690 :
1700 HOME
1710 FOR I = 1 TO N: V$ = " ": IF I > 9 THEN V$ = ""
1720 VTAB I + 3: PRINT V$: I
1730 PRINT TAB( 4): TE$(I): TAB( 26 - LEN ( STR$( Z(I) ))): Z(I): NEXT
1740 VTAB 10: HTAB 1: INPUT "LIGNE A CORRIGER (0 POUR FIN) "; LI$: IF LI$ = "0" THEN 110
1750 LI = VAL (LI$): IF LI < 1 OR LI > 12 THEN 1740
1760 L$ = TE$(LI): IF L$ = "" THEN L$ = "....."
1770 VTAB 20: HTAB 1: PRINT L$: HTAB 1: INPUT "": L$: IF L$ = "" THEN 1790
1780 TE$(LI) = LEFT$( L$, 12)
1790 VTAB 20: HTAB 1: CALL - 958: PRINT TE$(LI): TAB( 15): Z(LI): HTAB 15: INPUT "": L$: IF L$ = "" THEN 1860
1800 FOR I = 1 TO LEN (L$): IF ASC ( MID$( L$, I, 1)) < 48 OR ASC ( MID$( L$, I, 1)) > 57 THEN 1790
1810 NEXT I
1820 Z(LI) = VAL (L$)
1830 IF NS < 10 THEN 1850
1840 VTAB 21: PRINT "LE DIAGRAMME N'ACCEPERA QUE LES DIX MEILLEURES DONNEES"
1850 IF N < 12 AND LI > N THEN N = N + 1
1860 GOTO 1710
1870 :
1880 REM *****
1890 REM ***      LECTURE DES TITRES      ***
1900 REM *****
1910 :
1920 HOME : I = 0
1930 I = I + 1: T$ = TI$(I): IF T$ = "" THEN T$ = "....."
1940 PRINT : VTAB 5 * 2: PRINT T$: HTAB 1: INPUT "": T$: IF T$ = "" THEN 1960
1950 TI$(I) = LEFT$( T$, 40)
```

(Suite page 56)

```

1960 FOR J = 1 TO 2: VTAB 5 + I * 2: HTAB 1: CALL - 868
      : PRINT TI$(I): NEXT
1970 IF I < 2 THEN 1930
1980 GOTO 110
1990 :
2000 REM *****
2010 REM *** TRAITEMENT D'ERREURS ***
2020 REM *****
2030 :
2040 CALL 24693:ER = PEEK (222)
2050 IF ER = 6 THEN PRINT D$"OPEN"F$: PRINT D$"CLOSE"F$
      : PRINT D$"DELETE"F$: VTAB 21: PRINT : PRINT "FICHIE
      R INEXISTANT": FOR I = 1 TO 1500: NEXT : POKE 216,0:
      GOTO 1510
2060 L = PEEK (218) + PEEK (219) * 256
2070 VTAB 22: HTAB 1: CALL - 950: PRINT "ERREUR NUMERO
      ";; INVERSE : PRINT ER;; NORMAL : PRINT " LIGNE ";;
      INVERSE : PRINT L: NORMAL
2080 STOP
2090 DATA LIRE UN FICHIER,SAISIR DES DONNEES,MODIFIER L
      ES TITRES,TRACER LE GRAPHIQUE,CHANGER DE MODE,MEMORI
      SER LE GRAPHIQUE,MEMORISER LES DONNEES,IMPRIMER LE G
      RAPHIQUE,MODIFIER LA ROTATION,FIN DE TRAVAIL
2100 TEXT : HOME : END
2110 POKE 51,0: GOTO 110

```

Les algorithmes de notre diagramme doivent beaucoup à un programme de N.B. Mc Burney II (Créative Computing, 1980).

← En ProDOS, si une commande VERIFY n'aboutit pas, on retrouve le titre du fichier au catalogue... d'où la nécessité de l'effacer.

← Lignes du menu.

← Un GOTO 2110 permet de récupérer le DOS et les données.

IMPRIMANTES ET GRAPHISME

QUAND on possède une interface autorisant la copie des pages graphiques, il est facile d'utiliser des programmes tels que HISTOGRAMME (n°1) ou DIAGRAMME. En général, l'instruction-clé tient en une ou deux lettres.

Il était moins difficile, avec l'ancienne imprimante APPLE DMP (carte parallèle) d'utiliser, dans un programme Basic, la routine en langage machine qui était fournie par Apple.

Où les choses se compliquent, c'est avec l'Apple IIc et son imprimante *Image Writer*, une bonne machine, rapide et performante, mais fonctionnant avec une carte série. Elle dispose même de sa mémoire-tampon, très suffisante pour de la copie d'écran en mode TEXT et pour le listage.

En mode graphique, c'est une autre histoire, notamment avec *ProDOS* (encore mal connu, à cause du manque de documentation en français).

Un conseil : mémorisez chaque page graphique par un *BSAVE PAGE, A\$2000, L8192* ou *A\$4000, L8192*, suivant le cas, puis servez-vous de l'utilitaire Apple (DISQUETTE TOOL-KIT) qui accompagnait votre imprimante.

Le programmeur amateur qui, le premier, nous enverra une routine en assembleur, fonctionnant facilement sous DOS 3.3 et sous ProDOS, avec *Image Writer* ou *DMP*, aura bien mérité de tous les Apple-mordus de France et de Navarre ! NESTOR.



DIA.CARAC

Si vous possédez l'excellent utilitaire APPLISOFT TOOL KIT, rien ne vous oblige à taper DIA.CARAC. Les fontes de TOOL KIT sont parfaitement compatibles avec DIAGRAMME, mais leurs minuscules ne sont pas accentuées. Autre fonte utilisable : FAIRE.F1 de *Tremplin Micro* n°1. Lorsque vous aurez terminé la (longue) saisie de ces 768 octets, sauvez votre fonte par un :

FFONTE CARAC (CLAVIER FRANCAIS)

```

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

BSAVE DIA.CARAC, A\$6100, L\$400

6100- 00 00 00 00 00 00 00 00	6200- 8C 90 BE 00 BE B7 BE 00	6300- 10 10 00 10 00 00 00 00
6100- 30 30 30 30 00 30 00 00	6200- 3E 33 33 3F 33 33 33 00	6300- 00 00 3E 30 3E 33 3E 00
6110- 36 36 24 12 00 00 00 00	6210- 1F 33 33 1F 33 33 3F 00	6310- 03 03 1F 33 33 33 1F 00
6110- 98 2C 0C 1F 06 65 BA 00	6210- 1E 33 03 03 03 33 3E 00	6310- 00 00 1E 33 03 33 1E 00
6120- 00 1E 00 1E 20 1E 00 00	6220- 0F 10 33 33 33 33 3F 00	6320- 30 30 3E 33 33 33 3E 00
6120- 42 27 12 00 24 72 21 00	6220- 1F 33 03 1F 03 33 1F 00	6320- 00 00 1E 33 3F 03 1E 00
6130- 0C 16 02 04 2A 12 2C 00	6230- 1F 33 03 1F 03 03 03 00	6330- 1C 36 06 1F 06 06 06 00
6130- 10 10 10 00 00 00 00 00	6230- 1F 33 03 30 33 33 1F 00	6330- 00 00 3E 33 33 3E 30 1E
6140- 30 1C 0E 0E 0E 1C 30 00	6240- 33 33 33 3F 33 33 33 00	6340- 03 03 1F 33 33 33 33 00
6140- 0E 1C 30 30 30 1C 0E 00	6240- 3F 0C 0C 0C 0C 0C 3F 00	6340- 0C 00 0C 0C 0C 0C 0C 00
6150- 36 1C 77 1C 36 00 00 00	6250- 30 30 30 30 33 33 3E 00	6350- 30 00 30 30 30 30 33 1E
6150- 00 10 10 7E 7E 10 10 00	6250- 33 10 0F 07 0F 10 33 00	6350- 03 03 33 10 0F 10 33 00
6160- 00 00 00 00 10 10 10 00	6260- 03 03 03 03 03 03 3F 00	6360- 0C 0C 0C 0C 0C 0C 0C 00
6160- 00 00 00 7E 7E 00 00 00	6260- 33 3F 33 33 33 33 33 00	6360- 00 00 33 3F 20 23 23 00
6170- 00 00 00 00 10 10 00 00	6270- 33 33 37 30 33 33 33 00	6370- 00 00 1F 33 33 33 33 00
6170- 60 70 30 1C 0E 06 00 00	6270- 1E 33 33 33 33 33 1E 00	6370- 00 00 1E 33 33 33 1E 00
6180- 1E 33 30 3F 37 33 1E 00	6280- 1F 33 33 1F 03 03 03 00	6380- 00 00 1F 33 33 1F 03 03
6180- 0C 0E 0F 0C 0C 0C 1E 00	6280- 1E 33 33 37 30 33 2E 00	6380- 00 00 3E 33 33 3E 30 30
6190- 1E 33 30 10 06 03 3F 00	6290- 1F 33 33 1F 0F 10 33 00	6390- 00 00 1F 33 03 03 03 00
6190- 1E 33 30 1C 30 33 1E 00	6290- 1E 33 03 1E 30 33 1E 00	6390- 00 00 3E 03 1E 30 1F 00
61A0- 30 3C 36 33 3F 30 30 00	62A0- 3F 0C 0C 0C 0C 0C 0C 00	63A0- 06 06 1F 06 06 36 1C 00
61A0- 3F 03 03 1F 30 33 1E 00	62A0- 33 33 33 33 33 33 1E 00	63A0- 00 00 33 33 33 33 3E 00
61B0- 1C 06 03 1F 33 33 1E 00	62B0- 33 33 33 33 33 1E 0C 00	63B0- 00 00 33 33 33 1E 0C 00
61B0- 3F 33 10 0C 0C 0C 0C 00	62B0- 33 33 33 33 33 3F 33 00	63B0- 00 00 63 63 60 60 3E 00
61C0- 1E 33 33 1E 33 33 1E 00	62C0- 33 33 1E 0C 1E 33 33 00	63C0- 00 00 33 1E 0C 1E 33 00
61C0- 1E 33 33 3E 30 10 0C 00	62C0- 33 33 33 1E 0C 0C 0C 00	63C0- 00 00 33 33 33 3F 30 1E
61D0- 00 10 10 00 10 10 00 00	62D0- 3F 30 10 0C 06 03 3F 00	63D0- 00 00 3F 10 0C 06 3F 00
61D0- 00 10 10 00 10 10 10 00	62D0- 1E 1E 06 06 06 1E 1E 00	63D0- 90 0C 9E 03 0F 03 9E 00
61E0- 10 10 0C 0E 0C 10 10 00	62E0- 00 00 1E 33 03 33 1E 0C	63E0- 06 0C 33 33 33 33 3E 00
61E0- 00 00 7E 00 7E 00 00 00	62E0- 70 70 60 60 60 70 70 00	63E0- 06 0C 1E 33 3F 03 1E 00
61F0- 04 0C 10 30 10 0C 04 00	62F0- 06 06 06 06 06 06 06 06	63F0- 60 60 60 60 60 60 60 60
61F0- 3C 66 62 30 00 10 10 00	62F0- 00 00 00 00 00 00 7F 00	63F0- 00 00 00 00 00 00 00 00

COMPATIBILITÉ avec ProDOS

Nous vous recommandons le système ProDOS, beaucoup plus rapide que le DOS 3.3, et plus ouvert sur l'avenir. Méfiez-vous toutefois des premières versions (d'origine américaine), copiées (ou piratées) depuis la naissance de ProDOS, et pas forcément exemptes de bogues. Offrez-vous plutôt la version la plus récente... avec garantie d'origine. Elle vous coûtera quelques centaines de francs, mais vous évitera bien des déboires. La plupart des programmes du n°2 de *Tremplin Micro* fonctionnent sous ProDOS, après des modifications insignifiantes : noms des fichiers, par exemple. Si vous travaillez avec (ou sous) ProDOS, avec un programme en 40 colonnes faisant appel au CATALOG, n'oubliez pas de transformer la commande CATALOG en CAT.

```

*****
*
* TREMLIN MICRO *
* GUY-HACHETTE 85 *
* NUMERO 2 *
*
*****

```

```

*A 004 MENU
*A 017 MENU.RUN
*B 002 MENU,
*B 002 MENUASC
*B 002 MENUCOMP.C
T 005 JEU
T 005 JEU.
*A 009 LIST80.LMB
*B 002 LIST80.LMC
*A 012 FICHER.DIRECT
*A 020 BINDECHEX
*B 008 NBR2-23
*B 002 DMP40
*A 002 LISTAGE.FAITLIST
*A 006 LISTAGE.DEMO
*A 018 LISTAGE
T 002 LIST
*A 011 HOMME.LIBRE
*A 011 HOMME.LIBRE.2C
*B 002 PAIRIMP
*A 006 PAIRIMPAIR
*B 002 PAIR.LM+
*A 008 PAIRIMPAIR+
*A 002 CARACTERES.SOURIS
*A 005 DISSEQUER
*A 004 ONERR
*A 007 TRUCS
*A 004 LISTAGE.CONTROLE
*A 026 DIAGRAMME
*B 002 DIA.LM
*B 005 DIA.CARAC
*T 002 DIA.D1
*B 002 TRANS80.LM
*A 008 TRANS80.BASIC
*A 005 TRANS80.LM.DEMO

```

Pour obtenir la disquette contenant les programmes de *TREMLIN MICRO n°2* (DOS 3.3, APPLE IIe et IIc), utilisez le bulletin de commande de la page 62. Une version ProDOS est disponible, mais nécessite que l'on dispose du Basic-System ProDOS.

Vos menus à la carte (page 5)

MENU

BSAVE MENU,
A\$2000, L\$54
ou
BSAVE MEN,
A8192, L84

```

2000- A2 00 A9 20 20 F0 FB E8
2008- E0 20 D0 F6 60 A9 20 20
2010- F0 FB A2 27 86 24 A9 20
2018- 20 F0 FB A5 25 C9 16 D0
2020- EC 60 20 58 FC 20 00 20
2028- 20 0D 20 20 00 20 A9 08
2030- 05 25 20 54 20 A2 FF E8
2038- BD 5C 20 C9 AA D0 0A E6
2040- 25 E6 25 20 54 20 4C 37
2048- 20 C9 A3 F0 06 20 F0 FB
2050- 4C 37 20 60

```

MENUASCI

BSAVE MENUASCI,
A\$2054, L\$7E
ou
A8276, L126

```

2054- A9 0A 05 24
2058- 20 22 FC 60 A0 A0 A0 A0
2060- 0D 05 0E 15 20 04 15 20
2068- 0A 0F 15 12 AA 31 A0 D0
2070- D2 CF C7 D2 C1 CD CD C5
2078- A0 CE D5 CD C5 D2 CF A0
2080- B1 AA 32 A0 D0 D2 CF C7
2088- D2 C1 CD CD C5 A0 CE D5
2090- CD C5 D2 CF A0 B2 AA 33
2098- A0 D0 D2 CF C7 D2 C1 CD
20A0- CD C5 A0 CE D5 CD C5 D2
20A8- CF A0 B3 AA 34 A0 D0 D2
20B0- CF C7 D2 C1 CD CD C5 A0
20B8- CE D5 CD C5 D2 CF A0 B4
20C0- AA A0 A0 A0 A0 D2 C5 D0
20C8- CF CE D3 C5 A0 BA A0 20
20D0- AA A3

```

Vérifiez bien chaque ligne avant de passer à la ligne suivante. N'oubliez pas que vous devez d'abord taper un CALL-151 (lisez page 43).

CORRIGEZ VOTRE NUMÉRO 1 !

FICHE N°1 (la page zéro de l'Apple IIe)

L'adresse 214 (\$D6) n'est pas disponible... et si l'on tape un POKE 214,255, on s'expose à certains inconvénients. Lisez, sur ce sujet, la rubrique "Deux trucs pour votre Apple" de la page 40.

PAGE 27 (Transfert d'écrans en mémoire auxiliaire).

La ligne 10020, en bas de page, est erronée. Il faut lire POKE 66,0 et non POKE 66,35.

Apple II, II+ et compatibles Vous rencontrerez forcément des difficultés pour faire tourner certains programmes de *Tremplin Micro* sur les machines de la première et de la seconde génération, et plus encore sur des machines dites compatibles. C'est agaçant, mais plein d'enseignements. Vos recherches vous permettront de mieux comprendre le Basic en général, et l'Apple-soft en particulier. Ne vous laissez jamais décourager et faites profiter les autres de vos trucs et astuces !

NESTOR.

Vous avez écrit à TREMPLIN MICRO



** Dans le remplacement de DISK VOLUME, en mode direct, par mon identité, j'obtiens un message «ILLEGAL QUANTITE ERROR». Pourquoi ?*

TM Plusieurs réponses possibles, mais deux causes principales :

1. La variable A\$ (votre identité) ne comporte pas 11 signes, espaces compris (si votre nom est trop court, tapez x espaces).

2. Vous avez effectué plusieurs tentatives et omis de remettre la variable M à zéro.

Respectez exactement la procédure indiquée, en utilisant bien un DOS 3.3 d'origine, sans faire appel à des caractères illégaux (CTRL-, par exemple), ...et tout ira bien.

Si vous désirez supprimer le V de VOLUME, tapez $A\$ = A\$ + CHR\(32) , puis transformez le 46000 de la boucle en 45999.

** Je suis abonné à la revue POM'S et je crains que TREMPLIN MICRO n'aggrave encore ses difficultés de diffusion. Qu'en pensez-vous ?*

TM Sans POM'S (et aussi sans la presse américaine), les amateurs français de programmation en seraient encore aux balbutiements, notamment en assembleur ... et il n'y aurait pas de TREMPLIN MICRO.

POM'S est un magazine indispensable, que je recommande à tous ceux qui désirent réellement aller plus loin. Ce n'est pas un concurrent, mais un confrère auquel je tire mon chapeau. **GH.**

** J'ai été enthousiasmé par votre revue, et je l'ai fait acheter à mes amis. Bravo d'oser être didactique et de l'être bien.*

Pourrait-on avoir un jour la version commentée de FAIRE.LM ?

Très bien de faire de la publicité (gratuite) pour les assembleurs, mais il faudrait aussi mentionner les «débuggeurs», tellement efficaces pour mieux comprendre...

TM Souhaitons que plusieurs dizaines de milliers de lecteurs manifestent autant d'enthousiasme que vous.

Posséder un ordinateur personnel est une chose, mais en tirer la quintessence en est une autre.

La première partie de FAIRE.LM est une routine américaine permettant, comme vous avez pu le constater, d'écrire facilement en page graphique, et cela à partir de fontes que la plupart des programmeurs possèdent déjà (dans APPLESOFT TOOLKIT notamment). A quand un «débuggeur» français... comme PROCODE dans le domaine des assembleurs, par exemple ? Qu'il arrive, et il aura sa place dans TREMPLIN MICRO, c'est promis.

** J'ai tapé le programme SPECIAL BONJOUR, et il ne fonctionne pas sur mon APPLE IIc, (sous ProDOS). Je ne parviens pas davantage à écrire mon nom sur la disquette que j'ai initialisée.*

TM Cela ne nous surprend pas. Les deux programmes cités font appel à des routines et à des adresses qui ne concernent que le DOS 3.3. Par contre, la plupart des autres programmes sont facilement adaptables.

Il faut d'abord éliminer le $D\$ = CHR\$(13) + CHR\$(4)$ du DOS 3.3 et le remplacer par $D\$ = CHR\(4) . Un PRINT tiendra éventuellement lieu de $CHR\$(13)$. (ou un PRINT $CHR\$(13)$, mais isolé).

Ensuite, revoir les noms des fichiers, et noter que l'imprimante série se révèle parfois chatouilleuse quand ProDOS est là.

Dans les routines en langage machine, ne pas utiliser des adresses comme \$3EA et essayer de remplacer les six octets par EA EA EA.

Nous avons rassemblé, sur une disquette, les 4/5e des programmes de TREMPLIN MICRO n°1, et ils sont maintenant utilisables sous ProDOS... à condition de posséder ce système d'exploitation, car il ne s'agit que d'une disquette esclave.

SUITE DU COURRIER (PAGE 60)

VOUS AVEZ ÉCRIT À TREMLIN MICRO (suite)

** J'étais abonné à POM'S, mais j'ai abandonné pour cause de comprenette difficile. Avec vous, je m'attendais à mieux, peut-être à cause de mes expériences avec vos revues d'initiation aux mots croisés (c'est une vieille histoire !). Hélas ! pour moi, ici ou ailleurs, l'assembleur, c'est de l'hébreu... Pourtant, je vais poursuivre l'expérience pendant quelques numéros... à cause des mots croisés.*

TM Dans des revues comme POM'S ou comme TREMLIN MICRO, les programmes ne s'adressent pas tous à la totalité des lectrices et lecteurs. Si vous reprenez votre collection de POM'S dans un an, vous verrez les choses autrement... et vous commanderez les numéros qui vous manquent (non, je ne suis pas associé à eux, ...et ils ne me connaissent pas, mais je les aime bien).

On ne peut comprendre le langage machine qu'en le pratiquant, pas à pas. Je suis sûr que vous allez vous piquer au jeu... et je ne désespère pas de vous découvrir un jour parmi nos auteurs de programmes !

** Comment sauver, puis recharger un programme en langage machine ?*

TM Pour sauver un programme, aussi court soit-il, encore faut-il l'avoir écrit. Tapez CALL-151, puis enfoncez la touche RETURN. L'astérisque va apparaître. Votre moniteur attend votre programme. Recopiez alors, sans une seule erreur, les instructions qui vous intéressent.

Un détail : si l'on vous propose 0300-20 58 FC 60, ligne qui se contente d'effacer l'écran par un HOME, vous devrez taper :

* 300 : 20 58 FC 60, puis enfoncez la touche RETURN.

Si vous aviez mis un tiret à la place des deux points, ça n'aurait pas marché. Comment sauver cette mini-routine ?

Elle commence à l'adresse \$300 HEXA (qui correspond à 768 en DECIMAL) et elle comporte 4 octets (4 fois 2 caractères). Après le RETURN, vous avez vu apparaître un autre astérisque. Armez-vous de courage et tapez :

* BSAVE HOME, A\$300, L4 ... RETURN.

Ça ronronne côté disquette. Ça y est : HOME (ou un autre titre) est sauvé.

UN RESET (ou CTRL-C, RETURN) vous ramène au BASIC. Notez que vous auriez pu faire cela tout de suite, et sauver votre pro-

gramme après coup. Eteignez l'Apple. Rallumez. Faites un LIST pour afficher votre menu, puis tapez :

BRUN HOME ... RETURN

Ça ronronne de nouveau et, ô miracle !, l'écran se vide. Maintenant, pour vider l'écran, il vous suffira de taper CALL 768 (y compris dans le cadre d'un programme en BASIC).

Bien sûr, ce n'est qu'un exemple bête et méchant et je vous vois venir : un HOME ferait plus rapidement la même chose. D'accord, mais cent autres instructions pourraient suivre celle-ci ... sans frais supplémentaires !

** 1. Par quoi remplacer POMME OUVERTE et POMME FERMEE sur un APPLE II ?*

** 2. PEEK (105)+PEEK (106) * 256 ... me répond par SYNTAX ERROR. Pourquoi ?*

TM 1. Par vos manettes de jeu.
2. Tapez PRINT PEEK (105)+PEEK (106) * 256 (qui permet de savoir où se termine un programme) et vous aurez la réponse sans SYNTAX ERROR.

** Seriez-vous prêts à faire, dans vos colonnes, une optimisation par routines Assembleurs ou astuces Apple IIe, de petits sous-programmes utilitaires, écrits uniquement en Applesoft ?*

TM Assurément. Seriez-vous l'auteur de tels sous-programmes ? Tremplin Micro attend votre disquette !

** Certains programmes ne fonctionnent pas sur l'Apple II+, malgré la présence d'une carte 80 colonnes.*

TM Vous avez effectivement pu rencontrer quelques difficultés avec votre Apple II+, mais il doit être possible d'adapter tous les programmes à cette machine. A revoir : Pomme ouverte et Pomme fermée, ainsi que d'éventuels déplacements avec les flèches des Apple IIe et IIc. Quant à la carte 80 colonnes, il est indispensable de respecter les indications fournies par sa documentation (souvent, il faut notamment remplacer l'instruction HOME par PRINT CHR\$(12), et supprimer le mode INVERSE).

Lorsque vous nous écrivez, veuillez nous préciser si nous pouvons ou non citer votre identité, et le nom de votre ville. Merci.

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 cases de votre choix.

Depuis combien de temps possédez-vous un ordinateur personnel ?
Concevez-vous des programmes ? OUI NON
Langage BASIC PASCAL LOGO AUTRE Précisez :
Programmez-vous en LANGAGE MACHINE ? OUI NON
Utilisez-vous un assembleur ? OUI NON si OUI lequel ?

INTENTIONS D'ACHAT

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

RUBRIQUES PRÉFÉRÉES

Indiquez-moi le numéro des pages qui vous auront semblé les plus intéressantes dans ce numéro 2 :

Quelles sont VOS SUGGESTIONS :

Merci de bien vouloir répondre à ces questions :

1. Etes-vous pour ou contre les petites annonces dans TREMPLIN MICRO (Ventes et Achats) ?

POUR	<input type="checkbox"/>	CONTRE	<input type="checkbox"/>
------	--------------------------	--------	--------------------------
2. A votre avis, devons-nous ou non poursuivre la diffusion de disquettes DEMO, comme nous le faisons, pour GRIBOUILLE, dans le présent numéro ?

OUI	<input type="checkbox"/>	NON	<input type="checkbox"/>
-----	--------------------------	-----	--------------------------
3. Transformerez-vous votre Apple IIe avec le Kit prévu pour le rendre compatible IIc ?

OUI	<input type="checkbox"/>	NON	<input type="checkbox"/>
-----	--------------------------	-----	--------------------------
4. Etes-vous intéressé par des essais des nouveaux utilitaires tournant sur Apple ?

OUI	<input type="checkbox"/>	NON	<input type="checkbox"/>
-----	--------------------------	-----	--------------------------

Bulletin de commande et d'abonnement

Nom Prénom

Adresse complète

Code postal Ville

N'OUBLIEZ PAS DE COCHER LA VERSION DE LA DISQUETTE (DOS ou ProDOS)

1	Abonnement à 5 numéros (un an) : FRANCE	150 F
1	DISQUETTE de TREMPLIN MICRO N°1 (Pour APPLE IIe et IIc) DOS 3.3 <input type="checkbox"/> ProDOS <input type="checkbox"/>	105 F
1	DISQUETTE de TREMPLIN MICRO N°2 (Pour APPLE IIe et IIc) DOS 3.3 <input type="checkbox"/> ProDOS <input type="checkbox"/>	105 F
1	NUMÉRO 1 DE TREMPLIN MICRO	30 F (NET)
1	NUMÉRO 2 DE TREMPLIN MICRO	30 F (NET)
1	DISQUETTE DEMO DU TRAITEMENT DE TEXTE GRIBOUILLE	25 F (NET)

Rayer la ou les mentions inutiles

Frais d'envoi (gratuit pour les abonnés) + 10 F

Total à payer F

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

Mode de règlement choisi : Chèque Mandat-lettre Carte Bleue

N° de votre Carte Bleue

Date d'expiration Signature (obligatoire)

Montant à débiter F

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

