

tremplin micro

Auto :
le prix du kilomètre

Imprimante :
caractères géants
sur ImageWriter I
et ImageWriter II.

Graphisme :
percez ses mystères !

Dessin :
peinture automatique



A combien vous revient votre automobile ?
Comment colorer une figure ?
Dessiner des pyramides.
Jouer avec les noms des villes de France.

N°7 - Bimestriel - Deuxième année
3 Mars - 2 Mai 1986
254 FB - 11 FS - 33 F

Au sommaire du n° 8 de TREMPLIN MICRO

- LA BOÎTE AUX IDÉES (petite base de données)
 - Mastermind (version pour Apple IIe et IIc)
 - Translist... ou comment créer un fichier source à partir d'une simple routine en langage machine ?
 - Redimensionner vos variables
 - Variables dimensionnées (comprendre les...)
 - Ecran-damier en 40 et 80 colonnes (initiation)
 - AND.DÉMO (comment utiliser cette instruction)
 - Miroir (toutes les lignes d'écran inversées en un clin d'œil)
 - Tête en bas (encore de l'initiation !).
 - HISTO.TEXT (graphique en mode TEXT, avec les caractères souris... mais sur Apple IIc ou IIe + 65C02)
 - CATALOG PRODOS inédit
 - MIXAGE HGR1 et HGR2
 - ORGANISATION D'UN TOURNOI
- Etc., etc.

Parution le samedi 3 mai

(Disquette disponible à compter du 15 avril)

VISU-PUCE (n°6)

Un décalage dans la photocomposition de VISU-PUCE nous a obligés à ajouter deux octets nuls (NOP) en 91FF (ligne 307). Ce décalage ne modifie en rien le fonctionnement du programme, mais... VISU-CH

(Basic) aurait dû être rectifié de la même manière... ce qui n'a pas été le cas.

Deux solutions :

1° Elimination des NOP (EA EA), facile quand on utilise un assembleur... et VISU.CH est bon.

2° Ne rien changer, mais modifier les lignes 160 et 180 de VISU.CH, de la manière suivante :
692 au lieu de 690 — 665 au lieu de 663.

tremplin micro

7

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

BIMESTRIEL

Le numéro : 33 F
Abonnement d'un an 180 F
(6 numéros)

EDITIONS JIBENA

Direction-Rédaction :

Editions JIBENA

Guy-HACHETTE

La Petite Motte — Senillé
86100 CHÂTELLERAULT.

Téléphone :
49-93-66-66

PUBLICITÉ :
Joelle (même numéro)

Commission paritaire :
Demande en cours.

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

TREMPIN MICRO — Bimestriel —
C'est une publication des Editions JIBENA, 4, rue de la Cour-des-Noues, 75020 PARIS — S.A. au capital de 2600000 F — Imprimé par CITÉ-PRESS/PARIS — Dépôt légal à la date de parution — Inscription à la Commission Paritaire des Publications et Agences de Presse : en cours — Directeur de la Publication : Guy-Clément COGNÉ — Diffusion N.M.P.P.

Je persiste et signe...



EXISTENCE même de *TREMPIN MICRO* nous en administre une preuve irréfutable : la programmation passionne (au moins) un utilisateur d'Apple II... sur dix. Qui plus est, cet utilisateur est devenu un lecteur régulier de la revue. Je suis tenté d'ajouter... et un ami.

Je l'avais écrit sous le titre de notre numéro 1, il y a un peu plus d'un an : *Programmer devient un jeu*. C'était — et cela reste vrai aujourd'hui — un objectif pour *Tremplin Micro* : fournir aux amateurs de programmation le plus grand nombre possible de routines capables de les aider à "posséder vraiment leur Apple".

Depuis, *T.M.* a publié bon nombre de programmes, tantôt élémentaires, tantôt d'un niveau supérieur. Le Basic n'a pas été négligé, mais de fréquentes incursions dans le langage machine, celui que la machine comprend le mieux... et le plus rapidement, ont permis de débroussailler le terrain... encore à découvrir d'une programmation qui séduit déjà de nombreuses lectrices et de non moins nombreux lecteurs.

Spécialiste des jeux d'esprit (et créateur d'une soixantaine de revues), je maintiens que la programmation est un jeu. Mieux : une aventure dans laquelle on ne saurait s'engager sans avoir le désir intense de progresser... pour gagner.

Amusez-vous bien !

GUY-HACHETTE.

...la programmation est un jeu !

Sommaire

7

INITIATION

- ALLUMÉ ou ÉTEINT ? 19
- Sondage sur 8 bits 43

JEUX

- MÉMO-VISU
(un jeu visuel pour tester votre sens de la perception et apprendre à utiliser une petite forme) 17
- Villes de France
un jeu instructif de M. DEVAUX ... 37

DES TRUCS

- Adresses utiles 8
- Inversion de fonte 14
- Le point sur RND 41

AMUSETTE

LETTRES EN PLUIE

un programme pour les nostalgiques de l'effacement progressif 9

GRAPH

Une nouvelle version pour chiffres en dents de scie 23

COMMENT COLORER UNE FIGURE ?

graphisme en haute résolution 21

LE COÛT RÉEL
DE VOTRE
AUTOMOBILE



par Clément RENARD 47

TREMLIN MICRO Le numéro 8 paraîtra le 3 mai.

GRAPHISME

- IPO de Dominique LUGAT
(une nouvelle fonction graphique pour inverser la couleur, effacer le point, créer de petites animations, etc.) 3
- Construisez des pyramides 16
- ALLUMÉ ou ÉTEINT ?
Comment connaître l'état d'un point de l'écran 19
- Colorer une figure 21
- CADRE-DÉMO
en réponse à M. ANSELME 40

ORIGINAL !

- CARACTÈRES GÉANTS sur votre imprimante Apple (DMP et IMAGEWRITER I et II) 12
- CARAC.INV/DEMO 15

LOGICIEL

- CX HÉRAKLÈS 46

UTILITAIRES

- DATA-GÉNÉRATEUR
pour créer des lignes DATA au gré de sa fantaisie 30
- Où CATALOG devient CAT 33
- Remplacer une chaîne de caractères .. 35

LISEZ AUSSI :

- Bibliothèque informatique 11-29
- Logiciels de jeux sur Apple II :
MANDRAGORE, HOLD-UP, COLORIAGES, PUZZLES, ARCANÉ 42
- Utilisation de GRAF.Ø 52
- Le courrier des lecteurs 53 à 60
 - MULTIPLIER PAR 16 55
 - PRNTAX ET PRNTYX 56
 - INPUT, WRITE ET READ EN MODE DIRECT 56
 - ADRESSE D'UNE VARIABLE 56
 - POUR UN CADRE HGR 57
 - PRÉVENTION ROUTIÈRE 57
 - FORMATAGE IMAGEWRITER 58
 - GET.TIME 59
 - ON SGN(X) GOTO... 59
- Votre fiche n°7 61-62
- Bulletin de commande et d'abonnement 64

IPO

UNE NOUVELLE FONCTION GRAPHIQUE

Comme **HPlot X, Y**, la fonction **& IPO X, Y** trace le point correspondant aux coordonnées X et Y, mais là s'arrête la comparaison.

En effet, **& IPO** est capable de lire les caractéristiques existant déjà à l'adresse X, Y (même vide, votre écran est constitué de points).

Le programme **IPO** est écrit en assembleur, rapidité oblige, et peut être divisé en deux grandes parties :

- **IPO** proprement dit (dont nous vous fournissons le programme source complet).
- **IPO TO**, transposition modifiée de **HLINE**, dont seuls les codes vous sont donnés.

Ces deux parties sont réunies dans un seul fichier : **IPO.C**

UTILISATIONS POSSIBLES : Au gré de votre imagination, bien sûr, mais en voici une liste non exhaustive : inverser la couleur, effacer le point, le faire clignoter, déplacer un pointeur commandé par la souris, créer de petites animations, des effets spéciaux, etc.

PROCÉDURE : **IPO.C** est appelé via Ampersand, de la manière habituelle (\$3F5 : 4C 10 60 ou encore POKE 1013,76 : POKE 1014,16 : POKE 1015,96)

SYNTAXE : **HPlot X, Y** devient **& IPO X, Y** **HPlot TO X,Y** devient **& IPO TO X, Y**

PROGRAMME DÉMO

Un programme de démonstration, en Basic, vous montrera, mieux que de longues explications, comment utiliser **& IPO** dans le cadre de vos réalisations graphiques. Notez que la routine en langage machine n'est pas relogeable et n'oubliez pas, si vous employez de nombreuses variables, de protéger la totalité de vos programmes, ainsi que les pages graphiques par un **LOMEM : 25074**.

Si vous avez des questions à poser, téléphonez à Dominique !



IPO DÉMO

```
100 REM DEMONSTRATION & IPO X.Y ET & I
    PO TO X.Y
110 TEXT : NORMAL : D$ = CHR$ (4):G$ =
    CHR$ (7): PRINT CHR$ (21): HOME :
    PRINT G$
120 T$ = " : " : INVERSE
    : HTAB 11: VTAB 10: PRINT T$: HTAB
    11: VTAB 11: PRINT " DEMONSTRATI
    ON " : HTAB 11: VTAB 12: PRINT "
    &IPO &IPO TO " : HTAB 11: VTAB 13
    : PRINT T$: NORMAL : GOSUB 320: GO
    SUB 860
130 HOME : PRINT D$"BRUN IPO.C": CALL
    768: HGR : HCOLOR= 3: HTAB 1: VTAB
    22: PRINT " ON TRACE DEUX RECTANG
    LES QUELCONQUES..."
140 FOR I = 1 TO 10: H PLOT 80,80 + I T
    O 150,80 + I: NEXT : FOR I = 1 TO
    10: H PLOT 70,100 + I TO 170,100 +
    I: NEXT
150 GOSUB 320: HTAB 1: VTAB 22: PRINT
    "PUIS DES DROITES PARALELLES AVEC
    'H PLOT'"
157 :
159 REM *****
160 REM * H PLOT NORMAL *
161 REM *****
162 :
170 FOR J = 1 TO 6: FOR I = 50 TO 150:
    H PLOT 4 + I,I + J * 2: NEXT : NEX
    T
180 GOSUB 320: HTAB 1: VTAB 22: INVERS
    E : PRINT " & IPO ENTRE
    EN JEU " : NORMAL : GOSUB
    320: CALL 834: HTAB 1: VTAB 22: P
    RINT "L'EFFET DIMENSIONNEL EST EVI
    DENT..."
188 :
189 REM *****
190 REM * & IPO *
191 REM *****
192 :
200 GOSUB 210: GOTO 220
210 FOR J = 1 TO 6: FOR I = 50 TO 150:
    & IPO I + 30,I + J * 2: NEXT : NEX
    T : RETURN
220 GOSUB 320: HTAB 1: VTAB 22: PRINT
    * LE MECANISME EST SIMPLE...
228 :
229 REM *****
230 REM * ANIME *
231 REM *****
232 :
240 GOSUB 250: GOTO 260
250 FOR J = 1 TO 6: FOR I = 50 TO 150:
    & IPO I + 60,I + J * 3: NEXT : NEX
    T : RETURN
260 GOSUB 320: HTAB 1: VTAB 22: PRINT
    " IL PERMET DES PETITES ANIMATION
    S "
270 FOR J = 1 TO 6: FOR I = 50 TO 150:
    & IPO I + 60,I: NEXT : NEXT
278 :
279 REM *****
280 REM * EFFACEMENT *
281 REM *****
282 :
290 GOSUB 320: PRINT : VTAB 22: PRINT
    " BON... IL EST TEMPS D'EFFAC
    ER! " : PRINT G$: GOSUB 210:
    GOSUB 250: HOME : PRINT G$G$G$: HT
    AB 1: VTAB 22: INVERSE : PRINT " E
    T MAINTENANT, PLACE A L'ART...GAST
    ON " : GOSUB 320: NORMAL
300 GOTO 340
308 :
309 REM *****
310 REM * SP.IPO 222 *
311 REM *****
312 :
320 FOR I = 1 TO 2000: NEXT : FOR I =
    1 TO 20: B = PEEK ( - 16336): NEXT
    : RETURN
328 :
329 REM *****
330 REM * ART.GASTON AVEC &IPO TO *
331 REM *****
332 :
340 HOME : OC = 1: CALL 768: HGR2 : CAL
    L 791: CALL 768
350 FOR X = 5 TO 250 STEP X
360 GOSUB 430
370 IF OC = 1 THEN H PLOT X,100 TO 100,
    Y
380 IF OC = 2 THEN H PLOT 104,Y: & IPO
    TO X,102
390 NEXT X
400 OC = OC + 1: GOSUB 700: IF OC = 3
    THEN 470
410 GOTO 350
418 :
419 REM *****
```

```

420 REM *   CALCUL DES COORDONNEES   *
421 REM *****
422 :
430 Y = 3.14 * X ^ 2
440 Y = Y / 1100
450 RETURN
458 :
459 REM *****
460 REM *   DESSIN ALEATOIRE   *
461 REM *****
462 :
470 AI = 30:BI = 160: GOSUB 600: GOSUB
    600
480 Y = 130:Z = 90
490 FOR I = 1 TO 100
500 GOSUB 550
510 HPLLOT Y,Z
520 & IPO TO A,B
530 NEXT
540 GOSUB 700: GOSUB 720: GOTO 580
550 A = INT ( RND (120) * 100) + 160
560 B = INT ( RND (120) * 100) + 20
570 RETURN
580 HOME : CALL 768: HGR : HTAB 10: VT
    AB 22: PRINT "AU REVOIR...": FOR J
    = 1 TO 1000: GOSUB 590: POKE 855,
    AI: POKE 856,BI: CALL 834: NEXT :
    ONERR GOTO 730
590 GOSUB 650
600 FOR I = AI TO BI
610 HPLLOT AI + 50,I
620 & IPO TO BI + 50,I
630 NEXT
640 RETURN
650 AI = INT (( RND (1) * 1000) / 7)
660 BI = INT (( RND (1) * 1000) / 4)
668 :
670 IF BI < AI THEN 650
680 IF BI > 170 THEN 650
690 RETURN
700 FOR P = 1 TO 2000: NEXT : FOR P =
    1 TO 10:B = PEEK ( - 16336): NEXT
    : RETURN
708 :
709 REM *****
710 REM *   FEU D'ARTIFICES   *
711 REM *****
712 :
720 DIM A(30),B(30): CALL 768: HGR2 :
    CALL 768: HCOLOR= 3: FOR I = 1 TO
    8: FOR B = 1 TO 3:U = PEEK ( - 163
    36): NEXT : GOSUB 780: FOR J = 179
    TO Y STEP - 2: HPLLOT X,J: IF I <

```

```

4 THEN FOR P = 1 TO 20: NEXT
730 NEXT : GOSUB 740: NEXT : FOR P = 1
    TO 800: NEXT : CALL 768: HGR : RE
    TURN
740 P = 0: FOR P = 1 TO 8:B = PEEK ( -
    16336): NEXT : FOR K = 1 TO 20: G
    OSUB 820: HPLLOT X,Y: & IPO TO A(K)
    ,B(K): FOR U = 1 TO 2:B = PEEK ( -
    16336): NEXT : FOR U = 1 TO INT (
    RND (1) * 20): NEXT : NEXT
750 IF I < 3 THEN HCOLOR= 0: FOR J = 1
    79 TO Y STEP - 2: HPLLOT X,J: NEXT
    : FOR K = 1 TO 20: HPLLOT X,Y TO A(
    K),B(K): NEXT : HCOLOR= 3
760 RETURN
768 :
769 REM *****
770 REM *   RND (ECRAN)   *
771 REM *****
772 :
780 X = INT ( RND (1) * 1000) / 4:Y =
    INT ( RND (1) * 1000) / 5: IF X <
    30 OR X > 250 THEN 780
790 IF Y < 10 OR Y > 80 THEN 780
800 RETURN
808 :
809 REM *****
810 REM *   RND (ECLATE)   *
811 REM *****
812 :
820 A(K) = INT ((( RND (1) * 1000) / 8
    ) + X - 30):B(K) = INT ((( RND (1)
    * 1000) / 14) + Y - 30): IF A(K)
    < 1 OR A(K) > 279 THEN 820
830 IF B(K) < 1 OR B(K) > 150 THEN 820
840 RETURN
848 :
849 REM *****
850 REM *   ROUTINE CLEARHGR,SON   *
851 REM *****
852 :
860 FOR ADR = 768 TO 856: READ CODE: P
    OKE ADR,CODE: NEXT : RETURN
870 DATA 169,32,133,7,169,0,133,6,162,
    64,160,0,145,6,136,208,251,230,7,2
    02,208,244,96,169,32,133,7,169,0,1
    33,6,169,255,162,64,160,0,145,6,72
    ,138
880 DATA 72,32,57,3,104,170,104,136,20
    8,242,230,7,202,208,235,96,162,21,
    173,48,192,202,208,250,96,173,48,1
    92,136,208,5,206,88,3,240,9,202,20
    8,245,174,87,3,76,66,3,96,0,0

```

IPO

Si vous ne possédez pas d'assembleur, ne tapez que ces codes :

6000: A9 4C
6002: 8D F5 03
6005: A9 10
6007: 8D F6 03
600A: A9 60
600C: 8D F7 03
600F: 60

6010: A9 33
6012: 85 06
6014: A9 60
6016: 85 07

6018: A5 B8
601A: 48
601B: A5 B9
601D: 48
601E: A0 00
6020: 20 B7 00

6023: D1 06
6025: F0 01
6027: 60
6028: C8
6029: C0 03
602B: F0 09
602D: 20 B1 00
6030: 4C 23 60

```

1
2 INDEX EQU $6 ;Pour codes ASCII
3 VARCOUL EQU $8 ;Variable Couleur
4 POSHB EQU $19 ;Coord.horizontale
5 POSHH EQU $1A ;avec dépassement 256->279
6 POSV EQU $1B ;Coord.verticale
7 HCOLOR1 EQU $1C ;Var.en fonction de la 'HMASK'
8 HBASL EQU $26 ;Adresse du point HGR
9 HMASK EQU $30 ;Octet masque bit/point
10 CHRGET EQU $B1 ;Pour acquérir prochain carac.
11 CHRGOT EQU $B7 ;au curseur.
12 TXTPTR EQU $B8 ;Adresse du dernier caractère.
13 HCOLOR0 EQU $E4 ;Code couleur point
14 AMPERV EQU $3F5 ;Vecteur &
15 HPOSN EQU $F411 ;Calcul position point
16 HFNS EQU $F6B9 ;Transpose X-> XH XL
17 ORG $6000 ;Après les pages HGR
18
19 * Vectorisation &
20
21 DEBUT LDA $4C ;JMP
22 STA AMPERV ;sur le programme LM.
23 LDA <IPOAMP
24 STA AMPERV+1
25 LDA >IPOAMP
26 STA AMPERV+2
27 RTS ;Retour de BRUN ou CALL
28
29 IPOAMP LDA <ASCIIPO ;La table ASCII est lue
30 STA INDEX ;en index.
31 LDA >ASCIIPO
32 STA INDEX+1
33
34 * Conserver les pointeurs
35
36 LDA TXTPTR ;Pour revenir à l'interpréteur
37 PHA ;BASIC au bon endroit.
38 LDA TXTPTR+1
39 PHA
40 LDY $0
41 JSR CHRGOT ;Lecture du caractère après &
42
43 * Test de l'instruction & IPO
44
45 TEST CMP (INDEX),Y ;Comparons-le avec la table
46 BEQ SUIT ;...c'est bon -> SUIT
47 RTS ;Ce n'est pas pour nous.
48 SUIT INY ;Pour passer à la lettre
49 CPY $3 ;suivante sauf si 3e.
50 BEQ PARAMS
51 JSR CHRGET ;Lecture du code
52 JMP TEST ;et compare encore.
53

```


	54	* Table ASCII		
	55			
6033:	49 50 4F	56	ASCIPO ASC 'IPO'	
		57		
		58		
		59	* Paramètres fonctions	
		60		
6036:	20 B1 00	61	PARAMS JSR CHRGET	:Lecture du code
6039:	C9 C1	62	CMP £C1	:est-ce TO ?
603B:	F0 34	63	BEQ IPOTO	:Oui: fonction & IPO TO
603D:	68	64	PLA	
603E:	68	65	PLA	
603F:	20 B9 F6	66	JSR HFNS	:Il faut lire les paramètres
6042:	86 19	67	STX POSHB	:et les transférer.
6044:	84 1A	68	STY POSHH	
6046:	85 1B	69	STA POSV	
		70		
		71	* & IPO	
		72		
6048:	A5 E4	73	IPO LDA HCOLOR0	:Il faut conserver la couleur
604A:	85 08	74	STA VARCOUL	:pour la restituer en cas de
604C:	20 61 60	75	JSR ADRPLOT	:changement
604F:	D0 07	76	BNE NOCHANG	
6051:	A9 00	77	CHANG LDA £0	:Couleur noire pour tracer
6053:	85 E4	78	STA HCOLOR0	:sur une zone déjà dessinée
6055:	20 61 60	79	JSR ADRPLOT	
6058:	51 26	80	NOCHANG EOR (HBASL),Y	
605A:	91 26	81	STA (HBASL),Y	:Dessin du point après masques
605C:	A5 08	82	LDA VARCOUL	:successifs.
605E:	85 E4	83	STA HCOLOR0	:Rétablir la couleur d'entrée
6060:	60	84	RTS	:Retour principal
		85		
6061:	A6 19	86	ADRPLOT LDX POSHB	:Coordonnées écran HGR pour
6063:	A4 1A	87	LDY POSHH	:le SP 'HPOSN'
6065:	A5 1B	88	LDA POSV	
6067:	20 11 F4	89	JSR HPOSN	:On calcule l'adresse en fonction
606A:	A5 1C	90	LDA HCOLOR1	:des coordonnées 'POS'.
606C:	51 26	91	EOR (HBASL),Y	:Masque de couleur
606E:	25 30	92	AND HMASK	:Masque du bit/point
6070:	60	93	RTS	:bas les masques!
		94		
		95	* & IPO TO (voir codes IPO.C)	
		96		
		97	IPOTO EQU *	

SUITE ET FIN AU VERSO →

Table des symboles ordre alphabétique

-ADRPLOT...\$6061	-AMPERV....\$03F5	-ASCIPO....\$6033	? -CHANG.....\$6051
-CHRGET....\$00B1	-CHRGOT....\$00B7	? -DEBUT....\$6000	-HBASL.....\$0026
-HCOLOR0...\$00E4	-HCOLOR1...\$001C	-HFNS.....\$F6B9	-HMASK.....\$0030
-HPOSN.....\$F411	-INDEX.....\$0006	? -IPO.....\$6048	-IPOAMP....\$6010
-IPOTO.....\$6071	-NOCHANG...\$6058	-PARAMS....\$6036	-POSHB....\$0019
-POSHH.....\$001A	-POSV.....\$001B	-SUIT.....\$6028	-TEST.....\$6023
-TXTPTR....\$00B8	-VARCOUL...\$0008		

IPO TO (suite)

Suite et fin de la routine IPO en langage machine. Cette partie n'est qu'une transposition modifiée de HLINE... d'où l'absence du programme source (lequel exigerait d'ailleurs plusieurs pages).

*6071.61F1

```
6071- 68 68 20 B1 00 20 B9
6078- F6 84 9D A8 8A A6 9D 20
6080- 54 61 60 10 23 A5 30 4A
6088- B0 05 49 C0 85 30 60 88
6090- 10 02 A0 27 A9 C0 85 30
6098- 04 E5 A5 1C 0A C9 C0 10
60A0- 06 A5 1C 49 7F 85 1C 60
60A8- A5 30 0A 49 80 30 DD A9
60B0- 81 C8 C0 28 90 E0 A0 00
60B8- B0 DC 18 A5 D1 29 04 F0
60C0- 21 A9 7F 25 30 31 26 D0
60C8- 19 E6 EA A9 7F 25 30 10
60D0- 11 18 A5 D1 29 04 F0 0A
60D8- B1 26 45 1C 25 30 D0 02
60E0- E6 EA A5 D1 65 D3 29 03
60E8- C9 02 6A B0 96 30 30 18
60F0- A5 27 2C B9 F5 D0 22 06
60F8- 26 B0 1A 2C CD F4 F0 05
6100- 69 1F 38 B0 12 69 23 48
6108- A5 26 69 B0 B0 02 69 F0
6110- 85 26 68 B0 02 69 1F 66
6118- 26 69 FC 85 27 60 18 A5
6120- 27 69 04 2C B9 F5 D0 F3
```

```
6128- 06 26 90 18 69 E0 18 2C
6130- 08 F5 F0 12 A5 26 69 50
6138- 49 F0 F0 02 49 F0 85 26
6140- A5 E6 90 02 69 E0 66 26
6148- 90 D1 48 A9 00 85 E0 85
6150- E1 85 E2 68 48 38 E5 E0
6158- 48 8A E5 E1 85 D3 B0 0A
6160- 68 49 FF 69 01 48 A9 00
6168- E5 D3 85 D1 85 D5 68 85
6170- D0 85 D4 68 85 E0 86 E1
6178- 98 18 E5 E2 90 04 49 FF
6180- 69 FE 85 D2 84 E2 66 D3
6188- 38 E5 D0 AA A9 FF E5 D1
6190- 85 1D A4 E5 B0 05 0A 20
6198- 83 60 38 A5 D4 65 D2 85
61A0- D4 A5 D5 E9 00 85 D5 48
61A8- 8A 48 98 48 08 20 D2 61
61B0- 28 68 A8 68 AA 68 E8 D0
61B8- 04 E6 1D F0 14 A5 D3 B0
61C0- D5 20 ED 60 18 A5 D4 65
61C8- D0 85 D4 A5 D5 65 D1 50
61D0- D4 60 A5 E4 85 08 20 EB
61D8- 61 D0 07 A9 00 85 E4 20
61E0- EB 61 51 26 91 26 A5 08
61E8- 85 E4 60 A5 E4 51 26 25
61F0- 30 60
```

Après CTRL-C ou RESET, sauvez le tout (IPO + IPO TO) : **BSAVE IPO.C, A\$6000, L\$1F2**

VÉRIFICATION :

```
T = 0 : FOR I = 24576 TO 24575 + 498 : T =
T + PEEK(I) : NEXT : PRINT T
```

RÉSULTAT : 58634

ADRESSES UTILES

Pour connaître l'adresse à laquelle vient d'être chargé un fichier binaire, interrogez le DOS (ou ProDOS) par :

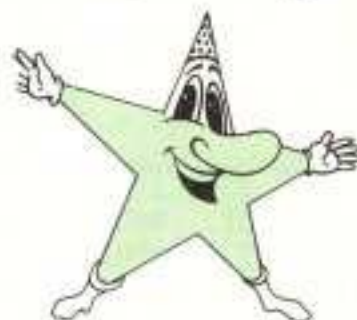
- PRINT PEEK(43634) + PEEK(43635) * 256 (DOS)
- PRINT PEEK(48825) + PEEK(48826) * 256 (ProDOS)

Vous connaîtrez la longueur (décimale) du fichier par :

- PRINT PEEK(43616) + PEEK(43617) * 256 (DOS)
- PRINT PEEK(48840) + PEEK(48841) * 256 (ProDOS)

AMPERSAND (\$3F5.3F6.3F7)

On installe les vecteurs par trois pokes (1013,X : 1014,X : 1015,X). Ainsi, si vous tapez POKE 1013,76 : POKE 1014,165 : POKE 1015,214, il vous suffira de taper & (+ return) pour lister votre programme.



LETTRES EN PLUIE

Vous ne trouvez pas un peu triste de taper "HOME" dans un programme basic ? tout s'efface d'un coup, vous laissant face à un grand écran vide...

Voici un programme pour les nostalgiques de l'effacement progressif. Chargez le programme par PRINT CHR\$(4);"BLOAD PLUIE". Lorsque vous voudrez effacer l'écran, vous taperez CALL 768, les lettres sembleront tomber du ciel !

Si vous ne possédez pas d'assembleur, tapez seulement les codes, après un CALL—151 :

300 : A5 22
302 : 48
303 : A9 00
305 : A2 D0
307 : 85 06
309 : 86 07

30B : A0 00
30D : E6 06
30F : D0 02
311 : E6 07

313 : A5 4E
315 : 71 06
317 : C8
318 : 85 4E
31A : 38
31B : E5 21
31D : B0 F9
31F : A5 4F
321 : 18
322 : 71 06
324 : 18

```

0 *
1 |
2 | PROGRAMME INÉDIT DE CLAUDE AUBRY |
3 |
4 POINTEUR EQU $06 ; Dans la ROM pour NB aléatoire
5 WNDLFT EQU $20 ; Marge GAUCHE
6 WNDWDTH EQU $21 ; LARGEUR fenêtre
7 WNDTOP EQU $22 ; HAUT de la fenêtre
8 WNDBTM EQU $23 ; BAS de la fenêtre
9 CH EQU $24 ; Curseur HORIZONTAL
10 CV EQU $25 ; Curseur VERTICAL
11 BASL EQU $28 ; Adresse de la LIGNE courante
12 RNDL EQU $4E ; Nombre aléatoire
13 ROM EQU $D000 ; Première adresse BASIC
14 VTAB EQU $FC22 ; Déplace le curseur en CV
15 *
16 ORG $300 ; Appelé par CALL 768
17 *
18 PLUIE EQU *
19 LDA WNDTOP ; On sauve le haut de la fenêtre
20 PHA ; ...en l'empilant
21 LDA £ < ROM ; On crée un pointeur (dans la
22 LDX £ > ROM ; ...rom les valeurs trouvées
23 STA POINTEUR ; serviront de nombre aléatoire)
24 STX POINTEUR+1
25 *
26 LDY £$00
27 INCPO INC POINTEUR ; Incrémente le pointeur
28 BNE LOOP ; HSB et LSB des nombres
29 INC POINTEUR+1 ; ...aléatoires
30 *
31 LOOP LDA RNDL ; Création du nombre aléatoire
32 ADC (POINTEUR),Y
33 INY ; ...avec valeurs prises en ROM
34 AGAIN0 STA RNDL ; Stocke
35 SEC ; ...avant soustraction
36 SBC WNDWDTH ; dans la fenêtre
37 BCS AGAIN0 ; Non : on soustrait
38 LDA RNDL+1 ; ...deuxième aléatoire
39 CLC ; Avant addition
40 ADC (POINTEUR),Y
41 AGAIN1 CLC ; Avant addition
  
```

325 : 65 22 42
 327 : 85 4F 43
 329 : 38 44
 32A : E5 23 45
 32C : B0 F6 46
 32E : A4 4E 47
 330 : 84 24 48
 332 : A5 4F 49
 334 : C5 22 50
 336 : 90 EC 51
 338 : 85 25 52
 33A : 20 22 FC 53
 33D : B1 28 54
 33F : C9 A0 55
 341 : F0 16 56
 343 : 48 57
 344 : A9 A0 58
 346 : 91 28 59
 348 : A5 25 60
 34A : C9 17 61
 34C : 90 03 62
 34E : 68 63
 34F : B0 08 64
 351 : E6 25 65

ADC WNDTOP : ...haut de fenêtre
 STA RNDL+1 : stocké
 SEC
 SBC WNDBTM : dans fenêtre
 BCS AGAIN1 : Non ?
 LDY RNDL : Récupère
 STY CH : ...colonne
 LDA RNDL+1 : ligne
 CMP WNDTOP : dans la fenêtre
 BCC AGAIN1 : Non ?
 STA CV
 JSR VTAB : Oui positionne BASL BASH
 LDA (BASL),Y : Caractère
 CMP £" " : Un espace ?
 BEQ ESPACE : Oui ?
 PHA : On sauve le caractère
 LDA £" " : On l'efface de l'écran
 STA (BASL),Y : ...en écrivant espace
 LDA CV : Ligne
 CMP £\$17 : La dernière ?
 BCC FINECRAN : Oui on passe ligne suivante
 PLA : Sauve le numéro de ligne
 BCS ESPACE : On va en première colonne
 INC CV : Ligne suivante

353 : 20 22 FC 66
 356 : 68 67
 357 : 91 28 68
 359 : A0 00 69
 35B : 84 24 70
 35D : A5 22 71
 35F : 85 25 72
 361 : 20 22 FC 73
 364 : B1 28 74
 366 : C9 A0 75
 368 : D0 A3 76
 36A : C8 77
 36B : C4 21 78
 36D : 90 F5 79
 36F : E6 22 80
 371 : A5 22 81
 373 : C5 23 82
 375 : 90 E2 83
 377 : 68 84
 378 : 85 22 85
 37A : 60 86
 87

PLUIE
 JSR VTAB : Positionne BASL BASH
 PLA : Récupère le caractère
 STA (BASL),Y : On l'écrit une ligne plus bas
 LDY £\$00 : Début de ligne
 STY CH : On recherche un caractère
 LDA WNDTOP : Haut de fenêtre
 STA CV : En curseur vertical
 JSR VTAB : Positionne BASL BASH
 LDA (BASL),Y : Prend caractère
 CMP £" " : Est-ce un espace ?
 BNE INCPO : Non : prochain aléatoire
 INY : Oui, on incrémente
 CPY WNDWDTH : = à la largeur ?
 BCC CHERCHE : Non : on continue la recherche
 INC WNDTOP : Terminé pour haut de fenêtre
 LDA WNDTOP : Récupère haut de fenêtre
 CMP WNDBTM : Compare haut et bas de fenêtre
 BCC ESPACE : < on continue ?
 PLA : Sinon terminé, on récupère
 STA WNDTOP : ...position de haut de fenêtre
 RTS : Retour au Basic

300 : A5 22 48 A9 00 A2 D0 85 06 86 07 A0
 00 E6 06 D0 02 E6 07 A5 4E 71 06 C8 85
 4E 38 E5 21 B0 F9 A5 4F 18 71 06 18 65 22
 85 4F 38 E5 23 B0 F6 A4 4E 84 24 A5 4F
 C5 22 90 EC 85 25 20 22 FC B1 28 C9 A0
 F0 16 48 A9 A0 91 28 A5 25 C9 17 90 03
 68 B0 08 RETURN
 351 : E6 25 20 22 FC 68 91 28 A0 00 B4 24
 A5 22 85 25 20 22 FC B1 28 C9 A0 D0 A3
 C8 C4 21 90 F5 E6 22 A5 22 C5 23 90 E2
 68 85 22 60 RETURN
 CTRL-C et RETURN

SAISIE-CONTRÔLE

Si vous êtes néophyte, tapez CALL—151, puis return. Ensuite, en 40 colonnes, tapez les valeurs ci-contre.

Pour terminer : BSAVE PLUIE.C, A\$300, L\$7B

VÉRIFICATION :

T=0 : FOR I = 768 TO 890 : T = T + PEEK(I) :
 NEXT : PRINT T Résultat : 14077

Votre bibliothèque INFORMATIQUE

• par Clément RENARD



• MS-DOS FACILE (Marabout)

Je sais : vous ne disposez que d'un modeste ordinateur personnel 8 bits, et j'ai l'air de retourner le fer dans la plaie en vous parlant (oh ! pas trop souvent !) des machines travaillant sur 16 bits. Mais elles existent... et votre prochain Apple aura cette inestimable qualité.

Faut-il voir, dans le MS-DOS, comme Ilya Virgatchik, auteur de ce nouveau Marabout-Service, "un système d'exploitation de seconde génération, en quelque sorte l'héritier du CP/M" ?

Sûrement, puisqu'il est aujourd'hui, pour les ordinateurs 16 bits, le seul vrai standard : le triomphe de Microsoft et d'IBM (en effet, PC-DOS et MS-DOS, même combat !).

Il faut lire MS-DOS FACILE, ne serait-ce que pour le comparer au ProDOS d'Apple. On constatera que les deux systèmes comportent de nombreux points communs, même si ProDOS, mais c'est tout à son honneur, tourne

sur une machine d'une autre génération (n'accablons pas les "anciens", surtout quand ils ont joué l'éminent rôle de défricheurs !). Personnellement, j'ai parcouru ce petit guide avec intérêt et j'avoue avoir un faible pour la présentation des ouvrages d'une collection au titre ambitieux : "Les Best-sellers de l'Informatique".

Tout un programme !

• GRAPHISME SCIENTIFIQUE (Masson)

Un livre sérieux dans une collection destinée à des lecteurs studieux : "Méthode + Programmes". Le sujet étudié ici est ambitieux : le graphisme scientifique sur micro-ordinateur (de la 2^e à la 3^e dimension).

Dans cette troisième édition, revue et corrigée, Robert Dony s'adresse à tous les possesseurs d'un ordinateur personnel ayant des possibilités graphiques, avec 50 applications résolues en Basic, et testées sur Apple II et Wang 2200.

J'aurais aimé publier ici, non pas l'une de ces 50 routines, mais une sélection des graphiques dont ce manuel est bourré, mais je préfère vous conseiller de les découvrir vous-même, devant votre Apple.

Ne vous attendez pas à une succession de programmes illustrés... à recopier sans y attacher autrement d'importance, mais bien à des "applications" longuement et minutieusement expliquées (de solides connaissances mathématiques ne sont pas inutiles quand on désire vraiment tirer profit d'un livre comme celui-ci). Cela étant, personne ne vous obligera à tout lire et les programmes peuvent évidem-

ment se suffire à eux-mêmes. L'auteur espère que cette nouvelle édition est définitivement expurgée des quelques coquilles et erreurs des deux précédentes. A vous de lui prouver le contraire : je suis persuadé qu'il vous en saura gré. C'est ainsi que l'on avance... dans notre petit monde de l'informatique !

• LE LANGAGE MACHINE SUR VOTRE APPLE II^e (Inter Editions)

L'auteur est Graham Keeler, professeur à l'Université de Salford (Grande-Bretagne). Il est traduit par Phong Tuan Nghiem. L'ouvrage constitue une bonne introduction au langage machine de l'Apple et à l'architecture du 6502, mais il va plus loin en se révélant accessible au profane.

L'initiation est menée rondement, mais sûrement. De plus, en annexe, l'auteur fournit les solutions des exercices choisis (à noter qu'une disquette, contenant toutes les routines du livre, peut être vendue séparément).

Clément RENARD.



TITRE DOUBLE FORMAT

TREMPIN MICRO

TITRE DOUBLE FORMAT
TREMPIN MICRO



Caractères géants sur votre imprimante Apple

Ces titres n'ont pas été agrandis et vous obtiendrez les mêmes (sur **DMP** et **IMAGEWRITER I** et **II**), à partir de la fonte que nous vous proposons.

Quant aux caractères en mode inverse, lisez un peu plus loin...
et mettez votre imprimante à contribution !

VOICI UN TITRE
ET UN AUTRE ELARGI
ON PEUT S'AMUSER !
AVEC UN APPLE
minuscules déconseillées!

TITRE DOUBLE FORMAT
TREMPIN MICRO

Comment obtenir des titres en double hauteur (16 bits) ?

Il est indispensable, dans un premier temps, de concevoir une fonte spéciale dont chaque lettre utilise deux caractères. Dans celle que je vous propose (CARAC.DOUBLE), la minuscule de chaque caractère sert de code à la partie supérieure de la lettre "double format" et la majuscule est le code de la partie inférieure.

Ainsi pour obtenir "TOTO", on tapera : PRINT "toto" : PRINT "TOTO". Naturellement, il faut définir un interlignage de 8 pts : CHR\$(27)"T08".

Vous comprenez pourquoi CARAC.DOUBLE ne comporte que des capitales.

Avec le programme CARAC.1.0 de notre précédent numéro, la composition d'une telle fonte ne pose aucun problème.

CHARGEMENT :

Non moins facile par le truchement de CARAC.CHARG (Tremplin Micro n°6).

Quant à la démonstration, la voici. Vous pourrez essayer les différentes possibilités que vous offre l'IMAGEWRITER, tant en mode expansé qu'en mode normal (essayez avec ESCAPE"N", "E", "q", "Q", etc.).

CARAC.DBLE

```
100 PRINT CHR$(4)"PRÉ1"  
110 PRINT CHR$(27)"!"  
120 PRINT CHR$(27)"p"  
130 PRINT CHR$(27)"*": REM UNIDIRECTIONNELLE  
140 PRINT CHR$(27)"T08": REM INTERLIGNE=8 PTS  
150 PRINT CHR$(27)"'": REM CARACTERES PERSONNAL  
ISES  
160 PRINT : PRINT  
170 PRINT "bonne annee": REM PARTIE SUPERIEURE  
180 PRINT "BONNE ANNEE": REM PARTIE INFÉRIEURE  
190 PRINT : PRINT : PRINT  
200 PRINT "titre double format"  
210 PRINT "TITRE DOUBLE FORMAT": PRINT : PRINT :  
PRINT  
220 PRINT "tremplin micro"  
230 PRINT "TREMPLIN MICRO"  
240 PRINT CHR$(27)"$"; CHR$(27)"A"; CHR$(27)  
+ CHR$(34); CHR$(27)"<"  
250 PRINT CHR$(15)  
260 PRINT CHR$(4)"PRÉ0"
```

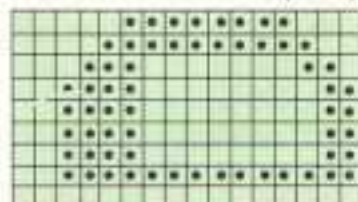
CARAC-DOUBLE

```
4000- 1B 2B 1B 49 62 50 00 00  
4008- FF FF FF FF 83 83 83 83  
4010- 83 83 C7 FF 7E 3C 42 50  
4018- 00 00 FF FF FF FF E1 E1  
4020- E1 E1 E1 E1 F3 FF 7E 3C  
4028- 6F 50 00 00 FC FE FF FF  
4030- 07 03 03 03 03 03 07 FF  
4038- FE FC 4F 50 00 00 3F 7F  
4040- FF FF F0 E0 E0 E0 E0 E0  
4048- F0 FF 7F 3F 6E 50 00 00  
4050- FF FF FF FF 1C 38 70 E0  
4058- C0 80 00 00 FF FF 4E 50  
4060- 00 00 FF FF FF FF 00 00  
4068- 00 00 01 03 07 0E FF FF  
4070- 61 50 00 00 F8 FC FE FF  
4078- 83 83 83 83 83 83 83 86  
4080- FC F8 41 50 00 00 FF FF  
4088- FF FF 03 03 03 03 03 03  
4090- 03 03 FF FF 65 50 00 00  
4098- FF FF FF FF 83 83 83 83  
40A0- 83 83 03 03 03 03 45 50  
40A8- 00 00 FF FF FF FF E1 E1  
40B0- E1 E1 E1 E1 E0 E0 E0 E0  
40B8- 6D 50 00 00 FF FF FF FF  
40C0- 38 70 E0 C0 E0 70 38 FF  
40C8- FF FF 4D 50 00 00 FF FF  
40D0- FF FF 00 00 00 01 00 00  
40D8- 00 FF FF FF 6C 50 00 00  
40E0- FF FF FF FF 00 00 00 00  
40E8- 00 00 00 00 00 00 4C 50  
40F0- 00 00 FF FF FF FF E0 E0  
40F8- E0 E0 E0 E0 E0 E0 E0 E0  
4100- 75 50 00 00 FF FF FF FF  
4108- 00 00 00 00 00 00 00 FF  
4110- FF FF 55 50 00 00 3F 7F  
4118- FF FF F0 E0 E0 E0 E0 E0  
4120- F0 FF 7F 3F 76 50 00 00  
4128- FF FF FF FF 00 00 00 00  
4130- 00 00 00 FF FF FF 56 50  
4138- 00 00 03 07 0F 1F 38 70  
4140- E0 E0 60 30 18 0F 07 03  
4148- 73 50 00 00 3C 7E FF FF  
4150- E7 C3 C3 C3 C3 C3 87  
4158- 0E 0C 53 50 00 00 10 30  
4160- 70 F1 E1 E1 E1 E1 E1 E1  
4168- F3 7F 3F 1E 74 50 00 00  
4170- 07 07 07 07 07 FF FF FF  
4178- FF 07 07 07 07 07 54 50  
4180- 00 00 00 00 00 00 00 FF  
4188- FF FF FF 00 00 00 00 00
```

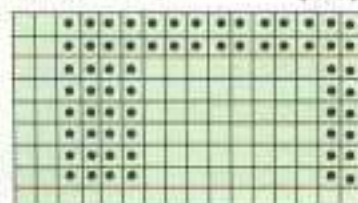
CARAC-DOUBLE

(suite et fin)

Partie (a) (Haut)



Partie (A) (Bas)



Exemple (caractères aA)

SAUVEZ VOTRE FONTE :

BSAVE CARAC.DOUBLE,
A\$4000, L\$397

VÉRIFICATION :

```
T = 0 : FOR I = 16384 TO
17302 : T = T + PEEK(I) :
NEXT : PRINT T
```

RÉSULTAT : 110821

4190-	72 50 00 00 FF FF FF FF	4298-	E7 E7 F7 FF 7F 3F 68 50
4198-	83 83 83 83 83 C3 E7 7F	42A0-	00 00 FF FF FF FF 80 80
41A0-	3E 1C 52 50 00 00 FF FF	42A8-	80 80 80 80 80 FF FF FF
41A8-	FF FF 03 03 07 0F 1F 39	42B0-	48 50 00 00 FF FF FF FF
41B0-	70 E0 C0 80 78 50 00 00	42B8-	03 03 03 03 03 03 03 FF
41B8-	03 07 0F 1F 3C F8 F0 E0	42C0-	FF FF 6A 4E 00 00 00 00
41C0-	F0 B8 1C 0F 07 03 58 50	42C8-	00 00 00 00 00 00 FF FF
41C8-	00 00 F0 F8 FC FE 0F 07	42D0-	FF FF 4A 4E 00 00 3E 7E
41D0-	03 01 03 07 0E FC F8 F0	42D8-	FE F0 E0 E0 E0 F0 FF FF
41D8-	69 47 00 00 FF FF FF FF	42E0-	7F 3F 6B 50 00 00 FF FF
41E0-	00 49 47 00 00 FF FF FF	42E8-	FF FF 80 80 C0 E0 F0 78
41E8-	FF 00 70 50 00 00 FF FF	42F0-	3C 1E 0F 07 4B 50 00 00
41F0-	FF FF 83 83 83 83 C3	42F8-	FF FF FF FF 03 03 03 07
41F8-	E7 7F 3E 1C 50 50 00 00	4300-	0F 1E 3C F8 F0 E0 71 50
4200-	FF FF FF FF 03 03 03 03	4308-	00 00 FC FE FF FF 07 03
4208-	03 01 00 00 00 00 63 50	4310-	03 03 03 03 07 FF FE FC
4210-	00 00 FC FE FF FF 07 03	4318-	51 50 00 00 3F 7F FF FF
4218-	03 03 03 03 07 1F 1E 1C	4320-	F0 E0 E4 EC FC F8 70 FF
4220-	43 50 00 00 3F 7F FF FF	4328-	FF BF 77 50 00 00 FF FF
4228-	F0 E0 E0 E0 E0 F0 F8	4330-	FF FF 00 00 80 80 80 00
4230-	78 38 64 50 00 00 FF FF	4338-	00 FF FF FF 57 50 00 00
4238-	FF FF 03 03 03 03 03 03	4340-	0F 1F 3F 7F E0 70 3F 1F
4240-	07 FF FE FC 44 50 00 00	4348-	3F 70 E0 7F 3F 1F 79 50
4248-	FF FF FF FF E0 E0 E0 E0	4350-	00 00 03 07 0F 1F 3C 78
4250-	E0 E0 F0 FF 7F 3F 66 50	4358-	F0 E0 F0 F8 3C 1F 0F 07
4258-	00 00 FF FF FF FF 83 83	4360-	59 50 00 00 1E 3F 7F FF
4260-	83 83 83 83 03 03 03 03	4368-	F0 E0 E1 E3 F7 FF 7E 3C
4268-	46 50 00 00 FF FF FF FF	4370-	18 00 7A 50 00 00 07 07
4270-	01 01 01 01 01 01 00 00	4378-	07 07 07 07 87 C7 E7 F7
4278-	00 00 67 50 00 00 FC FE	4380-	7F 3F 1F 0F 5A 50 00 00
4280-	FF FF 07 03 03 03 03 03	4388-	E0 F0 F8 FC FE FF F7 F3
4288-	07 1F 1E 1C 47 50 00 00	4390-	F1 F0 F0 F0 F0 F0 04
4290-	3F 7F FF FF F0 E0 E0 E0		

INVERSION DE FONTE

AVEC UN APPLE

```
100 TEXT : NORMAL : HOME
110 D$ = CHR$(4)
120 V$ = INPUT "TITRE DE LA FONTE: "; F$
130 ONERR GOTO 100
140 PRINT D$"BLOAD" F$, A16384
150 L = PEEK(43616) + PEEK(43617) * 256
160 REM PRODOS:48840 & 48841
```

► Si votre colonne de blanc est à gauche, utilisez CARAC.INV (lire page 16).

```
180 PRINT D$"BLOAD CARAC.INV": CALL 768
190 F$ = F$ + ".INV"
200 PRINT D$"BSAVE" F$, A16384, L * L
210 PRINT D$"RUN CARAC.CHARG"
```

Avec CARAC 1.0 (Tremplin Micro n°6), vous disposez d'un utilitaire qui vous permet de transformer les fontes graphiques du TOOL KIT Apple en polices de caractères personnalisés. Certaines d'entre elles permettent de très belles impressions en mode inverse. C'est le cas, par exemple, de COLOSSAL.SET. D'une manière générale, il faut, au départ, des caractères assez gras et il est souhaitable de ne pas abuser des minuscules. Si vous tapez d'abord ce petit programme en basic, l'opération sera automatique... ou presque !

Attention ! n'utilisez que des fontes obtenues à partir de CARAC.1.0.

La routine **CARAC.CHARG** (T.M. n°6) vous permettra de charger votre nouvelle fonte.

Vous ferez ensuite vos essais avec un embryon de programme ressemblant à **CARAC.INVDEMO** (ci-contre).

Changez tout ce que vous voudrez... sauf les codes !

PROBLÈME : **CARAC.INV** (ci-après) a été conçu pour inverser des fontes dans lesquelles le blanc (séparant la lettre de la suivante) est à droite du caractère, ce qui est normal.

Si vous avez conçu une fonte dans laquelle vous avez placé le blanc à gauche, utilisez **CARAC.INVØ** à la place de **CARAC.INV**.

CONSEIL : Pour obtenir de très bons résultats, concevez vous-même une police de caractères très gras. Place à l'imagination... et au talent !

Guy-HACHETTE.

CARAC.INVDEMO

```

100 D$ = CHR$(4):E$ = CHR$(27)
110 T$ = E$ + "N" + E$ + "X" + E$ +
    "'* + E$ + "!"
120 TEXT : NORMAL : HOME
130 PRINT D$"PR#1": PRINT
140 PRINT T$: PRINT
150 PRINT CHR$(15)
160 PRINT " VOICI UN TITRE "
170 PRINT CHR$(14)
180 PRINT " ET UN AUTRE ELARGI "
190 PRINT : PRINT "On peut s'amuser:
    "
200 PRINT : PRINT E$"T09";E$">"
210 L$ = "                ": PRINT L$
220 PRINT " AVEC UN APPLE ": PRINT L$
    $
230 PRINT : PRINT CHR$(15)
240 PRINT "Minuscules deconseillees!
    "
250 PRINT E$"Y";E$"$";E$ + CHR$(34)
    : CHR$(15);E$"<";E$"A"
260 PRINT D$"PR#0"
    
```

CARAC.INV

Si vous ne possédez pas d'assembleur, tapez seulement ces codes :

0300: A9 05
0302: 85 06
0304: A9 40
0306: 85 07

0308: A0 00
030A: B1 06
030C: 29 1F
030E: F0 04
0310: C9 11
0312: 90 01
0314: 60

0315: 85 08

```

0 *****
1 * CARAC.INV GUY-HACHETTE 15 DECEMBRE 85 *
2 * (INVERSE UNE FONTE DE CARACTERES PERSONNALISES) *
3 *-----*
4 * Fonctionne (automatiquement) avec CARAC.INVBAS *
5 *****
6 ORG $300
7 ADRB EQU $06 ; Pointeur adresse caractère
8 ADRH EQU $07 ; Partie haute même adresse
9 LONG EQU $08 ; Longueur du caractère
10 VAL EQU $09 ; Stockage temporaire
11 *-----*
12 LDA £$05 ; Adresse du premier...
13 STA ADRB ; ...traité (partie basse)
14 LDA £$40 ; Adresse de ce caractère,
15 STA ADRH ; mais partie haute
16 *-----*
17 DEBUT LDY £$0 ; 0 dans le registre Y pour...
18 LDA (Adb),Y ; ...lecture adres. indirect
19 AND £$1F ; Avec 00011111, $41/$61=$01
20 BEQ FIN ; Si le résultat est zéro...
21 CMP £$11 ; ...ou dépasse 16 (points),
22 BCC SUITE ; c'est la fin
23 FIN RTS
24 *-----*
25 SUITE STA LONG ; La longueur est sauvee,
    
```

CARAC.INV (suite et fin)

0317:	E6 08	26		INC	LONG	:	est incrémentée de 1
0319:	A0 01	27		LDY	£\$01	:	Boucle de lecture
031B:	B1 06	28	BCLE	LDA	(Adrb),Y	:	Premier octet
031D:	49 FF	29		EOR	£\$FF	:	Inversion de la valeur
031F:	38	30		SEC		:	Retenue forcée à 1...
0320:	2A	31		ROL		:	...pour une rotation gauche
0321:	C9 0D	32		CMP	£\$0D	:	Y a-t-il 0D dans l'accumul.?
0323:	F0 06	33		BEQ	TRAITE	:	Si oui: vers traitement
0325:	C9 8D	34		CMP	£\$8D	:	Pareil avec 8D
0327:	F0 02	35		BEQ	TRAITE		
0329:	D0 02	36		BNE	FAIT	:	Saut au-dessus du traitement
032B:	29 FE	37	TRAITE	AND	£\$FE	:	Masque "11111110" (moins 1)
032D:	48	38	FAIT	PHA		:	On empile provisoirement A,
032E:	A5 09	39		LDA	VAL	:	le temps de charger l'octet
0330:	91 06	40		STA	(Adrb),Y	:	précédent à l'adr. actuelle
0332:	68	41		PLA		:	Récupération sur la pile et
0333:	85 09	42		STA	VAL	:	sauvegarde dans VAL
0335:	C8	43		INY		:	Un de plus pour Y de boucle
0336:	C4 08	44		CPY	LONG	:	Est-ce la bonne longueur ?
0338:	90 E1	45		BCC	BCLE	:	Si "non": goto BCLE
033A:	A0 01	46		LDY	£\$01	:	Y chargé avec 1 pour mettre
033C:	A5 09	47		LDA	VAL	:	le contenu de VAL dans le
033E:	91 06	48		STA	(Adrb),Y	:	premier octet du caractère
0340:	E6 08	49		INC	LONG	:	Ne pas oublier la case conte-
0342:	A5 08	50		LDA	LONG	:	nant le code du caractère
0344:	18	51		CLC		:	avant de calculer l'adresse
0345:	65 06	52		ADC	ADRB	:	du caractère suivant
0347:	85 06	53		STA	ADRB		
0349:	A5 07	54		LDA	ADRH	:	Il peut y avoir une retenue,
034B:	69 00	55		ADC	£\$00	:	et il faut aussi y penser
034D:	85 07	56		STA	ADRH		
034F:	4C 08 03	57		JMP	DEBUT		

BSAVE CARAC.INV, A\$300, L\$52

Table des symboles ordre alphabétique

-ADRB.....\$0006	-ADRH.....\$0007	-BCLE.....\$031B	-DEBUT.....\$0308
-FAIT.....\$032D	-FIN.....\$0314	-LONG.....\$0008	-SUITE.....\$0315
-TRAITE....\$032B	-VAL.....\$0009		

*300.340

0300- A9 05 85 06 A9 40 85 07
 0308- A0 00 B1 06 29 1F F0 04
 0310- C9 11 90 01 60 85 08 A8
 0318- B1 06 49 FF 38 2A C9 0D
 0320- F0 06 C9 8D F0 02 D0 02
 0328- 29 FE 91 06 88 D0 E9 A5
 0330- 08 18 69 02 65 06 85 06
 0338- A5 07 69 00 85 07 4C 08
 0340- 03

CARAC.INV0

Mini-routine spéciale pour des polices dont la colonne de blanc est à l'extrême gauche.

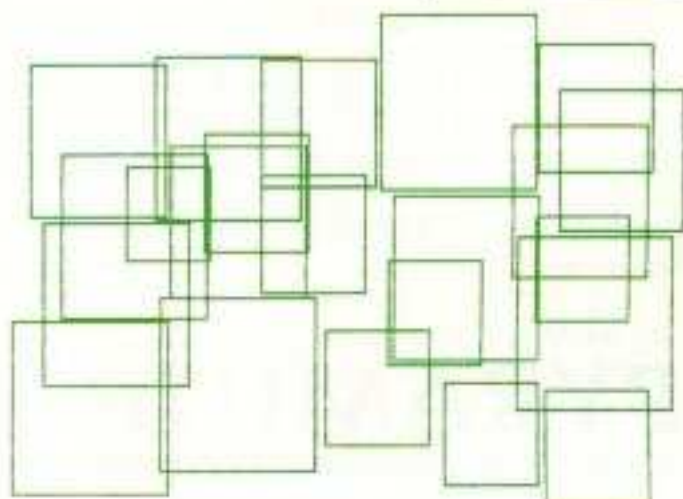
BSAVE CARAC.INV0, A\$300, L\$41

AVERTISSEMENT :

Ces programmes fonctionnent sur Apple IIe, IIc et IIe 65C02, avec imprimante DMP ou IMA-GEWRITER — sous ProDOS, tenez compte des indications mentionnées en REM.

MÉMO-VISU

B ON PIED... peut-être, mais bon œil ? Ce jeu visuel sans prétention vous permettra de tester votre sens de la perception (rien à voir avec la collecte des impôts, rassurez-vous !)... et aussi d'apprendre à utiliser une petite forme (ici : un carré).



■ **110** : HGR et HGR2 initialisent les deux pages graphiques... et les nettoient.

■ **120** : POKE 32,10 et POKE 33,20 permettent de créer une fenêtre au centre de l'écran.

■ **130** : CALL — 198 émet un Bip.

■ **150** : On réaffiche ici la page graphique HGR sans texte (et sans effacement).

■ **170** : SCALE fait varier la taille de la forme (aléatoirement, avec l'utilisation de RND).

■ **190** : POKE 230,64 permet d'écrire dans la page 2 (HGR2) et POKE 230,32 dans la page 1

■ **300** : Affichage de la page 2

■ ————— ■

Vous noterez que chaque carré est tracé dans la page 2 (sans effacement), tracé de nouveau dans la page 1... puis effacé par XDRAW après une courte boucle d'attente permettant de le visualiser. Cette boucle est fonction de la vitesse choisie (ligne 130).

```

100 TEXT : HOME : PRINT CHR$(21): GOSUB 390
110 HGR : HGR2 : HCOLOR= 3: ROT= 16: TEXT
120 T$ = " : " : POKE 32,10: POKE 33,
    20: VTAB 12: INVERSE : PRINT T$:" COMPTEZ LES CAR
    RES ";T$: NORMAL
130 PRINT : PRINT : CALL - 198: PRINT "VITESSE (1 A 4
    ) -> " : GET V$:V = VAL (V$): IF V < 1 OR V > 4 T
    HEN V = 1
140 TEXT : HOME
150 POKE - 16304,0: POKE - 16300,0: POKE - 16297,0: P
    OKE - 16302,0
160 N = 12 + INT ( RND (1) * 12)
170 SCALE= 8 + INT ( RND (1) * 8)
180 X = 32 + INT ( RND (1) * 215):Y = 32 + INT ( RND
    (1) * 127)
190 POKE 230,64: DRAW 1 AT X,Y: POKE 230,32
200 DRAW 1 AT X,Y
210 FOR Z = 1 TO 100 * V: NEXT
220 XDRAW 1 AT X,Y:F = F + 1: IF F < N THEN 170
230 GOSUB 380
240 VTAB 22: INVERSE : PRINT "COMBIEN Y A-T-IL EU DE
    CARRES ?": NORMAL
250 INPUT " $":R$
260 HOME : VTAB 22: INVERSE : IF VAL (R$) = N THEN PR
    INT "EXACT!": GOTO 280
270 PRINT "FAUX: ";F:
280 NORMAL : PRINT "      PRESSEZ UNE TOUCHE S.V.P"
290 GOSUB 370
300 POKE - 16302,0: POKE - 16299,0
310 GOSUB 370: GOSUB 380
320 PRINT : HOME : VTAB 22: CALL - 198: PRINT "<1> EN
    CORE <2> FIN <3> MENU DISQUETTE " : GET R$
330 IF R$ = "3" THEN TEXT : PRINT : PRINT CHR$(4)*RU
    N MENU"
340 IF R$ = "1" THEN RUN
350 IF R$ < > "2" THEN 320
    
```

```

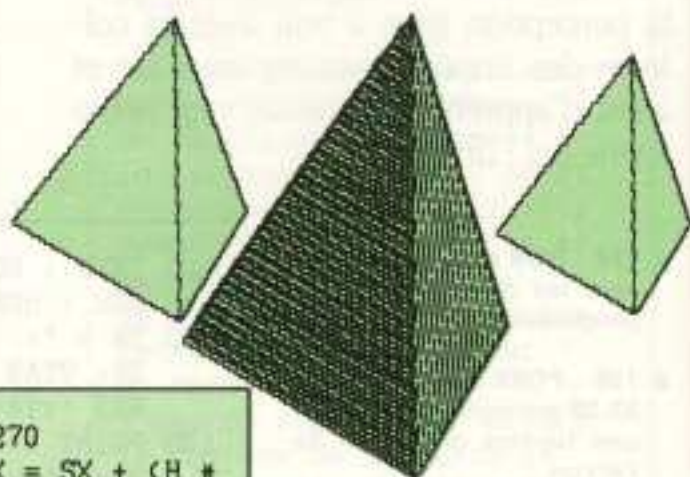
360 HOME : TEXT : END
370 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127:
    POKE - 16368,0: RETURN
380 POKE - 16304,0: POKE - 16300,0: POKE - 16297,0: P
    OKE - 16301,0: RETURN
390 FOR I = 768 TO 781: READ D: POKE I,D: NEXT : POKE
    232,0: POKE 233,3: RETURN
400 DATA 1,0,4,0,18,63,36,36,45,45,54,54,63,0

```

- 370 : Attente sans curseur.
- 380 : Affichage de la page 1 avec 4 lignes de texte.
- 390 : Initialisation de la table de forme (\$E8, \$E9).
- 400 : Forme (carré).

PYRAMIDE

Construisez des pyramides... à partir des paramètres que vous propose NESTOR.



```

100 TEXT : PRINT CHR$ (21): HOME : GOTO 270
110 CX = SX + (H * 0.32):CY = H * 0.66:BX = SX + (H *
    0.14):BY = H:AX = SX - (H * 0.45):AY = H * 0.77:
    RETURN
120 H PLOT SX,SY TO CX,CY TO BX,BY TO AX,AY TO SX,SY T
    O BX,BY: RETURN
130 SX = 55:SY = 10:H = 115: GOSUB 110: GOSUB 120
140 SX = 230:SY = 5:H = 95: GOSUB 110: GOSUB 120
150 SX = 145:SY = 0:H = - 1
160 H = H + 2: GOSUB 110
170 HCOLOR= 2: H PLOT CX,CY TO BX,BY: HCOLOR= 3: H PLOT
    BX,BY TO AX,AY TO SX,SY
180 IF H <= 157 THEN 160
190 GOSUB 120
200 VTAB 24: HTAB 10: PRINT "PRESSEZ UNE TOUCHE SVP":
    : VTAB 22: PRINT ""
210 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127:
    POKE - 16384,0
220 VTAB 24: HTAB 1: PRINT "<M>ENU DE DISQUETTE <E>NC
    ORE <T>ERMINE":
230 VTAB 19: PRINT : CALL - 198: GET R$: IF R$ = "T"
    THEN TEXT : HOME : END
240 IF R$ = "M" THEN PRINT : PRINT CHR$ (4)"RUN MENU"
250 IF R$ = "E" THEN 100
260 GOTO 220
270 POKE 8192,0:A$ = "2001<2000.3FFEM D823G": FOR I =
    1 TO LEN (A$): POKE 511 + I, ASC ( MID$ ( A$,I,1)
    ) + 128: NEXT : POKE 72,0: CALL - 144
280 HCOLOR= 3: HGR
290 VTAB 22: INVERSE : HTAB 14: PRINT " LA PYRAMIDE "
    : NORMAL : GOTO 130

```

- 110 : Paramètres totalement arbitraires, mais correspondant à une perspective de base exacte.
- 130 : SX et SY sont les coordonnées du sommet (55 pts à partir de la gauche et 10 à partir du haut de l'écran). H est la hauteur. Même chose aux lignes 140 et 150.
- 270 : Ce système permet d'effacer la page graphique, sans que cela soit visible. HGR produit le même effet, mais on a le temps de voir le contenu de l'écran avant l'effacement. Quand HGR a déjà été utilisé, l'effacement peut aussi être obtenu par : HCOLOR=0 : H PLOT 0,0 : CALL - 3082 (si on utilise HCOLOR=3, on colore l'écran).

ALLUMÉ ou ÉTEINT ?

Comment connaître l'état d'un point de l'écran ?

RAPPEL DE NOTIONS CONNUES

l'adresse 8192 à 16383 (\$2000-\$3FFF), la page HGR2 de 16384 à 24575 (\$4000-\$5FFF). Chaque point de l'écran (pixel) correspond à un octet de la mémoire. Cet octet est lui-même divisé en 8 bits.

Ainsi, la ligne 0 de votre écran HGR est constituée de $7 \text{ bits} \times 40 = 280 \text{ bits}$, points ou pixels. Mais où sont donc passés les 8^e bits ? Chacun d'eux est tout simplement utilisé pour déterminer la couleur des 7 autres. C'est relativement complexe... et clair à la fois. Qu'on en juge !

OÙ INTERVIENT LA COULEUR

est alors intéressant de consulter un tableau comme celui-ci

Quand un bit est à 0, la couleur est le noir. Le point est éteint et on n'y voit goutte. Quand il est à 1, le point est allumé... mais ça ne nous renseigne pas sur la couleur. Il

POINT ALLUMÉ 1	8 ^e BIT à :	COLONNE PAIRE	COLONNE IMPAIRE
	0	VIOLET	VERT
	1	BLEU	ROUGE

OÙ EST PASSÉ LE BLANC ?

fantin ? D'accord. Alors, expliquez-moi comment faire pour obtenir un écran uniformément bleu ?

Patience ! on y arrive. Quand deux bits consécutifs (sur une même rangée horizontale, si vous préférez) sont à 1, ils sont obligatoirement blancs. Pas d'exception ! Ça vous paraît enfantin ?

Réponse : Il faut que le 8^e bit soit à 1, que chaque bit des colonnes paires soit aussi à 1 et que, par contre, chaque bit des colonnes impaires soit à 0. En binaire, cela nous donne : $11010101 = \$D5$. Si, avec les mêmes données, le 8^e bit est à 0, on aura du violet, soit, en binaire $01010101 = \$55$.

TABLEAU DES COULEURS

0	NOIR	\$00	00000000	4	NOIR	\$80	10000000
1	VERT	\$2A	00101010	5	ORANGE	\$AA	10101010
2	VIOLET	\$55	01010101	6	BLEU	\$D5	11010101
3	BLANC	\$7F	01111111	7	BLANC	\$FF	11111111

HCOLOR

Cette instruction fixe la couleur Haute Résolution. Si vous tapez $HCOLOR = 4$, la couleur choisie est noire et l'adresse \$E4 (228) va contenir la valeur \$80 (128), soit 10000000 en binaire. Les 7 bits affichés par un $HPlot x,y$ sont nuls : le point va donc rester noir. On obtiendrait un résultat identique avec $HCOLOR = 0$.

(Suite page 20)

COLOR.DÉMO

```
100 TEXT : HOME : HGR
110 PRINT : VTAB 21: CALL - 958: PRINT
    "NUMERO DE COULEUR (0-7) -> " : : G
    ET CH$
120 IF CH$ = CHR$(27) THEN TEXT : HOM
    E : END
130 C = VAL (CH$): IF C > 7 THEN 110
140 RESTORE : FOR I = 0 TO C: READ C$,
    H$,B$: NEXT : HCOLOR= C
150 PRINT : VTAB 23: PRINT C$ "H$" "B
    $" :
160 IF C = 1 OR C = 3 OR C = 5 OR C =
    7 THEN 180
170 X = 0:X1 = 50:P = 0:P1 = 50:P$ = "
    COLONNES PAIRES": GOTO 190
180 X = 50:X1 = 100:P = 51:P1 = 100:P$
    = " COLONNES IMPAIRES"
190 FOR I = 1 TO 30: H PLOT X,P TO X1,P
    1:X = X + 1:X1 = X1 + 1: NEXT
200 IF C < > 0 AND C < > 3 AND C < > 4
    AND C < > 7 THEN PRINT P$ " ";
210 FOR I = 11 TO 19: H PLOT I,0 TO I +
    100,100: NEXT
220 CALL - 198: GET R$: GOTO 110
230 DATA NOIR,$00,00000000,VERT,$2A,00
    101010,VIOLET,$55,01010101,BLANC,$
    7F,01111111,NOIR,$80,10000000,ROUG
    E,$AA,10101010,BLEU,$D5,11010101,B
    LANC,$FF,11111111
```

COLOR.VAL

```
100 TEXT : HOME
110 PRINT "VALEUR DES DIFFERENTES COUL
    EURS: -----
    -----": PRINT
120 FOR I = 0 TO 7: HCOLOR= I: READ C$
    ,H$,B$: INVERSE : PRINT I: : NORMAL
    : PRINT " "C$: TAB( 10)H$:" "B$"
    " : PEEK (228): PRINT : NEXT
130 DATA NOIR,$00,00000000,VERT,$2A,00
    101010,VIOLET,$55,01010101,BLANC,$
    7F,01111111,NOIR,$80,10000000,ROUG
    E,$AA,10101010,BLEU,$D5,11010101,B
    LANC,$FF,11111111
```

FACULTATIFS Ces mini-programmes peu-
vent vous aider à mieux
comprendre le comportement de votre Apple...
face à la couleur. Notez que le rouge désigne
improprement l'orange. ■

POKE 228,128 ou POKE 228,0 provoquerait
le même effet. Vous pouvez lire la couleur
en cours, à n'importe quel moment, en mode
direct ou dans un programme, en tapant
PRINT PEEK(228)

ou C = PEEK(288) : PRINT C.

Si vous essayez d'éclairer un point avec une
couleur ne pouvant pas s'afficher sur la
colonne indiquée, vous n'obtiendrez rien. On
comprend facilement, en consultant notre
petit tableau qu'il n'est pas possible d'éclairer
certains points avec certaines couleurs.
Vous en aurez la preuve en tapant et en exécutant
ce court programme :

```
10 HCOLOR=1 : REM VERT
20 HGR
30 H PLOT 2,0 : REM COLONNE PAIRE
RUN... Il ne se passe rien
30 H PLOT 1,0 : REM COLONNE IMPAIRE
RUN... Le point est affiché
```

EN MODE HGR, UNE COULEUR NE PEUT
ÊTRE REPRODUITE QUE TOUTES LES DEUX
COLONNES.

Comment, dans ces conditions, obtenir un
bon tracé ? Trois solutions :

- ① Utiliser le blanc... qui fonctionne partout !
- ② Sélectionner la colonne en fonction de la couleur
- ③ Doubler l'épaisseur du tracé pour toutes les lignes verticales.

Ne pas oublier, en tout cas, que l'Apple va
automatiquement transformer en un double
point blanc... deux points allumés contigus
sur une même ligne horizontale...

quelle que soit leur couleur d'origine.

Les effets obtenus sont rarement heureux,
notamment quand on trace les fameux
camemberts, chers aux amateurs de
graphiques.

On croit pouvoir disposer de 8 couleurs...
mais avec ses deux noirs et ses deux blancs,
notre palette n'en compte déjà plus que 6.

En fait, avec les écrans trichromes, on n'utilise
que les trois couleurs fondamentales
(bleu, rouge et jaune) ou leurs complémentaires
(orange, vert et violet).

Nous n'allons pas nous plaindre... d'autant
moins que, en Double haute résolution, la
palette de l'Apple présente bel et bien 16
couleurs... mais c'est une autre histoire !



AUTORISÉ

①



NON AUTORISÉ

②

120 : Nettoyage préalable de la page HGR... puis HGR.

140 : Y1 = Sommet de la forme plus 1

Y2 = Base de la forme

160 : On peut utiliser une autre couleur (si vous essayez avec HCOLOR=4, vous modifierez légèrement le contour... sans rien colorer, évidemment).

240-250 : Si vous êtes curieux, tournez la page et vous aurez l'explication de ces deux lignes !

COLOSER UNE FIGURE

Si la coloration d'un polygone régulier... ou d'un cercle ne présente pas de difficulté particulière, il n'en va pas de même pour une figure plus ou moins biscornue.

Nous vous proposons ici une solution valable pour colorer une figure isolée obéissant à certains critères (impératifs).

- Elle ne doit pas s'afficher au-delà de la colonne 255 (la routine est évidemment modifiable, mais n'autorise pas de dépassement sous sa forme actuelle).
- Aucun de ses points ne doit être plus haut que le dernier point de la ligne précédente (cela vous paraît compliqué, mais les graphiques ci-contre vous donnent un exemple de "bonne" (1) et de "mauvaise" (2) figures).

COLOR.HGR < CLEMENT RENARD >

```

100 TEXT : HOME : PRINT CHR$(4)*PR#3
110 GOSUB 230
120 POKE 230,32: CALL 62450: HGR : HCOLOR= 3
130 HPLLOT 20,0 TO 120,40 TO 70,50 TO 95,80 TO 255,140
    TO 5,140 TO 40,70 TO 20,0
140 X = 0:Y1 = 1:Y2 = 140
150 POKE 6,Y1: POKE 7,X: POKE 8,0: POKE 25,Y2
160 HCOLOR= 2: REM PAR EXEMPLE
170 CALL 768: CALL - 198
180 VTAB 23: PRINT "<1> MENU DE DISQUETTE <2> LIST <3>
    > FIN *;; GET R$
190 PRINT : HOME : TEXT : IF R$ = "3" THEN END
200 IF R$ = "1" THEN PRINT CHR$(4)*"RUN MENU"
210 IF R$ = "2" THEN HOME : LIST : PRINT
220 GOTO 180
230 FOR I = 768 TO 838: READ R: POKE I,R: NEXT : RETU
    RN
240 DATA 169,0,133,9,32,55,3,240,6,230,9,230,7,208,24
    5,165,9,240,248,201,5,176,19,32,55,3,208,14,165,6
    ,166,7,164,8,32,87,244
250 DATA 230,7,208,238,96,169,0,133,7,230,6,165,6,197
    ,25,144,202,96,165,6,166,7,164,8,32,17,244,177,38
    ,37,48,41,127,96
    
```

COLOR-HGR les lignes 240-250 de la page précédente

0300-	A9 00	LDA	£\$00] 09 est un pointeur : il est à 0 tant que l'on n'a pas rencontré un point allumé. GOSUB le sous-programme de lecture d'écran.
0302-	85 09	STA	\$09	
0304-	20 37 03	JSR	\$0337	

CARREFOUR

0307-	F0 06	BEQ	\$030F] Si A = 0 saut à \$30F, sinon on incrémente les pointeurs 09, ainsi que 07 pour avancer d'une colonne, puis retour à la lecture d'écran via \$304.
0309-	E6 09	INC	\$09	
030B-	E6 07	INC	\$07] Le pointeur 09 est-il à 1 ? Si non, encore un tour ! Est-il plus grand que 5 ? Si oui saut. Si non, GOSUB lecture d'écran Si A = 1 on a rencontré une limite droite de la forme.
030D-	D0 F5	BNE	\$0304	
030F-	A5 09	LDA	\$09	
0311-	F0 F8	BEQ	\$030B	
0313-	C9 05	CMP	£\$05	
0315-	B0 13	BCS	\$032A	
0317-	20 37 03	JSR	\$0337	
031A-	D0 0E	BNE	\$032A	

H PLOT COLORE

031C-	A5 06	LDA	\$06] Si A=0, on va le colorer grâce à H PLOT (après avoir chargé la bonne adresse dans les 3 registres).
031E-	A6 07	LDX	\$07	
0320-	A4 08	LDY	\$08] Plus un point. Si on n'est pas à zéro, encore un tour.
0322-	20 57 F4	JSR	\$F457	
0325-	E6 07	INC	\$07	
0327-	D0 EE	BNE	\$0317	
0329-	60	RTS		

NOUVELLE LIGNE

032A-	A9 00	LDA	£\$00] Pointeur 07 remis à zéro pour lire une nouvelle ligne
032C-	85 07	STA	\$07	
032E-	E6 06	INC	\$06] +1 pour le nombre de lignes et comparaison avec la valeur-limite contenue dans \$19
0330-	A5 06	LDA	\$06	
0332-	C5 19	CMP	\$19	
0334-	90 CA	BCC	\$0300	Suite ou fin
0336-	60	RTS		

LECTURE D'ÉCRAN HGR

0337-	A5 06	LDA	\$06] Après avoir placé Y1 (ligne à lire) dans l'accumulateur et X (colonne) dans les registres X (partie basse) et Y (partie haute), on laisse à HPOSN (\$F411) le soin de faire tout le travail. A la sortie, si A est à 1, le point est allumé. S'il est à zéro, il est éteint.
0339-	A6 07	LDX	\$07	
033B-	A4 08	LDY	\$08	
033D-	20 11 F4	JSR	\$F411	
0340-	B1 26	LDA	(\$26), Y	
0342-	25 30	AND	\$30	
0344-	29 7F	AND	£\$7F	
0346-	60	RTS		

REMARQUE

Cette routine est perfectible, d'abord parce qu'elle n'est pas efficace dans tous les cas de figures, ensuite parce qu'elle ne sait pas colorer une forme au-delà de la 255^e colonne. C'est la modification la plus facile (les deux pointeurs existent : \$7 et \$8... et il suffit d'incrémenter le second quand le premier dépasse \$FF, ce qui ne pose pas de problème particulier).

Néanmoins, telle qu'elle se présente, la routine constitue à mon avis, une bonne approche du "coloriage automatique" des formes. A vous de jouer !

G.-H.

GRAPH

par NESTOR

CET utilitaire présente l'avantage (non négligeable) de reprendre des routines et polices de caractères déjà parues dans *Tremplin Micro*.

On peut aussi utiliser les fontes classiques de la disquette Apple DOS TOOL KIT (DOS 3.3).

Une courte routine en langage machine accélère considérablement l'affichage du graphe (texte notamment).

La première reprend le même principe que "Colorer une figure" (page 21).

GRAPH se compose de : *GRAPH* (programme en Basic), *GRAPH.LM* (routine installée à l'adresse \$300), *GRAPH.IMP* (à reprendre dans le n°6 de *Tremplin Micro*, sous le titre "IMP.HGR1"), *GRAPH.F* (fonte partielle de caractères).

Une routine *GRAPH.AIDE* permettra aux néophytes de récupérer et reloger automatiquement *GRAPH.IMP* et *GRAPH.F*.

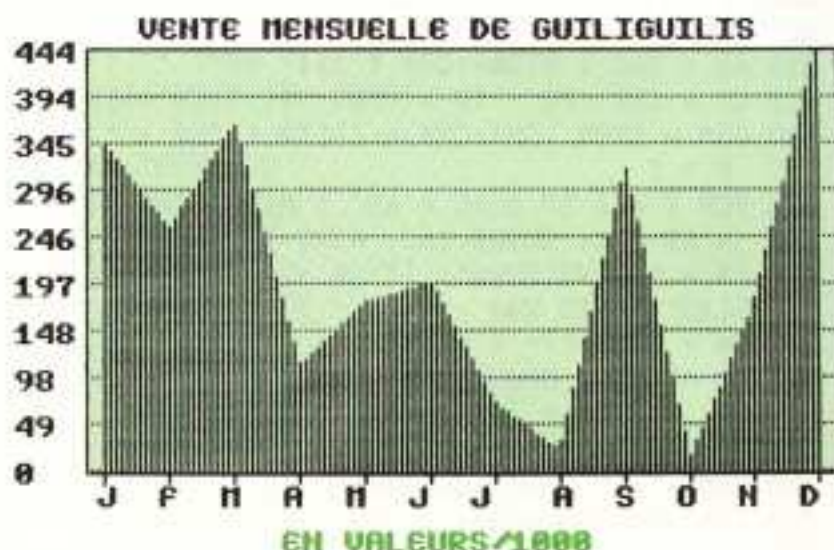
105 : LOMEM est fixé à 16384 pour protéger la page graphique.

125 : ONERR GOTO renvoie au menu.

Pour accélérer le tracé du graphique, on a placé le sous-programme de tracé en tête. Dans la version finale, il est conseillé d'éliminer les lignes de REM.

UNE NOUVELLE VERSION

POUR CHIFFRES EN DENTS DE SCIE



```

100 TEXT : HOME : PRINT CHR$(21)
105 LOMEM: 16384
110 DIM V(13),VF(13):D$ = CHR$(4)
115 PRINT D$"BLOAD GRAPH.F,A$1D4C": REM 7500
120 PRINT D$"BLOAD GRAPH.LM": PRINT D$"BLOAD GRAPH.IM
    P,A7250"
125 ONERR GOTO 595
130 GOTO 585: REM MENU
135 :
140 REM *****
145 REM * LES CHIFFRES ET LETTRES *
150 REM *****
155 :
160 POKE 8,AC: POKE 254,AD - INT (AD / 256) * 256: PO
    KE 255, INT (AD / 256)
165 CALL 829: RETURN
170 L = LEN (R$):AD = AD + INT (21 - L / 2)
175 FOR I = 1 TO L:AC = ASC ( MID$ (R$,I,1)) - 31
180 AD = AD + 1: GOSUB 160: NEXT I: RETURN
185 R$ = TI$:AD = 8191: GOSUB 170
  
```

GRAPH (BASIC)

SUITE

```
190 FOR I = 1 TO 12:AC = ASC ( MID$ ("JFMAMJJASOND".I
,1)) - 31:AD = 8785 + (I * 3.125): GOSUB 160: NEX
T
195 R = 0:DX = 1:CA$ = "83198575883190878359861588719
12783998655"
200 IF PG / DX > 999 THEN DX = DX * 10: GOTO 200
205 FOR I = 1 TO 40 STEP 4
210 AB = VAL ( MID$ (CA$,I,4))
215 CH = INT (PG - (R * (PG / 9))):CH = INT (CH / DX)
220 CH$ = STR$ (CH):CH$ = LEFT$ ((CH$ + "   " ),3):R =
R + 1
225 FOR K = 1 TO 3:AC = ASC ( MID$ (CH$,K,1)) - 31:AD
= AB + K: GOSUB 160: NEXT K: NEXT I
230 R$ = "EN VALEURS": IF DX = 1 THEN R$ = R$ + " REE
LLES": GOTO 240
235 R$ = R$ + "/*" + STR$ (DX)
240 AD = 9039: GOSUB 170: RETURN
245 :
250 REM *****
255 REM *   TRACE DU GRAPHIQUE   *
260 REM *****
265 :
270 IF NOT ND THEN R$ = " SAISIR LES DONNEES ": GOTO
365
275 IF NOT M THEN 285
280 IF M THEN POKE 230,32: CALL 62450
285 POKE 49236,0: POKE 49234,0: POKE 49239,0: POKE 49
232,0: HCOLOR= 3
290 IF NOT M THEN 355
295 HPLLOT 24,10 TO 24,155 TO 279,155 TO 279,10 TO 24,
10
300 IM = 30: HPLLOT IM,156 TO IM,159:D = 1
305 FOR I = 52 TO 279 STEP 22:D = D + 1
310 A = 154 - VF(D):A1 = 154 - VF(D - 1)
315 IF A1 = 155 THEN A1 = 154
320 IF A = 155 THEN A = 154
325 HCOLOR= 3: HPLLOT I,158 TO I,156
330 HCOLOR= 2: HPLLOT IM,A1 TO I,A
335 IM = I: NEXT
340 HCOLOR= 2: CALL 768
345 FOR I = 26 TO 154 STEP 16: HPLLOT 24,I TO 278,I: N
EXT
350 NG = 1: GOSUB 185
355 M = 0: CALL - 198: GET R$
360 PRINT : TEXT : GOTO 595
365 VTAB 21: CALL - 958: CALL - 198: HTAB 11: INVERSE
: PRINT R$: NORMAL : FOR I = 1 TO 2000: NEXT : G
OTO 595
370 :
375 REM *****
```

160 : POKE 8,AC indique le n° de caractère à la routine en langage machine.

L'adresse-écran où doit se faire l'affichage est elle-même poquée en 254 et 255 (\$FE et \$FF), toujours à l'intention de la routine en L.M.

165 : La routine L.M. commence à cette adresse.

190 : On peut imaginer de remplacer les initiales des mois par autre chose...

195 : Les adresses des lignes de la page HGR sont contenues dans la variable CA\$. On aurait pu les mettre en DATA.

De 185 à 240 : affichage de tout ce qui est texte.

270 : Quand ND=0, pas de données en mémoire.

280 : Effacement de la page HGR.

285 : Affichage de la page sans les 4 lignes de texte... et sans effacement.

Suivez pas à pas le tracé du graphique. Ce n'est pas très compliqué... et de longues explications paraissent superflues.

340 : CALL 768 envoie à la routine qui va colorer le graphique (avec HCOLOR=2).

Nous avons supprimé le tracé (en blanc) des dents de scie, mais il peut être rétabli :

330 : HPLLOT IM,A1 TO I,A
(Supprimer HCOLOR=2)

En cours de saisie, il est possible de revenir au menu en tapant un ?. Pour corriger la donnée précédente, taper l' *.

Si la valeur est bonne, se contenter d'un RETURN.

Le programme accepte les valeurs 0, mais vous vous planterez proprement s'il n'y a pas au moins une valeur supérieure à 0.

530 : Si le titre est bon, un simple RETURN suffira.

Attention ! on ne contrôle pas la longueur de ce fameux titre. Ne dépassez pas 39 caractères... ou modifiez le programme :

(531 IF LEN(TI\$) > 39 THEN 530).

Le signe §, après INPUT, indique la présence d'un CTRL-G... invisible. Ne tapez pas §, mais CTRL-G.

585-590 : Comptez bien vos espaces... sinon gare aux dégâts !

595 : L'utilisation des POKES évite HTAB.

```

380 REM *      SAISIE DES DONNEES      *
385 REM *****
390 :
395 TEXT : HOME : I = 0 : M = 0
400 I = I + 1
405 V$ = " " : IF I > 9 THEN V$ = ""
410 VTAB 3 + I : PRINT V$I". VALEUR -> "V(I):
415 HTAB 15: INPUT "":V$
420 IF V$ = "?" THEN 450
425 IF V$ = "*" AND I > 1 THEN I = I - 2: GOTO 400
430 IF V$ < > "" THEN V(I) = VAL (V$)
435 VTAB 3 + I: HTAB 15: PRINT V(I)
440 M = 1
445 IF I < 12 THEN 400
450 PP = V(1):PG = PP
455 FOR I = 1 TO 12
460 IF V(I) > PG THEN PG = V(I)
465 IF V(I) < PP THEN PP = V(I)
470 NEXT
475 :
480 IF PG < 144 THEN CO = 144 / PG:P = 1
485 IF PG > = 144 THEN CO = PG / 144:P = 2
490 FOR I = 1 TO 12
495 ON P GOTO 500,505
500 VF(I) = INT (V(I) * CO): GOTO 515
505 IF V(I) = 0 THEN VF(I)=0:GOTO 515
510 VF(I) = INT (V(I) / CO)
515 NEXT :ND = 1
520 IF LECT THEN LECT = 0: GOTO 595
525 :
530 VTAB 20: INVERSE : PRINT "TITRE:": NORMAL : PRINT
: VTAB 22: PRINT TI$: HTAB 1: INPUT "§":R$: IF
R$ < > "" THEN TI$ = R$
535 VTAB 20: CALL - 958: PRINT TI$
540 VTAB 22: PRINT "<1> ENREGISTRER LES DONNEES <0> N
ON " : CALL - 198: GET R$
545 PRINT : IF R$ = "0" THEN 595
550 IF R$ < > "1" THEN 540
555 GOTO 695
560 :
565 REM *****
570 REM *      MENU      *
575 REM *****
580 :
585 T$ = " : " : TMS = " M E N
U "
590 M$ = "1. LECTURE DE FICHIER 2. SAISIE/CORRECTI
ON 3. VOIR LE GRAPHIQUE 4. IMPRIMER LE GR
APHIQUE 5. FIN DE PROGRAMME"
595 HOME : POKE 32,10: POKE 33,20: VTAB 5: INVERSE :
PRINT T$TMS$: NORMAL : POKE 32,8: POKE 33,32: PR
INT
600 FOR I = 1 TO LEN (M$) STEP 25: PRINT MID$ (M$,I,2

```

GRAPH (BASIC)

SUITE et FIN

```
5): PRINT : NEXT
605 POKE 32,0: POKE 33,40
610 PRINT : VTAB 21: HTAB 11: INVERSE : PRINT *OPTION
    CHOISIE:      ":: NORMAL : HTAB 27: CALL - 198: P
    RINT "> <";: HTAB 28: GET R$
615 R = ASC (R$) - 48: IF R < 1 OR R > 5 THEN 610
620 PRINT R$
625 ON R GOTO 750,395,270,655,790
630 :
635 REM *****
640 REM *   IMPRESSION DU GRAPHIQUE *
645 REM *****
650 :
655 IF NOT NG THEN R$ = " OPTION GRAPH S.V.P *": GOTO
    365
660 POKE 6,1: POKE 7 1: POKE 253,0: POKE 254,23
665 POKE 49236,0: POKE 49234,0: POKE 49239,0: POKE 49
    232,0: CALL 7250: HOME : TEXT : GOTO 595
670 :
675 REM *****
680 REM *   ECRITURE DU FICHER *
685 REM *****
690 :
695 VTAB 20: CALL - 958: PRINT "TITRE -> GRAPH0 (PA
    R DEF AUT)": HTAB 10: INPUT "": F$: IF F$ > CHR$ (
    47) AND F$ < CHR$ (58) THEN F$ = "GRAPH" + F$
700 IF F$ = "?" THEN HOME : PRINT D$"CATALOG": PRINT
    : PRINT : PRINT : GOTO 695
705 IF F$ = "" THEN F$ = "GRAPH0"
710 VTAB 20: HTAB 10: INVERSE : PRINT F$: NORMAL
715 IF LECT THEN RETURN
720 PRINT D$"OPEN" F$: PRINT D$"WRITE" F$: PRINT TI$: F
    OR I = 1 TO 12: PRINT V(I): NEXT : PRINT D$"CLOSE
    " F$: GOTO 585
725 :
730 REM *****
735 REM *   LECTURE D'UN FICHER *
740 REM *****
745 :
750 HOME :LECT = 1: GOSUB 695
755 PRINT D$"OPEN" F$: PRINT D$"READ" F$: INPUT TI$: FO
    R I = 1 TO 12: INPUT V(I): NEXT : PRINT D$"CLOSE"
    F$
760 M = 1: GOTO 450
765 :
770 REM *****
775 REM *   FIN DE PROGRAMME *
780 REM *****
785 :
790 TEXT : HOME
795 VTAB 20: PRINT "$GOTO10000": VTAB 19: END
    10000 POKE 51,0: GOTO 595
```

Rappelons que la routine d'impression doit être reprise dans le n°6 de *Tremplin Micro (IMP.HGR1)*.

Utilisez notre programme *GRAPH.AIDE* pour la reloger (page 28). Ce sera plus facile car il y a quelques adresses à modifier.

Cette routine ne fonctionne qu'avec les imprimantes Apple, y compris ImageWriter II (voir configuration des DIPS dans T.M. 6).

695-715 : Tronc commun pour écriture et lecture du fichier. Le ? renvoie au catalogue de disquette.

Les fichiers s'appellent "GRAPH0,1,2, etc."

Pour obtenir GRAPH0 : RETURN

Pour un autre numéro, taper seulement le numéro + RETURN

En cas de pépin, GOTO 10000 vous renverra au menu

ProDOS Ne pas oublier de transformer *CATALOG* en *CAT*.

De plus, modifier ainsi la ligne 665 : Entre CALL 7250 et HOME, insérer : NOTRACE : PRINT :

GRAPH.LM

Après avoir tapé les codes de cette courte routine, vous la sauvez par un :

BSAVE GRAPH.LM, A\$300, L\$79

COLORER LE GRAPHE

0300-	A9 1C	LDA	£\$1C] Position horizontale (-2) du premier point à lire
0302-	85 07	STA	\$07	
0304-	A9 00	LDA	£\$00] 0 dans mémoire 08 (position horizontale, partie haute)
0306-	85 08	STA	\$08	
0308-	A9 9B	LDA	£\$9B] Limite verticale (+1) de lecture (on lira de 155 à ...)
030A-	85 06	STA	\$06	
030C-	E6 07	INC	\$07] +2 pour la partie basse
030E-	E6 07	INC	\$07	
0310-	D0 02	BNE	\$0314] Si on n'a pas obtenu 0 (\$FE+1+1), saut
0312-	E6 00	INC	\$08	
0314-	A5 07	LDA	\$07] Sinon, plus 1 pour partie haute
0316-	C9 12	CMP	£\$12	
0318-	D0 05	BNE	\$031F] 07 dans l'accumulateur
031A-	A5 08	LDA	\$08	
031C-	F8 01	BEQ	\$031F] Est-on à \$12 ?
031E-	60	RTS		
031F-	C6 06	DEC	\$06] Non : on continue
0321-	A5 06	LDA	\$06	
0323-	A6 07	LDX	\$07] Oui : \$8 est-il à 1 (\$112=274, limite horizontale de lecture) ?
0325-	A4 08	LDY	\$08	
0327-	20 11 F4	JSR	\$F411] Si \$8 est à zéro, on continue
032A-	B1 26	LDA	(£26), Y	
032C-	25 30	AND	\$30] Retour au Basic
032E-	29 7F	AND	£\$7F	
0330-	F8 ED	BEQ	\$031F] Compte à rebours vertical et mémoire \$6 dans A
0332-	A5 07	LDA	\$07	
0334-	A6 08	LDX	\$08] X et Y chargés avec l'adresse (basse et haute) du point à lire
0336-	A0 9A	LDY	£\$9A	
0338-	20 3A F5	JSR	\$F53A] HPOSN travaille pour nous. Sa réponse sera dans l'accumulateur. On aura 0 si le point est éteint et 1 s'il est allumé
033B-	4C 08 03	JMP	\$0308	

ECRITURE HGR

033E-	A9 44	LDA	£\$44] Adresse de base de la police de caractères (\$1D4C — 8 = \$1D44)
0340-	85 06	STA	\$06	
0342-	A9 1D	LDA	£\$1D] Numéro du caractère dans le registre Y
0344-	85 07	STA	\$07	
0346-	A4 08	LDY	\$08] Retenue à 0 avant l'addition
0348-	18	CLC		
0349-	A5 06	LDA	\$06] Adresse de base (partie basse) dans A
034B-	69 08	ADC	£\$08	
034D-	85 06	STA	\$06] +8 (nombre d'octets par caractère)
034F-	A5 07	LDA	\$07	
0351-	69 00	ADC	£\$00] Nouvelle valeur dans mémoire \$6
0353-	85 07	STA	\$07	
0355-	88	DEY] Adresse de base (partie haute) dans A
0356-	D0 F1	BNE	\$0349	
0358-	B1 06	LDA	(£06), Y] On lui ajoute la retenue éventuelle et on la remet dans \$7
035A-	91 FE	STA	(£FE), Y	

035C-	A2 00	LDX	£\$00	X à 0 pour écriture adressage indirect
035E-	C8	INY		Y = Y + 1
035F-	A5 FF	LDA	\$FF] Adresse-écran (partie haute) dans A pour lui ajouter \$4... ce qui correspond à + 1024 (\$400), puis remise dans \$FF.
0361-	69 04	ADC	£\$04	
0363-	85 FF	STA	\$FF] Lecture adresse caractère + Y et écriture adresse-écran + 0 (x=0)
0365-	B1 06	LDA	(<\$06),Y	
0367-	81 FE	STA	(<\$FE,X)	
0369-	C0 07	CPY	£\$07] Si Y n'est pas à \$7, on fait encore un tour
036B-	D0 F1	BNE	\$035E	
036D-	60	RTS		Retour au BASIC

GRAPH.AIDE

Si vous êtes rompu à la pratique du Basic, ne tapez pas ce programme. Une lecture attentive vous indiquera les diverses opérations à effectuer pour mettre rapidement au point

une fonte spéciale (GRAPH.F) et une routine d'impression (GRAPH.IMP)... Par contre, si vous manquez d'expérience, utilisez GRAPH.AIDE !

```

100 TEXT : PRINT CHR$(21):D$ = CHR$(4): HOME
105 GOSUB 205
110 PRINT "PLACEZ DISKETTE DOS TOOL KIT DANS VOTRE SLOT 1, PUIS FRAPPEZ UN S
    IMPLÉ (RETURN)"
115 GOSUB 180:F$ = "FLOW.SET"
120 ONERR GOTO 195
125 PRINT D$"BLOAD "F$",A7500"
130 GOSUB 190
135 PRINT D$"BSAVE GRAPH.F,A7500,L512"
140 HOME : PRINT "INSEREZ LA DISQUETTE (TREPLIN MICRO 6) (OU CELLE SUR LAQU
    ELLE FIGURE UNE COPIE DU PROGRAMME (IMP.HGR1)) DANS LE SLOT 1"
145 GOSUB 180
150 F$ = "IMP.HGR1"
155 PRINT D$"BLOAD "F$",A7250"
160 GOSUB 190
165 POKE 7432,133: POKE 7433,28: POKE 7292,24: POKE 7293,29: POKE 7305,48: P
    OKE 7306,29
170 PRINT D$"BSAVE GRAPH.IMP,A7250,L230"
175 VTAB 22: HTAB 6: INVERSE : PRINT "VOS DEUX ROUTINES SONT AU POINT": NORMA
    L : END
180 PRINT : VTAB 22: PRINT "          RETURN S.V.P. "; GET R$: IF R$ < > CHR
    $(13) THEN 180
185 PRINT : RETURN
190 HOME : VTAB 12: PRINT "PLACEZ MAINTENANT VOTRE DISQUETTE GRAPH DANS LE S
    LOT 1, PUIS PRESSEZ (RETURN)": GOSUB 180: PRINT : RETURN
195 PRINT : HOME : VTAB 12: IF PEEK(222) = 6 THEN PRINT "OU CE N'EST PAS LA
    BONNE DISQUETTE... OUVOUS NE POSSEDEZ PAS (<F$>)": GOTO 310
200 PRINT "IL Y A UNE ERREUR QUELQUE PART: CHERCHEZ"
205 VTAB 10: PRINT " SI VOUS DESIREZ UTILISER LES FONTES: (FAIRE.F1) ou (F
    AIRE.F2) DU NUMERO 1 DE TREPLIN REMPLACEZ (FLOW.SET), LIGNE 115     PA
    R (FAIRE.F1) ou (FAIRE.F2). "
210 VTAB 22: HTAB 6: PRINT "<1> CONTINUER <2> CORRIGER --> "; GET R$: PRINT
    : IF R$ = "1" THEN HOME : RETURN
215 IF R$ < > "2" THEN 210

```

FLOW.SET est, à notre avis, la police TOOLKIT la plus appropriée. Vous pouvez aussi utiliser FAIRE.F1 (Treplins Micro n°1).

Votre bibliothèque INFORMATIQUE

par Clément RENARD

• LA SÉCURITÉ INFORMATIQUE (Dunod)

Tout chef d'entreprise, utilisateur de l'informatique, devrait lire l'ouvrage de Jean-Marc Lamère... et en conseiller une lecture plus attentive encore au responsable du service intéressé.

Les risques existent. Ceux dont on parle, mais aussi les autres, moins connus, mais capables de semer la pagaille dans une entreprise. On sait déjà qu'ils coûtent, chaque année, plusieurs milliards de francs et sont une des causes de la disparition de nombreuses entreprises. L'auteur ne se contente pas d'énumérer les risques, mais propose une méthode pratique d'évaluation et de réduction de ces risques. A lire... ai-je écrit plus haut, mais également à étudier, puis à mettre en œuvre.

regroupant les meilleurs pirates, nous le démontre dans un livre qui est en même temps un guide de la sécurité informatique.

Vous avez bien compris : Bill Landreth, appréhendé par le F.B.I. et mis en accusation, vous livre aujourd'hui ses secrets.

Comment se protéger contre ces modernes "casseurs de systèmes" que sont les pirates de l'informatique ? Comment décourager leurs incursions indésirables... Comment les empêcher d'agir ou encore obtenir leur appui ? Si vous désirez en savoir plus, ou encore protéger votre serveur, lisez ce livre aussi captivant qu'un roman !



• INTELLIGENCE ARTIFICIELLE ET SYSTÈMES EXPERTS (Cedic/Nathan)

On parle beaucoup de l'intelligence artificielle... mais sans trop savoir où en sont les recherches, et en quoi consiste exactement cette nouvelle révolution informatique.

C'est probablement, actuellement en tout cas, l'un des secteurs les plus attractifs, sinon les plus prometteurs, de l'informatique de

demain. Qu'est-ce qu'un système expert ? Quels sont les grands principes déterminant la construction et l'exploitation d'un tel système ? Vous le saurez après avoir lu ce manuel d'Adrien Lescort. Je suis même persuadé que vous éprouverez l'envie irrésistible d'écrire, sur votre Apple, un programme reposant sur les grands principes régissant l'intelligence artificielle.

• SYSTÈME D'EXPLOITATION DES IBM PC (Dunod)

Demain... ou après-demain, votre Apple nouvelle manière vous permettra sans doute de franchir le fossé qui le sépare actuellement du standard IBM.

Gageons que vous aimerez alors mieux connaître le système d'exploitation du jeune frère (il a grandi très vite !). Il n'est d'ailleurs pas tellement différent de notre ProDOS.

G. Fodor, D. Bonifas et G. Tanguy vous aident ici, avec un guide clair et agréable à compiler, à comprendre les différences entre DOS 1.1, 2.0, 3.0, 3.1/PC-IX et XENIX.

Tout un programme !



• LE PIRATE DE L'INFORMATIQUE (Cedic/Nathan)

Nous sommes tous des victimes potentielles du piratage de l'informatique. Le "cracker", capable, à 16 ans, de percer les secrets des ordinateurs les mieux protégés... membre du "Cénacle", réseau

DATA-GÉNÉRATEUR

NOUS avons déjà publié (n°3 de *Tremplin Micro*), un programme générant des lignes DATA ou POKES, mais différent de celui de Nestor, puisqu'il partait de données existantes, déjà installées en mémoire par une routine en assembleur.

DATA-GÉNÉRATEUR permet de créer des lignes DATA au gré de sa fantaisie. Chaque ligne peut être facilement corrigée (voire remaniée), aussi longtemps qu'elle n'a pas été validée par un CTRL-F.

Point fort : quand on désire rentrer, sous forme décimale, une donnée que l'on a en hexa, il suffit de taper un CTRL-G... puis la valeur hexa ; le programme fait le reste. C'est amusant et pratique à la fois.

Autre point fort : il est souvent nécessaire d'inclure un espace au début d'une donnée alphanumérique. Lorsque celui-ci est inclus entre une virgule et la donnée, pas de problème. Par contre, s'il est en tête d'une ligne DATA, immédiatement après le mot "DATA", il ne sera pris en compte qu'en utilisant les guillemets. Exemple : 100 DATA " TOTO", LILI... Bien ! avec DATA.GÉNÉRATEUR, tapez votre espace... et laissez faire votre utilitaire !

PRINCIPALES VARIABLES de DATA.GÉNÉRATEUR

A	Nombre de lignes de DATA
DS	Le CHR\$ (4) bien connu
DA\$	Renferme le mot "DATA"
ES	Comme "espace" : CHR\$ (32)
G\$	Comme "guillemets" : CHR\$ (34)
H	Pour fixer HTAB (correction)
HE	Pointeur pour conversion Hexa-décimal
I	Nombre de données par ligne
L\$	Renferme la donnée en cours de saisie
L	Longueur de la ligne en cours
L1	Longueur de L\$ + S\$
LB	Longueur de base (n° de ligne + "DATA")
L\$	(Lignes créées)
MC	Pointeur pour écran de commandes
N\$ et N	Numéro de ligne
R\$ et R%	Chaque caractère saisi par GET
S\$	Est la virgule : CHR\$ (44)

```

100 TEXT : PRINT CHR$ (21): PRINT : C
    LEAR : GOTO 685
105 :
110 REM *****
115 REM *  ROUTINE DE SAISIE  *
120 REM *****
125 :
130 HOME :S$ = CHR$ (44):E$ = CHR$ (3
    2):DA$ = "DATA":G$ = CHR$ (34):D$
    = CHR$ (4)
135 HTAB 1: VTAB 1: CALL - 198: INPUT
    "NUMERO DE LIGNE ":N$: IF N$ = "
    " THEN N$ = "1000"
140 N = VAL (N$): IF N < 1 OR N > 640
    00 THEN 135
145 HTAB 1: VTAB 1: CALL - 958: INVER
    SE : PRINT "LIGNE " + N$: NORMAL
150 N = N - 1
155 N = N + 1:N$ = STR$ (N):A = A + 1
    : HTAB 15: VTAB 1: CALL - 868: IN
    VERSE : PRINT N: NORMAL :L1 = 0:L
    B = LEN (N$) + 4
160 L$(A) = L$(A) + N$ + DA$: IF S =
    1 THEN S = 0: GOTO 525
165 HTAB 1: VTAB 3: PRINT : CALL - 86
    8:I = 1 + 1: PRINT I: PRINT " >
  
```



```

*::PH = LEN ( STR$ (I)) + 4
170 HTAB PH: VTAB 4: PRINT L$: GET R
$:R% = ASC (R$)
175 IF R% = 7 AND NOT HE AND L$ = ""
THEN INVERSE :HE = 1: CALL - 198:
GOTO 170
180 IF NOT HE THEN 190
185 IF R% = 13 OR LEN (L$) = 4 THEN N
ORMAL : GOSUB 465:R% = 13:HE = 0
190 IF R% = 27 THEN GOSUB 725:MC = 1
195 IF G AND (R% = 13 OR R% = 44) THE
N L$ = L$ + G$:G = 0
200 IF NOT G AND I = 1 AND R% = 32 TH
EN L$ = G$ + E$:G = 1: GOTO 170
205 IF R% > 31 AND R% < 127 THEN L$ =
L$ + R$: GOTO 170
210 IF L$ = "" THEN 225
215 IF R% = 13 THEN 520
220 IF R% = 127 OR R% = 8 THEN L$ = M
ID$ (L$,1, LEN (L$) - 1): CALL -
1008: PRINT E$: GOTO 170
225 IF R% = 4 AND L > LB THEN 280
230 IF R% = 3 AND I < > 1 THEN 360
235 IF L$ < > "" OR L < = LB THEN 170
240 IF R% = 6 THEN 510
245 IF R% = 19 THEN 515
250 GOTO 170
255 :
260 REM *****
265 REM * DEL DERNIERE DONNEE *
270 REM *****
275 :
280 FOR K = L - 1 TO 1 STEP - 1: IF M
ID$ (L$(A),K,1) < > S$ THEN NEXT
285 IF K < LB THEN K = LB + 1
290 L$(A) = LEFT$ (L$(A),K - 1): IF L
EN (L$(A)) > LB THEN L$(A) = L$(A
) + S$
295 I = I - 2:L = LEN (L$(A)): GOTO 3
30
300 :
305 REM * NOMBRE DE DONNEES ? *
310 :
315 IF RIGHT$ (L$(A),1) < > S$ AND LE
N (L$(A)) > LB THEN L$(A) = L$(A)
+ S$
320 L = LEN (L$(A)):X = 0: FOR K = 1
TO L: IF MID$ (L$(A),K,1) = S$ TH
EN X = X + 1
325 NEXT :I = X
330 GOTO 535
335 :

```

```

340 REM *****
345 REM * CORRECTION *
350 REM *****
355 :
360 H = L + 1: IF MC THEN CALL - 958:
MC = 0
365 PRINT : VTAB 18: PRINT L$(A)" *::
L = LEN (L$(A))
370 PRINT : VTAB 18: HTAB H: GET R$:R
% = ASC (R$)
375 IF R% < > 127 THEN 400
380 IF H < = LB + 1 THEN H = LB + 1:
GOTO 370
385 IF H > L + 1 THEN 360
390 IF H = L + 1 THEN L = L - 1:L$(A)
= LEFT$ (L$(A),L): GOTO 360
395 L$(A) = LEFT$ (L$(A),H - 2) + RIG
HT$ (L$(A),L + 1 - H):H = H - 1:
GOTO 365
400 IF R% = 13 THEN L$ = "": GOTO 315
405 IF R% = 8 AND H > L1 THEN H = H -
1: GOTO 370
410 IF R% = 21 AND H < L + 1 THEN H =
H + 1: GOTO 370
415 IF R% = 9 AND L > = H THEN L$(A)
= LEFT$ (L$(A),H - 1) + " " + RIG
HT$ (L$(A),L + 1 - H): GOTO 365
420 IF R% < 32 OR R% > 126 THEN 370
425 IF H > = L THEN L$(A) = L$(A) + R
$:L = L + 1: GOTO 360
430 L$(A) = LEFT$ (L$(A),H - 1) + R$
+ RIGHT$ (L$(A),L - H):H = H + 1:
GOTO 365
435 GOTO 370
440 :
445 REM *****
450 REM * HEXA EN DECIMAL *
455 REM *****
460 :
465 L$ = RIGHT$ (("000" + L$),4):P =
0:Y = 0
470 FOR K = 4 TO 1 STEP - 1:X = ASC (
MID$ (L$,K,1)) - 48: IF X > 9 TH
EN X = X - 7
475 Y = Y + (X * 16 ^ P):P = P + 1: N
EXT
480 L$ = STR$ (Y): RETURN
485 :
490 REM *****
495 REM * DONNEE/LIGNE SUIVANTE *
500 REM *****
505 :

```



DATA-GENERATOR (FIN)

```
510 L$(A) = LEFT$(L$(A),L - 1): GOTO 580
515 L$(A) = LEFT$(L$(A),L - 1):I = 0 : GOTO 155
520 L1 = LEN (L$ + S$): IF L1 + PH > 40 THEN VTAB 4: CALL - 958
525 IF L1 + L > 239 THEN L$(A) = LEFT$(L$(A),L - 1):S = 1:I = 0: GOTO 155
530 L$(A) = L$(A) + L$ + S$:L$ = ""
535 IF MC = 1 THEN MC = 0: PRINT : VTAB 6: CALL - 958
540 PRINT : VTAB 18: PRINT L$(A):: CALL - 958:L = LEN (L$(A)): PRINT
545 VTAB 1: HTAB 37: CALL - 868: INVERSE : PRINT L: NORMAL
550 GOTO 165
555 :
560 REM *****
565 REM * FICHIER EXEC *
570 REM *****
575 :
580 HOME : FOR I = 1 TO A: PRINT L$(I) : NEXT
585 PRINT D$"OPEN F": PRINT D$"WRITE F": FOR I = 1 TO A: PRINT L$(I): NEXT : PRINT D$"CLOSE F"
590 :
595 REM *****
600 REM * MICRO MENU *
605 REM *****
610 :
615 HOME : INVERSE : FOR I = 10 TO 31 : VTAB 10: HTAB I: PRINT " ": VTAB 20: HTAB I: PRINT " ": NEXT
620 FOR I = 11 TO 19: VTAB I: HTAB 10 : PRINT " " : HTAB 31: PRINT " " : NEXT
625 VTAB 13: HTAB 12: PRINT 1: VTAB 15: HTAB 12: PRINT 2: VTAB 17: HTAB 12: PRINT 3: NORMAL
630 VTAB 13: HTAB 14: PRINT "EXEC-EFFACEMENT": VTAB 15: HTAB 14: PRINT "AUTRE SAISIE": VTAB 17: HTAB 14 : PRINT "FIN DE PROGRAMME"
635 VTAB 22: HTAB 10: PRINT "VOTRE CH OIX S.V.P. " : GET R$
640 PRINT :R = VAL (R$): IF R = 1 THEN 745
```

```
645 IF R = 2 THEN 100
650 IF R < > 3 THEN 635
655 HOME : END
660 :
665 REM *****
670 REM * EXPLICATION *
675 REM *****
680 :
685 HOME : HTAB 7: INVERSE : PRINT " GENERATEUR DE LIGNES DATA ": NORMAL : PRINT
690 PRINT : PRINT "CE PROGRAMME PERMET DE GENERER AUTANT DELIGNES DE DATA QU'ON LE DESIRE... POURVUQU'ON NE DEPASSE PAS DIM (50)"
695 PRINT : PRINT "<RETURN> VALIDE CHAQUE DONNEE - POUR UNECORRECTION, TAPER CTRL-C, PUIS UTILISER LES TOUCHES <DEL> POUR SUPPRIMER ET CTRL-I AVANT D'INSERER UN CARACTERE"
700 PRINT : PRINT "POUR EFFACER LA DERNIERE DONNEE: CTRL-D"
705 PRINT : PRINT "LES LIGNES VONT DE 1 EN 1 (DEPUIS LE NU-MERO DONNE AU DEPART). PASSAGE AUTOMATIQUE A LA LIGNE SUIVANTE DES QUE L'ON AT-TEINT 239 CARACTERES."
710 PRINT : PRINT "<ESC> PERMET D'AFFICHER UN RESUME DU TABLEAU DES DIFFERENTES COMMANDES."
715 POKE - 16368,0: WAIT - 16384,128,127: POKE - 16368,0: PRINT
720 PRINT : GOTO 130
725 PRINT : VTAB 10: PRINT "SAISIE: DEL et <- effacent un CARACTERE ----- CTRL-D efface DERNIERE DONNEE CTRL-S = ligne suivante et CTRL-F = FIN CTRL-C = correction dans la ligne DATA RETURN = validation de la DONNEE saisie"
730 PRINT : PRINT "CORRECTION: DEL efface - TAB ou CTRL I ----- ajoutent un espace - RETURN pour terminer."
735 PRINT : PRINT "CONVERTIR: Taper CTRL-G, puis la valeur ----- HEXA (= <$FFFF). "
740 RETURN
745 HOME : HTAB 2: VTAB 23: PRINT "EXEC F": HTAB 1: VTAB 22: DEL 100,745
```

Où CATALOG devient CAT

... et autres corrections mineures

VOUS venez d'écrire (ou de recopier) un programme fonctionnant sous DOS 3.3, et vous l'avez converti pour qu'il s'accommode du système ProDOS. Aucun problème, bien sûr, si ce n'est que la commande CATALOG, lancée en 40 colonnes, ne sera pas du meilleur effet... sauf si vous utilisez la routine CAT.ALLOG pour transformer instantanément les CATALOG de votre programme Basic en autant de CAT.

Il peut aussi y avoir incompatibilité entre certaines adresses du DOS et les adresses correspondantes de ProDOS.

En voulez-vous un exemple ? Après avoir chargé un programme en langage machine, on trouve l'adresse du premier octet en \$AA72-AA73 (43634-43635) et la longueur du programme en \$AA60-AA61 (43616-43617) alors que, sous ProDOS, ces adresses deviennent respectivement : \$BEB9-BEBA (48825-48826) et \$BEC8-BEC9 (48840-48841).

Pour remplacer (toujours instantanément) une valeur par une autre (de même longueur), vous utiliserez REC.REM (comme RECherches-REMPlacer !).

Tapez donc ces deux micro-programmes, puis testez-les, en respectant les indications données à la suite de chacun d'eux. Ils ont des ambitions limitées... mais ont le mérite de "marcher"... sans délai !

CAT.ALLOG

ASSEMBLAGE PAR ProCODE

Si vous ne possédez pas d'assembleur, seule cette partie vous concerne.

```

0300: A5 67
0302: 85 06
0304: A5 68
0306: 85 07
0308: A5 07
030A: C5 6A
030C: 90 0D
030E: A5 06
0310: C5 69
0312: 90 07
0314: A5 07
0316: C5 6A
0318: D0 01
031A: 60
    
```

```

0 * ELIMINATION DE "ALOG" DANS UN PROGRAMME BASIC *
1 * ----- *
2 * *
3 INDEXB EQU $06 : Adressage indexé (AdrBas)
4 INDEXH EQU $07 : Idem (AdrHaute)
5 * *
6 ORG $300
7 * *
8 PROG EQU *
9 LDA $67 : Début du programme (Ab)...
10 STA INDEXB
11 LDA $68 : et Adresse haute
12 STA INDEXH
13 BCL3 LDA INDEXH : Adresse haute dans A...
14 CMP $6A : comparée à LOMEM (AdrH)
15 BCC BCL1 : Si A plus petit, saut.
16 LDA INDEXB : Adresse basse dans A...
17 CMP $69 : comparée à LOMEM (AdrB)
18 BCC BCL1 : Si A plus petit, saut.
19 LDA INDEXH : Plus grand, alors on fait
20 CMP $6A : un nouveau test avec
21 BNE BCL1 : Lomem (AdrHaute)...Saut
22 RTS : ...ou fin de travail.
23 *
    
```

CAT.ALOG (suite et fin)

031B: E6 06	24	BCL1	INC	INDEXB	: + 1 pour pointeur AdrB
031D: D0 02	25		BNE	BCL2	: Si différent de 0: saut,
031F: E6 07	26		INC	INDEXH	: ou alors, + 1 pour AdrH
0321: A0 08	27	BCL2	LDY	£#08	: On a 8 caractères
0323: B1 06	28	BCL4	LDA	(#06),Y	: "CATALOG à tester
0325: D9 6E 03	29		CMP	£036E,Y	: en utilisant l'adressage
0328: D0 DE	30		BNE	BCL3	: indexé.
032A: 88	31		DEY		: Y = Y - 1
032B: D0 F6	32		BNE	BCL4	: Si Y <> de 0, suite
032D: A5 06	33		LDA	INDEXB	: On a trouvé "CATALOG
032F: 18	34		CLC		: Pas d'addition sans cela!
0330: 69 08	35		ADC	£#08	: On ajuste le pointeur...
0332: 85 06	36		STA	INDEXB	: qui n'a pas été...
0334: A5 07	37		LDA	INDEXH	: incrémenté à l'intérieur
0336: 69 00	38		ADC	£#00	: de la boucle
0338: 85 07	39		STA	INDEXH	: C'est fait!
033A: C8	40	BCL5	INY		: Y était à 0 (ici, +1)
033B: B1 06	41		LDA	(#06),Y	: On cherche les guillemets
033D: 48	42		PHA		: en stockant les valeurs
033E: C9 22	43		CMP	£\$22	: Guillemets ?
0340: D0 F8	44		BNE	BCL5	: Non: une autre lecture
0342: 84 FF	45		STY	£FF	: Oui: mémorisons Y
0344: A5 06	46		LDA	INDEXB	: On revient en arrière de
0346: 38	47		SEC		: 4 cases, par une petite
0347: E9 04	48		SBC	£#04	: soustraction (ALOG)
0349: 85 06	49		STA	INDEXB	: et on met à jour
034B: A5 07	50		LDA	INDEXH	: sans oublier la retenue
034D: E9 00	51		SBC	£#00	: éventuelle pour la
034F: 85 07	52		STA	INDEXH	: partie haute de l'adresse.
0351: 68	53	BCL6	PLA		: Avec cette boucle, on
0352: 91 06	54		STA	(#06),Y	: écrit les valeurs empilées
0354: 88	55		DEY		: sur ALOG
0355: D0 FA	56		BNE	BCL6	
0357: A5 06	57		LDA	INDEXB	: et on s'offre une autre
0359: 18	58		CLC		: mise à jour de l'index
035A: 65 FF	59		ADC	£FF	: sans oublier l'ancienne
035C: 85 06	60		STA	INDEXB	: valeur de Y (mémorisée
035E: A5 07	61		LDA	INDEXH	: plus haut), et pas
0360: 69 00	62		ADC	£#00	: davantage la fameuse
0362: 85 07	63		STA	INDEXH	: retenue... si elle existe!
0364: A9 20	64		LDA	£\$20	: A chargé avec l'espace
0366: C8	65	BCL7	INY		: Plus un pour Y
0367: 91 06	66		STA	(#06),Y	: Espace à l'adresse \$6-7+Y
0369: C0 04	67		CPY	£#04	: A-t-on mis 4 espaces ?
036B: D0 F9	68		BNE	BCL7	: Non: encore un tour!
036D: F0 99	69		BEQ	BCL3	: Oui: retour case départ.
	70	*			
036F: 22	71		DFB	'*	: C'est ce que l'on cherche
0370: 43	72		DFB	'C	: pour le transformer en un
0371: 41	73		DFB	'A	: simple CAT... suivi de
0372: 54	74		DFB	'T	: 4 espaces (dont notre
0373: 41	75		DFB	'A	: Apple adoré n'a que faire,

0374: 4C	76	DFB 'L	: mais dont il se moque
0375: 4F	77	DFB 'O	: éperdument.
0376: 47	78	DFB 'G	: Au revoir et Merci!

BSAVE CAT.ALOG, A\$300, L\$77

MODE D'EMPLOI

On ne peut plus simple. ① BLOAD CAT.ALOG — ② LOAD PROGRAMME BASIC — ③ CALL 768.
 Vous terminerez par SAVE PROGRAMME BASIC

REC.REM

Si vous ne possédez pas d'assembleur, contentez-vous de recopier ces codes.

0300:	20	58	FC
0303:	A9	15	
0305:	85	25	
0307:	20	22	FC
030A:	A0	03	
030C:	A9	76	
030E:	20	3A	DB
0311:	20	2C	D5
0314:	E8		
0315:	BD	00	02
0318:	9D	90	03
031B:	D0	F7	
031D:	86	FF	
031F:	F0	03	
0321:	20	1A	FC
0324:	A0	03	
0326:	A9	82	
0328:	20	3A	DB
032B:	20	2C	D5

0	*	-----*	*
1	*	RECHERCHER UNE CHAINE DANS UN PROGRAMME ET LA	*
2	*	REPLACER PAR UNE AUTRE DE LONGUEUR IDENTIQUE	*
3	*	-----*	*
4	*		*
5	INDEXB	EQU \$06	: Adres. indexé (Partie basse)
6	INDEXH	EQU \$07	: Idem Partie Haute
7	LONG	EQU \$FF	: Contient longueur du mot
8	CV	EQU \$25	: Position verticale curseur
9	BUFFER	EQU \$200	: Buffer d'entrée (0 à la fin)
10	CH1	EQU \$376	: Le mot "REPLACER: "
11	CH2	EQU CH1+12	: Le mot "PAR: "
12	VECDOS	EQU \$03D0	: REPRISE DES E/SORTIES
13	STROUT	EQU \$DB3A	: Affiche une chaine par Y,A
14	VP	EQU \$FC1A	: Remonte le curseur
15	VTAB	EQU \$FC22	: Déplace le curseur en CV
16	HOME	EQU \$FC58	: Efface l'écran
17	INLIN	EQU \$D52C	: Saisie texte sans prompt
18	*		*
19		ORG \$300	
20	*		*
21		JSR Home	: Le bel écran que voilà!
22		LDA £\$15	: On travaille sur la ligne...
23		STA CV	: 22 (CV va de 0 à 23; \$15=21)
24		JSR Vtab	: Mais renseignons l'Applesoft
25		LDY £>Ch1	: Partie haute dans Y...
26		LDA £<Ch1	: et partie basse dans A...
27		JSR Strout	: pour affichage "REPLACER: "
28		JSR Inlin	: Quel est le mot à remplacer?
29	BCL2	INX	: X=\$FF à la sortie de Inlin
30		LDA Buffer.X	: Lect. du mot dans le buffer
31		STA \$0390,X	: Stocké là ou ailleurs...
32		BNE Bcl2	: Si A n'est pas à zéro
33		STX Long	: X est sauvé dans Long (\$FF)
34		BEG Bcl3	: Ici, on passe...
35	BCL4	JSR Vp	: sans remontée de curseur
36	BCL3	LDY £>Ch2	: Pour affichage de "PAR: "
37		LDA £<Ch2	: on refait comme plus haut
38		JSR Strout	: et le mot s'affiche
39		JSR Inlin	: Input...X = \$FF en sortie.

REC.REM (suite et fin)

```

032E: B8          40 BCL5   INX          ; on lui ajoute 1 et X = 0
032F: BD 00 02   41         LDA  Buffer,X   ; Voir plus haut: c'est
0332: 9D A0 03   42         STA  $03A0,X  ; explications, mais le mot
0335: D0 F7          43         BNE  Bcl5     ; est stocké ailleurs
0337: E4 FF          44         CPX  Long     ; La longueur est-elle bonne ?
0339: D0 B6          45         BNE  Bcl4     ; Non: à refaire (VP)
46 *-----*
033B: A5 67          47         LDA  $67      ; Début du programme Basic
033D: 85 06          48         STA  IndexB   ; Stocké dans notre index
033F: A5 68          49         LDA  $68      ; Idem avec la partie haute
0341: 85 07          50         STA  IndexH
51 *-----*
0343: A5 07          52 SPG1   LDA  IndexH   ; Avons-nous dépassé Lomem
0345: C5 6A          53         CMP  $6A      ; partie haute ?
0347: 90 0F          54         BCC  Bcl6     ; Non: allons voir plus loin!
0349: A5 06          55         LDA  IndexB   ; Avons-nous dépassé Lomem
034B: C5 69          56         CMP  $69      ; partie basse ?
034D: 90 09          57         BCC  Bcl6     ; Non: on peut continuer
034F: A5 07          58         LDA  IndexH   ; Oui: est-on égal à Lomem
0351: C5 6A          59         CMP  $6A      ; partie haute ?
0353: D0 03          60         BNE  Bcl6     ; Non: on continue vraiment...
0355: 4C D0 03     61         JMP  Vecdos   ; Oui: retour au Dos. FIN
0358: B6 06          62 BCL6   INC  IndexB   ; Plus un pour l'index
035A: D0 02          63         BNE  Bcl7     ; Si l'on a 0 ($FF+1), il
035C: B6 07          64         INC  IndexH   ; faut incrémenter IndexH
035E: A4 FF          65 BCL7   LDY  $FF      ; Récupération de la longueur
0360: B1 06          66 BCL8   LDA  (IndexB),Y ; Lecture
0362: D9 BF 03     67         CMP  $03BF,Y ; Comparaison
0365: D0 DC          68         BNE  Spg1    ; Inégal ? retour case départ
0367: 88            69         DEY          ; Y = Y - 1
0368: D0 F6          70         BNE  Bcl8    ; Pas zéro ? encore un tour!
036A: C8            71 BCL9   INY          ; Maintenant, on va écrire
036B: B9 9F 03     72         LDA  $039F,Y ; le mot qui remplace sur
036E: 91 06          73         STA  ($06),Y ; celui qui doit disparaître
0370: C4 FF          74         CPY  $FF      ; La longueur est atteinte ?
0372: D0 F6          75         BNE  Bcl9    ; Non: on continue
0374: F0 CD          76         BEQ  Spg1    ; Retour à la case départ
0376: 52 45 4D
      50 4C 41
      43 45 52
      3A            77         DFB  'R','E','M','P','L','A','C','E','R', :
0380: 20 00          78         HEX  2000   ; Un espace + un zéro obli-
0382: 50 41 52
      3A            79         DFB  'P','A','R',':
0386: 20 00          80         HEX  2000   ; gatoire (fin de chaîne)
81 *-----*

```

BSAVE REC.REM, A\$300, L\$88

MODE D'EMPLOI

① BLOAD REC.REM — ② LOAD PROGRAMME BASIC. Pour chaque remplacement : CALL 768 (efface la page) ou CALL 771 (pas d'effacement). Indiquer le mot à chercher, puis celui à mettre à la place. Espace et ponctuation sont acceptés, mais pas un mot commençant par "INT" et pas davantage les commandes de DOS (CATALOG, par exemple). N'oubliez pas de sauver le programme transformé !

VILLES DE FRANCE

UN jeu instructif, conçu à l'intention des petits et des grands, par M. DEVAUX. La routine VILLES.LM est un peu longue, mais elle permet d'afficher instantanément la carte de France. Et puis, disons-le, ce n'est pas plus agaçant à recopier que n lignes DATA ! A vos claviers !



LE PROGRAMME EN BASIC

```

100 REM *****
110 REM ** VILLES DE FRANCE **
120 REM ** PAR M. DEVAUX **
130 REM *****
140 :
150 TEXT : NORMAL : PRINT CHR$(21):
HOME
160 PRINT CHR$(4);"BLOAD VILLES.LM"
170 LOMEM: 26500: DIM V(100)
180 T1$ = "* VILLES DE FRANCE *"
190 T2$ = "-----"
200 GOSUB 290
210 GOSUB 410
220 GOSUB 560
230 GOSUB 1000
240 HOME : END
250 REM DESSIN
260 HGR
270 CALL 24576
280 RETURN
290 REM PRESENTATION
300 PRINT CHR$(21): HOME
310 GOSUB 250
320 T$ = T1$:J = 1
330 VTAB 22: GOSUB 370
340 T$ = T2$: VTAB 23: GOSUB 370
350 FOR I = 1 TO 2000: NEXT
360 RETURN
370 REM FORMATAGE DES TITRES
380 T = (40 - LEN(T$)) / 2
390 HTAB T: PRINT T$
400 RETURN
410 REM CHOIX DU NOMBRE DE JOUEURS
420 FOR I = 1 TO 4
430 N$(I) = "":SC(I) = 0

```

```

440 TEXT : HOME
450 T$ = T1$: GOSUB 370:T$ = T2$
460 VTAB 3: GOSUB 370
470 VTAB 8: INPUT " Nombre de joue
urs ? ";N$:N = VAL(N$)
480 IF N < 1 OR N > 4 THEN VTAB 8: HT
AB 1: CALL - 958: GOTO 470
490 VTAB 12
500 FOR I = 1 TO N
510 PRINT " Nom du joueur n";I;:
INPUT " ? ";N$(I)
520 PRINT
530 NEXT
540 FOR I = 1 TO 1000: NEXT
550 RETURN
560 REM JEU
570 GOSUB 250
580 C = C + 1
590 FOR I = 1 TO N
600 HOME
610 VTAB 21: PRINT "Nom : ";N$(I);
620 HTAB 28: PRINT "Score : ";SC(I)
630 P = 10:AID$ = ""
640 H = INT ( RND (1) * 72) + 1
650 IF V(H) = 1 THEN 640
660 V(H) = 1
670 RESTORE
680 FOR J = 1 TO H
690 READ V$,A,B
700 NEXT
710 HCOLOR= 7
720 FOR W = B - 2 TO B + 2
730 HPLLOT A - 2,W TO A + 2,W
740 NEXT

```

(suite au verso)

VILLES DE FRANCE (suite)

```

750 S$ = "":R$ = ""
760 VTAB 23: PRINT "Nom de la ville
? ";AID$;
770 PRINT S$;: GET S$
780 IF ASC (S$) = 13 THEN 810
790 IF ASC (S$) = 127 OR ASC (S$) = 8
THEN VTAB 23: HTAB 1: CALL - 958
: GOTO 750
890 R$ = R$ + S$: GOTO 770
810 VTAB 23: HTAB 1: CALL - 958
820 IF R$ = V$ THEN 890
830 IF P = 2 THEN 870
840 P = INT (P / 2):AID$ = AID$ + LEF
T$ (V$,1)
850 V$ = RIGHT$ (V$, LEN (V$) - 1)
860 GOTO 750
870 VTAB 23: HTAB 3: PRINT "La répons
e était : ";AID$;V$
880 GOTO 920
890 VTAB 23: HTAB 17: PRINT "EXACT"
900 SC(I) = SC(I) + P
910 VTAB 21: HTAB 28: PRINT "Score :
";SC(I)
920 FOR T = 1 TO 1500: NEXT
930 HCOLOR= 0
940 FOR W = B - 1 TO B + 1
950 HPLLOT A - 1,W TO A + 1,W
960 NEXT
970 NEXT
980 IF C = 10 THEN RETURN
990 GOTO 580
1000 REM AFFICHAGE DES SCORES
1010 PRINT : TEXT : HOME
1020 T$ = T1$: GOSUB 370
1030 T$ = T2$: GOSUB 370
1040 VTAB 7: PRINT "SCORES : "
1050 PRINT "-----"
1060 VTAB 12
1070 FOR I = 1 TO N
1080 PRINT N$(I);: HTAB 20: PRINT "
";SC(I)
1090 PRINT
1100 NEXT
1110 VTAB 22: PRINT : INPUT "Une autr
e partie ? ";R$: IF LEFT$ (R$,1)
= "N" THEN RETURN
1120 IF R$ = "N" THEN RETURN
1130 IF LEFT$ (R$,1) < > "0" THEN 111
0
1140 RUN 170
1150 DATA DUNKERQUE,137,3,CALAIS,130,
5,ARRAS,143,14,LILLE,146,8
1160 DATA AMIENS,137,22,LE HAVRE,113,
28,ROUEN,122,30,BEAUVAIS,135,30
1170 DATA LAON,152,28,MEZIERES,165,24
,METZ,183,35,STRASBOURG,202,42
1180 DATA CAEN,105,34,SAINT-LO,97,35,
EVREUX,122,37,PARIS,137,41
1190 DATA CHALON-SUR-MARNE,161,39,NAN
CY,183,42,COLMAR,199,53
1200 DATA MULHOUSE,199,61,CHARTRES,12
6,48,BREST,54,45,QUIMPER,59,54
1210 DATA BOULOGNE-SUR-MER,131,9,RENN
ES,88,53,LE MANS,111,56
1220 DATA ORLEANS,132,58,SAINT-NAZAIR
E,81,67,NANTES,89,70
1230 DATA LA ROCHELLE,95,88,BORDEAUX,
101,113,NIORT,102,86
1240 DATA POITIERS,112,81,BLOIS,125,6
3,TOURS,117,67
1250 DATA TROYES,158,51,CHAUMONT,172,
53,VESOUL,184,61
1260 DATA BESANCON,184,69,DIJON,171,6
7,AUXERRE,153,59
1270 DATA ANGOULEME,110,98,PERIGUEUX,
117,107,LIMOGES,125,96
1280 DATA PAU,102,141,AGEN,114,124,CA
HORS,126,119
1290 DATA MONTAUBAN,124,128,TOULOUSE,
126,136,FOIX,128,148
1300 DATA PERPIGNAN,145,153,MONTPELLI
ER,158,136,NIMES,165,131
1310 DATA AVIGNON,172,129,MARSEILLE,1
88,139,TOULON,186,142
1320 DATA NICE,200,134,MONACO,205,131
,DIGNE,185,117
1330 DATA REIMS,148,31,GAP,186,117,GR
ENOBLE,181,106
1340 DATA VALENCE,171,111,LE PUY,157,
110,AURILLAC,139,112
1350 DATA RODEZ,140,122,CLERMONT-FERR
AND,146,96,LYON,169,96
1360 DATA ANNECY,185,93,MACON,168,87,
MOULINS,150,83
1370 DATA NEVERS,148,74,

```

Vous pouvez compléter la liste des villes et calculer vous-même les paramètres qui vous permettront de les positionner le plus exactement possible sur la carte de France.

VILLES.LM

(AFFICHAGE DE LA CARTE DE FRANCE)

6000- A9 33 85 86 A9 60 85 87
 6008- A2 87 28 EC F6 A0 81 B1
 6018- 86 F0 1F 85 88 C8 81 86
 6018- A6 88 A8 88 28 57 F4 A5
 6020- 86 18 69 82 85 86 98 87
 6028- A5 87 18 69 81 85 87 4C
 6038- 8D 68 68 15 88 88 8A 88
 6038- 89 81 88 81 87 81 86 82
 6040- 85 82 84 82 83 82 82 83
 6048- 81 83 81 84 88 85 88 86
 6058- 88 87 88 88 81 89 88 8A
 6058- 88 88 81 8C 88 8D 8E
 6068- 81 8F 81 10 88 11 7F 11
 6068- 7F 12 7F 13 7E 13 7D 14
 6078- 7C 15 7B 15 7A 15 79 15
 6078- 78 15 77 16 76 17 75 17
 6088- 74 17 73 17 72 18 71 19
 6088- 78 1A 6F 18 6F 1C 78 1D
 6098- 71 1D 72 1E 71 1F 78 1F
 6098- 6F 1F 6E 28 6D 28 6C 21
 60A8- 68 21 6A 21 69 28 68 28
 60A8- 67 28 66 28 65 28 64 1F
 60B8- 63 1F 62 1F 61 1F 68 28
 60B8- 5F 1F 5F 1E 5F 1D 5E 1C
 60C8- 5E 18 5E 1A 5E 19 5D 19
 60C8- 5C 19 58 1A 5A 19 59 18
 60D8- 58 18 57 18 57 19 58 1A
 60D8- 58 18 57 1C 57 1D 58 1E
 60E8- 59 1F 59 28 59 21 5A 21
 60E8- 58 22 58 23 58 24 58 25
 60F8- 58 26 58 27 58 28 58 29
 60F8- 58 2A 5C 28 5C 2C 58 2D
 6108- 5A 2D 59 2D 58 2D 58 2C
 6108- 57 28 56 28 56 2C 55 2D
 6118- 54 2C 53 2C 52 2C 52 28
 6118- 51 28 58 28 4F 2C 4E 2D
 6128- 4D 2E 4C 2D 4C 2C 48 28
 6128- 4A 2A 4A 29 49 28 49 27
 6138- 48 26 47 27 46 27 45 28
 6138- 44 28 43 27 42 28 42 29
 6148- 41 29 48 29 3F 29 3F 2A
 6148- 3E 2A 3D 29 3C 29 38 2A
 6158- 3A 2A 39 29 38 29 37 2A
 6158- 36 2B 35 28 34 2C 34 2D
 6168- 34 2E 34 2F 35 2F 36 2E
 6168- 37 2E 38 2E 38 2F 39 2F
 6178- 3A 38 39 31 38 31 37 31
 6178- 36 31 35 31 36 32 38 32
 6188- 39 33 38 34 37 34 36 34
 6188- 35 34 35 35 36 36 37 37
 6198- 37 38 38 39 39 39 3A 38
 6198- 38 38 3C 38 3D 38 3E 39
 61A8- 3F 39 48 39 48 3A 41 38
 61A8- 42 3A 43 38 44 3A 45 38
 61B8- 45 3C 45 3D 45 3E 46 3F
 61B8- 47 3E 48 3E 49 3D 4A 3D
 61C8- 48 3D 4C 3E 48 3F 4A 3F
 61C8- 48 48 4C 48 4D 48 4E 48
 61D8- 4E 41 4E 42 4D 43 4E 44
 61D8- 4F 45 58 45 51 44 52 44
 61E8- 53 44 54 44 55 44 56 45
 61E8- 55 46 54 46 53 46 52 46

61F8- 51 47 52 48 53 48 54 49
 61F8- 54 4A 53 48 53 4C 53 4D
 6208- 53 4E 54 4F 55 58 55 51
 6208- 56 52 57 53 58 54 59 55
 6218- 5A 56 58 56 5C 56 5D 56
 6218- 5E 56 5E 57 5D 58 5D 59
 6228- 5D 5A 5E 58 5E 5C 5E 5D
 6228- 5E 5E 5D 5F 5D 68 5C 61
 6238- 5D 62 5E 62 5F 63 68 64
 6238- 61 65 62 66 63 67 63 68
 6248- 63 69 64 6A 64 68 64 6C
 6248- 64 6D 63 6D 62 6C 62 68
 6258- 61 6A 61 69 61 68 68 67
 6258- 5F 66 5E 65 5E 66 5D 67
 6268- 5D 68 5D 69 5D 6A 5D 68
 6268- 5D 6C 5C 6D 5C 6E 5C 6F
 6278- 5C 78 5C 71 5C 72 5C 73
 6278- 5D 73 5E 73 5E 74 5D 75
 6288- 5C 75 5C 76 5C 77 5C 78
 6288- 5C 79 5B 7A 58 7B 58 7C
 6298- 58 7D 5A 7E 5A 7F 5A 88
 6298- 5A 81 59 82 59 83 59 84
 62A8- 59 85 58 86 58 87 58 88
 62A8- 57 89 57 8A 56 88 55 88
 62B8- 54 8C 55 8D 56 8D 57 8D
 62B8- 58 8E 58 8F 58 98 57 91
 62C8- 57 92 58 92 59 92 5A 92
 62C8- 58 93 5C 93 5D 93 5E 94
 62D8- 5F 94 68 94 61 94 62 95
 62D8- 62 96 63 97 64 97 65 97
 62E8- 66 96 67 97 68 98 69 98
 62E8- 6A 98 68 98 6C 98 6D 98
 62F8- 6E 98 6F 99 78 99 71 99
 62F8- 72 99 73 99 74 99 74 98
 6308- 74 97 75 97 76 97 77 97
 6308- 78 98 79 98 7A 99 78 99
 6318- 7C 99 7D 99 7D 9A 7E 98
 6318- 7F 9A 88 9A 81 9A 82 98
 6328- 83 9C 83 9D 84 9E 85 9F
 6328- 86 A8 87 9F 88 9F 89 9E
 6338- 8A 9E 88 9E 8C 9F 8E 9F
 6338- 8F 9F 98 9E 91 9E 92 9D
 6348- 93 9D 94 9E 95 9E 96 9E
 6348- 97 9D 96 9C 95 98 94 9A
 6358- 94 99 94 98 94 97 95 96
 6358- 94 95 94 94 94 93 95 92
 6368- 96 91 97 98 98 8F 99 8F
 6368- 9A 8F 98 8E 9C 8D 9D 8C
 6378- 9E 88 9F 8A 88 8A 81 89
 6378- A2 89 A3 8A A4 88 A5 88
 6388- A6 88 A7 88 88 88 A9 8C
 6388- AA 8D 88 8D 8C 8D 88 88
 6398- AE 88 AE 8A AF 89 88 89
 6398- B1 8A B1 88 88 8C B1 8D
 63A8- B2 8D B3 8E 84 8E 85 8F
 63A8- B6 8F B7 8F 88 8F B9 98
 63B8- BA 98 88 98 8C 98 8D 98
 63B8- BD 91 8E 91 8F 98 C8 98
 63C8- C1 98 C2 8F C3 8E C4 8E
 63C8- C5 8D C4 8C C4 88 C5 88

63D8- C6 8A C7 89 C8 89 C8 88
 63D8- C9 87 CA 87 C8 86 C8 85
 63E8- CC 84 CD 85 CE 84 CF 83
 63E8- CF 82 CF 81 CF 88 CF 7F
 63F8- D8 7E D1 7D D1 7C D1 7B
 63F8- D8 7B CF 7C CE 7C CD 78
 6408- CC 7B CB 7B CA 7B C9 7A
 6408- C8 7A C7 79 C6 78 C6 77
 6418- C6 76 C5 75 C5 74 C6 74
 6418- C6 73 C7 72 C8 71 C7 78
 6428- C7 6F C6 6F C5 6F C4 6E
 6428- C3 6D C2 6C C2 68 C2 6A
 6438- C3 6A C4 6A C5 6A C6 69
 6438- C6 68 C7 67 C8 66 C8 65
 6448- C8 64 C7 63 C6 63 C6 62
 6448- C6 61 C5 68 C4 68 C3 5F
 6458- C3 5E C4 5D C5 5C C5 58
 6458- C5 5A C4 59 C3 58 C2 57
 6468- C3 56 C3 55 C2 54 C3 53
 6468- C2 53 C1 53 C8 53 BF 54
 6478- BE 54 BD 55 BC 55 BC 56
 6478- BC 57 88 58 8A 58 89 58
 6488- B9 57 89 56 8A 55 8A 54
 6488- B9 53 89 52 8A 51 8A 58
 6498- BB 4F 88 4E BC 4D 8D 4C
 6498- BD 48 8D 4A BE 49 BF 48
 64A8- C8 47 C1 46 C2 45 C3 44
 64A8- C4 43 C5 42 C4 41 C3 41
 64B8- C2 41 C3 48 C4 3F C5 3F
 64B8- C6 48 C7 48 C8 48 C9 3F
 64C8- C9 3E C9 3D C9 3C CA 38
 64C8- CA 3A CA 39 CA 38 CA 37
 64D8- CA 36 C9 35 C9 34 C9 33
 64D8- CA 32 CA 31 C8 38 C8 2F
 64E8- CB 2E CC 2D CC 2C CC 28
 64E8- CC 2A CD 29 CE 28 CE 27
 64F8- CF 26 CF 25 D8 24 D8 23
 64F8- CF 22 CE 22 CD 22 CC 21
 6508- CB 21 CA 21 C9 21 C8 21
 6508- C7 21 C6 28 C5 21 C4 21
 6518- C3 21 C2 21 C1 28 C8 28
 6518- BF 28 BE 21 8D 28 8D 1F
 6528- BC 1E BC 1D 88 1C BA 1B
 6528- B9 1C 88 18 B7 18 B6 1C
 6538- B5 1C B4 1C B3 18 B2 1A
 6538- B1 18 88 18 88 18 AF 1B
 6548- AE 18 AD 1A AC 19 AB 19
 6548- AA 18 A9 17 A8 16 A8 15
 6558- A7 14 A7 13 A7 12 A7 11
 6558- A6 18 A5 11 A4 11 A4 12
 6568- A4 13 A3 13 A2 14 A1 14
 6568- A8 14 9F 13 9E 12 9F 11
 6578- 9F 18 9E 8F 9E 8E 9D 8E
 6578- 9C 8E 98 8D 9A 8D 9D 8C
 6588- 98 88 97 8A 96 89 95 8A
 6588- 94 8A 94 88 94 87 94 86
 6598- 93 86 92 86 91 85 98 86
 6598- 8F 86 8F 85 8E 85 8D 84
 65A8- 8C 84 8C 83 8C 82 88 81
 65A8- 88 88 88 88 FF FF

BSAVE VILLES.LM, A\$6000, L\$5AE

VERIFICATION : T = 0 ; FOR I = 24576 TO 26029 : T = T + PEEK(I) : NEXT I : PRINT T **Résultat : 153025**

CADRE-DÉMO

en réponse à monsieur ANSELME

NOS lecteurs se passionnent pour les cadres en langage machine, mais les résultats obtenus ne leur donnent pas toujours satisfaction. Exemple : l'essai (guère plus rapide que le Basic) de notre ami Bernard ANSELME. Le voici... expliqué, dans sa version originale (on poque le caractère à utiliser à l'adresse 6 : 32 pour un espace en mode inverse, 170 pour l'astérisque, etc.)

0300-	20 58 FC	JSR	\$FC58	HOME efface l'écran
0303-	A5 06	LDA	\$06	On récupère en 06 le caractère à afficher
0305-	A2 29	LDX	£\$29	On va afficher 41 caractères (1 ligne + 1)
0307-	20 ED FD	JSR	\$FDED	COUT sort un caractère
030A-	CA	DEX		X = X - 1
030B-	D0 FA	BNE	\$0307	Si X est supérieur à 0, boucle
030D-	A0 16	LDY	£\$16	Pour FOR Y = 1 TO 22 (\$16)
030F-	A2 26	LDX	£\$26	Si X = \$26 (38), la routine...
0311-	20 4A F9	JSR	\$F94A	...PRBL2 va envoyer 38 espaces
0314-	A5 06	LDA	\$06	On récupère en 06 le caractère-cadre
0316-	20 ED FD	JSR	\$FDED	Affichage en bout de ligne, puis...
0319-	20 ED FD	JSR	\$FDED	... au début de la ligne suivante
031C-	88	DEY		Y = Y - 1
031D-	D0 F0	BNE	\$030F	Si Y supérieur à 0, boucle
031F-	A2 26	LDX	£\$26	Pour FOR X = 1 TO 38 (\$26)
0321-	20 ED FD	JSR	\$FDED	Affichage
0324-	CA	DEX		X = X - 1
0325-	D0 FA	BNE	\$0321	Si X différent de 0, encore un tour
0327-	84 25	STY	\$25	Ramener le curseur en haut de l'écran
0329-	8D F7 87	STA	\$07F7	POKE le dernier caractère pour...
032C-	68	RTS		... éviter le SCROLL

Bien sûr, on le sait, nous avons publié plusieurs routines capables de tracer (instantanément) tous les cadres désirés, mais j'ai voulu montrer, avec la modification ci-dessous (le nombre d'instructions est le même, et la longueur est identique à un octet près), qu'il est possible, très simplement, en utilisant les sous-programmes de l'Apple, d'accélérer considérablement un programme.

0300-	20 58 FC	JSR	\$FC58	Dans la routine de Bernard ANSELME,
0303-	A5 06	LDA	\$06	le manque de rapidité est dû à PRBL2
0305-	A2 29	LDX	£\$29	(\$F94A) qui affiche successivement x
0307-	20 ED FD	JSR	\$FDED	espaces (38 dans le cas étudié)... mais
030A-	CA	DEX		22 fois !
030B-	D0 FA	BNE	\$0307	Avec ma version (totalement identique
030D-	A2 16	LDX	£\$16	

par ailleurs), \$FC62 (CR) envoie un retour chariot, puis \$FC10 (BS) recule le curseur, ce qui le ramène à la fin de la ligne précédente. Les deux \$FDED affichent alors le caractère de fin de ligne... puis le premier de la ligne suivante, et ainsi de suite.

80 COLONNES

Le programme fonctionne de la même manière en 80 colonnes, mais il faudra modifier le contenu des adresses \$306 (\$51... 81, au lieu de \$29... 41) et \$320 (\$48... 78 au lieu de \$26... 38).

030F-	20 62 FC	JSR	\$FC62
0312-	20 10 FC	JSR	\$FC10
0315-	A5 06	LDA	\$06
0317-	20 ED FD	JSR	\$FDED
031A-	20 ED FD	JSR	\$FDED
031D-	CA	DEX	
031E-	D0 EF	BNE	\$030F
0320-	A2 26	LDX	\$26
0322-	20 ED FD	JSR	\$FDED
0325-	CA	DEX	
0326-	D0 FA	BNE	\$0322
0328-	06 25	STX	\$25
032A-	00 F7 07	STA	\$07F7
032D-	60	RTS	

AFFICHAGE NORMAL : POKE 6,32 (par exemple), puis CALL 768 — **CADRE SANS HOME** : CALL 771 — **EFFACEMENT DU CADRE** : POKE 6,160, puis CALL 771.

Merci de votre attention et bonjour chez vous !

Votre NESTOR.

Le POINT sur : RND

C'est, vous le savez, la traduction de RANDOM (hasard). On utilise souvent cette fonction dans des jeux où intervient le hasard. Comment ?

$$X = 1 + \text{INT}(\text{RND}(1) * 9)$$

Au départ, avec RND(1), on obtient un nombre compris entre 0 et 1. En le multipliant par 9, on aura donc un nombre aléatoire compris en 0 et 8. En lui ajoutant 1, il sera compris entre 1 et 9.



DÉ À JOUER

Pour simuler le jet d'un dé à jouer, utilisez

$$X = 1 + \text{INT}(\text{RND}(1) * 6)$$

ARGUMENT NÉGATIF

Dans cet exemple, — où RND(-1) est le premier appel de la fonction —, on obtiendra toujours la même liste de nombres... aléatoires.

```
100 X = RND (-1)
110 FOR I = 1 TO 20
120 PRINT RND(1)
130 NEXT
```

QUESTION

Comment obtenir un nombre compris entre 21 et 30 ?

RÉPONSE

$$X = 21 + \text{INT}(\text{RND}(1) * 10) \dots$$

0 + 21 à 9 + 21.

RND(0) permet de sortir le nombre aléatoire obtenu à partir du précédent RND.

JEU sur APPLE

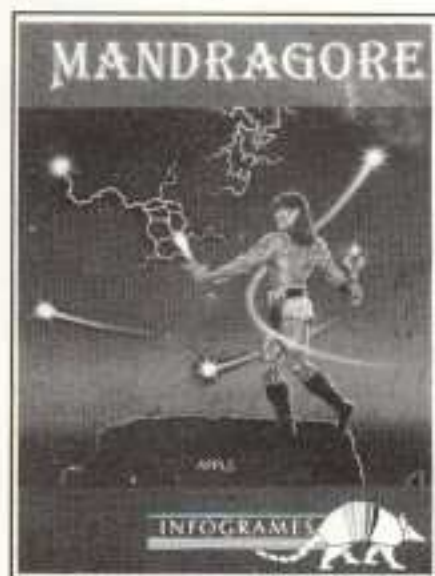


• MANDRAGORE de Marc Cecchi (Infogrames)

Neuf donjons... neuf énigmes à élucider pour affronter Yarod-Nor dans son propre donjon. Mais qui est cet étrange personnage ? Un infâme tyran, bien sûr, qui fait régner le mal et la terreur sur le pays de Mandragore. Comment en est-on arrivé là, alors que calme et bonté caractérisaient le riant royaume du roi Jorland ? L'affaire remonte au moment où une pluie d'étoiles filantes s'abat sur la contrée et met dramatiquement fin au règne du bon roi... aussitôt remplacé par un seigneur venu de nulle part, le sinistre Yarod-Nor.

Avec un Apple II, doté d'un minimum de 48 K, vous pourrez, après avoir constitué une équipe de quatre "héros", partir à l'attaque des neuf donjons...

(Catégorie "Aventure", difficile, 10-20 ans. Un livret + une disquette double face).



• HOLD-UP d'E. N'Guyen Van Huong, J.-D. Blanc et C. Quéant (Infogrames)

Attention ! dans ce jeu d'aventure (10-15 ans), il va y avoir un cerveau, un clochard, un receleur, une passante, plusieurs flics et — évidemment ! — quelques indics. Le décor est enchanteur : la Côte d'Azur, ses palmiers, sa Promenade des Anglais, sa plage... et sa capitale ensoleillée, Nice.

Ça se passe en pleine canicule, au mois d'août, pendant le week-end et il s'agit — vous vous en doutiez un peu, non ? — de dévaliser une banque, puis de s'enfuir à l'étranger. Bon, ce n'est pas très moral, d'accord. Ce n'est qu'un jeu, et puis le cerveau — VOUS... mais oui ! — aura tout à craindre de la police et de ses indics... surtout s'il n'a pas su s'entourer de fidèles complices, bien préparer son coup... et trouver les objets nécessaires à sa réalisation. Je ne vous souhaite pas bonne chance... car ce serait encourager le vice !

• COLORIAGES (Version Soft)

Vingt coloriages (insectes, animaux de la jungle, animaux familiers), sur deux disquettes : de quoi occuper les artistes en herbe pendant plusieurs heures ! L'enfant dispose d'une palette de tubes de couleur et d'un pinceau. A lui, en utilisant la souris, de tremper le pinceau dans la couleur de son choix pour obtenir le meilleur effet possible ! Jeu éducatif s'il en est. A conseiller.

Fonctionne sur Apple IIc ou IIe 128 K, avec la souris.



• PUZZLES (Version Soft)

Dans sa "Collection Souris", Version Soft propose aux enfants un excellent choix de 15 puzzles (animaux de la ferme, de la jungle et de la préhistoire). Utilisation on ne peut plus simple : l'enfant choisit un sujet, clique dessus, et l'image apparaît (le graphisme est fort réussi). Quand il estime avoir bien observé l'animal, il clique de nouveau... et l'image éclate en morceaux. Le jeu consiste, on s'en doute, à reconstituer l'animal qui, en guise de récompense, a la gentillesse de s'animer.

Fonctionne sur Apple IIc et Apple IIe 128 K... avec la souris.



• ARCANE par Luc et Viviane Sérard (Version Soft)

Ce logiciel utilise l'interpréteur graphique Arlequin (Chat Mauve) et il fonctionne sur Apple IIc ou IIe (révision B), avec une carte 80 colonnes étendue. Un moniteur couleur n'est pas indispensable, mais conseillé.

Avec Arcane, votre Apple se transforme en un véritable devin — avec Mireille, une voyante extra-lucide. Arcane a été réalisé par des spécialistes du tarot de Marseille et, comme vous le savez, là-bas, nos amis ne plaisantent que dans les histoires dites marseillaises.

A s'offrir pour animer une soirée (680 F TTC).

SONDAGE

SUR 8 BITS

Ce court programme n'est qu'une démonstration. Son but : vous montrer qu'il est facile — et économique ! — de mémoriser sur un seul octet la réponse à huit questions. Comment ? tout simplement en utilisant les huit bits de chaque octet. Dans notre exemple, cela se passe de la manière suivante :

7] 0 = Femme 1 = Homme
6	
5] Un seul sera à 1, celui correspondant à la tranche d'âge choisie
4	
3	
2	
1] 0 ou 1 selon la réponse aux questions 1 et 2
0	

Une mini-routine en langage machine transforme le nombre binaire (RE\$) en un octet HEXA, mais on obtiendrait le même résultat en basic (revoir, à ce sujet, le programme de conversion paru dans *Tremplin Micro n°6*).

NOTA : les résultats sont totalisés au fur et à mesure (lignes 255-260).

Les données sont mémorisées sous forme de fichier B (de 8176 à ...x). Les 16 premiers octets servent à mémoriser les résultats (deux octets par item). A partir de 8192, on trouve la valeur Hexa correspondant à chaque questionnaire. Le dernier octet est à zéro.

ANALYSE DES RÉSULTATS

On peut imaginer un programme capable d'analyser les résultats en sélectionnant tel ou tel bit de chaque octet. Nous donnons plus loin un exemple montrant comment compter les hommes. Sachant que le bit 7 est à 1 pour un homme et à 0 pour une femme, il est évident que tout octet supérieur à 7F (01111111) désigne un homme (le 7^e bit passe à 1 à partir de la valeur \$80).

A vous, amie Lectrice, ami Lecteur, de plancher un peu sur ces problèmes !

SONDAGE.BAS

```

100 TEXT : NORMAL : PRINT CHR$(21):
    HOME
105 GOTO 350: REM ROUTINE LM ET FICHI
    ER BINAIRE
110 FOR I = 1 TO 10:T$ = T$ + "----":
    NEXT
115 PRINT "SAISIE DES RESULTATS": VTA
    B 2: PRINT T$: POKE 34,3
120 POKE 32,27: POKE 33,13: PRINT : H
    OME
125 PRINT "HOMME = 1": PRINT "FEMME =
    0"
130 PRINT : PRINT "18-24 ANS : 1";"25
    -34 ANS : 2";"35-49 ANS : 3";"50-
    64 ANS : 4";"65 ANS &+ : 5"
135 PRINT "QUESTION 1  QUESTION 2
    0=NON  1= OUI"
140 PRINT : PRINT "FIN = ESCAPE"
145 POKE 32,0: POKE 33,25: PRINT
150 Q = Q + 1: HOME : INVERSE : PRINT
    Q: NORMAL
155 RE$ = ""
160 PRINT : VTAB 6: PRINT "SEXE *;: G
    ET R$: IF R$ < CHR$(48) OR R$ >
    CHR$(49) THEN 160
165 PRINT R$:RE$ = RE$ + R$

```

SONDAGE

SUR 8 BITS (suite)

```
170 PRINT : VTAB 8: PRINT "AGE " ; G
    ET R$: IF R$ < CHR$ (49) OR R$ >
    CHR$ (53) THEN 170
175 PRINT R$
180 FOR I = 1 TO 5
185 IF I = VAL (R$) THEN RE$ = RE$ +
    "1": GOTO 195
190 RE$ = RE$ + "0"
195 NEXT
200 PRINT : VTAB 10: PRINT "QUESTION
    1 " ; GET R$: IF R$ < CHR$ (48) O
    R R$ > CHR$ (49) THEN 200
205 PRINT R$: RE$ = RE$ + R$
210 PRINT : VTAB 12: PRINT "QUESTION
    2 " ; GET R$: IF R$ < CHR$ (48) O
    R R$ > CHR$ (49) THEN 210
215 PRINT R$: RE$ = RE$ + R$
220 PRINT : INVERSE : PRINT RE$: NORM
    AL
225 PRINT : VTAB 22: PRINT "<1> SUITE
    <0> ANNULE " ; GET R$
230 IF R$ = CHR$ (27) THEN 245
235 IF R$ < CHR$ (48) OR R$ > CHR$ (4
    9) THEN 225
240 PRINT R$: ON VAL (R$) + 1 GOTO 15
    5,245: GOTO 275
245 FOR I = 1 TO 8: V = VAL ( MID$ (RE
    $, I, 1))
250 J = (I * 2) - 2
255 P = PEEK (RE + J) + V: IF P = 256
    THEN P = 0: POKE RE + J + 1, PEE
    K (RE + J + 1) + V
260 POKE RE + J, P
265 POKE I + 784, V: NEXT
270 CALL 768: POKE AD + Q, PEEK (6)
275 IF R$ < > CHR$ (27) THEN 150
280 HOME
285 INVERSE : PRINT Q" QUESTIONNAIRES
    " ; NORMAL : PRINT
290 PRINT : P = 2: GOSUB 345: PRINT "
    de 18-24 ans"
295 P = 4: GOSUB 345: PRINT " de 25-3
    4 ans"
300 P = 6: GOSUB 345: PRINT " de 35-4
    9 ans"
305 P = 8: GOSUB 345: PRINT " de 50-6
    4 ans"
310 P = 10: GOSUB 345: PRINT " de 65
    ans &+"
315 PRINT : PRINT "QUESTION 1 = " ; P
    = 12: GOSUB 345: PRINT " OUI"
320 PRINT : PRINT "QUESTION 2 = " ; P
    = 14: GOSUB 345: PRINT " OUI"
325 PRINT : PRINT : P = 0: GOSUB 345:
    PRINT " hommes - " ; Q - PE ; " femme
    s"
330 POKE 32,0: POKE 33,40: PRINT : PR
    INT T$
335 Q = Q + 1: POKE 8191 + Q,0: Q = Q
    + 16
340 PRINT : PRINT CHR$ (4)"BSAVE SOND
    AGE,A 8176,L"Q: END
345 PE = PEEK (RE + P) + PEEK (RE + P
    + 1) * 256: PRINT PE ; RETURN
350 FOR I = 768 TO 784: READ R: POKE
    I,R: NEXT
355 AD = 8191: RE = 8176
360 ONERR GOTO 375
365 PRINT CHR$ (4)"BLOAD SONDAGE": Q =
    ( PEEK (43616) + PEEK (43617) *
    256) - 17: REM 48840-41 SOUS PROD
    OS
370 GOTO 110
375 IF PEEK (222) = 6 THEN Q = 0: FOR
    I = RE TO AD: POKE I,0: NEXT : G
    OTO 110
380 DATA 169,0,162,0,24,126,17,3,42,2
    32,224,8,208,246,133,6,96
```

Combien d'hommes ?

0300-	A9 00	LDA	£\$00
0302-	85 06	STA	\$06
0304-	85 08	STA	\$08
0306-	85 09	STA	\$09
0308-	A9 20	LDA	£\$20
030A-	85 07	STA	\$07

Ce court programme compte les octets dont le bit 7 est à 1 (Homme).

L'adresse de départ est écrite en \$6-7 (0020).

Deux octets sont prévus pour le comp-
teur : \$8-9

030C-	A2 00	LDX	£\$00	X = 0 pour adressage indirect
030E-	A1 06	LDA	(<\$06,X)	Lecture de la mémoire indiquée en \$6-7
0310-	F0 10	BEQ	\$0322	Si c'est un zéro, on a terminé
0312-	10 06	BPL	\$031A	Si n = 0, c'est une femme
0314-	E6 08	INC	\$08	Si c'est un homme, on incrémente le compteur
0316-	D0 02	BNE	\$031A	A-t-on obtenu 0 (\$FF+1) ?
0318-	E6 09	INC	\$09	Si oui, il faut incrémenter l'adresse haute
031A-	E6 06	INC	\$06	+1 pour l'index (adresse basse)... et +1 pour la partie haute si on a dépassé \$FF
031C-	D0 02	BNE	\$0320	
031E-	E6 07	INC	\$07	Et on recommence
0320-	D0 EC	BNE	\$030E	Fin de travail.
0322-	60	RTS		

Après un CALL 768, on lira le résultat par un $N = \text{PEEK}(8) + \text{PEEK}(9) * 256$. Quand on est sûr de ne pas dépasser 255 (\$FF), il n'est pas nécessaire de prévoir 2 octets pour le compteur.

AUTRE EXEMPLE

On désire savoir combien de personnes de sexe féminin (bit 7 à 0), appartenant à la première tranche d'âge (bit 6 à 1) ont répondu par oui à la question 1. Pour faciliter la compréhension, nous testons (bit 1 à 1), les trois bits à tour de rôle (il existe un autre moyen).

0300-	A9 00	LDA	£\$00] Voir la première routine. Là encore, on a prévu deux octets pour le résultat : \$8-9
0302-	85 06	STA	\$06	
0304-	85 08	STA	\$08	
0306-	85 09	STA	\$09	
0308-	A9 20	LDA	£\$20	
030A-	85 07	STA	\$07	

030C-	A2 00	LDX	£\$00	
030E-	A1 06	LDA	(<\$06,X)	
0310-	F0 1A	BEQ	\$032C	
0312-	30 10	BMI	\$0324	Si n = 1, c'est un homme : on saute !

0314-	29 40	AND	£\$40	\$40, c'est 01000000, pour tester le bit 6. Si on obtient 0, on passe...
0316-	F0 0C	BEQ	\$0324	
0318-	A1 06	LDA	(<\$06,X)	

031A-	29 02	AND	£\$02	\$02, c'est 00000010, pour tester le bit 1. Si on obtient 0, on passe...
031C-	F0 06	BEQ	\$0324	

031E-	E6 08	INC	\$08] Mêmes explications que pour l'exemple n°1
0320-	D0 02	BNE	\$0324	
0322-	E6 09	INC	\$09	
0324-	E6 06	INC	\$06	
0326-	D0 02	BNE	\$032A	
0328-	E6 07	INC	\$07	
032A-	D0 E2	BNE	\$030E	
032C-	60	RTS		

QUESTION : Pouvait-on raccourcir le programme ? **RÉPONSE :** Oui **COMMENT ?**



COMPTABILITÉ GÉNÉRALE

CX HÉRAKLÈS *

POMME D'OR 1983 du meilleur logiciel Apple (catégorie gestion), CX HÉRAKLÈS a fait ses preuves depuis longtemps.

Capable, sans disque dur, de gérer 500 comptes, il s'adresse à bon nombre de nos lecteurs.

• SIMPLICITÉ D'ABORD

Destiné aux artisans, commerçants et travailleurs indépendants, CX HÉRAKLÈS se devait d'être facile à mettre en œuvre. En fait, dès le premier contact avec ce logiciel, on ressent la volonté de dialogue. On sait que telle opération est en cours, mais on apprend aussi ce qu'il est nécessaire de faire pour passer à l'étape suivante. L'utilisateur est guidé pas à pas. Le temps d'apprentissage ne dépasse pas quelques heures (1 heure affirme la documentation, mais elle est tout de même optimiste !).

• POINTS FORTS

- Les tris sont effectués à la fin d'une période de saisie, afin de ne pas interrompre celle-ci et de minimiser le temps passé à attendre devant l'écran.
- En cours de saisie, il est facile de changer de journal.
- Le chevauchement de deux exercices est automatique.
- Toutes les commandes sont dirigées à partir d'un nombre minimum de touches, toujours les mêmes : ESCAPE, RETURN, OUI et NON.
- Exploitation des fichiers par d'autres programmes (Visicalc, Apple Works, Business Graphics).

- Numérotation interne chronologique de chaque écriture, pour une réelle crédibilité fiscale.
- Journaux à contre-parties multiples automatiques (jusqu'à 6).
- Edition de comptes, journaux, balances (à l'écran ou sur imprimante).
- Edition journal par journal du total des mouvements créditeurs et débiteurs en une seule opération.

• PRÉSENTATION DES JOURNAUX

Les journaux sont présentés en colonnes, exactement comme sur les livres comptables. Une clé de contrôle permet de s'assurer de la validité de chaque numéro de compte.

En comptabilité, il est agréable de posséder un document unique, facile à manier, regroupant toutes les écritures pour l'année passée pour un compte particulier. CX HÉRAKLÈS facilite l'assemblage chronologique de toutes les pages d'un même compte. En effet, dans l'édition du grand livre, il fait un saut de page à chaque changement de compte et pagine le compte à partir du dernier numéro de page enregistré lors de la dernière clôture périodique du même compte.

• SÉCURITÉ

Si vous espérez bidouiller votre comptabilité avec CX HÉRAKLÈS, n'allez pas plus loin. Le programme n'autorise absolument pas les corrections... illégales. La seule façon de rectifier une erreur est de passer une contre-écriture, puis de resaisir la bonne information.

La sécurité, c'est aussi l'impossibilité, pour une personne non autorisée, d'accéder aux commandes. Le maître, c'est le responsable numéro un de la comptabilité — Vous ! Il a son mot de passe et c'est lui qui attribue ou non un code à tel ou tel utilisateur. Toutefois, dans tous les cas, il peut se réserver un accès exclusif aux clôtures... ou encore n'autoriser que certaines opérations à un utilisateur donné.

NOTONS AU PASSAGE :

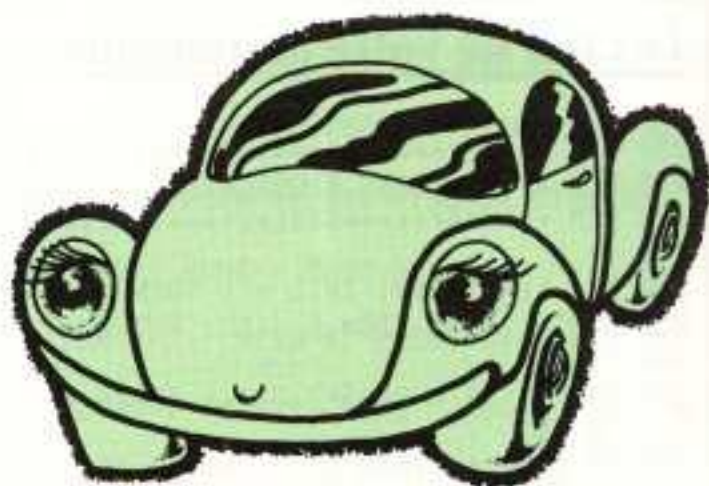
- une réparation automatique des fichiers en cas de coupure de courant.
- la sauvegarde du grand livre et l'édition automatique du journal d'ouverture lors de la clôture annuelle.
- l'impossibilité de supprimer un compte ou un journal s'ils contiennent des écritures.

C.R.

* Version-Soft
94 rue Lauriston PARIS 75116

Le coût réel de votre automobile

par Clément RENARD



TOUT, vous compterez tout pour savoir à combien vous revient, au kilomètre bien sûr, votre taxi personnel. Et pas question, ici, de diviser par 10000 l'érosion de votre capital-auto... sous prétexte que vous avez prévu de parcourir 10000 km dans l'année à venir. Ce qui est dépensé est en général irrécupérable. Essayez plutôt — en cas d'accident grave, par exemple — de vous faire rembourser une part inutile de vignette ou d'assurance ! Quant aux réparations importantes et aux pneus neufs, nous savons tous qu'ils n'entrent que pour une faible part dans le prix de vente d'un véhicule d'occasion.

Conclusion : Un prix de revient kilométrique réel ne doit pas tenir compte des plus-values éventuelles. Et voici ce que cela donne :



POUR MÉMOIRE

POKE 32,2 fixe la fenêtre d'écran à 2 (ce qui correspond à HTAB 3).

POKE 33,74 limite l'écran à 74 lignes de largeur, à partir de la ligne fixée par POKE 32,n

POKE 34,2 évite l'effacement des lignes 0 et 1 de l'écran.

POKE 35,23 protège la ligne 23.

POKE 36,3 : Position horizontale du curseur

POKE 37,1 : Position verticale du curseur.

POKE 2039,32 affiche, en bas et à droite de l'écran, un carré en mode inverse.

```

100 NORMAL : D$ = CHR$ (4): PRINT D$"PR13": PRINT : VTAB
      1
105 GOSUB 920
110 ONERR GOTO 120
115 GOSUB 840: GOTO 150
120 POKE 216,0:LF = 0
125 :
130 REM *****
135 REM *          TRACE DU CADRE          *
140 REM *****
145 :
150 PRINT : HOME
155 INVERSE : FOR I = 1 TO 9: PRINT "          " :: NEXT
160 FOR I = 2 TO 23: POKE 1403,79: PRINT "          " :: NEXT
165 FOR I = 1 TO 13: PRINT "          " :: NEXT
170 NORMAL : POKE 37,1: POKE 36,3: POKE 2039,32
175 :
180 REM PROTECTION DE CE CADRE
185 POKE 34,2: POKE 35,23: POKE 32,2: POKE 33,74
190 :
195 REM TITRE DE LA PAGE
200 R$ = " COUT REEL DE MON AUTOMOBILE "
205 INVERSE : PRINT : PRINT LEFT$ (V$, LEN (R$)): PRINT
      R$: PRINT LEFT$ (V$, LEN (R$)): NORMAL
210 PRINT : PRINT "          FICHE DE SITUATION"
215 PRINT "          -----"
220 GOTO 265
225 :

```

Le coût de votre automobile

(suite)

SAISIE

Lors de la première utilisation, il faut indiquer la date d'achat du véhicule, son kilométrage (pas de 0 S.V.P.), son prix d'achat, sa cote actuelle et son kilométrage actuel.

Pendant la saisie, utilisez la touche DELETE pour effacer (ou revenir en arrière lorsqu'il s'agit de la date).

On peut revenir à l'item précédent en pressant l'une des touches ← ↑.

Quand la valeur est bonne, inutile de la retaper : un simple RETURN est suffisant.

Remarques : la date d'achat doit être antérieure (ou identique) à la date actuelle. La valeur actuelle doit être inférieure (ou égale) au prix d'achat.

Le programme détermine le temps qui s'est écoulé entre la date de l'achat et la date du jour, mais cela ne sert strictement à rien dans le calcul du coût au kilomètre.

```
230 REM *****
235 REM * FICHE SIGNALETIQUE *
240 REM *****
245 :
250 L = LEN (DS$): IF L = 1 THEN DS$ = "": RETURN
255 DS$ = LEFT$ (DS$,L - 1): RETURN
260 :
265 SA = 0:S$ = S$(SA):DA$ = "DATE D'ACHAT" : ":V = 10:H
= 16
270 IF S$ = "" THEN S$ = "00.00.1985"
275 PRINT : VTAB V: PRINT DA$ + S$:
280 DS$ = ""
285 IF H = 18 OR H = 21 THEN H = H - 1: GOSUB 250
290 IF H = 18 OR H = 21 THEN H = H + 1:DS$ = DS$ + "."
295 POKE 1403,H: GET R$: IF (R$ = CHR$(8) OR R$ = CHR$(11)) AND (SA) THEN H = 16: GOTO 385
300 IF R$ = CHR$(13) THEN 320
305 IF R$ = CHR$(127) AND H > 16 THEN H = H - 1: GOSUB 250: GOTO 285
310 IF R$ < CHR$(48) OR R$ > CHR$(57) THEN POKE - 1636 8,0: GOTO 290
315 DS$ = DS$ + R$: PRINT R$;:H = H + 1: GOTO 290
320 H = 16: IF DS$ = "" THEN 335
325 IF LEN (DS$) = 10 THEN S$ = DS$: GOTO 335
330 S$ = DS$ + RIGHT$ (S$,10 - LEN (DS$))
335 PRINT : VTAB V: POKE 1403,H: PRINT S$
340 J = VAL ( LEFT$ (S$,2)):M = VAL ( MID$ (S$,4,2)):AN = VAL ( RIGHT$ (S$,4))
345 IF (J < 1 OR J > 31) OR (M < 1 OR M > 12) OR (AN < 1950 OR AN > 1999) OR MID$ (S$,3,1) < "." OR MID$ (S$,6,1) < "." THEN S$ = "":H = 16: GOTO 270
350 IF SA = 5 AND AN < VAL ( RIGHT$ (S$(0),4)) THEN 270
355 S$(SA) = S$: IF SA = 5 THEN RETURN
360 AM = AN:MM = M:JM = J
365 :
370 SA = 1:S$ = "KILOMETRAGE" : ": GOTO 460
375 SA = 2:S$ = "PRIX D'ACHAT" : ": GOTO 460
380 SA = 3:S$ = "COTE ACTUELLE" : ": GOTO 460
385 SA = 4:S$ = "KILOMETRAGE" : ": GOTO 460
390 IF VAL (S$(SA)) < VAL (S$(1)) THEN 385
395 PRINT :SA = 5:V = 20:DA$ = "DATE ACTUELLE" : ":S$ = S$(SA)
400 GOSUB 270
405 :
410 MR = 12 - MM + M - 1: IF (AN - AM - 1) > 0 THEN MR = MR + (12 * (AN - AM - 1))
415 JR = (30 - JM) + J
420 IF JR > 29 THEN MR = MR + 1:JR = JR - 30: GOTO 420
425 IF AN = AM THEN MR = MR - 12
430 IF MR < 0 OR JR < 0 THEN 395
```



POKE 1403,n remplace
POKE 36,n... et HTAB.
Telle qu'elle est écrite,
cette version du pro-
gramme fonctionne sur
Apple IIe, Apple IIc et
Apple IIe avec 65C02

Sur Apple II+, il faudra
nécessairement revoir
l'affichage sur 80 col-
onnes, en fonction de la
carte utilisée.

```

435 VTAB 22: PRINT "  " : INVERSE : PRINT MR" MOIS & " J
R" JOURS"
440 FOR I = 3 TO 22: POKE 1403,28: VTAB I: PRINT " ": NE
XT : NORMAL
445 POKE 33,0: POKE 32,35: POKE 33,44: VTAB 3: HTAB 36:
PRINT
450 POKE 34,2: POKE 35,22: HOME : GOTO 540
455 :
460 PRINT :V = 10 + (SA * 2): VTAB V: PRINT S$:S$ = S$(
SA)
465 POKE - 16368,0: PRINT : VTAB V: POKE 1403,H: PRINT S
$: GET R$
470 P = PEEK ( - 16384)
475 IF PEEK (37) < > V - 1 THEN 465
480 IF P > 47 AND P < 58 AND LEN (S$) < 6 THEN S$ = S$ +
R$
485 IF P = 127 THEN S$ = STR$ ( INT ( VAL (S$) / 10)): P
RINT CHR$ (8):" "
490 IF S$ = "0" THEN S$ = ""
495 S$(SA) = S$
500 IF P = 11 OR P = 8 THEN ON SA GOTO 265,370,375,380
505 IF P = 13 AND S$ < > "" THEN ON SA GOTO 375,380,385,
395
510 GOTO 465
515 :
520 REM *****
525 REM * GRANDS POSTES DE DEPENSES *
530 REM *****
535 :
540 ONERR GOTO 700
545 PRINT "1. STATION-SERVICE      " : I = 1: GOSUB 570
550 PRINT "2. REVISIONS-REPARATIONS " : I = 2: GOSUB 570
555 PRINT "3. VIGNETTE & ASSURANCE  " : I = 3: GOSUB 570
560 PRINT "4. AUTRES DEPENSES      " : I = 4: GOSUB 570
565 PRINT "-----": GO
TO 575
570 PRINT LEFT$(V$,12 - LEN ( STR$ ( INT (T(I))))):T(I)
: RETURN
575 PRINT : INVERSE : PRINT "CODE": NORMAL : PRINT " (F
IN/ERREUR SAISIE=CONTROLE-C) " : GET R$
580 IF R$ = CHR$ (3) THEN 735
585 R = ASC (R$) - 48: IF R < 1 OR R > 4 THEN PRINT : CA
LL - 998: CALL - 998: GOTO 575
590 POKE 34,10
595 PRINT : HOME
600 POKE 1403,0: PRINT "DATE:      " : S$:
605 POKE 1403,9: INPUT "":D1$
610 IF D1$ = "" THEN D1$ = S$: GOTO 625
615 L = LEN (D1$): IF L = 10 THEN 625
620 D1$ = D1$ + RIGHT$(S$,10 - L)
625 PRINT RL$: POKE 1403,9: PRINT D1$:S$ = D1$
630 POKE 1403,0: PRINT "OBJET:      .....
..":

```



DÉPENSES

Vous comptabiliserez les
réparations résultant
d'accidents dans le poste
4. Vous pourrez par exem-
ple avoir une facture de
"Tôlerie-peinture" de
4500 F ... puis un rem-
boursement de 4000 F de
votre assurance, à enre-
gistrer en moins (-4000).

Remarque : les centimes
sont acceptés.

Le coût de votre automobile

(suite)

```
635 POKE 1403,9: INPUT "":OB$
640 IF OB$ = "" THEN PRINT RL$:: GOTO 630
645 POKE 1403,0: INPUT "MONTANT: ":MO$
650 IF MO$ = "" OR VAL (MO$) = 0 THEN PRINT RL$:: GOTO 6
45
655 POKE 1403,0: PRINT : PRINT "<C>ORRECT <R>EFAIRE <S
>UITE " : GET R$: POKE - 16368,0
660 IF R$ = CHR$ (3) THEN 680
665 IF R$ = "C" THEN 680
670 IF R$ = "R" THEN 595
675 IF R$ < > "S" THEN PRINT CHR$ (31):: GOTO 655
680 T(R) = T(R) + VAL (MO$):E = E + 1:E$(E) = STR$ (R) +
" " + D1$ + " " + OB$ + LEFT$ (V$,26 - LEN (OB$))
685 P = 0:L = LEN (MO$): FOR I = L TO 1 STEP - 1: IF MID
$(MO$,I,1) = "." THEN P = L + 1 - I
690 NEXT : IF P < 3 THEN MO$ = MO$ + LEFT$ (V$,3 - P)
695 E$(E) = E$(E) + LEFT$ (V$,10 - LEN (MO$)) + MO$
700 POKE 34,7: POKE 1403,0: VTAB 8: CALL - 198: IF R$ =
CHR$ (3) THEN 735
705 GOTO 575
710 :
715 REM *****
720 REM *      COUT AU KILOMETRE      *
725 REM *****
730 :
735 NK = VAL (S$(4)) - VAL (S$(1)): REM KILOMETRES PARCO
URUS
740 AM = VAL (S$(2)) - VAL (S$(3)): REM PERTE SUR CAPITA
L
745 DT = 0: FOR I = 1 TO 4:DT = DT + T(I): NEXT
750 DT = DT + AM
755 DK = DT / NK:DK = VAL ( STR$ (( INT (DK * 100) / 100
)))
760 PRINT : POKE 34,17: HOME : PRINT : INVERSE : PRINT "
LE KILOMETRE ACTUEL COUTE:" : NORMAL : PRINT " ":DK
765 IF R$ = CHR$ (3) THEN 795
770 :
775 REM *****
780 REM *      ECRITURE DU FICHIER 1      *
785 REM *****
790 :
795 PRINT : PRINT D$"OPEN AUTO": PRINT D$"WRITE AUTO": F
OR I = 0 TO 5: PRINT S$(I): NEXT : FOR I = 1 TO 4: P
RINT T(I): NEXT : PRINT LF + E: PRINT D$"CLOSE AUTO"
800 IF E = 0 THEN 810
805 PRINT D$"OPEN AUTO.D.L52": PRINT D$"WRITE AUTO.D.R0"
: PRINT LF + E: FOR I = 1 TO E: PRINT D$"WRITE AUTO.
D.R"LF + I: PRINT E$(I): NEXT : PRINT D$"CLOSE AUTO.
D"
810 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: PO
KE - 16368,0: TEXT : HOME : END
```

CONTRÔLE-C

Pendant la saisie des dépenses, un CTRL-C vous permettra de revenir à la ligne de CHOIX.

Pour terminer la saisie, il suffit aussi de taper CTRL-C.

COU-T-KM

On est ou pas d'accord avec ce calcul... qui peut évidemment être modifié suivant les critères de chaque utilisateur.

FICHER AUTO

On y trouve la fiche du véhicule, le total des dépenses pour chaque poste... et le nombre d'enregistrements du fichier direct AUTO.D

FICHER AUTO.D

Rien ne vous interdit de l'exploiter autrement par la suite, notamment pour réaliser une analyse plus pointue des dépenses.

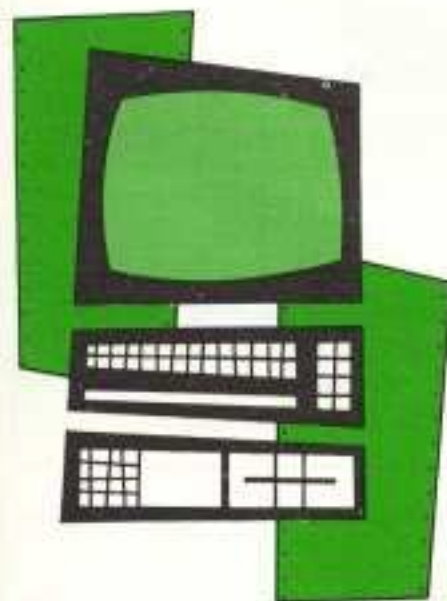


AFFICHAGE DES DÉPENSES

Possible dès la deuxième utilisation. Certains aimeront peut-être y ajouter une routine d'impression : rien de plus facile !

```
PRÉ1 : FOR I = 1 TO LF :  
PRINT E$(I) : NEXT : PRÉ0.
```

```
820 REM *****  
825 REM *   LECTURE DU FICHER 1   *  
830 REM *****  
835 :  
840 PRINT : PRINT D$"OPEN AUTO": PRINT D$"READ AUTO": F  
OR I = 0 TO 5: INPUT S$(I): NEXT : FOR I = 1 TO 4: I  
NPUT T(I): NEXT : INPUT LF: PRINT D$"CLOSE AUTO"  
845 VTAB 12: PRINT "DESIREZ-VOUS LIRE LA LISTE ACTUELLE  
DES DEPENSES ?": PRINT  
850 PRINT "<O>ui ou <N>on ? " : : CALL - 198: GET R$: PRIN  
T : IF R$ = "O" OR R$ = "o" THEN 890  
855 IF R$ < > "N" AND R$ < > "n" THEN PRINT RL$RL$: GOTO  
850  
860 RETURN  
865 :  
870 REM *****  
875 REM *   LECTURE AFFICHAGE FICHER2 *  
880 REM *****  
885 :  
890 HOME :L = 0  
895 PRINT D$"OPEN AUTO.D.L52": PRINT D$"READ AUTO.D,R0":  
INPUT LF: FOR I = 1 TO LF: PRINT D$"READ AUTO.D,R" I  
: INPUT E$(I): NEXT : PRINT D$"CLOSE AUTO.D"  
900 FOR I = 1 TO LF:L = L + 1: PRINT E$(I)  
905 IF L = 22 THEN L = 0: GET R$: POKE - 16368,0: HOME  
910 NEXT I  
915 GET R$: RETURN  
920 FOR I = 1 TO 10:V$ = V$ + "   ": NEXT : REM Ligne v  
ide  
925 RL$ = CHR$(31): REM Remonte d'une ligne  
930 RETURN
```



À VOUS DE JOUER !

Les programmes de *TREMPIN MICRO* sont testés sur trois types d'APPLE : *Ile*, *Ilc* et *Ile 65C02* (avec imprimantes *DMP*, *IMAGEWRITER* et *IMAGEWRITER II*).

Les listages publiés sont ceux des programmes essayés. Ils ne devraient donc pas comporter d'erreurs... mais il faut compter avec la photocomposition (pas toujours), la mise en page, et les méchantes punaises (bugs).

Bref, il arrive que les erreurs soient sans influence sur le fonctionnement du programme. Ce n'est pas une raison pour ne pas les signaler.

Par contre, lorsqu'elles perturbent le programme, il convient d'agir sans tarder. En tout cas, à partir de ce numéro, nous enverrons **5 disques vierges** au premier Lecteur (ou à la première Lectrice) qui aura découvert et corrigé une véritable erreur. Qu'on se le dise !

Utilisation de GRAF.0

DE nombreux lecteurs nous écrivent ou nous téléphonent pour des problèmes d'impression. C'est vrai pour la SCRIBE (dont on peut demander, chez les revendeurs Apple, la documentation en français — échange gratuit), mais c'est aussi — et souvent — le cas pour l'IMAGEWRITER.

On sait que le kit vendu avec l'imprimante comprend une disquette TOOL KIT permettant toutes les impressions graphiques. Nous avons déjà expliqué, à plusieurs reprises, comment extraire la routine d'impression, et l'utiliser à partir d'un programme en Basic.

Mais le temps passe, le matériel évolue... et les programmes aussi. Certains lecteurs confondent une routine GF... et une autre intitulée GRAF.0. Il est évident que les résultats sont désastreux.

Le programme ci-dessous vous permettra, sous ProDOS, d'utiliser facilement GRAF.0 (qui aura au préalable été copié sur votre propre disquette).

GRIMP.0

```

60000 REM UTILISE GRAF.0 DE LA DISQUETTE TOOL KIT PRODOS APPLE
60010 TEXT : HOME : HIMEM: 36864
60020 D$ = CHR$(4)
60030 IM = 1: REM SLOT DE VOTRE IMPRIMANTE
60040 PRINT D$"BLOAD GRAF.0"
60050 POKE 6,IM: POKE 252,16 * IM
60060 HOME : PRINT "NOM DE VOTRE IMAGE (? POUR CATALOGUE) : -----
-----"
60070 PRINT : CALL - 198: INPUT "-> ";TI$: IF TI$ = "?" THEN PRINT D$"C
AT": GET R$: PRINT : GOTO 60060
60075 IF NOT LEN (TI$) THEN 60140
60080 CALL - 198: PRINT : PRINT "HGR 1 OU 2 ? -> ";: GET P$: IF P$ < >
"1" AND P$ < > "2" THEN 60060
60090 PRINT : PRINT D$"BLOAD"TI$,A$"2000 + 2000 * (VAL(P$)=2)
60100 PRINT : PRINT "MODE (0=Simple blanc sur noir 1=Noir surblanc 2=Doub
le blanc sur noir 3=Comme 1)"
60110 PRINT : CALL - 198: PRINT "CHOIX MODE -> ";: GET R$: PRINT : IF R$
< "0" OR R$ > "3" THEN 60080
60120 POKE 7, VAL (R$) + 4 * (VAL(P$)=2)
60130 CALL 38636: CALL 38156
60131 END : REM A ELIMINER POUR UTILISER LA SUITE...
60135 REM UNE LIGNE COMME CELLE-CI PEUT VOUS RENVOYER AU PROGRAMME BASIC
EN COURS,SANS PERDRE LES VARIABLES (LA MEME OPERATION DOIT ETRE PREVU
E POUR QUITTER LE DIT PROGRAMME...EN DIRECTION DE GRIMP.0
60140 PRINT D$"CHAIN GRAPHE,à1635": REM RETOUR AU PROGRAMME GRAPHE

```

Attention ! pour appeler GRIMP.0, votre programme Basic devra comporter une ligne ainsi rédigée :
1635 PRINT CHR\$(4) "CHAIN GRIMP.0,à60000"

Naturellement, 1635 a été choisi arbitrairement. Le vrai numéro sera celui de la ligne de votre propre programme.

Vous avez écrit à TREMPLIN MICRO



• Malgré mon inexpérience, je viens de m'offrir ProCODE, car j'essaie, désespérément, d'apprendre. J'ai soif de savoir...

J'ai acheté : AU-DELÀ DU BASIC AVEC L'ASSEMBLEUR (Ed. Radio), LES RESSOURCES DE L'APPLE IIc...

Je devore tout ce que je trouve et qui pourrait m'apprendre à programmer en assembleur, mais je ne trouve que des auteurs qui n'apprennent la pratique de l'assembleur qu'à ceux qui la connaissent déjà (ou suis-je trop bête pour comprendre ?).

Pouvez-vous m'indiquer un livre dont l'auteur me prendrait par la main en partant du principe que je suis idiot et se mettrait dans la tête de remplir la mienne ?

Pour l'instant, je recopie des programmes et lorsqu'ils ne fonctionnent pas, je suis incapable de les rectifier comme je le fais en basic lorsque cela arrive. Par exemple j'ai recopié, sous ProCODE : WHAT TIME IS IT, MISS MOUSE ?

Résultat : ProCODE me répond : LIGNE 32 — Opérande non en page 0 et même réponse à la ligne 33.

C.-J. L. (06160 JUAN LES PINS)

TM La Programmation du 6502 (Ed. Dunod) devrait vous aider à mieux comprendre... mais astreignez-vous à en taper un à un tous les exemples. Un autre conseil : revenez de temps en temps sur des routines que vous n'aviez pas comprises lors d'un premier examen : fort de vos connaissances nouvelles, vous les verrez d'un autre œil !

Ceci dit, il y a deux omissions dans le listage source de WHAT TIME IS IT MISS MOUSE ?

```
0304 : A9 74 32 LDA £ < INTERUP
0306 : A2 03 33 LDA £ > INTERUP
```

Il faut en excuser notre service de photocomposition (qui ne connaît pas les signes plus petit et plus grand que... ajoutés manuellement).

• Existe-t-il, sur l'Apple IIc, une instruction permettant la renumérotation automatique d'un programme ? Peut-être pourriez-vous publier une telle routine ?

G. R. (07140 LES VANS)

TM Avec le DOS 3.3, vous avez RENUMBER (sur la disquette MASTER), mais rien (gratuitement) avec ProDOS. Deux solutions (il y en a d'autres... moins évidentes) :

— Convertir vos programmes en DOS 3.3, utiliser RENUMBER... puis reconvertir.

— Vous offrir BASIC PROGRAMMING WITH ProDOS qui comporte, entre autres utilitaires, le programme APA, version ProDOS.

• Pourquoi l'Apple IIc passe-t-il en mode TRACE quand je me mets en 80 colonnes en utilisant : PRINT CHR\$(4);"PR£3" ?

P. M. (13003 MARSEILLE)

TM Un conseil : ajoutez un PRINT après cette commande...

• Je tire mon chapeau à votre revue qui est l'une des rares à être spécialisées dans Apple. Avant tout, je voudrais corriger votre programme TIMER dans T. M. n°5 page 3. Vous avez dit que la carte souris génère une interruption tous les 60ème de seconde. Or c'est faux. La fréquence exacte est 51ème de seconde. Il faut donc corriger le programme à l'adresse \$386 (902) pour y mettre £\$32 (50) au lieu de £\$3B (59). Ainsi, le décalage que j'ai constaté après seulement 20 secondes entre mon chronomètre et le TIMER n'existe plus.

Je voudrais maintenant m'adresser à tous les lecteurs de T.M. Je possède un Apple IIc avec Souris et une imprimante à jet d'encre. J'ai constaté que les codes qui arrivaient à mon imprimante étaient toujours supérieurs à 127 (bit de poids fort à 1). Comment empêcher cela ?

J'aimerais que vous publiiez mon adresse complète afin qu'un lecteur puisse me répondre ou m'écrire pour échanger des idées.

Bruno PACI (20 rue Erik Satie — Appt. 610 — 31100 TOULOUSE)

TM Attention ! la fréquence horloge du 6502 est en fonction des cycles d'une instruction (ce n'est pas si simple...). Le temps d'exécution d'un cycle est égal à :

$$T \times V/F$$

(Suite au verso)

Vous avez écrit à TREMLIN MICRO



TM T = Nombre de cycles-horloge (pourquoi pas plusieurs ?)

V = Fréquence Microprocesseur 1MHz base

F = Fréquence horloge de votre micro.

Ainsi l'Apple IIc tourne plus vite que le IIe !

• *Disposant d'une imprimante SEIKOSHA GP-500 A, reliée par une interface CENTRONICS V.24 à un Apple IIc, je viens vous demander de m'aider. Comment imprimer mes graphiques ?*

D. J. (86000 POITIERS)

TM Pour parler d'une imprimante, il faut disposer de son manuel d'utilisation (en français de préférence).

Visiblement, les importateurs de cette imprimante n'ont pas estimé utile d'offrir à leurs clients un utilitaire de recopie graphique... et c'est navrant. Mais aussi, pourquoi chercher tant de complications, sur un Apple IIc... alors qu'il existe une ImageWriter ? Le prix ? Hum ! où est l'économie quand on n'obtient pas de sa machine ce que l'on est en droit d'en attendre ?

Nous sommes tout à fait disposés, ici, à évoquer, GRATUITEMENT, toutes les initiatives des diffuseurs d'imprimantes compatibles, pourvu qu'elles servent les intérêts de nos lecteurs.

Guy-HACHETTE.

Vous pouvez aussi téléphoner à SEIKOSHA : (16-1) 45.34.75.35.

• *Est-il réellement indispensable, sous ProDOS, de taper d'interminables CAT/DISK0/SOUSDISK (par exemple) pour obtenir le catalogue (et rien que cela) SOUSDISK de la disquette DISK0 ?*

D. V. (33000 BORDEAUX)

TM Dans votre exemple, si le PREFIX est DISK0, vous pouvez obtenir le sous-catalogue SOUSDISK en tapant un simple CATSOUSDISK.

• *Pouvez-vous m'indiquer comment initialiser une disquette pour que celle-ci démarre sur un programme en langage machine ?*

J. M. (32000 AUCH)

TM Sous DOS 3.3, tapez d'abord POKE 40514,52 (la valeur normale est 6) puis INIT NOM (DU PROGRAMME). Ensuite, DELETE NOM (DU PROGRAMME) puis BSAVE NOM, A..., L...

TM Essayez par exemple de placer les codes à l'adresse \$300 :

300 : 20 60 FB A9 02 85 22 20 6E A5 4C D0 03
puis BSAVE NOM, A768, L13

Quand vous lancerez le système, celui-ci affichera "APPLE II", le CATALOG... puis vous renverra à l'Applesoft.

• *Comment peut-on cacher un listing d'un programme en Basic ? Pourriez-vous publier un programme capable de colorier une zone de l'écran (polygone) ?*

S. G. (78310 ELANCOURT)

TM Un POKE 214,255 annule certaines commandes du DOS 3.3 (LIST, NEW, CALL, POKE) et provoque un RUN quand on essaie de les utiliser. Nous publions, page 19, un article qui devrait déjà vous éclairer sur la façon de colorier une figure, en page HGR.

• *1° Pourriez-vous donner la solution d'extension au programme EXAMEM.LM (T.M.1) ? Par exemple pour l'affichage des chiffres ?*

2° La programmation est-elle obligatoire pour du Hard-copy sur Apple IIc ?

T. B. (59560 COMINES)

TM 1. Changer 3B5 : 20 7D 0E par 3B5 : 20 CF 03
Ajouter ensuite :
3CF : C9 30 90 05 E9 5B 38 E9 A5 60
Sauver la nouvelle version : BSAVE EXAMEM.LM, A768, L217

2. Oui, il est indispensable de passer par un programme, aussi bien en mode TEXT qu'en mode HGR... à moins de disposer d'une interface (ou d'une imprimante) autorisant la copie directe...

• *J'éprouve toujours une certaine satisfaction à recevoir les nouveaux numéros de TREMLIN MICRO, qui traitent en profondeur les "secrets" de la "micro APPLE". Cependant, comme le dit si bien un correspondant (T.M. n°5 page 57 signé Jean-Luc M. 54600 VILLERS LES NANCY), APPLE FRANCE se fout de nous, et c'est peu dire. Pourquoi ? Tout simplement, car la documentation "officielle" c'est du commerce, pour ne pas dire "la carotte", et quant aux stages renvoyés de mois en mois (et cela fait 2 ans) dans la province, il N'Y EN A PAS, et il N'Y EN AURA JAMAIS (les mots sont pesés). D'ailleurs, je ne me suis jamais gêné pour le leur dire avec le CLUB APPLE (500 F par an de cotisation — remboursée 75 F je crois pour éviter les JUGEMENTS SÉVÈRES).*

J'ai suivi des cours sur T07 — ARMSTRAD... aussi je me trouve toujours un peu désemparé lorsqu'il s'agit de programmer APPLESOFT : mon niveau n'est pas élevé, mais j'arrive toutefois à program-

mer avec ma fille de 14 ans qui fulmine devant les "acrobaties" qu'il faut faire avec APPLE IIc. Ne serait-il pas possible d'aborder l'APPLESOFT, dans votre revue T.M., seulement quelques pages du type LED MICRO, pour satisfaire les "moins" initiés, et ne pas les décourager ? car votre revue, si elle est bien faite, est d'un niveau que beaucoup de jeunes trouvent élevé. Un ouvrage comme "Pratique de l'APPLESOFT par LILEN" a été lu, relu et "exécuté". Je pense que vous auriez bonne audience auprès de certains lecteurs.

Et puis, pourquoi ne pas réaliser des programmes d'un niveau moyen du type rencontré dans d'autres revues, mais adaptés à l'APPLESOFT — et pour les citer :

- "Retrouver et réapprendre le LATIN"
- "La multiplication égyptienne"...

Jean-Pierre M. (26240 SAINT-VALLIER SUR RHÔNE)

TM Vous êtes un peu sévère avec Apple dont les documentations — sans être des modèles du genre — n'ont rien à envier à celles des autres constructeurs.

Il y a bien place pour l'initiation dans les colonnes de T.M., mais la composition de chaque numéro résulte d'un choix fondé sur les demandes de nos Lectrices et Lecteurs.

D'autre part, il n'est pas possible, dans chacun des numéros d'une revue bimestrielle, de revenir sur des notions de base qu'une multitude d'ouvrages (très complets) ont déjà traités. Notre rôle, je crois, consiste à apporter un plus... mais ai-je raison ? Au lectorat de Tremplin Micro de répondre !

G.-H.

• Notre club a été créé en Septembre 1984 et compte actuellement 145 adhérents.

Nous nous permettons de vous solliciter pour l'obtention de spécimens de votre revue ainsi que de remise sur le prix de l'abonnement à votre publication pour nos adhérents.

Nous souhaiterions nous faire connaître dans la région et vous demandons de bien vouloir informer vos lecteurs de l'existence du club informatique FOR I NEXT. Le club a reçu en don un ordinateur SKS 2500 ; pour l'instant, nous n'avons pas pu l'utiliser, notre généreux donateur ne nous ayant pas fourni la documentation ni aucun système d'exploitation. L'importateur français, la société SODIEPIE à Charenton, n'existe plus et il nous est impossible de connaître le nouvel importateur ni de pouvoir joindre le fabricant Allemand : l'adresse que nous possédons doit être incomplète, notre courrier est revenu. Nous nous permettons d'abuser de votre gentillesse et vous sollicitons pour l'obtention des nom et adresse du nouvel importateur ou de l'adresse du fabricant en Allemagne : à moins que

vous connaissiez une personne possédant ce type d'ordinateur et qui serait prête à nous aider.

Club Informatique FOR I NEXT
chez M. Philippe BARRAULT
15, square Dieudonné COSTES BONDOUFLE (91000 EVRY)
Tél. du Club : 64.97.67.09.

TM Voilà qui est fait... et il est certain que les Lecteurs de T.M. prendront contact avec vous.

Spécimens : T.M. n'est jamais distribué gratuitement et aucune remise n'est consentie sur le prix de l'abonnement. C'est la rançon de la liberté !

SKS 2500 : Votre généreux donateur doit être content d'avoir réussi à se débarrasser de la bête. Lui devrait pouvoir vous renseigner... à moins que l'un de nos Lecteurs... Qui sait ?

MULTIPLIER PAR 16 ?

• Dans un programme en assembleur, je suis amené à multiplier des valeurs par 8 ou par 16. Comment y parvenir ?

Louis BLÉROT (AVIGNON)

TM Vous pouvez utiliser la petite routine ci-après. Le multiplicande est stocké sur deux octets, aux adresses \$06 et \$07. Avec ce sous-programme, le résultat ne doit pas dépasser \$FFFF et le plus grand multiplicande sera donc égal à 4095 (4095 × 16 = 65520 = \$FFF0). Naturellement, il est stocké suivant le principe habituel : partie basse en 06 et partie haute en 07. Avec 4095 (\$FFF), on aura donc :

*6 : FF 0F
*300 G (Exécution)
*6.7
0006 — F0 FF (soit \$FFF0)

Voici maintenant la routine :

0300-	A2 04	LDX	£\$04
0302-	06 06	ASL	\$06
0304-	26 07	ROL	\$07
0306-	CA	DEX	
0307-	D0 F9	BNE	\$0302
0309-	60	RTS	

Question : Quelle doit être la valeur de x pour multiplier par 8 ?

Réponse : x = 3 pour multiplier par 8, 2 pour multiplier par 4 et 1 pour multiplier par 2... mais on fera alors plus court en utilisant :

MULTIPLIER PAR 2 :				MULTIPLIER PAR 4 :			
300 :	06 06	ASL	\$06	300 :	06 06	ASL	\$06
302 :	26 07	ROL	\$07	302 :	26 07	ROL	\$07
304 :	60	RTS		304 :	06 06	ASL	\$06
				306 :	26 07	ROL	\$07
				308 :	60	RTS	

Vous avez écrit à TREMPLIN MICRO



PRNTAX et PRNTYX

* Je ne comprends pas comment on affiche parfois des nombres hexadécimaux par quelques POKES, suivis d'un CALL. Exemple : POKE 69,160 : POKE 70,90 : CALL — 1727... et on obtient \$A05A

Elise N. (79120 LEZAY)

TM Comme vous y allez ! dans votre exemple, 160 (\$A0) est très bien mis dans l'Accumulateur (\$45=69) et 90 (\$5A) dans le registre X (\$46=70), mais un clin d'œil à la routine PRNTAX (\$F941) ne produira pas l'effet escompté. Il faut passer par une courte routine en langage machine, comme ci-après, et poquer au préalable vos valeurs dans deux mémoires-relais :

POKE 6,160 : POKE 7,90 : CALL 768 (\$300)

Et voici le sous-programme :

```
300 : A5 06      LDA $06
302 : A6 07      LDX $07
304 : 4C 41 F9   JMP $F941
```

Notez que \$F940 (PRNTYX) affiche les registres Y-X, comme son nom l'indique. Si vous avez la curiosité de taper CALL — 151, puis \$F940L, vous constaterez que la première instruction de PRNTYX commence par un TYA (qui met Y dans A). Autrement dit, PRNTYX confie tout le travail à PRNTAX !

Remarque : comme on a déjà pu le lire dans T.M., c'est un moyen d'afficher en HEXA une valeur DÉCIMALE. Ainsi, il suffit de poquer 24576 aux adresses 6-7 : POKE 6, 24576 — INT(24576/256) * 256 : POKE 7, INT(24576/256)... puis de taper CALL 768 pour obtenir l'affichage de 6000.

INPUT, WRITE ET READ EN MODE DIRECT

* Pour des raisons trop longues à vous expliquer, j'aimerais pouvoir faire des INPUT en mode direct... à l'intérieur d'une boucle, puis stocker mes données sur la disquette, toujours en mode direct. Est-ce possible et comment ?

Luc SERVAN (Liège — Belgique)

TM INPUT : Possible, en utilisant le fameux POKE 118,0 (numéro de ligne en cours d'exécution

TM qui est normalement à 255... en mode direct).
Exemple : POKE 118,0 : INPUT M\$(sur la même ligne)... ou encore :
POKE 118,0 : FOR I = 1 TO 10 : INPUT M\$(I) : NEXT

Si ce nombre de données est supérieur à 10, tapez au préalable un DIM M\$(... nombre de données).

Attention ! Si vous avez déjà "joué" avec la variable M\$(n), vous vous exposerez à un REDIM'D ARRAY ERROR... que vous éviterez par un CLEAR.

Longueur maximum des mots saisis : 13 (qui nous expliquera pourquoi ?).

FICHER SANS PROGRAMME : Bien sûr que ça marche (avec le DOS en mémoire, évidemment), comme ceci :

```
POKE 118,0 : POKE 51,0:PRINT CHR$(4) "OPEN
ESSAI" : FOR I = 1 TO 10 : PRINT CHR$(4)
"WRITE ESSAI" : PRINT M$(I) : NEXT : PRINT
CHR$(4) "CLOSE ESSAI"
```

Et vous pourrez vérifier votre essai, toujours en direct, de la manière suivante :

```
POKE 118,0 : POKE 51,0 : PRINT CHR$(4)
"OPEN ESSAI" : FOR I = 1 TO 10 : PRINT CHR$(4)
"READ ESSAI" : INPUT S$(I) : NEXT : PRINT
CHR$(4) "CLOSE ESSAI"
```

Puis contrôler la présence des données dans la variable S\$(n) par :

```
FOR I = 1 TO 10 : PRINT M$(I) : TAB (13) : S$(I)
: NEXT
```

ADRESSE D'UNE VARIABLE

* J'ai lu quelque part (mais où ?) qu'il était facile de trouver, par le truchement de l'Ampersand, l'adresse d'une variable... Pouvez-vous éclairer ma lanterne ?

Marc DONIZOT (95160 MONTMORENCY)

TM Première opération : indiquer à l'Ampersand que votre sous-programme est à l'adresse \$300. Pour cela : POKE 1013,76 : POKE 1014,0 : POKE 1015,3.

Ensuite, il faut avoir des variables en mémoire, mais on peut taper A\$ = "TOTO"... pour essayer ! Voici la routine en L.M :

```
0300- 20 E3 DF   JSR   $DFE3
0303- AA                TAX
0304- 20 40 F9   JSR   $F940
0307- A9 A0      LDA   $A0
0309- 20 ED FD   JSR   $FDED
030C- 9B                TYA
030D- 4C 24 ED   JMP   $ED24
```

TM

Deuxième opération :

Pour avoir la réponse, taper (en mode direct ou dans le cadre d'un programme) :

& A\$.... et l'adresse contenant le descriptif de la variable s'affiche en HEXA et en DÉCIMAL.

Si cette adresse est \$810, on pourra lire (à partir de cette adresse) :

— un octet contenant la longueur (en Hexa)

— deux octets contenant l'adresse où est mémorisé le mot "TOTO"

Il suffit de modifier (si peu) cette petite routine pour qu'elle se révèle capable d'afficher la longueur d'une variable :

```

0300- 20 E3 DF JSR $DFE3
0303- AA TAX
0304- 20 40 F9 JSR $F940
0307- A9 A0 LDA $A0
0309- 20 ED FD JSR $FDED
030C- 86 06 STX $06
030E- 84 07 STY $07
0310- A2 00 LDX $00
0312- A1 06 LDA ($06,X)
0314- AA TAX
0315- A9 00 LDA $00
0317- 4C 24 ED JMP $ED24

```

Essayez-la :

```

POKE 118,0 : FOR I = 1 TO 5 : INPUT $$ (I) :
NEXT
BERNADETTE
LILIANE
RENE
ANTOINETTE
BERNARD

```

```

FOR I = 1 TO 5 : & $$ (I) : PRINT : NEXT
082 F 10
0832 7
0835 4
0838 10
083B 7

```

POUR UN CADRE HGR

* Je me suis attaqué à votre routine de tracé d'un cadre en L.M à la page 47 du n°5. Ça me décevait un peu qu'elle soit à peine plus rapide que la version Basic. Vous trouverez ci-dessous une routine plus courte (7 octets) ; quant à la vitesse, je vous laisse apprécier... Cette fois, la version Basic devient vraiment laborieuse ! Une seule limitation : comme on travaille à l'octet, l'abscisse va de 7 en 7, ce qui fait que mon cadre est un peu plus épais (14 pts).

Maurice CHAVELLI (13012 MARSEILLE)

```

300- A9 20 LDA $20
302- 85 E6 STA $E6
304- AD 54 C0 LDA $C054
307- AD 52 C0 LDA $C052
30A- AD 57 C0 LDA $C057
30D- AD 50 C0 LDA $C050
310- A9 2A LDA $2A
312- 20 F4 F3 JSR $F3F4
315- A9 0C LDA $0C
317- 85 06 STA $06
319- 20 17 F4 JSR $F417
31C- A9 00 LDA $00
31E- A0 02 LDY $02
320- 91 26 STA ($26),Y
322- C8 INY
323- C0 26 CPY $26
325- D0 F9 BNE $0320
327- E6 06 INC $06
329- A5 06 LDA $06
32B- C9 B3 CMP $B3
32D- D0 E8 BNE $0317
32F- 60 RTS

```

TM

Quand j'écrivais : "Gageons qu'il se trouvera, parmi nos Lectrices et Lecteurs, quelque génie capable de rendre la routine en langage machine plus performante"... je ne me trompais pas. Bravo !

PRÉVENTION ROUTIÈRE

* La Prévention routière a dû publier plusieurs logiciels à l'intention des enfants des écoles, mais je ne réussis pas à les obtenir chez mon revendeur Apple. Pouvez-vous m'indiquer les titres de ces logiciels et me fournir l'adresse de leur éditeur ?

Marius S., Instituteur (30000 NÎMES)

TM

Vous trouverez "En allant à l'école — Le Piéton" et "En allant à l'école — Le Cycliste" dans la collection des Logiciels Nathan.

Chaque série se présente sous la forme habituelle : boîte plastique, livret illustré et deux cassettes. Age conseillé : 4-8 ans et 8-11 ans.

Il est certain que ces logiciels éducatifs sauront faire prendre conscience aux enfants des dangers qui les guettent dans la rue.

C'est de la bonne besogne, mais il y a un hic : ils ne fonctionnent que sur T07 et M05... ce qui pose bien des problèmes dans l'enseignement privé... où Apple est, semble-t-il, bien implanté.

Souhaitons que Nathan édite rapidement une version sur Apple !

Cedic/Nathan — 6 bd Jourdan — 75014 PARIS.

Vous avez écrit à TREMPLIN MICRO



FORMATAGE IMAGEWRITER

* Chers Amis de T.M., cette lettre pour vous communiquer le programme ci-joint qui pourra, je pense, résoudre le problème d'un de vos correspondants de T.M. n°6, M. Guy V. (92370 CHAVILLE), qui éprouve des difficultés avec son IMAGEWRITER. Le problème de l'interligne d'abord. Votre lecteur désire 6 lignes par pouce : c'est la configuration par défaut de l'imprimante. Il n'a donc pas à s'inquiéter. Quant à la longueur de page et aux marges haute et basse, l'utilitaire ci-joint lui permettra de venir à bout de ses difficultés et lui permettra, en prime, de choisir l'une des huit polices de caractères disponibles sur l'IMAGEWRITER. Ce programme a été conçu pour les listings les plus courants sur le marché, c'est-à-dire, 11 et 12 pouces. Mais, s'il le désire, votre lecteur pourra utiliser des feuilles de longueur différente en changeant la valeur du paramètre LP à la ligne 260. Je pense qu'une valeur de 97 devrait convenir pour son papier de 10,7 pouces. A lui de faire des essais !

Claude RIVIÈRE (13009 MARSEILLE)

```

10 TEXT : PRINT CHR$(21): HOME : INVERSE
   : PRINT " FORMATAGE IMAGEWRITER (APPLE
   //E //C) *; NORMAL : PRINT "-----
   PAR CLAUDE RIVIERE -----"
20 VTAB 6: PRINT "L'IMPRIMANTE EST-ELLE BR
   ANCHÉE (O/N) ?"; GET R$: PRINT R$
30 IF R$ = "O" OR R$ = "o" THEN 50
40 IF R$ = "N" OR R$ = "n" THEN VTAB 9: HT
   AB 10: PRINT "ALORS, BRANCHEZ LA!"; VTA
   B 12: PRINT "ET FRAPPEZ UNE TOUCHE POUR
   CONTINUER..."; GET W$: GOTO 50
45 GOTO 20
50 HTAB 3: VTAB 15:SL$ = " 1": PRINT "CART
   E SERIE DANS LE CONNECTEUR N°"SL$; CAL
   L - 1000: INPUT " ";R$: IF R$ = "" THEN
   R$ = SL$
60 SL = VAL (R$): VTAB 18:PR$ = "LONGUEUR
   DU PAPIER UTILISÉ "; GOSUB 340: HTAB 1
   8: INPUT "11 OU 12 POUCE ? ";LP$: IF L
   P$ < > "11" AND LP$ < > "12" THEN 60
70 IF LP$ = "11" THEN LP = 104: GOTO 90
80 LP = 120
90 HOME : VTAB 2: HTAB 12: INVERSE : PRINT
   "MODE D'IMPRESSION": NORMAL

```

```

100 VTAB 5: PRINT "- ETENDU (9 CAR/IN, 72 C
   OL).....1": PRINT "- PICA (10 CA
   R/IN, 80 COL).....2"
110 PRINT "- ELITE (12 CAR/IN, 96 COL).....
   .....3": PRINT "- PICA PROPORTIONNEL
   (96 COL).....4"
120 PRINT "- ELITE PROPORTIONNEL (100 COL).
   .....5": PRINT "- 1/2 CONDENSE (13,4
   CAR/IN, 107 COL)..6"
130 PRINT "- CONDENSE (15 CAR/IN, 120 COL).
   .....7": PRINT "- ULTRA-CONDENSE (17
   CAR/IN, 132 COL)..8"
140 VTAB 22: HTAB 10: PRINT "VOTRE CHOIX (1
   -> 8) ? *; GET CH$: PRINT CH$:CH = VA
   L (CH$): IF CH < 1 OR CH > 8 THEN 140
150 PRINT CHR$(4)*PR$SL: ON CH GOSUB 160,
   170,180,190,200,210,220,230: GOTO 250
160 PRINT CHR$(27)*"n": PRINT CHR$(9)*"72N"
   : RETURN
170 PRINT CHR$(27)*"N": PRINT CHR$(9)*"80N"
   : RETURN
180 PRINT CHR$(27)*"E": PRINT CHR$(9)*"96N"
   : RETURN
190 PRINT CHR$(27)*"p": PRINT CHR$(9)*"98N"
   : RETURN
200 PRINT CHR$(27)*"P": PRINT CHR$(9)*"100N"
   *: RETURN
210 PRINT CHR$(27)*"e": PRINT CHR$(9)*"107N"
   *: RETURN
220 PRINT CHR$(27)*"q": PRINT CHR$(9)*"120N"
   *: RETURN
230 PRINT CHR$(27)*"Q": PRINT CHR$(9)*"132N"
   *: RETURN
240 :
241 REM *****
242 REM * DEP.HAUT FEUILLE *
243 REM *****
244 :
250 PRINT CHR$(29); CHR$(65); CHR$(64);
260 FOR I = 1 TO LP: PRINT CHR$(64);: NEXT
   : REM REGLAGE LONG.FEUILLE
270 :
271 REM *****
272 REM * ARRET BAS FEUIL. *
273 REM *****
274 :
280 PRINT CHR$(67); CHR$(64);
290 FOR I = 1 TO 12: PRINT CHR$(64);: NEXT
   : REM REGLAGE NOMBRE DE LIGNES BLANCHE
   S POUR SAUT DE PAGE
300 :
301 REM *****
302 REM * DEPART FEUILLE S.*
303 REM *****

```

```

304 :
310 PRINT CHR$ (65); CHR$ (64); CHR$ (30)
320 HOME : PRINT CHR$ (4)*PR$: VTAB 2:PR$
  = "FORMATAGE TERMINE": INVERSE : GOSUB
  340: NORMAL
330 VTAB 6:PR$ = "FAITES MAINTENANT :": GOS
  UB 340: VTAB 10:PR$ = "PREN" SLOT, <RETUR
  N>": GOSUB 340: VTAB 12:PR$ = "LOAD NOM
  .PROGRAMME, <RETURN>": GOSUB 340: VTAB 1
  4:PR$ = "LIST, <RETURN>": GOSUB 340: END

340 W = INT ((40 - LEN (PR$)) / 2): HTAB W:
  PRINT PR$: RETURN : REM CENTRAGE TITRE
  S

```

GET.TIME

* J'ai un problème ! Débutant en informatique, je me heurte à un problème dont je n'ai trouvé la solution sur aucun de mes livres.

Comment faire en sorte qu'un INPUT ne dure qu'un certain temps ? (pour jouer sur des temps de réponses, par exemple).

Pourriez-vous me fournir la réponse, ou m'indiquer où la découvrir ?

Daniel D. (25000 BESANÇON)

TM Cette petite routine devrait faire votre bonheur...

```

100 REM * GET.TIME

110 TEXT : PRINT CHR$ (21): HOME :D$ = CHR$
  (4):R$ = CHR$ (13):G$ = CHR$ (7):NS =
  B:CYCLES = 70: REM NBR.SECONDE ET CYCLE
  S SELON LE MICROPROCESSEUR (EXP.65C02)
120 HTAB 4: PRINT "LE MOT ";: INVERSE : PRI
  NT "AUTOMATE";: NORMAL : PRINT " VIENT.
  ..": HTAB 4: VTAB 4: PRINT "....D'ARABI
  E (A)": HTAB 4: VTAB 6: PRINT "....DE
  GRECE (B)"
125 HTAB 4: VTAB 8: PRINT "....DE CHINE (
  C)": HTAB 4: VTAB 11: PRINT "VOUS AVEZ
  8 SECONDES POUR REpondre"
130 HTAB 16: VTAB 14: FLASH : PRINT "A";: N
  ORMAL : PRINT ", ";: FLASH : PRINT "B";:
  NORMAL : PRINT " ou ";: FLASH : PRINT
  "C": NORMAL : PRINT G$: VTAB 20
131 :
132 REM *****
133 REM * "CHRONO SECONDES" *
134 REM *****
135 :
140 C = C + 1: IF C / 30 = INT (C / 30) THE
  N FOR Z = 1 TO 3:B = PEEK ( - 16336): N

```

TM

```

EXT : FOR P = 1 TO CYCLES: NEXT : PRINT
  ".":S = S + 1: IF S = NS THEN 200
160 IF PEEK ( - 16384) < 127 THEN 140: REM
  TOUCHE PRESSEE?
165 :
166 REM *****
167 REM * SAISIE IMMEDIATE *
168 REM *****
169 :
170 GET CAR$: PRINT G$: HTAB 12: VTAB 20
180 IF CAR$ = "B" OR CAR$ = "b" THEN PRINT
  "EN EFFET.": GOTO 210
190 PRINT "REponse FAUSSE!...": END
200 PRINT G$G$: HTAB 10: VTAB 20: PRINT "TE
  MPS LIMITE, TROP TARD!"
210 END

```

ON SGN(x) GOTO...

* Peut-on utiliser SGN(x) GOTO en Applesoft ? Il me semble que cela pourrait être intéressant dans certains programmes...

G. NIVEAU (03200 VICHY)

TM Voici un exemple :

```

10 HOME :X = 5
20 X = X - 1
30 ON SGN (X) + 1 GOTO 50,60
40 PRINT "NEGATIF": GOTO 70
50 PRINT "NEUTRE": GOTO 20
60 PRINT "POSITIF": GOTO 20
70 END

```

```

$RUN
POSITIF
POSITIF
POSITIF
POSITIF
NEUTRE
NEGATIF

```

Ne pas oublier que SGN
donne :

1	si + (positif)
0	si 0
-1	si - (négatif)

* Pour répondre à la question de Jean S. (04860 PIERREVENT) sur la SEIKOSHA-GP 100A, parue dans T.M. n°6, page 57, je peux vous indiquer que, pour obtenir une impression sur N colonnes (N <= 80), il faut faire : POKE - 12312,N (avec 0 < N <= 80) C'est ainsi que je me suis constitué un fichier EXEC avec les commandes suivantes :

```

POKE 33,33      LIST
POKE - 12312,80 PRE0
PRE1            POKE 33,40

```

qui me permet d'obtenir un listing sur imprimante en 80 colonnes.

Stéphane L. (45120 CHALETTE)

Vous avez écrit à TREMPLIN MICRO



* *Monsieur Nestor, grosses bises, avec mes meilleurs vœux pour 1986. Je cherche un logiciel qui me permettrait d'utiliser ma SCRIBE à fond, surtout la couleur. J'ai déjà THE PRINT SHOP, mais ce n'est pas géant. D'avance beaucoup de mimis, et merci ! Au fait, voulez-vous gagner 30 F en un quart d'heure ? Simple : le ruban de la SCRIBE coûte 69 F (à Lyon). Or, il suffit d'ouvrir le boîtier (les encoches sont d'accès très facile) et de décrocheter le ressort du ruban. Ensuite — de préférence pendant un cours — on rembobine avec un crayon (pour l'inertie) et on remet le ressort en place. Reste à remettre la cassette dans l'imprimante en prenant soin de couvrir la tête de la cellule photo-électrique qui surveille le bon déroulement du ruban (un bout de sparadrap fait l'affaire !). Gain : 34,50 F pour 15'30 de "travail" ! Super, non ?*

Alexandra (69007 LYON)

TM Ah ! ce que je suis content ! vrai : je retrouve mes dix-huit ans, et je deviens jaloux de ton interlocuteur préféré (quand même), un Apple IIc. Mais je n'aurai pas droit aux mimis because... rien à te proposer. Si les Lecteurs de Tremplin Micro ont de bons tuyaux, je te promets de te les faire suivre, foi de Nestor !

* *J'aimerais bénéficier de la double haute résolution avec mon Apple IIc. Comment faire ?*

Nicolas N. (57740 LONGUEVILLE)

TM Il est probable que T.M. se penchera un jour sur ce problème. Notez qu'il existe un logiciel sur les disquettes accompagnant la Carte Chat Mauve vendue pour le IIc.

* *Je suis à la recherche d'un logiciel (Apple IIe) ainsi que du câblage périphérique concernant l'émission-réception en C.W. (morse) et RTTY (pas du matériel, mais des informations). Une annonce ou un regroupement de personnes ayant le même intérêt pour ce mode de transmission, est chose concevable... si vous m'aidez en ce sens.*

Christian G. (60140 LIANCOURT) — (Ecrire au journal).

TM Voilà qui est fait... Tenez-nous au courant et merci !

MANETTES DE JEUX

* *Le petit programme ci-contre aurait dû prendre place dans le numéro 6 de T.M. (voir courrier, page*

67), mais il n'est jamais trop tard pour bien faire.

TM L'état des boutons des manettes de jeux est vérifiable en lisant la zone d'entrées/sorties de l'Apple, aux adresses \$C061, \$C062 : PEEK(49249) et PEEK(49250). Le résultat est supérieur à 127 si un des boutons est pressé. En d'autres termes, le BIT 7 passe de 0 à 1.

```

100 REM * MANETTES EN MODE GRAPHISME
110 TEXT : NORMAL : PRINT CHR$(21): HOME
    : HGR : HTAB 1: VTAB 22: PRINT "ESC) P
    ETOUR MENU":D$ = CHR$(4):G$ = CHR$(7
    ):ML = 279:MH = 150: PRINT G$: REM MAX
    .ECRAN
120 :
130 REM *****
140 REM * LECTURE PARAMET. *
150 REM *****
160 :
170 X = PDL(0):Y = PDL(1): REM COORDONNE
    ES MANETTE
180 IF PEEK(-16287) > 127 THEN PRINT G$
    :S = 1: HTAB 1: VTAB 22: PRINT "BOUTON
    2 ";; INVERSE : PRINT "POINT/TRACE":
    NORMAL
190 IF PEEK(-16286) > 127 THEN PRINT G$
    : HTAB 1: VTAB 22: PRINT "BOUTON 1 ";;
    INVERSE : PRINT "EFFACEMENT";; NORMAL
    : PRINT " ";; HGR :DR = 0
200 IF S = 1 THEN DR = ABS(DR - 1):S = 0:
    REM INVERSEUR TRACE OU NON (BOUTON 2)
210 IF PEEK(49152) = 155 THEN PRINT D$*RU
    N MENU": REM ESC)
220 :
230 REM *****
240 REM * TRACE AUX COORDS *
250 REM *****
260 :
270 X = INT(X): IF X > ML THEN X = ML: RE
    M LIMITE ECRAN
280 Y = INT(Y): IF Y > MH THEN Y = MH
290 IF DR = 0 AND P = 0 THEN HCOLOR= 0: HP
    LOT AX,AY: HCOLOR= 3: HPLLOT X,Y: REM O
    N EFFACE LE POINTEUR LORS D'UN DEPLACE
    MENT
300 IF DR = 1 THEN FOR I = 1 TO 2: HPLLOT A
    X + 1,AY + I TO X + 1,Y + I: NEXT I: R
    EM TRACE DEUX DROITES POUR L'EPAISSEUR
310 HTAB 23: VTAB 22: PRINT X" ";; HTAB 33
    : VTAB 22: PRINT Y" ";; REM ECRITURE D
    ES COORDONNEES
320 AX = X:AY = Y: GOTO 170: REM ON CONSER
    VE X,Y POUR UN TRACE AX->X AY->Y
    
```

ECRAN DE 80 COLONNES

MÉMO ET COPIE

MATÉRIELS : Apple IIe ou IIc — Imprimante DMP ou IMAGEWRITER (I et II)

OBJET : Mémorisation des 24 lignes d'un écran de 80 colonnes sous forme de fausses variables (blocs de 80 caractères suivis d'un octet ZÉRO).

ADRESSE : 8192 (\$2000).

INTÉRÊT : Permet de comprendre le mécanisme de lecture et de mémorisation d'une page-écran. Autorise le réaffichage de ce même écran... mais la 24^e ligne provoque un scroll (déroulement des lignes d'écran vers le haut).

300 :	A9	20	LDA	£\$20	}	Initialisation de l'adresse de base pour la mémorisation des données (\$2000 = 8192)
302 :	85	07	STA	\$07		
304 :	A9	00	LDA	£\$00	}	CV : Position verticale du curseur
306 :	85	06	STA	\$06		
308 :	85	25	STA	\$25	}	BASCALC range dans \$28-\$29 l'adresse de base
30A :	E6	25	INC	\$25		
30C :	20	C1	JSR	\$FBC1	}	Initialisation de la boucle avec Y
30F :	A0	00	LDY	£\$00		
311 :	8D	55	STA	\$C055	}	Lecture page 2
314 :	B1	28	LDA	(\$28),Y		
316 :	C9	20	CMP	£\$20	}	Lecture de l'octet à l'adresse indiquée dans \$28-\$29 + Y Est-ce plus grand que \$20 ?
318 :	80	03	BCS	\$031D		
31A :	18		CLC		}	Si plus grand ou égal \$20, on passe ou alors on ajoute \$C0
31B :	69	C0	ADC	£\$C0		
31D :	91	06	STA	(\$06),Y	}	Mémo à l'adresse indiquée dans \$6-7 + Y Plus un pour adresse mémo
31F :	E6	06	INC	\$06		
321 :	D0	02	BNE	\$0325	}	Si différent de 0, pas de retenue Dans le cas contraire, + 1 pour partie haute
323 :	E6	07	INC	\$07		
325 :	8D	54	STA	\$C054	}	Lecture page 1
328 :	B1	28	LDA	(\$28),Y		
32A :	C9	20	CMP	£\$20	}	Même processus que pour la page 2 (plus haut).
32C :	80	03	BCS	\$0331		
32E :	18		CLC		}	Y = Y + 1 pour la boucle Est-on à 40 (\$28) ?
32F :	69	C0	ADC	£\$C0		
331 :	91	06	STA	(\$06),Y	}	Non : on fait un autre tour.
333 :	C8		INY			
334 :	C0	28	CPY	£\$28	}	Oui : on stocke le caractère 0 (fin de chaîne) à la fin de la ligne
336 :	D0	D9	BNE	\$0311		
338 :	A9	00	LDA	£\$00	}	Retenue annulée Lecture de la mémoire \$6 + \$29 (28 et 1 octet nul)
33A :	91	06	STA	(\$06),Y		
33C :	18		CLC		}	Écriture mémoire \$6 S'il n'y a pas de retenue, saut.
33D :	A5	06	LDA	\$06		
33F :	69	29	ADC	£\$29	}	Dans le cas contraire, plus 1 pour partie haute
341 :	85	06	STA	\$06		
343 :	90	02	BCC	\$0347	}	A-t-on lu les 24 lignes (\$18) ? Non : encore un tour.
345 :	E6	07	INC	\$07		
347 :	A5	25	LDA	\$25	}	
349 :	C9	18	CMP	£\$18		
34B :	D0	BD	BNE	\$030A		

(ICI LA MÉMORISATION EST TERMINÉE)

COPIE DE L'ÉCRAN (suite)

34D :	48			PHA		}	On empile les valeurs des registres avant d'utiliser l'imprimante			
34E :	8A			TXA						
34F :	48			PHA						
350 :	98			TYA		}	C'est le PR£1 du Basic			
351 :	48			PHA						
352 :	A9	01		LDA	£\$01					
354 :	20	95	FE	JSR	\$FE95	}	Les codes envoyés à l'imprimante sont stockés à partir de \$396 (à l'envers)			
357 :	A2	0C		LDX	£\$0C					
359 :	BD	95	03	LDA	\$0395,X					
35C :	20	ED	FD	JSR	\$FDED	}	Paramètres de l'adresse de base (mémorisation)			
35F :	CA			DEX						
360 :	D0	F7		BNE	\$0359					
362 :	A2	20		LDX	£\$20	}	Nombre de lignes (24)			
364 :	86	07		STX	\$07					
366 :	A2	00		LDX	£\$00					
368 :	86	06		STX	\$06	}	Adresse de la ligne dans A et Y, puis affichage par STROUT (la chaîne se termine par 0 ou par ")			
36A :	A2	18		LDX	£\$18					
36C :	86	08		STX	\$08					
36E :	A5	06		LDA	\$06	}	+ retour chariot, à remplacer par EA EA EA EA EA avec certaines imprimantes			
370 :	A4	07		LDY	\$07					
372 :	20	3A	DB	JSR	\$DB3A					
375 :	A9	0A		LDA	£\$0A	}	Moins 1 au nombre de lignes Si zéro... terminé			
377 :	20	ED	FD	JSR	\$FDED					
37A :	C6	08		DEC	\$08					
37C :	F0	0D		BEQ	\$038B	}	Annulation de la retenue			
37E :	18			CLC						
37F :	A5	06		LDA	\$06					
381 :	69	51		ADC	£\$51	}	+ 81 (\$51) pour lire la ligne suivante			
383 :	85	06		STA	\$06					
385 :	90	02		BCC	\$0389					
387 :	E6	07		INC	\$07	}	On n'a pas zéro donc on saute			
389 :	D0	E3		BNE	\$036E					
38B :	A9	00		LDA	£\$00					
38D :	20	95	FE	JSR	\$FE95	}	PR£0 du Basic			
390 :	68			PLA						
391 :	A8			TAY						
392 :	68			PLA		}	Récupération des valeurs empiées			
393 :	AA			TAX						
394 :	68			PLA						
395 :	60			RTS		}	Retour au Basic... mais il faudra un RESET			
396 :	0D	30	31	30	4C			1B	}	Codes envoyés à l'imprimante ; lus à l'envers, ils donnent : CHR\$(9) 80N ; CHR\$(27) E ; CHR\$(27) L010 ; CHR\$(13)
39C :	45	1B	4E	30	38			09		

Si vous installez la courte routine ci-après, un CALL 930 provoquera le réaffichage des 24 lignes d'écran (ou des 23 lignes... sans SCROLL, avec \$3AE = \$17)

3A2 : 20 58 FC A2 20 86 07 A2 00 86 06 A2 18 3BC : 18 A5 06 69 51 85 06 90 02 E6 07 D0 E8
86 08 A5 06 A4 07 20 3A DB C6 08 F0 0D 86 25 20 22 FC 60

Les heureux possesseurs de

MINIE

ont déjà reçu un premier programme
d'application utilisant les ressources du
logiciel de Claude Aubry.

Prochainement, ils pourront participer à un
jeu qui récompensera précisément les
auteurs des deux meilleures applications.

Chasseur d'Images

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



**Chez votre
marchand de journaux !**