

# *tremplin micro*

*Demandez l'heure  
à votre Apple !*

*Un indicatif  
musical*

*La pseudo page 3*

*Un calendrier  
perpétuel*

*Dessinez des étoiles !  
Condensez vos écrans !*



Bimestriel — Première année  
15 Novembre 1985 - 8 Janvier 1986  
Numéro 5 — 30 F - 231 FB - 10 FS



---

# tremplin micro

# 5

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

## BIMESTRIEL

Le numéro : 30 F  
Abonnement d'un an 180 F  
(6 numéros)

## EDITIONS JIBENA

Direction-Rédaction :  
Editions JIBENA  
Guy-HACHETTE

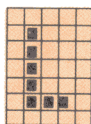
La Petite Motte — Senillé  
86100 CHÂTELLERAULT.

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.

## Vous êtes formidables !



ES LECTRICES et LECTEURS de *Tremplin Micro* sont réellement formidables, et j'en ai quotidiennement la preuve, en lisant leurs lettres ou en leur parlant... au téléphone.

J'en connais pourtant trois qui vont m'accuser de ne voir que le bon côté des choses, et de ne point publier leurs critiques. Pourquoi, en vérité, vanterais-je les "mérites" que s'attribue l'un d'eux ; heureux de pirater logiciels et disquettes, et de photocopier — dit-il — les soixante pages d'une revue inintéressante (la nôtre)... plutôt que de l'acheter. Que de temps et de papier perdus pour des informations dépourvues d'intérêt ! Je n'aimerais pas être à la place du cochon de payant (car il existe) victime de cette inutile et onéreuse hémorragie de feuilles volantes !

Oui, vous êtes formidables, et nous commençons dans ce numéro, à publier certains de vos programmes. Nous avons d'ailleurs, dans ce domaine, beaucoup d'autres projets. Nous en reparlerons bientôt.

La majorité des Lectrices et Lecteurs de *Tremplin Micro* est favorable à la présence, dans la revue, d'une publicité informative (voir formative).

*Tremplin Micro* ouvrira bientôt ses colonnes aux annonceurs, mais le nombre des pages rédactionnelles n'en souffrira pas, au contraire. La publicité doit apporter un plus... ou ne pas être.

Une bonne nouvelle pour 1986 : votre magazine paraîtra 6 fois par an (au lieu de 5). Le prochain numéro sera mis en vente le 9 janvier.

Et maintenant, place à la programmation !

Amicalement

GUY-HACHETTE.



# Sommaire

# 5

## Initiation

- USR(?) ..... 24
- La pseudo page 3 graphique ..... 27
- Déplacement et comparaison des zones de mémoire ..... 51

## Des trucs

- Trois mystérieuses instructions de ProCODE ..... 19

## Micro-basic

- Un calendrier perpétuel proposé par Michel Edelin ..... 36

## What time is it Miss Mouse ?

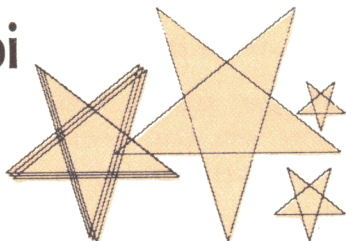
- par Claude Aubry •

Demandez l'heure à la souris  
APPLE ..... 3

## GRAPHISME

Tracer un cadre dans une page graphique ..... 46

## Dessine-moi des étoiles



Une méthode à la fois simple et efficace pour tracer des étoiles ..... 49

## PROGRAMMES UTILITAIRES

- Correction avancée ..... 11
- Des pokes utiles pour modifier le catalog ProDOS ..... 29
- Ecrans condensés, un programme de Maurice Chavelli .. 40
- RAM.COPIE ou comment transférer vos programmes-disquette sur /RAM. 52

MUSIQUE • par Jacques Saussard •  
Un indicatif musical ..... 30

## ESSAIS DE LOGICIELS

- Il y a un abonné au numéro que votre Apple a demandé ..... 10 (Version Com — Telplus — Transcan)
- Gestion II de Version Soft ..... 23

SON • avec Richard Thibert •  
De quoi mettre une certaine animation dans vos jeux ..... 14

## LISEZ AUSSI :

- Livres ..... 8
- Questions et réponses ..... 8
- Votre bibliothèque informatique ..... 20—39—48

## NOS LECTEURS ONT DES IDÉES

- Caractères français transposés en page graphique par M. Devaux ..... 33
- Votre fiche n°5 ..... 21—22
- Le courrier des lecteurs ..... 55
- Index ..... 60
- Bulletin de commande et d'abonnement ..... 62

## TREMPIN MICRO

Le numéro 6 paraîtra le 9 janvier.



# What time is it Miss Mouse ?

par Claude AUBRY



Quelle heure est-il ? Demandez-le à la **souris APPLE**, elle vous répondra en affichant constamment, précisément, et **imperturbablement** l'heure, la minute et la seconde dans un coin de votre écran.

Pour afficher le compteur, il suffit d'appeler le programme par CALL 768. Il restera présent sur l'écran tant que vous ne taperez pas CTRL RESET, et comptera soigneusement le temps quelle que soit votre activité (il marque tout de même une pause durant les accès disque, le temps de respirer, en somme !).

Pour le mettre à l'heure, il faut "POKER" les valeurs convenables aux adresses "COMPTEUR".

## LE PROGRAMME

Le court programme en assembleur 6502 (il "tient" dans la zone \$300-3CF) utilise la possibilité pour l'interface souris de générer des interruptions tous les soixantièmes de seconde lorsqu'on choisit le mode interruption à chaque rafraîchissement d'écran.

Lorsque ce mode est choisi, l'ordinateur interrompt la tâche qu'il était en train d'accomplir et va exécuter un programme dont l'adresse est donnée en \$3FE-3FF. Il suffit de fournir à

ce vecteur l'adresse de notre programme d'horloge et le tour est gentiment joué.

Cette horloge peut être utilisée avec n'importe quel programme ne faisant pas appel à la souris et j'espère qu'elle vous amusera : vous pourrez compter le temps passé à répondre à un INPUT, fabriquer un réveil, etc.

Naturellement, toute routine utilisant la plage \$300-3CF détraquera irrémédiablement votre Apple-chrono...



## Assemblage par ProCODE

```

0 * _____ *
1 | WHAT TIME IS IT MISS MOUSE ? |
2 | |
3 | |
4 | |
5 | Une utilisation originale de la
6 | souris pour Apple IIc et IIe.
7 | Claude AUBRY (septembre 1985) |
8 * _____ *
9 *
10 TMP EQU $6 ; POINTEUR TEMPORAIRE
11 *
12 SETMOUSE EQU $12
13 SERVEMOUSE EQU $13
14 INITMOUSE EQU $19
15 *
16 WARM EQU $3D0 ; RESTART PRODOS
17 VECINT EQU $3FE ; VECTEUR interruption
18 ECRAN EQU $400 ; Début de l'écran
19 *
20 COUT EQU $FDED ; SORTIE DE 1 CARACTERE
21 *
22 * _____ *
23 | DÉBUT DU PROGRAMME |
24 * _____ *
25 *
26 * Programme appelé par CALL 768
27 *
28 ORG $300
29 *
300 : 78 30 BEGIN SEI ; Interdit les interruptions
301 : 20 35 03 31 JSR CHECKSLOT ; OÙ EST LA CARTE SOURIS
304 : A9 74 32 LDA £ INTERUP ; LSB programme inter
306 : A2 03 33 LDX £ INTERUP ; HSB programme inter
308 : 8D FE 03 34 STA VECINT ; LSB VECTEUR interruption
30B : 8E FF 03 35 STX VECINT+1 ; HSB
36 *
30E : A0 19 37 LDY £INITMOUSE ; Pour syncro
310 : 20 1C 03 38 JSR CALLCARD
313 : A0 12 39 LDY £SETMOUSE ; Pour mode
315 : A9 09 40 LDA £%00001001 ; Sélection mode inter
317 : 20 1C 03 41 JSR CALLCARD ; Par screen refresh
42 *
31A : 58 43 CLI ; Rétablit interruptions
31B : 60 44 RTS ; Retour basic
45 *
46 * _____ *
47 | SOUS PROGRAMME CALLCARD |
48 * _____ *
49 *
31C : 48 50 CALLCARD PHA ; on sauve l'accu
31D : B1 06 51 LDA (TMP),Y ; recherche offset
31F : AE 30 03 52 LDX CN ; adresse (HSB) carte
322 : AC 31 03 53 LDY N0 ; et N° de slot
325 : 8D 33 03 54 STA TOCARD+1 ; vectorisé
328 : 8E 34 03 55 STX TOCARD+2 ; pour saut indirect
32B : 68 56 PLA ; recupère accu
32C : 20 32 03 57 JSR TOCARD ; saut indirect à la carte
32F : 60 58 RTS ; et retour
59 *

```

(Suite page 5)



```

330 : 00          60 CN          HEX 00          ; stock HSB AD carte
331 : 00          61 N0         HEX 00          ; stock Numéro slot
332 : 4C 00 00   62 *
63 TOCARD      JMP $0000      ; opérand modifié par prog
64 *
65 * _____ *
66 |
67 | Recherche du slot où est la carte MOUSE
68 |
69 |
70 | Doc. page 54 Apple mouse user's manual
71 | _____ *
72 *
335 : A2 07      73 CHECKSLOT LDX £$07      ; on teste 7 slots
337 : A9 00      74          LDA £$00      ; on commence la recherche
339 : 85 06      75          STA TMP       ; aux adresses hautes
33B : A9 C8      76          LDA £$C8      ; soit $C700
33D : 85 07      77          STA TMP+1    ; en fait C800 décrémente
33F : C6 07      78 LOOP1     DEC TMP+1    ; on décrémente
341 : CA         79          DEX         ; X sert de compteur
342 : 30 21      80          BMI EXIT     ; si 0 erreur 0 carte
344 : A0 0C      81          LDY £$0C     ; octet de signature
346 : B1 06      82          LDA (TMP),Y  ; doit être là
348 : C9 20      83          CMP £$20     ; et égal à $20
34A : D0 F3      84          BNE LOOP1    ; pas là ! slot précédent
34C : A0 FB      85          LDY £$FB     ; 2' test de signature
34E : B1 06      86          LDA (TMP),Y  ; en CN $FB
350 : C9 D6      87          CMP £$D6     ; est-ce bien $D6 ?
352 : D0 EB      88          BNE LOOP1    ; non slot précédent
354 : A5 07      89          LDA TMP+1    ; oui on sauve l'adresse
356 : 8D 34 03   90          STA TOCARD+2 ; pour plus tard
359 : 8D 30 03   91          STA CN       ; utilisé par callcard
35C : 0A         92          ASL         ; on décale
35D : 0A         93          ASL         ; de 4 octets
35E : 0A         94          ASL         ; pour faire passer
35F : 0A         95          ASL         ; HSB en LSB
360 : 8D 31 03   96          STA N0      ; numéro de slot
363 : D0 0E      97          BNE START    ; et retour
98 *
365 : A0 05      99 EXIT     LDY £$05     ; message d'erreur
367 : B9 CA 03   100 ERRLOOP LDA NOCARDMSG-1,Y ; si pas de carte
36A : 20 ED FD   101          JSR COUT     ; on envoie le message
36D : 88         102          DEY         ; ECARD et on arrête
36E : D0 F7      103          BNE ERRLOOP
104 *
370 : 4C D0 03   105 WARMSTART JMP WARM
106 *
373 : 60         107 START    RTS
108 *
109 * _____ *
110 |
111 | PROGRAMME D'INTERRUPTION
112 |
113 | _____ *
114 *
115 * Ce programme est effectué 60 fois
116 * par seconde.
117 *
118 INTERUP     EQU *
119          SEI         ; On interdit les interruptions
120 *
374 : 78         121          PHA         ; SAUVEGARDE DU CONTEXTE
375 : 48         122          TXA
376 : 8A

```

(Suite page 6)

377 :	48	123	PHA		
378 :	98	124	TYA		
379 :	48	125	PHA		
		126 *			
37A :	A0 13	127	LDY	£SERVEMOUSE	; La souris est-elle cause inter ?
37C :	20 1C 03	128	JSR	CALLCARD	
37F :	EE C4 03	129	INC	SOIXANT	; Compte les 60e de seconde
382 :	AD C4 03	130	LDA	SOIXANT	; Cela fait-il une seconde ?
		131 *			
385 :	C9 3B	132	CMP	£59	
387 :	D0 29	133	BNE	SKIP	; Pas encore : on saute
		134 *			
389 :	A9 00	135	LDA	£0	; Oui : on remet à 0
38B :	8D C4 03	136	STA	SOIXANT	; Le compteur des soixantièmes
		137 *			
38E :	A2 05	138	LDX	£5	; 6 DIGITS
		139 *			
		140	LOOPTIME	EQU *	
		141 *			
390 :	FE C5 03	142	INC	TEMPS,X	; Et on incrémente les unités
393 :	BD C5 03	143	LDA	TEMPS,X	; Combien d'unités ?
		144 *			
396 :	C9 3A	145	CMP	£'9+1	; + de 9 ?
398 :	90 18	146	BCC	SKIP	; Non : on saute la suite
		147 *			
39A :	A9 30	148	LDA	£'0	; Sinon on remet les unités...
39C :	9D C5 03	149	STA	TEMPS,X	; ... à 0...
39F :	CA	150	DEX		
3A0 :	FE C5 03	151	INC	TEMPS,X	; ... et on incrémente les dizaines
3A3 :	BD C5 03	152	LDA	TEMPS,X	; Combien de dizaines ?
		153 *			
3A6 :	C9 36	154	CMP	£'6	; + de 6 ?
3A8 :	D0 08	155	BNE	SKIP	; Non : saute la suite
		156 *			
3AA :	A9 30	157	LDA	£'0	; Sinon on remet les DZ à 0
3AC :	9D C5 03	158	STA	TEMPS,X	
		159 *			
3AF :	CA	160	DEX		; Les 6 chiffres ?
3B0 :	10 DE	161	BPL	LOOPTIME	; Non : on boucle
		162 *			
		163 *			
		164	SKIP	EQU *	
		165 *			
		166 *			
		167			
		168			
		169			
		170 *			
		171 *			
		172 *			On peut ruser et supprimer l'affichage (mais non le cumul du temps)
		173 *			en poquant un code bidon à la place du STA ECRAN,X en le chan-
		174 *			geant en LDA par exemple.
		175 *			
		176 *	ainsi	POKE 951,189	; arrête l'affichage
		177 *		POKE 951,157	; LE RÉTABLIT
		178 *			
3B2 :	A2 05	179	LDX	£5	; 6 CHIFFRES
3B4 :	BD C5 03	180	LDA	TEMPS,X	; On récupère...
3B7 :	9D 00 04	181	STA	ECRAN,X	; ... et on affiche
3BA :	CA	182	DEX		; Encore ?
3BB :	10 F7	183	BPL	LOOP	; Oui
		184 *			
3BD :	68	185	PLA		

### PROGRAMME D'AFFICHAGE

(Suite page 7)



3BE : 98	186	TYA		; ON RESTITUE LE CONTEXTE
3BF : 68	187	PLA		
3C0 : 8A	188	TXA		
3C1 : 68	189	PLA		
	190 *			
3C2 : 58	191	CLI		; On rétablit les inter...
3C3 : 40	192	RTI		; ... et on repart
	193 *			
	194 *			
	195			
	196			
	197			
	198 *			
	199 *			
3C4 : 00	200	SOIXANT	HEX	00
	201 *			
3C5 : 30 30 30	202	TEMPS	ASC	'000000' ; POKE 965+Y,X pour mise à l'heure
30 30 30	203 *			
	204 *			
	205			
	206 *			
	207 *			
3CB : C4 D2 C1	208	NOCARDMSG	ASC	"DRAC£" ; MESSAGE si pas de carte
C3 A3				

### COMPTEURS

### MESSAGE ERREUR

Si vous ne possédez pas d'assembleur, vous pouvez saisir les 26 lignes ci-après. Tapez, d'abord **CALL-151**, suivi de RETURN, puis la valeur indiquée, sans oublier un RETURN à la fin de chaque ligne.

Quand vous aurez terminé, enregistrez votre programme sur disquette par un BSAVE MOUSE.TIMER,A768,L208 (A\$300,L\$D0).

### VERIFICATION

(en mode direct, après CTRL-RESET) :

T=0 : FOR I = 768 TO 975 : T = T + PEEK(I) : NEXT I : PRINT T

Le résultat devra être 21212.

```

300 : 78 20 35 03 A9 74 A2 03
308 : 8D FE 03 8E FF 03 A0 19
310 : 20 1C 03 A0 12 A9 09 20
318 : 1C 03 58 60 48 B1 06 AE
320 : 30 03 AC 31 03 8D 33 03
328 : 8E 34 03 68 20 32 03 60
330 : 00 00 4C 00 00 A2 07 A9
338 : 00 85 06 A9 C8 85 07 C6
340 : 07 CA 30 21 A0 0C B1 06
348 : C9 20 D0 F3 A0 FB B1 06
350 : C9 D6 D0 EB A5 07 8D 34
358 : 03 8D 30 03 0A 0A 0A 0A
360 : 8D 31 03 D0 0E A0 05 B9

```

```

368 : CB 03 20 ED FD 88 D0 F7
370 : 4C D0 03 60 78 48 8A 48
378 : 98 48 A0 13 20 1C 03 EE
380 : C4 03 AD C4 03 C9 3B D0
388 : 29 A9 00 8D C4 03 A2 05
390 : FE C5 03 BD C5 03 C9 3A
398 : 90 18 A9 30 9D C5 03 CA
3A0 : FE C5 03 BD C5 03 C9 36
3A8 : D0 08 A9 30 9D C5 03 CA
3B0 : 10 DE A2 05 BD C5 03 9D
3B8 : 00 04 CA 10 F7 68 98 68
3C0 : 8A 68 58 40 00 30 30 30
3C8 : 30 30 30 C4 D2 C1 C3 A3

```

### MOUSE.DEMO

```

1 PRINT CHR$(4);"PR£0"
5 PRINT CHR$(13); CHR$(4); "BRUN MOUSE TIMER"
10 HOME
15 VTAB 6: HTAB 7: PRINT "PROGRAMME DE MISE À L'HEURE"
20 VTAB 12: PRINT "Donnez l'heure sous la forme HHMMSS"
25 VTAB 17: HTAB 17

```

```

30 FOR I = 0 TO 5
40 GET H$: IF H$ < "0" OR H$ > "9" THEN 40
45 PRINT H$;
50 POKE 965 + I, ASC(H$)
60 NEXT

```

## QUESTIONS

Nous répondons ici aux questions de lecteurs néophytes.

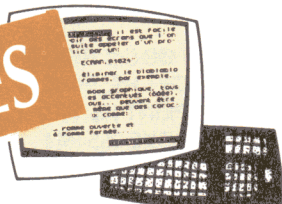
- L'Apple IIe 65C02 est-il vraiment compatible avec ses prédécesseurs ? Est-il plus intéressant et plus rapide ?

Il est entièrement compatible avec l'Apple IIc dont il utilise notamment les nouveaux logiciels basés sur le concept graphique souris (à condition d'être équipé de la souris et d'une carte 80 colonnes, bien entendu). Pour le reste, Apple annonce une compatibilité à 93% avec les anciens Apple IIe, toutes versions confondues, ce qui n'est pas si mal. Incontestablement, quand il est totalement utilisé, le 65C02 est plus rapide que le 6502. Il dispose en effet de 10 instructions supplémentaires et de 27 nouvelles possibilités d'adressage, autorisant des routines plus courtes.

- Quels sont les pokes permettant d'écrire dans les pages graphiques 1 et 2, puis de les afficher ?

```
10 HGR : HGR2 : TEXT : HOME
20 HCOLOR = 3: POKE 230,32: REM ECRITURE PAGE 1
30 HPL0T 0,0 TO 279,191
40 POKE 230,64: REM ECRITURE PAGE 2
50 HPL0T 0,191 TO 279,0: GOSUB 200:
VTAB 22: PRINT "PAGE 1 AVEC TEXTE"
60 REM PAGE 1 AVEC 4 LIGNES DE TEXTE
70 POKE 49236,0: POKE 49235,0: POKE
49239,0: POKE 49232,0
80 GOSUB 200: REM PAGE 1 SANS TEXTE
90 POKE 49234,0
100 GOSUB 200: REM PAGE 2
110 POKE 49237,0
120 GOSUB 200: REM PAGE 1
130 POKE 49236,0
140 GOSUB 200: REM PAGE 1 AVEC TEXTE
150 POKE 49235,0: HOME : VTAB 22: PRINT
"PAGE 1 AVEC TEXTE"
160 END
200 VTAB 1: CALL — 198: GET R$: PRINT :
RETURN
```

## REponses



## LIVRES



### GRAPHISME ET SON SUR APPLE \*

Avez-vous déjà traité le graphisme sur votre Apple (II+, IIe ou IIc) ? Quelle que soit votre réponse, sachez que, de toute façon, vous apprendrez quelque chose dans le troisième volet de "PRATIQUE DES APPLE", consacré au graphisme et au son.

J.-C. Fantou, auteur de cet ouvrage, mérite des éloges. Voilà de la bonne besogne ! Tout est clair, exact, documenté. Gageons que, grâce aux conseils et routines de ce spécialiste, bon nombre de programmeurs (débutants ou non) vont s'intéresser au graphisme et au son.

Cela nous promet, pour les mois à venir, d'inédits listages, capables de générer de fantastiques images... ou les mesures inattendues de compositeurs en herbe. \* EDITIONS RADIO 9, rue Jacob — 75006 PARIS (145 F)

### LE FUTUR AUJOURD'HUI

On ne présente pas Albert Ducrocq, professeur à la Fondation nationale des Sciences politiques, journaliste, écrivain et chroniqueur écouté à Europe 1.

On ne présente pas davantage son livre, accueilli par des critiques élogieuses dans toute la presse, lorsqu'il a paru, en 1984, chez Plon.

Les 15 années qui vont changer notre vie : l'informatique, l'électronique, les robots, mais aussi la bio-industrie, etc.

Le Futur aujourd'hui, publié en livre de poche, dans la collection Marabout Université, va toucher de nouveaux et nombreux lecteurs... vous peut-être !

Clément RENARD.



# Il y a un abonné au numéro que votre Apple a demandé...

**Branché, câblé : la communication est à l'ordre du jour. Malheureusement, nos beaux ordinateurs n'aiment guère communiquer, et le jour où l'on pourra raccorder une machine au téléphone aussi simplement qu'on branche aujourd'hui une prise de courant ou un moniteur vidéo ne semble pas encore arrivé...**

Connecter son Apple au téléphone c'est, soudain, pouvoir communiquer avec le monde entier comme avec son voisin, pour transmettre, recevoir ou échanger informations, graphismes ou programmes.

Pour cela, il faut disposer d'une interface RS-232 (carte Série) et d'un modem ou, solution de luxe, d'une carte-modem directement enfichable dans l'Apple. A cela il faut ajouter un logiciel de communication sans lequel votre machine serait incapable du moindre dialogue intelligent. Car la communication, ça se gère.

Si certains modems, comme la carte Appletell par exemple, sont directement livrés avec un logiciel performant, il n'en est pas de même pour d'autres. Aussi les concepteurs de softs se sont-ils penchés sur la question et proposent-ils désormais des solutions.

Pour vous, nous venons de tester Version Com, de la turbulente société française Version Soft, et Telplus de Micro-mat, une autre société française.

## Version-Com

Un joli coffret, un manuel copieux, une disquette : il n'en faut pas plus pour gérer les communications sur un Apple. Mais pas n'importe lequel : Version-Com ne reconnaît que deux machines, l'Apple IIc et le nouvel Apple IIe (version 65C02). Si vous pos-

édez un Apple plus ancien, il vous faudra donc retourner à la case départ et passer chez votre revendeur pour faire monter le kit de mise à niveau.

Côté modems, Version-Com est plus souple et supporte tous les modems externes raccordés à la carte SSC Super-Série ou à la sortie série du IIc, ainsi que les modems Apple-Sectrad, Digitelec et Ultec. Il reconnaît également les cartes Appletell, Apple-Cat II (de Novation) et les MID Supercom ou S2. Un bel exploit quand on sait que pratiquement aucune de ces cartes ne fonctionne sur les mêmes paramètres !

Avant d'acheter Version-Com, il sera donc prudent de vous assurer que le matériel dont vous disposez déjà est bien compris dans la liste ci-dessus. Il vous suffira alors de vous reporter au chapitre qui lui est consacré dans la documentation livrée avec le logiciel pour savoir dans quel slot insérer la carte (c'est parfois essentiel !) ou comment régler les micro-switches de votre interface. Les explications de l'auteur, bien que parfois un peu disséminées, ont le mérite d'amener très vite au réglage idéal et donc à un fonctionnement correct... Bravo !

Fidèle au concept de la souris, Version-Com vous obligera à cliquer. Quelques secondes après l'installation du logiciel, une ligne de commandes apparaît en haut de l'écran et permet d'ouvrir des fenêtres pour choisir la fonction

désirée : numérotation automatique, opérations sur fichier, édition, transfert, appel de procédure, etc. On ne jongle donc plus avec des codes-contrôle et cette conception est bien pratique. Version-Com est aussi capable de travailler tout seul. Pour cela, il faut d'abord cliquer l'option *Commencer l'apprentissage*; le programme mémorise alors toutes les opérations que vous exécutez à partir de ce moment : il enregistre la procédure. Laquelle peut ensuite être stockée sur disquette et être exécutée automatiquement par la suite. On retrouve là l'une des fonctions ayant contribué au succès de la carte Appletell, et il faut reconnaître que c'est très pratique, même s'il y a des limitations (par exemple un numéro qui ne répond pas, alors que vous êtes en mode apprentissage).

Version-Com autorise le transfert de fichiers en émission comme en réception, la mémorisation, l'impression et permet, en fait, de travailler à distance avec une autre machine sans se préoccuper de la liaison téléphonique. La seule limitation concernera évidemment la vitesse de transmission : 300, 600 ou 1200 bauds, ce n'est pas bien rapide !

## Communiquer avec qui ?

Version-Com se révèle très utile dans tous les cas où il est nécessaire de relier deux machines, même de marques différentes, pour des échanges d'informations. C'est alors un logiciel souple et fiable.

En revanche, dès que l'on commence à vouloir varier les plaisirs en multipliant le nombre de correspondants, on se heurte à de petits problèmes. Par exemple, ne comptez pas transformer votre

Apple IIc en Minitel : l'interface série du IIc refusant la norme V 23 (75/1200 bps). Même chose, d'ailleurs, sur le IIe et la plupart des modems : seules les cartes Appletell et Novation permettent l'accès aux serveurs à la norme Vidéo-tex. Les serveurs américains (norme Bell 103) sont, quant à eux, réservés aux possesseurs d'une carte Appletell, la seule qui accepte le 300 bauds full duplex. Autant de limitations parfois bien frustrantes, qui s'expliquent davantage par les limitations des modems que par celles du logiciel, évidemment incapable d'appeler des fonctions qui n'existent pas.

L'autre problème rencontré concerne la numérotation et la réponse automatique. Malgré son prix élevé, le modem Apple, fabriqué par Sectrad, ne sait pas composer les numéros ni se connecter automatiquement. Adieu donc les procédures de Version-Com !

Version-Com met enfin en évidence les dramatiques limitations de l'Apple IIc pour ce qui concerne les communications, de nombreuses options, prévues par le programme, étant interdites par les lacunes de la machine.

## Telplus

Produit par Micromat, Telplus ne fonctionne que sur les Apple II et IIe dotés d'une carte Super-Série et d'un modem ou de la carte Adaptel Micromat.

— version série : la SSC Apple doit être raccordée à tout modem conforme aux normes V21 (Transpac 300 bauds), V23 (émission 75 bps, réception 1200 bps) ;

— version Adaptel : cette carte spécifique, reliée à un Minitel, permet d'utiliser le modem intégré du Minitel en mode V22 (1200/1200 bps).

Cette dernière solution est fort attirante pour tous ceux qui habitent en zone annuaire électronique et qui disposent donc d'un Minitel gratuit. La carte Adaptel est annoncée par Micromat au prix de 450 F... Nous ne vous en dirons

### Prix & adresses

**Version-Com** : chez tous les revendeurs Apple. Prix : 900 à 1 000 F.

Version-Soft, 19 rue Ganneron, 75018 Paris. Tél.: (1) 43-87-94-87.

**Telplus** : en vente chez International Computer (26 rue du Renard, 75004 Paris), Microshop (6 rue de Chateaudun, 75009 Paris), IVS (10 rue de Montesson, 95870 Bezons) ou par correspondance chez Micromat. Telplus : 800 F. Transcan : 450 F.

Micromat, BP 63, 93130 Noisy-le-Sec. Tél.: (1) 48-49-16-29.

pas plus sur son fonctionnement car il nous a été impossible de nous la procurer.

Si le coffret de Telplus est prometteur, la doc livrée avec le logiciel est, elle, à la limite de l'indigence. Au lieu d'indiquer en quelques lignes les réglages à effectuer sur la carte Super-Série pour une configuration rapide, on se contente de vous renvoyer à son mode d'emploi. Léger...

Heureusement, une fois l'interface paramétrée, les menus (au nombre de deux) sont clairs. On choisit une option, le programme se charge et s'exécute. Le démarrage d'une communication est simple, de même que le choix d'un mode de communication ou d'une configuration. Avec deux drives, ça va même très vite : disquette programme dans l'un, disquette fichier dans l'autre, tout tourne sans la moindre intervention. Avec un seul drive, en revanche, on n'arrête pas de changer de disquettes. Enervant.

Vendu 800 F, Telplus a des ambitions limitées et se contente d'une émulation Minitel simplifiée. Les touches de fonction du Minitel sont simulées par l'association de Escape et d'une lettre clé (ESC-E = envoi, ESC-S = sommaire, etc.). Quant au graphisme, il est «simplifié», ce qui, en clair, veut dire qu'il est remplacé par des signes étranges, sans signification. Si le serveur Minitel appelé n'envoie que des pages graphiques, bonjour les dégâts ! S'il envoie beaucoup de texte, vous pourrez l'utiliser. C'est le cas de l'annuaire électronique et de certaines banques de données.

Telplus permet, en mode terminal, d'imprimer les pages affichées ou de les stocker sur disquette en DOS 3.3. Il est aussi livré avec un utilitaire qui assure la conversion des fichiers Wordstar, Applewriter ou Visicalc, ce qui ouvre de nombreuses possibilités dans le cas d'échanges avec un autre utilisateur doté d'une configuration identique.

## Transcan

Dans la foulée, nous avons aussi testé Transcan. C'est un petit logiciel sans prétentions (450 F), construit sur les mêmes bases que Telplus et dont le seul rôle consiste à se connecter à Transpac et à rechercher des numéros d'accès aux banques de données. On indique au logiciel le numéro de départ et il «scanne» jusqu'à ce qu'une connexion intervienne. Transcan signale alors sa réussite, stocke le numéro sur disque et continue. Vous pouvez ainsi vous constituer un véritable annuaire

des banques de données, y compris de celles dont le numéro est confidentiel. Reste à savoir comment vous l'exploiterez ensuite. N'oubliez pas que cette recherche suppose que vous soyez d'abord connecté à Transpac et que les PTT commencent à facturer la communication à ce moment-là. Un «scanning» de deux heures risque donc de coûter cher.

## Rendre les modems intelligents

Un Apple tout seul ou avec modem ne sait pas communiquer. Un logiciel d'exploitation est donc indispensable dès lors que l'on souhaite imprimer, mémoriser ou transmettre des données.

Version-Com est le premier logiciel réellement universel... ou presque. Facile à paramétrer et à utiliser, il résout bien des problèmes de compatibilité et constitue un excellent outil pour qui souhaite relier plusieurs machines par téléphone interposé. Il autorise aussi l'accès aux banques de données à vocation professionnelle. En revanche, les applications grand public, normalement accessibles par Minitel, restent difficiles à exploiter et, faute d'une carte Appletell, il ne faut pas espérer afficher les graphismes que diffusent les serveurs comme le Parisien Libéré, AZ et les autres.

Telplus peut, quant à lui, constituer un pont entre deux machines, d'autant que le logiciel n'est pas protégé et peut donc être ouvert et modifié par tout bon programmeur. Sa fonction émulation Minitel permet effectivement de dialoguer avec des serveurs grand public, à condition de se satisfaire de l'affichage sans graphisme.

Si vous connaissez vos correspondants, Version-Com vous permettra de disposer d'un excellent utilitaire de communication pour transmettre ou recevoir des données, même en votre absence. Les fans de Calvados apprécieront aussi les fonctions d'apprentissage et la politique «souris» qui, associées aux possibilités de mémorisation des pages, permettent de diminuer les temps de transmission, donc les coûts et d'amortir ainsi le logiciel.

En revanche, si votre seul but est de transformer votre Apple en Minitel pour accéder aux services Télétel, continuez à utiliser l'horrible joujou des PTT... ou pensez à la carte Appletell, solution de luxe peut-être, mais presque parfaite.

Guy-Michel Cogné



# CORRECTION AVANCÉE

Insérer ou supprimer un caractère (et même plusieurs) dans une variable n'est pas d'une réelle simplicité... notamment en Basic, mais il est possible de traiter le problème autrement. Voici une première approche. Gageons qu'elle nous vaudra des développements intéressants (c'est à dessein que je ne reprends pas, une à une, les explications du programme en L.M ... cherchez un peu !)

**10** X\$ doit être la première variable du programme.

**30** Au départ, on initialise les pointeurs 26 et 27 (\$1A-\$1B)

**110** CALL-998 monte le curseur (\$FC1A-VP)

**130** CALL-922 envoie un saut de ligne (\$FC66-LF)

**150** CALL-1036 avance le curseur (\$FBF4-ADVANCE)

**220** CALL-1008 le recule (\$FC10-BS)

**250** CALL-198, c'est BELL (\$FF3A)

**300** Dans cette ligne, à l'affichage, tous les caractères insérés entre les symboles § et §§ apparaîtront en mode inverse (voir ligne 330)

```

10 X$ = " "; PRINT CHR$(21); TEXT : HOME : NORMAL : GO
    SUB 320
20 N = 6: REM 4 LIGNES DE TEXTE MAXIMUM (VTAB 3 A 6)
30 POKE 26,0: POKE 27,2
40 GOSUB 360:X$ = F$
50 VTAB 3: HTAB 1: CALL 768
60 VTAB PEEK(27) + 1: HTAB PEEK(26) + 1
70 GET R$:A = ASC(R$):V = PEEK(37)
80 IF A = 9 THEN CALL 815: GOTO 50
90 IF A = 127 THEN CALL 946: GOTO 50
100 IF A > 31 THEN CALL 872: GOTO 50
110 IF A = 11 AND V > 2 THEN CALL - 998: GOTO 70
120 IF V = 2 AND PEEK(36) = 0 AND A = 8 THEN 70
130 IF V < N AND A = 10 THEN CALL - 922: GOTO 70
140 IF V = N AND (A = 10 OR PEEK(36) = 39) AND A = 21
    THEN 70
150 IF A = 21 THEN CALL - 1036: GOTO 70
160 IF A > 8 THEN 70
170 ON A GOTO 180,190,230,200,290,210,70,220
180 HTAB 1: GOTO 70
190 HTAB 40: GOTO 70
200 VTAB 3: HTAB 1: GOTO 70
210 VTAB 3: HTAB PEEK(24): GOTO 70
220 CALL - 1008: GOTO 70
230 CALL - 198: VTAB 22: HTAB 6: PRINT " <§1§§> SUITE
CORRECTION <§2§§> FIN "; GET R$: PRINT R$
240 IF R$ = "1" THEN 50
250 CALL - 198: VTAB 22: HTAB 1: PRINT "(§E§§)NCORE (§
M§§)ENU DE DISQUETTE (§T§§)ERMINE "; GET R$: PRINT
: IF R$ = "T" THEN HOME : END
260 IF R$ = "E" THEN VTAB 22: CALL - 868: GOTO 230
270 IF R$ = "M" THEN PRINT CHR$(4)"RUN MENU"
280 GOTO 250
290 VTAB 14: PRINT : CALL - 958: FOR I = 1 TO 10: PRIN
T "____"; NEXT : PRINT
300 PRINT "INSERER : CTRL-§I§§ SUPPRIMER : §DEL§§
CTRL-§A§§: Début ligne CTRL-§B§§: Fin ligne CTRL
-§D§§: Début texte CTRL-§F§§: Fin texte
CTRL-§C§§: Terminé
CTRL-§§E§§: Dépannage"

```

```

310 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127
: POKE - 16368,0: VTAB 14: PRINT : CALL - 958: GOT
0 60
320 PRINT CHR$(4)"BLOAD COR.LM": CALL 936: REM ROUTIN
E CORRIGER DE 768 A 835 ET ROUTINE INVERSE DE 936 A
968
330 VTAB 13: PRINT " UTILISEZ LES $QUATRES$ FLECHES DE
L'APPLE MAIS NE DEPASSEZ PAS LE POINT... FINAL"
340 VTAB 22: PRINT " $CTRL-E$$ = TABLEAU DES COMMAN
DES"
350 RETURN
360 F$ = "CECI EST UN SIMPLE ESSAI, ET CETTE VARIABLE DE
VRA, PAR LA SUITE, PROVENIR D'UN FICHER DIRECT DONT
LES ENREGISTREMENTS NE DEPASSERONT PAS 4 LIGNES DE
TEXTE. ALLONS-Y."
370 RETURN

```

**310** SÉQUENCE  
D'ATTENTE BIEN  
CONNUE

**320** CALL 936 met  
en service la routine  
d'affichage en mode  
inverse (obtenue,  
dans le programme,  
par l'insertion d'un  
CTRL-I, symbolisé,  
dans notre listage,  
par \$, et stoppée par  
un CTRL-N, symbolisé  
par deux \$\$).

## COR.LM (PREMIÈRE PARTIE)

0300-	A5 69	LDA	\$69	0342-	C4 19	CPY	\$19
0302-	18	CLC		0344-	D0 F5	BNE	\$033B
0303-	69 02	ADC	£\$02	0346-	A9 20	LDA	£\$20
0305-	85 06	STA	\$06	0348-	99 00 20	STA	\$2000,Y
0307-	A5 6A	LDA	\$6A	034B-	4C 7B 03	JMP	\$037B
0309-	69 00	ADC	£\$00	034E-	20 8C 03	JSR	\$038C
030B-	85 07	STA	\$07	0351-	C6 18	DEC	\$18
030D-	A0 01	LDY	£\$01	0353-	A4 19	LDY	\$19
030F-	B1 06	LDA	(£\$06),Y	0355-	B9 01 20	LDA	\$2001,Y
0311-	48	PHA		0358-	99 00 20	STA	\$2000,Y
0312-	C8	INY		035B-	C8	INY	
0313-	B1 06	LDA	(£\$06),Y	035C-	C4 18	CPY	\$18
0315-	A8	TAY		035E-	D0 F5	BNE	\$0355
0316-	68	PLA		0360-	A9 20	LDA	£\$20
0317-	85 08	STA	\$08	0362-	99 00 20	STA	\$2000,Y
0319-	84 09	STY	\$09	0365-	4C 7B 03	JMP	\$037B
031B-	20 3A DB	JSR	\$DB3A	0368-	AD 00 02	LDA	\$0200
031E-	A0 00	LDY	£\$00	036B-	48	PHA	
0320-	B1 06	LDA	(£\$06),Y	036C-	20 8C 03	JSR	\$038C
0322-	85 18	STA	\$18	036F-	A4 19	LDY	\$19
0324-	A4 18	LDY	\$18	0371-	68	PLA	
0326-	88	DEY		0372-	99 00 20	STA	\$2000,Y
0327-	B1 08	LDA	(£\$08),Y	0375-	20 F4 FB	JSR	\$FBF4
0329-	99 00 20	STA	\$2000,Y	0378-	4C 8C 03	JMP	\$038C
032C-	D0 F8	BNE	\$0326	037B-	A0 00	LDY	£\$00
032E-	60	RTS		037D-	A5 18	LDA	\$18
032F-	20 8C 03	JSR	\$038C	037F-	91 06	STA	(£\$06),Y
0332-	E6 18	INC	\$18	0381-	C8	INY	
0334-	A4 18	LDY	\$18	0382-	A9 00	LDA	£\$00
0336-	A9 00	LDA	£\$00	0384-	91 06	STA	(£\$06),Y
0338-	99 00 20	STA	\$2000,Y	0386-	C8	INY	
033B-	88	DEY		0387-	A9 20	LDA	£\$20
033C-	B9 FF 1F	LDA	\$1FFF,Y	0389-	91 06	STA	(£\$06),Y
033F-	99 00 20	STA	\$2000,Y	038B-	60	RTS	



038C-	A5 24	LDA	24
038E-	85 1A	STA	1A
0390-	A5 25	LDA	25
0392-	85 1B	STA	1B
0394-	38	SEC	
0395-	E9 02	SBC	02
0397-	F0 09	BEQ	03A2
0399-	AA	TAX	
039A-	A9 00	LDA	00
039C-	18	CLC	
039D-	69 28	ADC	28
039F-	CA	DEX	
03A0-	D0 FA	BNE	039C
03A2-	18	CLC	
03A3-	65 24	ADC	24
03A5-	85 19	STA	19
03A7-	60	RTS	

## DEUXIÈME PARTIE (AFFICHAGE MODE INVERSE)

03A8-	A9 B3	LDA	03B3
03AA-	85 36	STA	36
03AC-	A9 03	LDA	0303
03AE-	85 37	STA	37
03B0-	4C EA 03	JMP	03EA
03B3-	C9 89	CMP	89
03B5-	D0 05	BNE	03B0
03B7-	A9 3F	LDA	033F
03B9-	85 32	STA	32
03BB-	60	RTS	
03BC-	C9 8E	CMP	8E
03BE-	D0 05	BNE	03C0
03C0-	A9 FF	LDA	03FF
03C2-	85 32	STA	32
03C4-	60	RTS	
03C5-	20 F0 FD	JSR	03FD
03C8-	60	RTS	

### \$300-\$32E :

Cette routine lit l'adresse de la variable X\$ (qui est la première du programme). Elle affiche la chaîne pointée par Y et A (grâce au sous-programme \$DB3A-STROUT), puis elle la recopie à partir de l'adresse \$2000.

### \$32F-\$38B :

Du basic, on y va par un CALL 815. C'est la routine d'insertion. Notez qu'elle utilise un sous-programme (\$38C-\$3A7) facile à comprendre quand on sait lire, en \$24 la position horizontale du curseur et en \$25 sa position verticale. Le JMP de la ligne \$34B saute la routine de suppression de caractères (CALL 846) qui commence à l'adresse \$34E.

En \$368 (CALL 872) on lit le caractère qui est dans le buffer (mais seulement si celui-ci dépasse la valeur décimale 31, ce qui élimine les caractères de contrôle), on le recopie à l'adresse 2000 + Y, puis on avance le curseur grâce au sous-programme \$FBF4. On notera tous les sauts à \$38C où, nous l'avons déjà vu, la position du curseur est dûment sauvegardée en \$1A et \$1B (adresses lues à partir du Basic, à la ligne 60).

### Deuxième partie :

Elle concerne l'affichage en mode inverse, obtenu, dans la liste du programme, par un CTRL-I (symbolisé par §) et interrompu par un CTRL-N (symbolisé par §§). Pour sortir de ce mode d'affichage, faites un RESET.

Avec ProDOS, remplacez \$3EA (ligne 3B0) par \$F26F (4CF26F). De plus, dans le programme en Basic, placez un CALL 936 au début des lignes 230, 250 et 300.

## COR.LM

BSAVE  
COR.LM,  
A\$300,  
L\$C9

0300-	A5 69 18 69 02 85 06 A5
0308-	6A 69 00 85 07 A0 01 B1
0310-	06 48 C8 B1 06 A8 68 85
0318-	08 84 09 20 3A DB A0 00
0320-	B1 06 85 18 A4 18 88 B1
0328-	08 99 00 20 D0 F8 60 20
0330-	8C 03 E6 18 A4 18 A9 00
0338-	99 00 20 88 B9 FF 1F 99
0340-	00 20 C4 19 D0 F5 A9 20
0348-	99 00 20 4C 7B 03 20 8C
0350-	03 C6 18 A4 19 B9 01 20
0358-	99 00 20 C8 C4 18 D0 F5
0360-	A9 20 99 00 20 4C 7B 03

0368-	AD 00 02 48 20 8C 03 A4
0370-	19 68 99 00 20 20 F4 FB
0378-	4C 8C 03 A0 00 A5 18 91
0380-	06 C8 A9 00 91 06 C8 A9
0388-	20 91 06 60 A5 24 85 1A
0390-	A5 25 85 1B 38 E9 02 F0
0398-	09 AA A9 00 18 69 28 CA
03A0-	D0 FA 18 65 24 85 19 60
03A8-	A9 B3 85 36 A9 03 85 37
03B0-	4C EA 03 C9 89 D0 05 A9
03B8-	3F 85 32 60 C9 8E D0 05
03C0-	A9 FF 85 32 60 20 F0 FD
03C8-	60

## VERIFICATION

T = 0 : FOR I = 768 TO 968 : T = T + PEEK(I) : NEXT : PRINT T  
Résultat exact: 20248

```

0 *****
1 *
2 *          ****          ****          *          *
3 *          *          *          **          *
4 *          *          *          *          *
5 *          ****          *          *          *
6 *          *          *          *          *
7 *          *          *          *          *
8 *          ****          ****          *          *
9 *
10 *      Voici tout un ensemble de routines
11 *      très bruyantes: de quoi mettre une
12 *      certaine animation dans vos jeux
13 *
14 *****
15
16          ORG    $9000
17
18 * Ce programme s'assemble avec ProCODE, en une seule fois
19 * Vous pourrez insérer ces petites routines en assembleur
20 * où bon vous semblera, mais je vous recommande, pour une
21 * utilisation depuis un programme en Basic, de les placer
22 * au-dessus de HIMEM ($73, $74), puis de descendre HIMEM.
23
24 **      MACRO SON      **
25
26          DO      0
27 SON      MAC
28          LDA    $0
29          LDX    $1
30          LDY    $2
31          JMP    SON0
32          EOM
33          FIN
34
35 *****
36 *
37 *      DES INDICATIFS
38 *
39 *****
40
41 * Voici quelques sons très courts qui vont vous permettre
42 * de bruyter vos jeux, ou de remplacer un bip aussi connu
43 * que bref, le fameux CTRL-G (BELL).
44
45 INDICATIF SON $67,$32,$82
46 MUSIQUE     SON $10,$78,$01
47 PROMPT     SON $01,$FF,$23
48 SIRENE     SON $01,$FF,$FF
49 TUUT       SON $01,$FF,$01
50 SLURP      SON $80,$06,$FE
51

```

Avec  
Richard  
THIBERT  
auteur de ProCODE







```

52 *****
53 *           S O N O           *
54 *****
55
56 * Ce sous-programme génère les sons. Pour y parvenir, il
57 * lui faut 3 paramètres, contenus dans (a) (x) et (y):
58
59 * (a) nombre de répétitions de la cellule de base
60 * (x) fréquence de base
61 * (y) incrément pour modifier la fréquence de base
62
63 * Donc, si vous entrez par SONO1, les paramètres doivent
64 * être placés dans CONT INCR et FREQ.
65

```

```

66 CONT      DS      1
67 INCR      DS      1
68 FREQ      DS      1
69 P          DS      1
70
71 SPK       =      $C030
72
903A: 8D 36 90 73 SONO      STA  CONT      ; sauve les paramètres
903D: 8C 37 90 74           STY  INCR
9040: 8E 38 90 75           STX  FREQ
76
9043: A9 00 77 SONO1     LDA  #0          ; init. pointeur d'incrément
78
9045: 8D 39 90 79           STA  P
9048: 08 80           PHP          ; pas d'interruption...
9049: 78 81           SEI          ; ...si on utilise la souris
82
904A: AE 38 90 83 §1      LDX  FREQ      ; génération du son
904D: AD 39 90 84 §2      LDA  P
9050: 6D 37 90 85           ADC  INCR
9053: 8D 39 90 86           STA  P
9056: A8 87           TAY
9057: 88 88 §3      DEY          ; boucle d'attente
9058: D0 FD 89           BNE  §3
905A: AD 30 C0 90           LDA  SPK      ; un crack
905D: 8A 91           TXA
905E: A8 92           TAY
905F: 88 93 §4      DEY          ; boucle d'attente
9060: D0 FD 94           BNE  §4
9062: AD 30 C0 95           LDA  SPK
9065: CA 96           DEX
9066: D0 E5 97           BNE  §2
9068: CE 36 90 98           DEC  cont
906B: D0 DD 99           BNE  §1
906D: 28 100          PLP          ; rétablit les interruptions
906E: 60 101          RTS
102

```



# SON avec Richard THIBERT (suite)



```

103 *****
104 *
105 *   UNE VRAIE SIRENE   *
106 *
107 *****

```

```

108
109 * Cette routine imite une sirène classique avec une modu-
110 * lation de la fréquence qui monte, puis qui descend. Une
111 * simple pression sur une touche interrompra la sirène.

```

```

112
113 * Vous pouvez essayer STfreq = $20
114 *                               Lfreq = $20
115 *                               Hfreq = $30

```

```

116 * Le son est beaucoup plus désagréable, parfaitement!

```

```

117
118 STFREQ = $A0
119 LFREQ  = $50
120 HFREQ  = $A0

```

```

121
122 KBD    = $C000
123 KBDST  = $C010

```

```

124
906F: 08   125 SIR    PHP                ; pas d'interruption
9070: 78   126        SEI
9071: AD 10 C0 127        LDA KBDST          ; le clavier remis à zéro
9074: A2 A0   128        LDX $STFREQ       ; fréquence de départ
129
9076: 8A   130 $1    TXA                ; (x) est sauvé dans (a)
9077: CA   131 $2    DEX                ; boucle d'attente
9078: EA   132        NOP
133
9079: D0 FC   133        BNE $2
907B: AA   134        TAX                ; restore (x)
907C: AD 30 C0 135        LDA SPK            ; un crack
907F: AD 00 C0 136        LDA KBD            ; une touche ?
9082: 30 0E   137        BMI $4                ; oui
9084: 90 06   138        BCC $3                ; (c) dit si on monte ou pas
139
9086: CA   140        DEX                ; on descend
9087: E0 50   141        CPX $LFREQ         ; est-on trop bas ?
142
9089: 4C 76 90 143        JMP $1                ; (c) est positionné
144
908C: E8   145 $3    INX                ; de même pour le haut
908D: E0 A0   146        CPX $HFREQ
908F: 4C 76 90 147        JMP $1
148
9092: 28   149 $4    PLP
9093: 60   150        RTS
151

```



```

152 *****
153 *                               *
154 *   ETES-VOUS JEUNE ?   *
155 *                               *
156 *****
157

```

```

158 * La faculté d'audition de hautes fréquences diminue avec
159 * l'âge, et d'une manière très sensible. Pour tester vos
160 * propres facultés d'audition, voici une routine qui vous
161 * permettra de connaître la fréquence émise par le haut-
162 * parleur de votre Apple.
163 * On imagine déjà le programme Basic qui, par une suite
164 * de tests, déterminera vos limites et votre âge (tout à
165 * fait approximatif, rassurez-vous, Madame!).
166

```

```

167 * F est la fréquence. Nous avons la correspondance avec F
168 *  $511500 / (20 + 5 * F) = F$  en Hz (F valeur de l'octet). Je
169 * vous signale que je détecte un (faible) sifflement pour
170 * 14613 Hz, mais plus rien à 17000. Une mesure plus fine
171 * m'a permis de conclure que j'entends jusqu'à 14900 Hz.
172 * Bien sûr, ce n'est qu'une indication, compte tenu de la
173 * bande passante du haut-parleur de l'Apple.
174

```

```

175 * Définissez les paramètres et faites un JMP test ou CALL
176 * test (TEST étant bien sûr remplacé par l'adresse de la
177 * routine).
178

```

```

179 T      DS  1      ; détermine la durée
180 F      DS  1      ; la fréquence
181

```

```

9096: 08
9097: 78

```

```

182 TEST   PHP
183        SEI
184

```

```

9098: AD 30 C0
909B: AC 95 90
909E: 88
909F: D0 FD

```

```

185 $1    LDA  SPK      ; 4 cycles
186        LDY  F        ; 4
187 $2    DEY           ; 2
188        BNE  $2       ; 2,3
189

```

```

90A1: CA
90A2: D0 07

```

```

190        DEX           ; 2
191        BNE  $3       ; 2,3
192

```

```

90A4: CE 94 90
90A7: D0 EF

```

```

193        DEC  T        ; 6
194        BNE  $1       ; 3
195

```

```

90A9: 28
90AA: 60

```

```

196        PLP
197        RTS
198

```

```

90AB: EA

```

```

199 $3    NOP           ; 2

```

```

90AC: 24 00
90AE: 4C 98 90

```

```

200        BIT  $0      ; 3 pour 3 cycles
201        JMP  $1      ; 3

```



# SON avec Richard THIBERT (suite)

## Table des symboles ordre alphabétique

-CONT.....\$9036	-F.....\$9095	-FREQ.....\$9038	-HFREQ.....\$00A0
-INCR.....\$9037	? -INDICATIF.\$9000	-KBD.....\$C000	-KBDST.....\$C010
-LFREQ.....\$0050	? -MUSIQUE...\$9009	-P.....\$9039	? -PROMPT....\$9012
? -SIR.....\$906F	? -SIRENE....\$901B	? -SLURP.....\$902D	M-SON.....\$1310
-SONO.....\$903A	? -SONO1.....\$9043	=SPK.....\$C030	=STFREQ....\$00A0
-T.....\$9094	? -TEST.....\$9096	? -TUUT.....\$9024	

Si vous ne disposez pas d'un assembleur, voici les 177 octets générés par ProCODE :

9000- A9 67 A2 32 A0 82 4C 3A	9060- D0 FD AD 30 C0 CA D0 E5
9008- 90 A9 10 A2 78 A0 01 4C	9068- CE 36 90 D0 DD 28 60 08
9010- 3A 90 A9 01 A2 FF A0 23	9070- 78 AD 10 C0 A2 A0 8A CA
9018- 4C 3A 90 A9 01 A2 FF A0	9078- EA D0 FC AA AD 30 C0 AD
9020- FF 4C 3A 90 A9 01 A2 FF	9080- 00 C0 30 0E 90 06 CA E0
9028- A0 01 4C 3A 90 A9 80 A2	9088- 50 4C 76 90 E8 E0 A0 4C
9030- 06 A0 FE 4C 3A 90 00 00	9090- 76 90 28 60 00 00 08 78
9038- 00 00 8D 36 90 8C 37 90	9098- AD 30 C0 AC 95 90 88 D0
9040- 8E 38 90 A9 00 8D 39 90	90A0- FD CA D0 07 CE 94 90 D0
9048- 08 78 AE 38 90 AD 39 90	90A8- EF 28 60 EA 24 00 4C 98
9050- 6D 37 90 8D 39 90 A8 88	90B0- 90
9058- D0 FD AD 30 C0 8A A8 88	

BSAVE SON.C, A36864, L177

### VERIFICATION

Taper, en mode direct :

T = 0 : FOR I = 36864 TO 36863 + 177 : T = T + PEEK(I) : NEXT I : PRINT : PRINT T

Le résultat devra être **22342**

## SON.DEMO (pour essayer les routines !)

```

10 HIMEM: 36863: TEXT : HOME : PRINT CHR$(4)"BLOAD SON.C"
20 INVERSE : PRINT "SONO": NORMAL : RESTORE
30 VTAB 22: PRINT "APPUYEZ SUR UNE TOUCHE APRES CHAQUE SON": VTAB 3
40 READ A$: PRINT A$: READ CA: CALL CA
50 POKE - 16368,0: WAIT - 16384,128: POKE - 16368,0: PRINT
60 X = X + 1: IF X < 6 THEN 40
70 HOME : INVERSE : PRINT "SIRENE": NORMAL : VTAB 22: PRINT "APPUYEZ SUR U
  NE TOUCHE POUR FAIRE CESSER"
80 CALL 36975: HOME : INVERSE : PRINT "ETES-VOUS JEUNE ?": NORMAL :F = 8
90 VTAB 3:F = F - 1: POKE 37013,F: VTAB 7: CALL - 958: PRINT "FREQUENCE =
  "; INT (511500 / (20 + 5 * F))" Hz": CALL 37014
100 IF F > 2 THEN 90
200 DATA "INDICATIF SON $67,$32,$82 (103 50 130)",36864
210 DATA "MUSIQUE SON $10,$78,$01 ( 16 120 1)",36873
220 DATA "PROMPT SON $01,$FF,$23 ( 1 255 35)",36882
230 DATA "SIRENE SON $01,$FF,$FF ( 1 255 255)",36891
240 DATA "TUUT SON $01,$FF,$01 ( 1 255 1)",36900
250 DATA "SLURP SON $80,$06,$FE (128 06 254)",36909
  
```





# • Complément au programme SON

## • Trois mystérieuses instructions de ProCODE

En complément au programme son, je propose ici quelques informations très utiles au musicien programmeur. Vous savez tous que chaque note de la gamme correspond à une fréquence définie.

Il existe des tableaux de conversion, mais ils sont toujours difficiles d'emploi, car non compris dans le principe de calcul. Voici la méthode pour calculer les relations notes et fréquences. Sachez d'abord que la plus petite division du ton est le COMA : dans un ton il y a 9 comas et, bien sûr, comme les nombres décimaux ne sont pas utilisés, on divise ce ton en deux types de demi-tons : le demi-ton diatonique 4 comas et le chromatique 5 comas.

Nous allons chercher la valeur absolue du coma. Vous savez que l'oreille perçoit une variation de fréquence logarithmiquement. Donc une variation d'un coma se traduit par un coefficient pour multiplier la fréquence de la note précédente. Tout ceci a l'air compliqué, mais la réalité est fort simple.

Une variation d'une octave correspond à une multiplication par 2 de la fréquence. Or une octave comprend 5 tons et 2 demi-tons diatoniques donc 53 comas.

On a, si X est la valeur absolue du coma

$$X^{53} = 2 \text{ d'où on tire } X = 10^{\wedge} (\text{LOG}2/53)$$

Finalement on a  $F * 1.013164^{\wedge} N = F$  où N est la distance en comas des deux notes.

On peut alors calculer le tableau suivant :

0	la	444 Hz	66	do	1053 Hz
9	si	499 Hz	75	ré	1184 Hz
13	do	526 Hz	84	mi	1332 Hz
22	ré	592 Hz	88	fa	1403 Hz
31	mi	666 Hz	97	sol	1579 Hz
35	fa	702 Hz	106	la	1776 Hz
44	sol	789 Hz	115	si	1998 Hz
53	la	888 Hz	119	do	2105 Hz
62	si	999 Hz			

Voici pour finir quelques indications très confidentielles sur ProCODE. En effet, le programme, dans sa version 1.23, comprend 3 instructions non explicitées dans le manuel, et ceci pour des raisons

de délai, la documentation étant terminée bien avant la mise en vente, et pour des raisons de fiabilité, une longue période de test étant de rigueur.

Passons aux choses sérieuses... voilà les 3 instructions méconnues : SHK ON, SHK OFF, NODSK.

Pour les utilisateurs de ProCODE voici, sous la même forme que dans le manuel, la description de ces instructions.

### SHK ON

— SHK ON — ProCODE permet de gérer un double octet de contrôle, appelé SHECK SUM en anglais ; l'instruction SHK ON indique à ProCODE que chaque octet de code généré va être additionné à une variable prédéfinie sur 16 bits : à, dans cette variable, la valeur actuelle de la somme est toujours accessible.

### SHK OFF

— SHK OFF — Cette instruction demande l'arrêt de mise à jour de la variable à qui contient le SHECK SUM.

Exemple d'utilisation de ces instructions ; le programme est entre test et fin s'il y a un octet différent lors de l'appel de vérif il n'y aura pas de retour.

```

      SHK ON      §0 LDA TEST,X      CMP £ < à
TEST  LDA £$10   ADC P              BNE ERR
      LDA £$24   STA P              LDA P+1
FIN   =  *      BCC §1             CMP £ > à
      SHK OFF   §1 INC P+1         BNE ERR
VERIF LDX £0    CPX £FIN-TEST      RTS
      STX P     BCC §0             ERR JMP ERR
      STX P+1   LDA P

```

### NODSK

— NODSK — Cette instruction s'utilise avec DSK, elle permet d'arrêter la génération du code sur disque, ou par exemple de créer plusieurs fichiers code avec la même source.

Exemple : ce petit programme : créer à l'assemblage 2 fichiers objet1 et objet2.

```

DSK   OBJET1
ASC   "FICHIER OBJET NUMERO 1"
NODSK
DSK   OBJET2
ASC   "FICHIER OBJET NUMERO 2"
NODSK

```



# Votre bibliothèque INFORMATIQUE

par Clément RENARD

## ● *Apprivoiser l'ordinateur : pourquoi pas ?*

Dans *Vous et l'ordinateur Apple*, Edward H. Carlson se propose d'aider les débutants à apprivoiser leur machine (Cedic/Nathan).

L'auteur n'est probablement pas médecin, mais il a choisi de répartir cette initiation sur 33 leçons très pédagogiques et réellement à la portée de toutes et de tous.

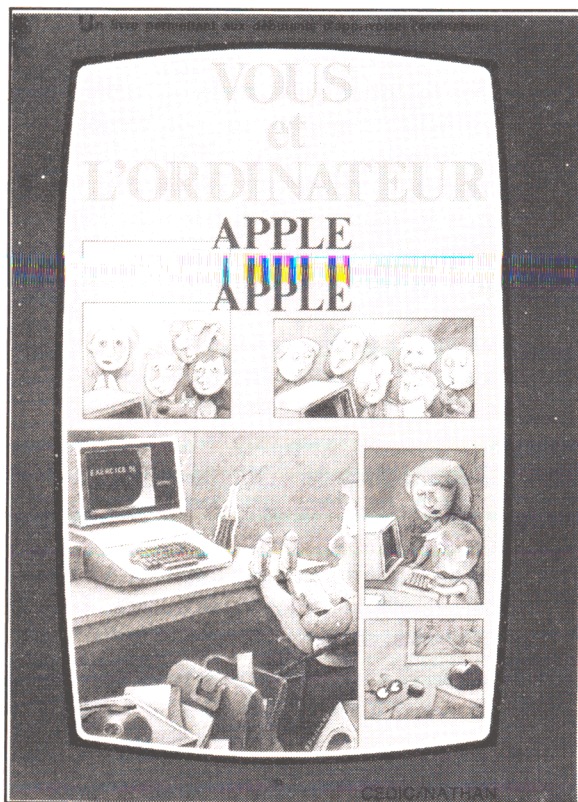
et réellement à la portée de toutes et de tous.

Naturellement, le langage utilisé est le Basic. Chaque leçon comprend, outre des notes destinées au professeur (ou aux parents de l'enfant) de nombreuses questions, explications et embryons de programmes.

Est-il possible d'apprendre à programmer à partir des exemples fournis dans cet ouvrage ? La réponse est oui.

Personnellement, j'ai aimé le style alerte de l'auteur et sa manière d'aborder chaque instruction du Basic. Je pense que *Vous et l'ordinateur Apple* mérite de figurer, en bonne place, dans la bibliothèque des classes de micro-informatique.

C'est un abc qui s'adresse surtout aux jeunes (il est illustré de nombreux dessins amusants), mais qui se révélera capable d'intéresser les adultes.



## ● *Une auto-formation bien comprise*

Avez-vous eu la curiosité de feuilleter, dans la collection *Micropratiques* (Hachette Informatique), *Les Ecrans de l'Apple IIe* ? L'éditeur annonce "un langage simple, accessible à tous. Une présentation claire et utilisant largement la couleur pour une initiation progressive au Basic".

Je m'attendais à retrouver, sous une présentation luxueuse, une nouvelle compilation des manuels diffusés par Apple. Nous en avons l'habitude, pas vrai ?

Surprise, et de taille : si le luxe est bien présent dans les deux livrets de cet ouvrage d'auto-formation (mais qui s'en plaindrait ?), on a le plaisir — même quand on connaît déjà son Apple IIe — d'y trouver de très bons exemples de programmation. On se prend à regretter que tout cela soit l'œuvre d'un sujet de Sa Gracieuse Majesté : Phil Robinson (traduction signée par Denis Pellerin). Bravo ! Phil ! c'est de la belle besogne, et j'espère que les nouveaux possesseurs d'Apple (ainsi que ceux pour lesquels l'Applesoft ressemble encore aux caractères étrusques) s'offriront ce double guide de la programmation sur Apple IIe.

Un regret : j'aurais souhaité la même chose, mais sous forme de fiches...

C. R.



## ■ GESTION II de Version Soft

**G**ESTION II intéressera bon nombre de Lectrices et Lecteurs de *Tremplin Micro*, mais seulement s'ils sont les heureux possesseurs d'un Apple IIc ou IIe... équipé du 65C02 (n'oublions pas qu'il existe un kit de mise à niveau très bon marché).

*Gestion II* s'adresse, en priorité, aux membres des professions libérales placés sous le régime des **Bénéfices Non Commerciaux** (médecins, avocats, etc.), mais aussi aux responsables de multiples clubs, associations et syndicats (type loi de 1901).

Ce logiciel, facile à maîtriser, est d'ailleurs capable de s'adapter, après quelques heures d'apprentissage, aux besoins de quiconque désire réaliser une gestion logique, sûre et approfondie de ses divers revenus.

### Un large éventail de comptes

L'utilisateur dispose de 60 comptes (dont plusieurs comptes bancaires), de 100 écritures mensuelles et de 10 mouvements automatiques (très pratique pour prévoir le remboursement d'un emprunt !).

La souris commande tous les menus déroulants et la gestion des fenêtres.

Avec *Gestion II*, on retrouve les facilités déjà offertes par *Version Graph* et les autres produits de Version Soft.

Toutes les saisies peuvent être modifiées ou supprimées par des commandes aussi simples que... *Modifier...* et *Supprimer*.

Avec *Gestion II*, il est commode de prévoir son budget pour l'année à venir, puis de comparer les prévisions aux dépenses réelles.

### Des états... à imprimer

Tous les rapports peuvent être imprimés tels qu'ils apparaissent sur l'écran. On pourra ainsi éditer, suivant les besoins :

- Une liste mensuelle des écritures (en totalité ou par catégorie) ;
- La liste mensuelle des écritures retenues pour une future déclaration de revenus ;

- La liste des soldes de toutes les catégories ;
- Une analyse détaillée de revenus, avec calcul du revenu net ;
- Un bilan complet avec le résultat.

A noter que *Gestion II* permet l'établissement d'un budget prévisionnel détaillé, pour une année complète, puis la comparaison des prévisions avec les dépenses réelles.

Le contrôle des relevés bancaires n'a pas été oublié ; quant à la documentation, elle est fidèle à la philosophie de Version Soft : claire, concise, mais capable d'être comprise sans effort, grâce à des exemples judicieusement choisis.

Demandez une démonstration à votre revendeur !

Clément RENARD.

### Considérations générales

- Comment peut-on encore vendre — en 1985 — un Apple IIe sans carte 80 colonnes, alors que celle-ci est incorporée dans l'Apple IIc ? Certes, la plupart des revendeurs la proposent à leurs clients, mais elle reste — hélas ! — une option (à noter que la carte texte Apple a récemment subi une baisse de plus de 200 F).
- On constate que la plupart des nouveaux logiciels travaillent non seulement en 80 colonnes, mais utilisent la totalité des ressources du 65C02, et exigent 128 K.
- Indéniablement, le concept graphique souris fait recette et devient même le standard Apple. Les menus déroulants remplacent les classiques présentations d'hier. C'est beau. C'est pratique, mais gourmand en mémoire !
- La double haute résolution devient une règle dans les logiciels très élaborés. C'est notamment vrai chez *Version.Soft*, l'un des meilleurs compagnons de route de l'Apple II.
- L'ordinateur personnel de papy a vécu, mais il existe encore et se révèle parfois capable, avec ses slots bien garnis, et grâce à de multiples bidouillages, de rivaliser avec l'Apple IIc. NESTOR.

# USR (?)

En recopiant attentivement ce programme de démonstration, le néophyte comprendra certainement à quoi sert

l'instruction USR(X) de son Apple, mais il sera bien inspiré en réalisant lui-même plusieurs routines d'application.

```

10 TEXT : NORMAL : PRINT CHR$(12);
   CHR$(21); HOME
15 P = 1; Q = P
20 INVERSE : T$ = " FONCTION USR (PAGE ZERO) "; GOSUB 280
25 PRINT "POKE 10,76
   ($0A)POKE 12,ADR/256
   ($0C)POKE 11,ADR-PEEK(12)*256
   ($0B)"
30 PRINT : PRINT "Ces 3 POKES initialisent la fonction USR
   De cette manière, un appel à USR renverra automatiquement à un sous-programme en langage machine débutant à l'adresse ADR."
35 PRINT : PRINT "On sait que cette fonction permet d'établir un lien entre un programme en Basic et un autre en assembleur, auquel elle transmet des paramètres avant de retourner au programme principal."
40 GOTO 285
45 INVERSE : PRINT "EXEMPLES:"; NORMAL : PRINT : PRINT : PRINT "100 PRINT USR(8) (ARGUMENT=8)
   500 X=2*USR(Y+1)"
50 PRINT : PRINT "Exemple absurde, supposons qu'une petite routine en langage machine soit chargée d'afficher l'argument multiplié par 10, comme ceci:"
55 PRINT : PRINT "300 20 39 EA JSR $EA39 (FAC * 10) 303 60
   RTO (Basic)"
60 PRINT : PRINT "POKE10,76:POKE12,768/256:POKE11,768-PEEK(12)*256 va initialiser la fonction USR"
65 PRINT : PRINT "Ensuite, si l'on tape un RUN 100, c'est le nombre ";
   : INVERSE : PRINT 80; : NORMAL : PRINT " qui s'affichera.": GOTO 285

```

Lorsque vous lancerez votre programme, vous obtiendrez successivement ces écrans :

## FONCTION USR (PAGE ZERO)

```

POKE 10,76 ($0A)
POKE 12,ADR/256 ($0C)
POKE 11,ADR-PEEK(12)*256 ($0B)

```

Ces 3 POKES initialisent la fonction USR

De cette manière, un appel à USR renverra automatiquement à un sous-programme en langage machine débutant à l'adresse ADR.

On sait que cette fonction permet d'établir un lien entre un programme en Basic et un autre en assembleur, auquel elle transmet des paramètres avant de retourner au programme principal.

PRESSEZ UNE FLECHE ← OU → POUR SUITE

## EXEMPLES:

```

100 PRINT USR(8) (ARGUMENT=8)
500 X=2*USR(Y+1)

```

Exemple absurde, supposons qu'une petite routine en langage machine soit chargée d'afficher l'argument multiplié par 10, comme ceci :

```

300 20 39 EA JSR $EA39 (FAC * 10)
303 60 RTS (Basic)

```

POKE10,76:POKE12,768/256:POKE11,768-PEEK(12)\*256 va initialiser la fonction USR

Ensuite, si l'on tape un RUN 100, c'est le nombre 80 qui s'affichera.

PRESSEZ UNE FLECHE ← OU → POUR SUITE



```

70 PRINT "La valeur de l'argument se
   trouve dans le registre accumula
   teur ($9D à $A2), où la routine en
   langage machine devra donc la récup
   érer."
75 PRINT : PRINT : PRINT "La NOUVELL
   E VALEUR de la fonction USR sera
   celle figurant dans l'accumulateu
   r, au moment du RTS renvoyant au Ba
   sic."
80 GOTO 285
85 PRINT "FAITES UN ESSAI AVEC CE PR
   OGRAMME, EN DONNANT DES VALEURS
   DIFFERENTES A B:" : LIST 100,140: G
   OTO 285
90 LIST 100,140
100 POKE 10,76: POKE 12,768 / 256: PO
   KE 11,768 - PEEK (12) * 256
110 POKE 768,32: POKE 769,57: POKE 77
   0,234: POKE 771,96: REM ROUTINE L
   M
120 VTAB 22: HTAB 2: CALL - 198: PRI
   NT "B = " ; : GET R$:S = 1:R = ASC
   (R$): GOSUB 290
130 PRINT R$; : INPUT " "; B$:B = VAL (
   R$ + B$): IF NOT B THEN 340
140 VTAB 22: HTAB 20: CALL - 868: PR
   INT USR (B): GOTO 120
200 VTAB 3: PRINT "Dans notre exemple
   , on pouvait accéder à la routine $
   EA39 (MUL10) sans passer par le pro
   gramme de l'adresse $300, de cette
   manière:"
205 PRINT : PRINT "POKE 10,76: POKE 1
   1,57: POKE 12,234: REM $4C 39 EA"
210 PRINT : PRINT "Temps avec:
   -----
   FOR I
   = 1 TO 5000:A=10: A = A * 10:NEXT
   38 SECONDES"
215 PRINT : PRINT "Temps avec:
   -----
   FOR I
   = 1 TO 5000:A=10: A = USR(A):NEXT
   27 SECONDES"
220 GOTO 285
225 VTAB 3: HTAB 28: INVERSE : PRINT
   "AUTRE EXEMPLE": NORMAL : VTAB 3
230 T$ = "300 20 63 EB JSR $EB63303 2
   0 80 EB JSR $EB80306 20 82 E9 JS
   R $E982309 60 RTS " : F

```

La valeur de l'argument se trouve dans le registre accumulateur (\$9D à \$A2), où la routine en langage machine devra donc la récupérer.

La NOUVELLE VALEUR de la fonction USR sera celle figurant dans l'accumulateur, au moment du RTS renvoyant au Basic.

PRESSEZ UNE FLECHE <- OU -> POUR SUITE

FAITES UN ESSAI AVEC CE PROGRAMME, EN DONNANT DES VALEURS DIFFERENTES A B:

```

100 POKE 10,76: POKE 12,768 / 25
   6: POKE 11,768 - PEEK (12) *
   256
110 POKE 768,32: POKE 769,57: POKE
   770,234: POKE 771,96: REM R
   OUTINE LM
120 VTAB 22: HTAB 2: CALL - 198
   : PRINT "B = " ; : GET R$:S =
   1:R = ASC (R$): GOSUB 290
130 PRINT R$; : INPUT " "; B$:B = VAL
   (R$ + B$): IF NOT B THEN 34
   0
140 VTAB 22: HTAB 20: CALL - 86
   8: PRINT USR (B): GOTO 120

```

PRESSEZ UNE FLECHE <- OU -> POUR SUITE

Dans notre exemple, on pouvait accéder à la routine \$EA39 (MUL10) sans passer par le programme de l'adresse \$300, de cette manière:

```

POKE 10,76: POKE 11,57: POKE 12,234: REM
   $4C 39 EA

```

Temps avec:

```

-----
FOR I = 1 TO 5000:A=10: A = A * 10:NEXT
38 SECONDES

```

Temps avec:

```

-----
FOR I = 1 TO 5000:A=10: A = USR(A):NEXT
27 SECONDES

```





```

OR I = 1 TO LEN (T$) STEP LEN (T
$) / 4: PRINT MID$(T$,I, LEN (T$
) / 4): NEXT
235 PRINT : PRINT "$EB63 recopie l'ac
cumulateur en $A5-$AA": PRINT
240 PRINT "$EB80 détermine les signes
des nombres (de $9D à $A2
et de $A5 à $AA)": PRINT
245 PRINT "$E982 multiplie le contenu
de l'accumu- lateur par cel
ui de la zone allant de $A5 à
$AA, puis place le résul- ta
t dans l'accumulateur"
250 PRINT : PRINT "Comme on le voit,
la routine élève l'ar-gument au ca
rré..."
255 POKE 768,32: POKE 769,99: POKE 77
0,235: POKE 771,32: POKE 772,128:
POKE 773,235: POKE 774,32: POKE 77
5,130: POKE 776,233: POKE 777,96
260 VTAB 22: HTAB 2: CALL - 198: PRI
NT "B = ";: GET R$:S = 1:R = ASC
(R$): GOSUB 290
265 PRINT R$;: INPUT "":B$:B = VAL (
R$ + B$): IF NOT B THEN 340
270 VTAB 22: HTAB 20: CALL - 868: PR
INT USR (B): GOTO 260
275 GOTO 285
280 PRINT : VTAB 1: HTAB (41 - LEN (
T$)) / 2: INVERSE : PRINT T$: NORM
AL : POKE 34,2: VTAB 3: RETURN
285 VTAB 24: HTAB 1: PRINT "PRESSEZ U
NE FLECHE (- OU -) POUR SUITE ";:
GET R$:R = ASC (R$): POKE - 163
68,0
290 IF R = 8 THEN P = P - 1: GOTO 305
295 IF R = 21 THEN P = P + 1: GOTO 30
5
300 IF S = 1 THEN S = 0: RETURN
305 IF NOT P THEN P = 1
310 IF S = 1 THEN S = 0: POP
315 IF R = 6 THEN TEXT : HOME : END
320 IF R = 3 THEN VTAB 22: PRINT :
PRINT CHR$(4)"RUN MENU"
325 IF P = 9 THEN P = 8
330 IF P = 0 THEN 285
335 PRINT :Q = P: VTAB 3: CALL - 958
: ON P GOTO 25,45,70,85,90,200,225
,345
340 P = P + 1: GOTO 330
345 VTAB 12: HTAB 6: PRINT "CTRL-C PO
UR MENU DE DISQUETTE
CTRL-F POUR TERMINER": GOTO 285

```

#### AUTRE EXEMPLE

```

300 20 63 EB JSR $EB63
303 20 80 EB JSR $EB80
306 20 82 E9 JSR $E982
309 60 RTS

```

\$EB63 recopie l'accumulateur en \$A5-\$AA

\$EB80 détermine les signes des nombres  
(de \$9D à \$A2 et de \$A5 à \$AA)

\$E982 multiplie le contenu de l'accumu-  
lateur par celui de la zone allant  
de \$A5 à \$AA, puis place le résul-  
tat dans l'accumulateur

Comme on le voit, la routine élève l'ar-  
gument au carré...

B = 45

2025

- La ligne 255 poque les 10 octets de la routine en langage machine que vous montre l'écran ci-dessus.
- A la ligne 265, si R est égal à 0, on passe à la page suivante.
- Les flèches de l'Apple permettent d'aller de l'avant... ou de revoir les pages précédentes.

CTRL-C POUR MENU DE DISQUETTE  
CTRL-F POUR TERMINER



# I LA PSEUDO PAGE 3 GRAPHIQUE

Il existe une pseudo PAGE 3 (que vous pouvez protéger par un LOMEM : 32768) et vous allez la rencontrer...

En effet, alors que POKE 230,32 et 230,64 permettent respectivement d'écrire dans les pages 1 et 2, un POKE 230,96 vous ouvre des horizons nouveaux... puisqu'il vous autorise à écrire dans une pseudo PAGE 3.

Ne jubilez pas trop vite : pas question d'afficher instantanément cette fameuse **PAGE 3**. Il est indispensable de la recopier dans l'une des deux pages graphiques autorisées par votre Apple.

Mais cela est simple, grâce à la routine MOVE(\$FE2C)... poquée par les lignes 310 et 320 de notre programme (mais on prend la peine de vous les expliquer !).

Pour le reste, tapez ce petit programme de démonstration (vous pouvez vous faire aider si vous avez un poil dans la main). Vous comprendrez tout, je vous le promets !

**Nestor.**

```

10 TEXT : NORMAL : PRINT CHR# (12); CHR# (21): HOME
20 GOSUB 310: REM ROUTINE TRANSFERT PAGE3 VERS PAGE2
30 POKE 49234,0: REM MODE GRAPHIQUE PLEINE PAGE
40 GOSUB 240: HGR : HGR2 :X = 0
50 POKE 230,96: REM PERMET DE TRACER DANS UNE PSEUDO PAGE3
60 POKE 49236,0: POKE 49233,0: REM PASSAGE EN MODE TEXT PAGE1
70 HCOLOR= 0: HPLLOT 0,0: CALL - 3082: REM EFFACE LA PAGE3
80 HCOLOR= 3: HPLLOT 1,1 TO 279,189: HPLLOT 279,1 TO 1,189: REM TRACE PAGE3
90 LIST 30,60: LIST 70,80: LIST 100,110: VTAB 23: INVERSE : HTAB 6: PRINT
  "PENSEZ A ESCAPE POUR TERMINER": NORMAL : GOSUB 200
100 POKE 49237,0: POKE 49232,0: REM PASSAGE EN MODE GRAPHIQUE PAGE2
110 CALL 768: REM TRANSFERT PAGE2
120 PRINT : PRINT CHR# (4)"BLOAD ETO,A#2000"
130 FOR I = 1 TO 10: POKE 49237,0: GOSUB 230: POKE 49236,0: GOSUB 230: NEXT
  I
140 HOME : POKE 49235,0: VTAB 22: PRINT "HGR2 VA EFFACER LA PAGE2": GOSUB 2
  00: HGR2
150 POKE 49236,0: HOME : POKE 49235,0: VTAB 22: PRINT "CALL 768 VA RECUPERE
  R LA PAGE3": GOSUB 200: POKE 49234,0: POKE 49237,0: CALL 768
160 GOSUB 200: POKE 49236,0: REM PAGE1
170 GOSUB 200: POKE 49237,0: REM PAGE2
180 X = X + 1: IF X < 8 THEN 160
190 GOSUB 200: GOTO 190
200 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: POKE - 16368,0:R
  = PEEK ( - 16384)
210 IF R = 27 THEN 250
220 RETURN

```





```

230 FOR J = 1 TO 400: NEXT : RETURN
240 HCOLOR= 0: FOR I = 1 TO 3: X = 32 * I: POKE 230,X: HPLLOT 0,0: CALL - 30
    82: NEXT : RETURN : REM EFFACEMENT DES PAGES1.2.3
250 POKE 49236,0: POKE 49235,0: HOME :
260 VTAB 22: PRINT "(M)ENU DE DISQUETTE (E)NCORE (T)ERMINE ";: GET R$: PRI
    NT
270 IF R$ = "M" THEN PRINT : PRINT CHR$(4)"RUN MENU"
280 IF R$ = "E" THEN RUN
290 IF R$ < > "T" THEN HOME : POKE 49234,0: RETURN
300 TEXT : HOME : END
310 POKE 768,160: POKE 769,0: POKE 770,132: POKE 771,60: POKE 772,169: POKE
    773,96: POKE 774,133: POKE 775,61: POKE 776,169: POKE 777,255: POKE 778
    ,133: POKE 779,62: POKE 780,169
320 POKE 781,127: POKE 782,133: POKE 783,63: POKE 784,169: POKE 785,64: POK
    E 786,133: POKE 787,67: POKE 788,132: POKE 789,66: POKE 790,32: POKE 791
    ,44: POKE 792,254: POKE 793,96: RETURN

```

### TRANSFERT PAGE 3 DANS PAGE 2 (LES POKES)

0300-	A0 00	LDY	£#00	] Page 3 (début de la zone à copier) \$6000 (24576)	
0302-	84 3C	STY	#3C		
0304-	A9 60	LDA	£#60		
0306-	85 3D	STA	#3D		
0308-	A9 FF	LDA	£#FF		
030A-	85 3E	STA	#3E		
030C-	A9 7F	LDA	£#7F		] Page 3 (fin de la zone à copier) \$7FFF (32767)
030E-	85 3F	STA	#3F		
0310-	A9 40	LDA	£#40		
0312-	85 43	STA	#43		
0314-	84 42	STY	#42	] Destination Page 2 \$4000 (16384)	
0316-	20 2C FE	JSR	#FE2C		
0319-	60	RTS			

MOVE déplace la zone \$3C-3D (\$6000), finissant en \$3E-3F (\$7FFF) vers la zone commençant en \$42-43 (\$4000).

## Des adresses nullement mystérieuses, et fort utiles !

- \$E4 (228)** Contient le code de la couleur en HGR
- \$E6 (230)** On y trouve \$20 pour HGR, \$40 pour HGR2... et \$60 pour la pseudo PAGE 3  
En Basic, cela se traduit par Poke 230,32 — Poke 230,64 — Poke 230,96
- \$E7 (231)** Valeur de SCALE
- \$E8-\$E9 (232-233)** Adresse de la table de formes

• Pour effacer la page courante, en haute résolution, vous aurez une alliée puissante dans la routine BKGND (\$F3F4). **Par exemple :**

```

300 : A9 55   LDA £$AA   —» $55 = Bleu
302 : 20 F4 F3 JSR $F3F4 —» Routine BKGND
305 : 60

```



# DES POKES UTILES

## pour modifier le catalog ProDOS

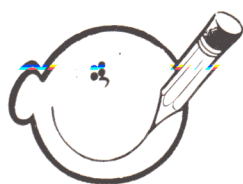
Ouvrez dès maintenant un répertoire des POKES-ProDOS et si vous disposez de la version 1.1.1 (Septembre 1984) de ce système d'exploitation, essayez le petit programme DEMO :

```

10 TEXT : HOME : PRINT CHR$(4)"PRÉ3"
12 PRINT " CES POKES NE SONT UTILISABLES QU'AVEC LA VERSION 1.1.1 DU SYSTEM
   E PRODOS APPLE "
14 VTAB 2: FOR I = 1 TO 20: PRINT "----";: NEXT : PRINT : POKE 34,3
16 PRINT "POKE 42198,0 (128) = TITRES DES FICHIERS EN MODE INVERSE"
18 POKE 42198,0: GOSUB 56: POKE 42198,128
20 PRINT "POKE 42316,96 (162) = ELIMINE LES DATES DU CATALOG (NON DU CAT)
   "
22 POKE 42316,96: GOSUB 56: POKE 42316,162
24 PRINT "POKE 42344,173 (170) = LOCK (*) DEVIENT (-)"
26 POKE 42344,173: GOSUB 56: POKE 42344,170
28 PRINT "POKE 42402,0 (18) = REMPLACE LES DATES PAR <NO DATE>"
30 POKE 42402,0: GOSUB 56: POKE 42402,18
32 PRINT "POKE 42403,96 (152) = ELIMINE LA MENTION <NO DATE>"
34 POKE 42403,96: GOSUB 56: POKE 42403,152
36 PRINT "POKE 42490,175 (173) = CHANGE LE / DES DATES POUR UN TIRET (-)"
38 POKE 42490,175: GOSUB 56: POKE 42490,173
40 PRINT "POKE 45201,155 (131) = ESCAPE STOPPE LE CATALOGUE "
42 POKE 45201,155: PRINT CHR$(4)"CATALOG": PRINT : PRINT : PRINT "ENFONCE
   Z UNE TOUCHE POUR FAIRE UN ESSAI ";: GET R$: PRINT : HOME : GOSUB 56
44 POKE 45201,131
46 VTAB 22: PRINT "(M)ENU DE LA DISQUETTE DRIVE1 (R)ECOMMENCER (T)ERMINE
   -> ";: GET R$
48 IF R$ = "M" THEN PRINT : PRINT CHR$(4)"- / TM5/MENU"
50 IF R$ = "R" THEN RUN
52 IF R$ < > "T" THEN PRINT : GOTO 46
54 TEXT : HOME : END
56 POKE 34,6: PRINT CHR$(4)"CATALOG": GET R$: PRINT : POKE 34,3: HOME : R
   ETURN

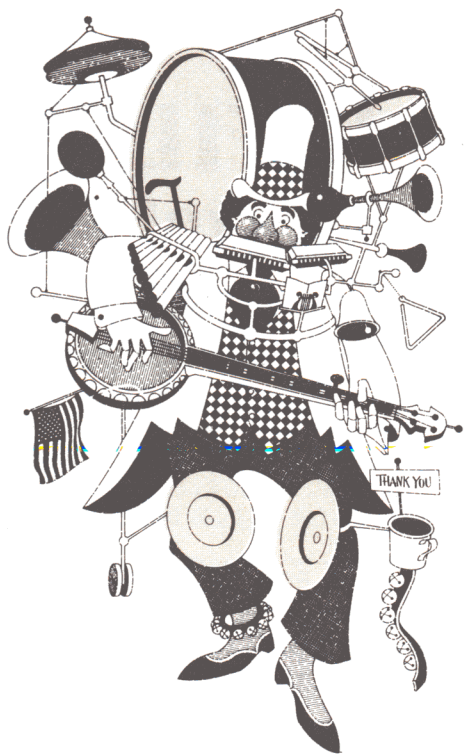
```

### RESUME



Nous indiquons entre parenthèses la valeur normale

POKE 42198,0	(128)	:	Titres des fichiers en mode inverse
POKE 42316,96	(162)	:	Les dates ne sont plus visibles
POKE 42344,173	(170)	:	* devient un tiret (fichiers protégés)
POKE 42402,0	(18)	:	Les dates sont remplacées par "NO DATE"
POKE 42403,96	(152)	:	La mention "NO DATE" n'apparaît plus
POKE 42490,175	(173)	:	Troque le tiret des dates pour un /
POKE 45201,155	(131)	:	ESCAPE stoppe le catalogue.



# INDICATIF

## (ô combien !) musical

• par Jacques Saussard

Ce programme utilise DIA.LM et DIA.CARAC (*Tremplin Micro n°2*), mais fonctionne parfaitement en mode TEXT.

On ne transfère le texte en page graphique... que pour l'amour de l'art !

```

10 REM INDICATIF (RETROUVE D'OREILLE)
20 REM "MUSIC.INIT" (D'APRES N.BREAUD-POULIQUEN, PSI)
   EST DISSIMULE DANS MUSIC.DIATM, A PARTIR DE L'ADRE
   SSE 24752
30 PRINT CHR$(4)"BLOAD MUSIC.DIATM": REM "DIA.LM +
   DIA.CARAC" DE "TREMPLIN MICRO 2"
40 TEXT : PRINT CHR$(21): HOME : HCOLOR= 0: POKE 23
   0,64: CALL - 3086: HCOLOR= 3: REM EFFACEMENT DE H
   GR2
50 D$ = CHR$(4):TR = 24576:IN = 24752:IO = IN + 1:IP
   = IN + 2
60 POKE 49239,0: POKE 49232,0: POKE 49237,0: POKE 492
   34,0
70 VTAB 1: HTAB 11:T$ = "           ": INVE
   RSE : FOR I = 1 TO 3: HTAB 11: PRINT T$: NEXT : VTA
   B 2:T$ = "           ": PRINT T$:: HTAB 31: PRINT T$
80 VTAB 24: FOR I = 39 TO 1 STEP - 1: PRINT " " :: NE
   XT : FOR I = 23 TO 2 STEP - 1: VTAB I: HTAB 1: PRI
   NT " " :: HTAB 40: PRINT " " :: NEXT : POKE 2039,32:
   VTAB 2: HTAB 12: NORMAL : PRINT " I N D I C A T I F
   "
90 CALL TR: REM PAGE TEXT COPIEE EN HGR2
100 VTAB 6: HTAB 3: PRINT "VERSION ORIGINALE": CALL TR
110 READ I,J: IF J = 0 THEN 130
120 POKE IN,I: POKE IO,J: CALL IP: GOTO 110
130 RESTORE : VTAB 9: HTAB 3: PRINT "VERSION MANDOLINE
   ...": CALL TR
140 READ I,J: IF J = 0 THEN 160

```

Si vous ne possédez pas *Tremplin Micro n°2*, supprimez la partie graphique du programme... ou commandez-le !

TR = Adresse de la routine de transfert (\$6000)

IN = Routine Music (\$60B0)

La ligne 60 vous montre comment passer en mode HGR2 (sans rien effacer).



Vous apprécierez tout particulièrement la version enrôlée !



Quant à la version bourrée (rien à voir avec la bourrée auvergnate !) je vous la recommande pour les jours sans alcool...



France-Culture, c'est tout de même quelque chose, non ?

```

150 FOR K = 1 TO (J / 16): POKE IN,I: POKE IO,12: CALL
    IP: NEXT : GOTO 140
160 RESTORE : VTAB 12: HTAB 3: PRINT "VERSION ENROUEE
    !": CALL TR
170 READ I,J: IF J = 0 THEN 190
180 FOR L = 1 TO (J / 16) + 2: POKE IN,I: POKE IO1,6:
    CALL IP: NEXT : GOTO 170
190 RESTORE : VTAB 15: HTAB 3: PRINT "VERSION JAZZ...
    CA SWINGUE PRESQUE !": CALL TR
200 READ I,J: IF J = 0 THEN 260
210 IF J > 32 THEN POKE IN,I: POKE IO,90: CALL IP:M =
    0: GOTO 200
220 IF J < 32 THEN POKE IN,I: POKE IO,30: CALL IP:M =
    0: GOTO 200
230 IF M = 0 THEN POKE IN,I: POKE IO,60: CALL IP:M =
    1: GOTO 200
240 IF M = 1 THEN POKE IN,I: POKE IO,26: CALL IP:M =
    0
250 GOTO 200
260 RESTORE : VTAB 18: HTAB 3: PRINT "VERSION BOURREE
    (SAOUL-MUSIC !)": CALL TR
270 READ I,J: IF J = 0 THEN 350
280 N = INT ( RND (1) * 30)
290 Y = Y + 1
300 POKE IN,I: POKE IO,16: CALL IP
310 IF Y > (N / 2) THEN FOR TI = 1 TO 200: NEXT TI: G
    OSUB 420:Y = 0
320 POKE IN,I + N: POKE IO,I + N: CALL IP
330 GOTO 270
340 GOSUB 420
350 GOSUB 420
360 POKE 32,2: POKE 33,36: POKE 34,4: POKE 35,22: PRIN
    T : CALL - 198: GET R$: HOME : VTAB 7: LIST 440,49
    0: FOR I = 7 TO 13: VTAB I: PRINT "      ": NEXT
    : POKE 32,0: POKE 33,40: POKE 34,0: POKE 35,24: CAL
    L - 3086: CALL TR
370 VTAB 20: HTAB 10: CALL - 198: PRINT "(E)NCORE (M)
    ENU (T)ERMINE ";: CALL TR: GET R$
380 IF R$ = "E" THEN RESTORE : HOME : GOTO 40
390 IF R$ = "M" THEN PRINT : PRINT D$"RUN MENU"
400 IF R$ = "T" THEN TEXT : HOME : END
410 GOTO 370
420 FOR P = 1 TO 20:W = PEEK (49200):Y = PEEK ( - 16
    336): NEXT : RETURN

430 REM *****
440 REM L'INDICATIF EST CELUI
450 REM DE L'EMISSION
460 REM ALLEGRO MA NON TROPPO
470 REM DIFFUSEE A 11H30 (!)
480 REM SUR LES CHAINES DE
490 REM FRANCE-CULTURE

```

LE DIMANCHE !

SUITE AU VERSO

## INDICATIF (suite)

```
500 DATA 35,32,34,32,30,32,30,32,30,32,30,32,26,64,30
,32,34,32,30,32,30,32,35,32,40,32,45,64,35,32,34,32
,30,32,30,32,30,32,22,32,23,64,26,32,26,32,30,64
510 DATA 34,32,30,32,26,32,23,32,22,32,19,32,17,64,16
,32,17,32,19,32,23,32,17,32,19,32,22,64,26,32,34,32
,35,32,30,32,22,32,35,32,34,64,23,32,23,32,22,32,22
,32,22,64
520 DATA 45,16,35,16,30,16,22,32,0,0
```

Sans ces lignes de DATA, je ne vous promets plus rien !

## MUSIC.DIATM

Si vous possédez les programmes *DIAGRAMME* du n°2 de *Tremplin Micro*, inutile de retaper DIA.LM et DIA.CARAC.

Pour obtenir *MUSIC.DIATM*, contentez-vous de taper ces quelques lignes, puis de lancer le programme. Les lignes de Pokes installeront la routine *MUSIC* à l'adresse 24752 (\$60B0). Tout est automatique !

```
10 TEXT : HOME : PRINT "DISQUETTE DIAGRAMME SVP " ; GET R# : PRINT
20 PRINT CHR# (4)"BLOAD DIA.LM,A#6000" : PRINT CHR# (4)"BLOAD DIA.CARAC,
A#6100"
30 GOSUB 50 : PRINT : PRINT "DISQUETTE MUSIC SVP " ; GET R# : PRINT
40 PRINT CHR# (4)"BSAVE MUSIC.DIATM,A#6000,L#400" : END
50 POKE 24752,164 : POKE 24753,255 : POKE 24754,173 : POKE 24755,48 : POKE 24
756,192 : POKE 24757,136 : POKE 24758,208 : POKE 24759,5 : POKE 24760,206 :
POKE 24761,177 : POKE 24762,96
60 POKE 24763,240 : POKE 24764,9 : POKE 24765,202 : POKE 24766,208 : POKE 247
67,245 : POKE 24768,174 : POKE 24769,176 : POKE 24770,96 : POKE 24771,76 : P
OKE 24772,178 : POKE 24773,96 : POKE 24774,96 : RETURN
```

• Maintenant, pour celles et ceux qui se moquent du graphisme, voici les mêmes POKES pour *MUSIC.INIT*, à installer à l'adresse 768 (\$300).

Bien sûr, il conviendra d'éliminer tous les CALL TR, ainsi que les accès à la page graphique HGR2. De plus on aura IN = 768. Le programme ne s'en portera pas plus mal, je vous l'affirme, foi de Nestor !

## MUSIC.INIT

Remplacez la ligne 30 par : 30 GOSUB 530

```
530 POKE 768,164 : POKE 769,255 : POKE 770,173 : POKE 771,48 : POKE 772,192 : P
OKE 773,136 : POKE 774,208 : POKE 775,5 : POKE 776,206 : POKE 777,1 : POKE 7
78,3
540 POKE 779,240 : POKE 780,9 : POKE 781,202 : POKE 782,208 : POKE 783,245 : P
OKE 784,174 : POKE 785,0 : POKE 786,3 : POKE 787,76 : POKE 788,2 : POKE 789,3
: POKE 790,96 : RETURN
```



# Caractères français

transposés en page graphique par M. Devaux

Appeler la routine par un BLOAD FONTE.LM avant de faire un RUN FONTE.DEMO

**POKE 49235,0**  
rétablit les 4 lignes de texte dans la page HGR

**CALL — 998**  
monte le curseur

**CALL — 1008**  
recule le curseur

**CALL 16384**  
est l'adresse de la routine LM (\$4000)

**CALL — 990**  
déplace le curseur à la ligne figurant à l'adresse \$25 (CV)

## FONTE.DEMO

```

10 TEXT : NORMAL : PRINT CHR$ (21)
20 HGR : TEXT : HOME :T = 1
30 VTAB 21: PRINT "CTRL-T = TEXT - CTRL-F = FINI DE J
   " "CTRL-C = TERMINE ET LA PAGE HGR APPARAIT" +
   " UTILISEZ LES QUATRE FLECHES DE L'APPLE "
40 VTAB 1: HTAB 1: IF T = 0 THEN T = 1: TEXT : GOSUB 21
   0
50 GET A$:A = ASC (A$)
60 IF A = 6 THEN POKE 49235,0: GOTO 160
70 IF A = 20 THEN TEXT : GOSUB 210: GOTO 50
80 IF A = 11 THEN CALL - 998
90 IF A = 127 THEN : CALL - 1008:A$ = " ":D = 1
100 PRINT A$;: IF D = 1 THEN D = 0: CALL - 1008
110 V = PEEK (37): IF V = 19 THEN CALL - 958
120 H = PEEK (36)
130 IF A < > 3 THEN 50
140 POKE 49236,0: POKE 49234,0: POKE 49239,0: POKE 49232
   ,0:T = 0
150 CALL 16384: GOSUB 210: GOTO 50
160 VTAB 22: HTAB 1: PRINT "(M)ENU DE DISQUETTE (E)NCORE
   (T)ERMINE";: GET R$: PRINT
170 IF R$ = "M" THEN PRINT CHR$ (4)"RUN MENU"
180 IF R$ = "E" THEN 40
190 IF R$ < > "T" THEN 160
200 TEXT : HOME : END
210 POKE 36,H: POKE 37,V: CALL - 990: RETURN
    
```

Nous publions in-extenso le programme (inspiré de DIA.LM) que nous a adressé M. Devaux, mais nous l'avons complété par une petite routine de démonstration qui se suffit à elle-même.

**UTILISATION :** Sauvez FONTE.LM (pages 34-35) par un BSAVE FONTE.LM, A\$4000, L\$680.

**ATTENTION !** Ce programme ne supporte pas les modes INVERSE et FLASH.

### VERIFICATION

T = 0 : FOR I = 16384 TO 18047 : T = T + PEEK(I) : NEXT : PRINT T 65526

## FONTE.LM 1

4000- A9 01 85 E7 EA EA EA EA  
 4008- 20 C1 FB A0 00 B1 28 20  
 4010- 22 40 C8 C0 28 D0 F6 A5  
 4018- 25 18 69 01 C9 18 D0 E6  
 4020- 60 00 84 06 85 07 C9 A0  
 4028- D0 01 60 A5 25 A0 00 18  
 4030- 65 25 C8 C0 07 00 F8 85  
 4038- 00 A5 06 A0 00 18 65 06  
 4040- C8 C0 06 D0 F8 85 09 A9  
 4048- 24 38 E5 06 30 07 A9 00  
 4050- 85 0A 4C 59 40 A9 01 85  
 4058- 0A A6 09 A4 0A A5 08 20  
 4060- 11 F4 A9 80 A0 40 A2 A1  
 4068- E4 07 F0 0A E8 18 69 10  
 4070- 90 F6 C8 4C 68 40 AA A9  
 4078- 01 20 5D F6 A4 06 A9 A0  
 4080- 91 28 60 00 00 00 00 00  
 4088- 00 00 00 00 00 00 00 00  
 4090- 09 36 0D 24 07 00 00 00  
 4098- 00 00 00 00 00 00 00 00  
 40A0- 09 2D DE 33 15 3F 0E BE  
 40A8- 0D 2D 2D 00 00 0B 00 00  
 40B0- 49 8D 3F BF 0D 15 3F 97  
 40B8- 75 24 0D 1E 06 00 00 00  
 40C0- 31 25 4D BA 17 17 17 4D  
 40C8- 35 3F 00 00 00 00 00 00  
 40D0- 09 15 B6 0D 1E 0E DE 38  
 40D8- 1C 2C E0 3C 00 00 00 00  
 40E0- 49 36 06 00 00 00 00 00  
 40E8- 00 00 00 00 00 00 00 00  
 40F0- 49 1E 1E 36 0E 0E 05 00  
 40F8- 00 00 00 00 00 00 00 00  
 4100- 49 0E 0E 36 1E 1E 07 00  
 4108- 00 00 00 00 00 00 00 00  
 4110- 49 8D 1F 1F 0E 2D 1E 17  
 4118- 2D 15 1F 1F 4E 05 00 00  
 4120- 49 32 8D 3F 3F 4E 2E 00  
 4128- 00 00 00 00 00 00 00 00  
 4130- 49 92 32 1E 07 00 00 00  
 4138- 00 00 00 00 00 00 00 00  
 4140- 91 2A 2D 2D 00 00 00 00  
 4148- 00 00 00 00 00 00 00 00  
 4150- 49 92 92 07 00 00 00 00  
 4158- 00 00 00 00 00 00 00 00  
 4160- 49 89 17 17 17 17 07 00  
 4168- 00 00 00 00 00 00 00 00  
 4170- 09 2D 15 DF 33 36 36 29  
 4178- 2D 20 24 F7 1E 06 00 00  
 4180- 49 3E 0E 36 AE 3F 07 00  
 4188- 00 00 00 00 00 00 00 00  
 4190- 0A 0C 2D 15 BE BF 17 2E  
 4198- 2D 2D 00 00 00 00 00 00

41A0- 29 2D 35 17 3E 4E F6 3F  
 41A8- 07 20 00 00 00 00 00 00  
 41B0- 49 31 BF 0D FE 33 2D 2D  
 41B8- 1E 36 00 00 00 00 00 00  
 41C0- 92 52 0E 2D 05 20 E4 3F  
 41C8- 27 2C 2D 2D 00 00 00 00  
 41D0- 49 2D DE BB 2E 2D 15 F6  
 41D8- 3F 07 20 05 00 00 00 00  
 41E0- 29 2D 35 1E 1E 1E 36 06  
 41E8- 00 00 00 00 00 00 00 00  
 41F0- 09 2D 15 B6 F6 3F 07 20  
 41F8- 4D 38 3F 20 07 00 00 00  
 4200- 09 2D 15 36 3F 27 18 B6  
 4208- 12 2D 05 28 38 00 00 00  
 4210- 49 12 16 06 00 00 00 00  
 4218- 00 00 00 00 00 00 00 00  
 4220- 49 12 16 F6 07 00 00 00  
 4228- 00 00 00 00 00 00 00 00  
 4230- 49 F1 1E 1E 0E 0E 0E 06  
 4238- 00 00 00 00 00 00 00 00  
 4240- 91 2D 2D 16 3F 3F 06 00  
 4248- 00 00 00 00 00 00 00 00  
 4250- 09 0E 0E 0E 1E 1E 1E 06  
 4258- 00 00 00 00 00 00 00 00  
 4260- 0A 0C 2D 15 1E 1E B6 07  
 4268- 00 00 00 00 00 00 00 00  
 4270- 52 29 25 A8 16 36 3E 3F  
 4278- 07 60 2D 04 00 00 00 00  
 4280- 91 36 36 4D 21 3C 3F 4C  
 4288- 21 1C 1C 17 06 00 00 00  
 4290- 31 36 36 2E 2D 05 20 1C  
 4298- 3F 4C 21 1C 3F 06 00 00  
 42A0- 09 2D 15 DF 33 36 76 2D  
 42A8- 05 20 00 00 00 00 00 00  
 42B0- 31 36 36 2E 2D 05 20 24  
 42B8- E4 3F 06 00 00 00 00 00  
 42C0- 29 2D F5 DB 36 2D F5 18  
 42C8- 36 2D 2D 04 00 00 00 00  
 42D0- 29 2D F5 DB 36 2D DE 36  
 42D8- 06 00 00 00 00 00 00 00  
 42E0- 09 2D F5 DB 36 36 0E 2D  
 42E8- 25 3C 07 00 00 00 00 00  
 42F0- 31 36 36 2E 49 24 3C 3F  
 42F8- 4C 21 3C 00 00 00 00 00  
 4300- 09 2D 1E 36 36 15 3F 07  
 4308- 00 00 00 00 00 00 00 00  
 4310- 49 09 36 36 F6 3F 1C 06  
 4318- 00 00 00 00 00 00 00 00  
 4320- 31 36 36 6E 09 1C 1C 1C  
 4328- 0C 0C 0C 05 00 00 00 00  
 4330- 31 36 36 2E 2D 2D 00 00  
 4338- 00 00 00 00 00 00 00 00



## FONTE.LM 2

4340- 31 36 36 6E 09 24 24 24  
 4348- 17 D6 E4 04 00 00 00 00  
 4350- 31 36 36 6E 09 24 E7 1C  
 4358- 0C 48 36 2E 00 00 00 00  
 4360- 11 36 36 0E 2D 05 20 24  
 4368- 24 3B 37 00 00 00 00 00  
 4370- 31 36 36 2E 08 08 60 E4  
 4378- 3F 96 25 00 00 00 00 00  
 4380- 11 36 36 0E 6D 1C 07 68  
 4388- 24 E4 3F 06 00 00 00 00  
 4390- 29 2D 15 B6 12 07 38 28  
 4398- 38 B7 1A 24 24 3C 00 00  
 43A0- 09 2D 15 DF 33 0E 2D 15  
 43A8- F6 3F 07 20 00 00 00 00  
 43B0- 29 2D F5 33 36 36 07 00  
 43B8- 00 00 00 00 00 00 00 00  
 43C0- 31 36 36 0E 2D 05 20 24  
 43C8- 24 07 00 00 00 00 00 00  
 43D0- 31 36 AE 15 05 60 24 24  
 43D8- 07 00 00 00 00 00 00 00  
 43E0- 31 36 36 2E 20 08 AE 0E  
 43E8- 24 24 24 07 00 00 00 00  
 43F0- 31 0E 16 17 6E 09 E4 1C  
 43F8- 0C 0C 3C 00 00 00 00 00  
 4400- 31 0E 96 0A 24 64 0C 3C  
 4408- 00 00 00 00 00 00 00 00  
 4410- 29 2D 35 17 17 17 17 2E  
 4418- 2D 2D 00 00 00 00 00 00  
 4420- 49 15 1E 07 20 00 00 00  
 4428- 00 00 00 00 00 00 00 00  
 4430- 52 29 2D DE 1B 76 2D 35  
 4438- BB 07 00 00 00 00 00 00  
 4440- 09 2D 35 DB 73 2D 15 1E  
 4448- 3F 1C 96 2D 2D 38 00 00  
 4450- 92 0A 0C 0C 0E 0E 06 00  
 4458- 00 00 90 00 00 00 00 00  
 4460- 92 92 2A 2D 2D 25 00 00  
 4468- 00 00 00 00 00 00 00 00  
 4470- 09 0E 0E 06 00 00 00 00  
 4478- 00 00 00 00 00 00 00 00  
 4480- 12 09 2D 15 36 3E 3F 07  
 4488- 20 29 25 00 00 00 00 00  
 4490- 31 36 36 2E 2D 05 20 24  
 4498- 3B 27 00 00 00 00 00 00  
 44A0- 12 09 2D 35 12 3A 3F 07  
 44A8- 20 3C 00 00 00 00 00 00  
 44B0- 49 09 36 36 36 3F 3F 20  
 44B8- 24 29 25 00 00 00 00 00  
 44C0- 91 32 36 29 2D 04 20 1C  
 44C8- 3F 16 2D 04 00 00 00 00  
 44D0- 89 36 37 31 2E 08 18 28  
 44D8- 0C 20 3B 07 00 00 00 00

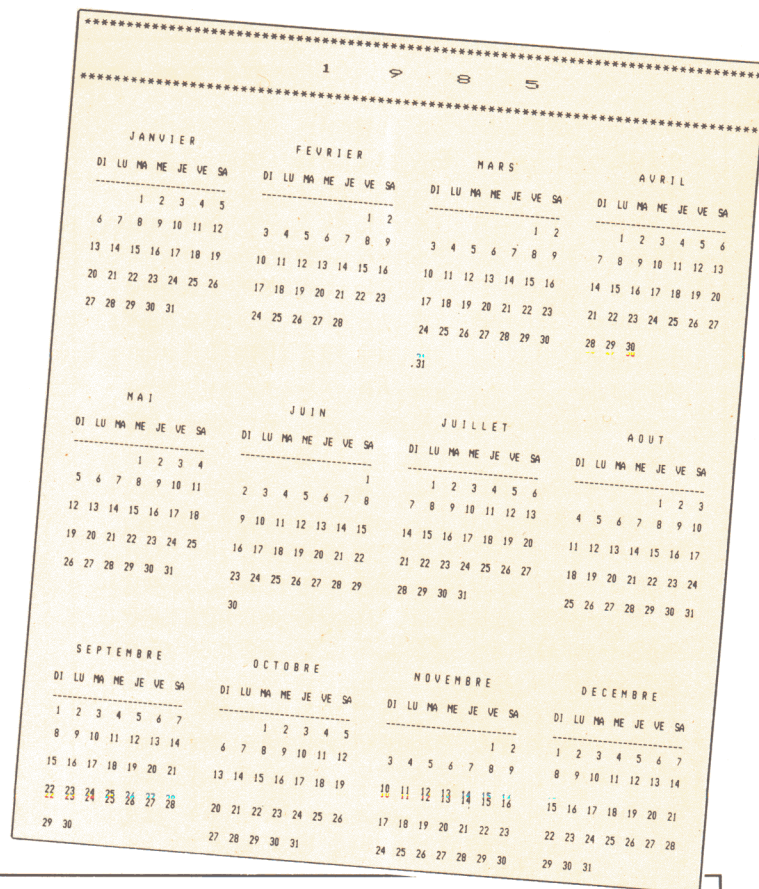
44E0- 91 32 96 29 2D 20 24 1C  
 44E8- 3F 96 2D 06 00 00 00 00  
 44F0- 31 36 36 2E 49 24 24 3B  
 44F8- 27 00 00 00 00 00 00 00  
 4500- 49 16 77 36 2E DE 38 00  
 4508- 00 00 00 00 00 00 00 00  
 4510- 49 B1 77 36 F6 3F 20 00  
 4518- 00 00 00 00 00 00 00 00  
 4520- 31 36 36 2E 08 38 4C 0C  
 4528- 96 3A 20 00 00 00 00 00  
 4530- 09 35 36 36 3E 4E 20 00  
 4538- 00 00 00 00 00 00 00 00  
 4540- 91 36 36 0D 20 3C 28 29  
 4548- 36 36 06 00 00 00 00 00  
 4550- 91 36 36 4D 21 24 1C 3F  
 4558- 06 00 00 00 00 00 00 00  
 4560- 91 32 76 2D 05 20 24 3B  
 4568- 37 00 00 00 00 00 00 00  
 4570- 91 2D AD F6 3F D6 24 24  
 4578- 05 00 00 00 00 00 00 00  
 4580- 91 32 0E 2D 56 24 24 3C  
 4588- 3F 06 00 00 00 00 00 00  
 4590- 91 36 36 45 08 08 38 BF  
 4598- 06 00 00 00 00 00 00 00  
 45A0- 92 61 2D B5 3B BF 2A 2D  
 45A8- 05 28 00 00 00 00 00 00  
 45B0- 09 36 77 36 0E 2D 20 18  
 45B8- 38 06 00 00 00 00 00 00  
 45C0- 91 36 76 6D 3C 0C 24 04  
 45C8- 00 00 00 00 00 00 00 00  
 45D0- 91 36 0E 0E 05 28 20 24  
 45D8- 00 00 00 00 00 00 00 00  
 45E0- 91 36 36 2D 20 95 25 24  
 45E8- 24 00 00 00 00 00 00 00  
 45F0- 91 15 15 17 17 4D E1 04  
 45F8- 28 20 00 00 00 00 00 00  
 4600- 91 36 0E 2D DE 2A 2D 20  
 4608- 24 24 00 00 00 00 00 00  
 4610- 91 2D 2D 1E 1E 1E 17 2D  
 4618- 2D 05 00 00 00 00 00 00  
 4620- 91 32 76 2D 25 20 04 08  
 4628- 18 17 3E B7 2D 04 00 00  
 4630- 91 36 76 6D 3C 0C 24 04  
 4638- 1B 1C 04 00 00 00 00 00  
 4640- 91 32 76 2D 25 20 04 20  
 4648- 33 3A B7 2D 04 00 00 00  
 4650- 69 09 05 00 00 00 00 00  
 4658- 00 00 00 00 00 00 00 00  
 4660- 0A 2D 2D 3E 3F 37 2D 2D  
 4668- 3E 3F 37 2D 2D 3E 3F 3F  
 4670- 00 00 00 00 00 00 00 00  
 4678- 00 00 00 00 00 00 00 00

Michel EDELIN  
vous propose son

# CALENDRIER PERPÉTUEL

Chaque année,  
vous l'éditez sur :

- **IMAGEWRITER** ou
- **EPSON RX 80** (voir page 38)



```

10 TEXT : NORMAL : PRINT CHR$(21); CHR$(12): HOME
15 PRINT "*****"
20 PRINT "***** CALENDRIER *****"
25 PRINT "***** PERPETUEL *****"
30 PRINT "*****"
35 FOR I = 1 TO 1500: NEXT I
40 GOTO 485
45 :
50 REM SOUS PROGRAMME D'ECRITURE DE CHAINES DE CARACTERES
55 :
60 ES$ = " ": FOR X = 1 TO ES:ES$ = ES$ + E$: NEXT X: RETURN
65 :
70 REM CALCUL ET POSITION DU 1 JOUR
75 :
80 P = (U - 1) / 100:Q = INT (P / 4):M = 36 + INT ((5 * (U - 1)) / 4) - I
  NT (P) + Q
85 Y = M - (7 * INT (M / 7)) + 1: IF INT (U / 4) = U / 4 THEN B = 1:M(2) =
  29
90 A(1) = M(1):C(1) = Y: FOR W = 2 TO 11:A(W) = A(W - 1) + M(W):C(W) = Y + (
  A(W - 1) - (INT (A(W - 1) / 7) * 7)): IF C(W) > 7 THEN C(W) = C(W) - 7
95 NEXT W
100 IF Y = < 0 THEN Y = 7 + Y
105 N = M(I): IF I = 2 AND B = 1 THEN N = 29
110 PA$ = " PATIENTE "
115 HOME : VTAB 12: FOR PA = 1 TO 30 STEP 10: INVERSE : HTAB 15: PRINT MID$(
  (PA$,PA,10): NEXT : NORMAL
120 IF Y = 1 THEN E1$ = " ": GOTO 130
125 ES = Y - 1:E$ = " ": GOSUB 60:E1$ = ES$

```

Sous ProDOS, remplacer  
PRE1 par PRINT CHR\$(4);  
"PRE1" (ligne 215), puis  
PRE0 par PRINT CHR\$(4);  
"PRE0" (ligne 350)



```

130 EE$ = " ";MO$(I) = ""
135 FOR K = 1 TO N: IF LEN ( STR$(K) ) = 1 THEN K$ = " " + STR$(K): GOTO
    145
140 K$ = STR$(K)
145 MO$(I) = MO$(I) + K$ + EE$
150 NEXT K
155 NM = LEN (MO$(I) + E1$):ES = 168 - NM:E$ = " ": GOSUB 60:E2$ = ES$
160 MO$(I) = E1$ + MO$(I) + E2$
165 I = I + 1:Y = Y + (N - 28): IF Y > 7 THEN Y = Y - 7
170 IF I = 13 THEN GOSUB 185: GOSUB 245: GOTO 490
175 GOTO 105
180 :
185 REM COUPURE DU MOIS EN SEMAINE
190 :
195 FOR G = 1 TO 12:P = 1: FOR D = 1 TO 6:MI$(G,D) = MID$(MO$(G),P,28):P =
    P + 28: NEXT D: NEXT G: RETURN
200 :
205 REM EDITION DE L'ANNEE
210 :
215 PR# 1: PRINT CHR$(27);"N"; CHR$(15); CHR$(9);"136N"
220 U$ = "": FOR I = 1 TO 3:U$ = U$ + MID$(STR$(U),I,1) + " ": NEXT :U$
    = U$ + RIGHT$(STR$(U),1)
225 E$ = "*":ES = 80: GOSUB 60: PRINT ES$: PRINT
230 PRINT CHR$(27);"N"; CHR$(14); TAB(15);U$: PRINT : PRINT CHR$(15);E
    S$: PRINT : PRINT : PRINT
235 RETURN
240 :
245 REM EDITION DU CALENDRIER
250 :
255 FOR H = 1 TO NE: IF H > 1 THEN PRINT CHR$(12)
260 GOSUB 205
265 FOR G = 1 TO 12 STEP ML: IF ML = 4 THEN 295
270 IF ML = 2 THEN 285
275 TA = 12: PRINT CHR$(27);"N"; CHR$(14): IF (G / 2) - INT (G / 2) = 0 T
    HEN PRINT : PRINT : PRINT : GOTO 300
280 IF G > 1 THEN PRINT CHR$(12): GOSUB 205: PRINT CHR$(14): GOTO 300
285 IF G = 7 THEN PRINT CHR$(12): GOSUB 205
290 TA = 11:ES = 23:E$ = " ": GOSUB 60: GOTO 300
295 TA = 10:ES = 16:E$ = " ": GOSUB 60: PRINT CHR$(27);"Q"; CHR$(15);
300 PRINT TAB(TA);MA$(G);: IF ML = 1 THEN PRINT "": PRINT : PRINT : PRINT
    :TA = 8:ES$ = "": GOTO 315
305 PRINT ES$;MA$(G + 1);: IF ML = 2 THEN PRINT "": PRINT :TA = 6:ES = 12:
    GOSUB 60: GOTO 315
310 PRINT ES$;MA$(G + 2);ES$;MA$(G + 3): PRINT :TA = 6:ES = 5: GOSUB 60
315 PRINT TAB(TA);: FOR Z = 1 TO ML: PRINT JO$;ES$;: NEXT Z: PRINT : IF ML
    = 4 THEN PRINT CHR$(27)"r": PRINT CHR$(27)"f";
320 PRINT TAB(TA): FOR Z = 1 TO ML: PRINT SO$;ES$;: NEXT Z: PRINT : IF ML
    = 4 THEN PRINT CHR$(27)"r": PRINT CHR$(27)"f";
325 FOR D = 1 TO 6
330 PRINT TAB(TA);MI$(G,D);: IF ML = 1 THEN PRINT "": PRINT : PRINT : GOT
    O 345
335 PRINT ES$;MI$(G + 1,D);: IF ML = 2 THEN PRINT "": PRINT : GOTO 345
340 PRINT ES$;MI$(G + 2,D);ES$;MI$(G + 3,D): PRINT

```

(SUITE PAGE 38)

```

345 NEXT D: IF ML = 4 THEN PRINT : PRINT
350 NEXT G: NEXT H: PRÉ 0: RETURN
355 :
360 REM ENTREE DES DONNEES
365 :
370 B$ = " ENTREE L'ANNEE
    E CONSIDEREE "
375 HOME : VTAB 2: FOR J = 1 TO 100 STEP 20: HTAB 10: INVERSE : PRINT MID$(B$,J,20): NEXT J
380 VTAB 12: HTAB 16: PRINT "-> ": NORMAL
385 VTAB 12: HTAB 20: PRINT "1900";: HTAB 20: INPUT "":U$:U = VAL(U$): IF U > 2000 THEN 385
390 IF NOT U THEN TEXT : HOME : END
395 HOME : VTAB 4: HTAB 11: INVERSE : PRINT "POUR L'ANNEE ":U: NORMAL
400 VTAB 8: PRINT : PRINT "NOMBRE DE MOIS PAR LIGNE (1/2/4) -> ": GET ML$:M L = VAL(ML$): IF ML < 1 OR ML = 3 OR ML > 4 THEN 400
405 PRINT ML$
410 IF ML = 1 THEN NP = 6:S$ = "S": GOTO 425
415 IF ML = 2 THEN NP = 2:S$ = "S": GOTO 425
420 NP = 1:S$ = ""
425 PRINT : PRINT "EDITION DU CALENDRIER SUR ";NP;" PAGE";S$
430 PRINT : INPUT "NOMBRE D'EDITION DESIREE -> ";NE
435 NP = NP * NE: IF NP > 1 THEN S$ = "S"
440 PRINT : PRINT "CHARGER AU MOINS ";NP;" FEUILLE";S$
445 VTAB 20: HTAB 1: PRINT "VALIDATION (O/N) ": GET A$
450 IF A$ < > "O" AND A$ < > "N" THEN 445
455 PRINT A$
460 IF A$ = "N" THEN 360
465 GOTO 80
470 :
475 REM VARIABLES DIVERSES
480 :
485 DIM M(12),A(12),C(12),MO$(12),MI$(12,12),MA$(12)
490 B = 0:M(1) = 31:M(3) = 31:M(5) = 31:M(7) = 31:M(8) = 31:M(10) = 31:M(12) = 31:M(4) = 30:M(6) = 30:M(9) = 30:M(11) = 30:M(2) = 28:I = 1
495 JO$ = "DI LU MA ME JE VE SA ":SO$ = "-----"
500 MA$(1) = " J A N V I E R ":MA$(9) = " S E P T E M B R E ":MA$(3) = " M A R S ":MA$(4) = " A V R I L ":MA$(5) = " M A I ":M A$(6) = " J U I N ":MA$(8) = " A O U T "
505 MA$(7) = " J U I L L E T ":MA$(10) = " O C T O B R E ":MA$(11) = " N O V E M B R E ":MA$(12) = " D E C E M B R E ":MA$(2) = " F E V R I E R "
510 GOTO 360

```

### MODIFICATIONS POUR EPSON RX 80

```

215 PRÉ1 : PRINT CHR$(27); "a"; CHR$(27); "G"
220 U$ = STR$(U) : E$ = " " : U$ = LEFT$(U$,1) + E$ + MID$(U$,2,1) + E$ + MID$(U$,3,1) + E$ + RIGHT$(U$,1)
230 PRINT CHR$(27); "W"; CHR$(1); TAB(15); U$; CHR$(27); "W"; CHR$(0) : PRINT : PRINT E$: PRINT : PRINT : PRINT
275 TA = 12 : PRINT CHR$(27); "W"; CHR$(1) : IF (G/2) - INT(G/2) = 0 THEN PRINT : PRINT : PRINT : GOTO 300
280 IF G = 1 THEN PRINT CHR$(12) : GOSUB 205 : PRINT

```

```

CHR$(27); "W"; CHR$(1) : GOTO 300
285 IF G = 7 THEN PRINT CHR$(12) : GOSUB 205 : GOTO 290
290 TA = 11 : ES = 23 : E$ = " " : GOSUB 60 : PRINT CHR$(18); : GOTO 300
295 TA = 10 : ES = 16 : E$ = " " : GOSUB 60 : PRINT CHR$(15); :
315 PRINT TAB(TA); : FOR Z = 1 TO ML : PRINT JO$; E$; : NEXT Z : PRINT
320 PRINT TAB(TA); : FOR Z = 1 TO ML : PRINT SO$; E$; : NEXT Z : PRINT ""

```



# Votre bibliothèque INFORMATIQUE

par Guy-HACHETTE

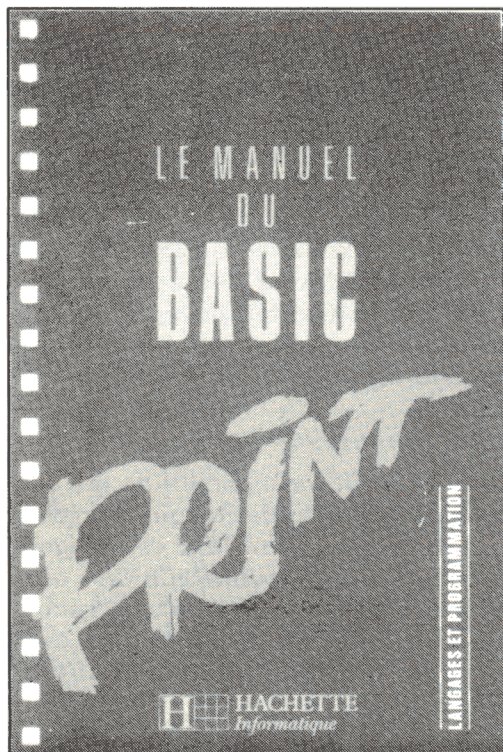
## ● *Créez vos jeux d'aventure sur micro-ordinateur*

C'est le titre du livre d'un auteur français, J.-M. PEZERET (Eyrolles), et cela méritait d'être souligné, à mon avis du moins.

Comment élaborer un scénario, choisir un thème, structurer les données et le programme lui-même, utiliser le graphisme ? Le sujet est vaste et le projet ambitieux !

L'ouvrage se propose de donner un support technique à votre imagination, et de décrire une méthodologie de programmation pour la réalisation de jeux d'aventure, notamment au travers d'un exemple concret, que vous bâtirez de A à Z. Il semble qu'il y parvienne, mais le Basic est celui du TO7 et du MO5, c'est-à-dire un langage qui, notamment sur le plan du graphisme, diffère très sensiblement de celui de l'Apple. Il serait intéressant de réaliser le même ouvrage en Applesoft.

Pourtant, si vous le pouvez, lisez le livre de J.-M. Pézeret et notamment son chapitre 6, consacré aux modules de base.



## ● *Le Basic vu par des pédagogues*

Connaître le Basic est à l'ordre du jour, notamment dans l'enseignement. Les ouvrages en français sont déjà nombreux, mais ne résultent pas toujours de travaux récents. Est-ce la raison qui a conduit Hachette Informatique à publier une excellente traduction de ce *Manuel du Basic* (par M. Waite et M. Pardee) ? Peut-être.

Il reste que les Français disposent maintenant d'un véritable cours qui leur permettra de se familiariser avec le Basic ou d'en approfondir la pratique.

Un détail : le chapitre 7 y est consacré à un programme de simulation du Rubik's cube sur ordinateur personnel.

*Le Manuel du Basic* comporte de nombreuses routines développées avec le Basic IBM. On sait qu'il s'agit du Basic Microsoft. Ce n'est évidemment pas l'idéal pour qui possède un Apple, mais la découverte d'un autre langage, surtout lorsqu'il est très proche du vôtre, se révèle généralement passionnante, n'est-il pas vrai ?

G.-H.



# ÉCRANS CONDENSÉS

## UN PROGRAMME DE MAURICE CHAVELLI

DANS la série "Ecrans condensés", le numéro 3 de *Tremplin Micro* présentait le programme "ECRANLM" permettant d'économiser de la place sur les disquettes pour mémoriser des masques en 40 colonnes. Le présent programme va plus loin sur deux plans :



- 1 Les masques sont codés suivant un principe simple : tout caractère qui se répète est détecté et à partir du troisième, on se contente de compter ces caractères. A la fin du comptage, on inscrit le nombre obtenu. Cette technique permet d'économiser facilement la moitié de la mémoire. Pour ne pas trop compliquer le programme, j'ai "ignoré" le cas où un caractère se répète plus de 256 fois. Si cela se produit sur un masque, vous aurez des mélanges. Si tous vos masques comportent un cadre, il n'y aura aucun problème.
- 2 Les masques sont regroupés dans une TABLE qui est ensuite mémorisée sur disquette. Il suffit donc de caser cette table quelque part en mémoire, l'appel d'un masque se faisant simplement en indiquant son numéro.

### CREATION

Le programme basic ECR.CREER permet de créer la TABLE à partir de masques(\*) déjà stockés sur disquette. Il suffit de suivre les instructions du programme ; la table se construit à partir de l'adresse \$2000 (8192 en décimal). La routine ECR.IN est chargée du codage et du transfert, elle a été agrémentée de nombreux commentaires et, pour ne pas dépayser le lecteur, elle utilise le même principe d'exploration de l'écran que ECRANLM du numéro 3. J'ai d'ailleurs supprimé la référence à l'adresse \$25 qui correspond à la position verticale du curseur (VTAB), celle-ci est en effet inutile car elle est ignorée par la routine BASCALC (\$FBC1) qui utilise "A" comme indicateur de ligne... le hasard fait parfois bien les choses ! La table contient en tête les adresses de début de chaque masque (2 octets par masque), puis les masques codés à la file. Attention à l'occupation mémoire ! Le programme vous l'indique en permanence.

### LECTURE

Le programme basic ECR.LIRE vous montre comment utiliser la routine ECR.OUT qui permet de lire une TABLE. Le programme est prévu pour lire à l'adresse \$2000 (il faut donc au préalable charger une table par BLOAD TABLE,A\$2000), mais il est très facile à modifier (ligne 55) ; d'ailleurs la TABLE est relogeable, mais attention à ne pas "écraser" un programme ! Pour changer un peu, la routine ECR.OUT écrit les lignes en bas de l'écran, puis fait un scolling (montée de l'écran d'une ligne), ce qui est plus agréable à l'œil ! Vous pouvez imaginer d'autres transferts : de haut en bas ; de droite à gauche... si vous avez du temps et du courage.

Maurice CHAVELLI.

(\*) Commencez donc par les créer, en utilisant le programme paru dans *Tremplin Micro* n°1 (de la page 17 à la page 23). Attention ! il s'agit d'écrans TEXT et non de pages graphiques.



## ● ECR.CREER

```
10 TEXT :D$ = CHR$(4): POKE 6,0: POKE 7,33:N = 0:AD = 8194: POKE AD - 2,
0: POKE AD - 1,33
14 REM -----
15 REM CHARGEMENT ROUTINE
16 REM -----
20 PRINT D$"BLOAD ECR.IN"
24 REM -----
25 REM MENU
26 REM -----
30 GOSUB 370: VTAB 20: INVERSE : PRINT "ENTRER 'C' POUR LE CATALOGUE": HTA
B 6: PRINT "ENTRER 'F' POUR TERMINER ": NORMAL : VTAB 16: HTAB 6: INP
UT "NOM DU MASQUE ? ";A$
40 IF A$ = "C" THEN HOME : PRINT D$"CATALOG": PRINT : PRINT "TOUCHE QUELC
ONQUE POUR LA SUITE": GOSUB 570: GOTO 30
50 IF A$ = "F" THEN 140
60 B = PEEK (6):C = PEEK (7)
70 ONERR GOTO 530
74 REM -----
75 REM CHARGEMENT MASQUE
76 REM -----
80 PRINT D$"BLOAD";A$: POKE 216,0
90 POKE 6,B: POKE 7,C
100 GOSUB 570: IF A$ = CHR$(27) THEN 30
110 IF A$ = CHR$(13) THEN 130
120 CALL - 198: GOTO 100
124 REM -----
125 REM APPEL ROUTINE
126 REM ET MISE A JOUR ADRESSE
127 REM -----
130 CALL 768:N = N + 1: POKE AD, PEEK (6): POKE AD + 1, PEEK (7):AD = AD +
2: GOTO 30
140 GOSUB 620: HTAB 12: PRINT "(S)AUEGARDE": PRINT : HTAB 12: PRINT "(A)NN
ULATION"
150 GOSUB 570: IF A$ = "A" THEN RUN
160 IF A$ = "S" THEN 180
170 CALL - 198: GOTO 150
180 GOSUB 620: HTAB 6: INPUT "SOUS QUEL NOM? ";A$
184 REM -----
185 REM COMPACTAGE TABLE
186 REM -----
190 POKE 60,0: POKE 61,33: POKE 62, PEEK (6): POKE 63, PEEK (7): POKE 66,N
* 2: POKE 67,32: CALL 865
200 FOR E = 0 TO 2 * N - 2 STEP 2:V = PEEK (8192 + E) + PEEK (8193 + E) *
256 - 256 + 2 * N:H = INT (V / 256): POKE 8192 + E,V - H * 256: POKE 8
193 + E,H: NEXT
204 REM -----
205 REM SAUEGARDE TABLE
206 REM -----
210 PRINT D$"BSAVE";A$;" ,A$2000,L";D
220 GOSUB 620: HTAB 6: PRINT "UNE AUTRE TABLE ?"
230 GOSUB 570: IF A$ = "0" THEN RUN
```

Copiez la routine **ECR.IN** (page 43) et sauvez-la sur disquette par un **BSAVE ECR.IN, A\$300, L\$67**

Sous **ProDOS**, transformez ainsi la ligne 80 :

```
80 POKE 48984,0 : PRINT D$
"BLOAD";A$: POKE 216,0
```

Cela vous évitera l'inévitable message d'erreur :

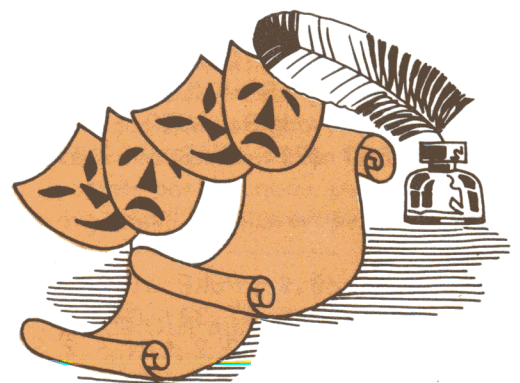
**NO BUFFERS AVAILABLE**

SUITE PAGE 42

```

240 IF A$ = "N" THEN 290
250 CALL - 198: GOTO 230
260 REM -----
270 REM POUR LE PLAISIR
280 REM -----
290 A$ = "ALORS SALUT... A LA PROCHAINE... ET BONJOUR CHEZ TOI...":B$ = "":
HOME : VTAB 10
300 FOR C = 1 TO LEN (A$): FOR D = 1 TO 50: NEXT D: IF PEEK ( - 16384) >
128 THEN 330
310 IF LEN (B$) < 39 THEN B$ = B$ + MID$ (A$,C,1): HTAB 40 - LEN (B$): P
RINT B$;: NEXT C
320 B$ = RIGHT$ (B$, LEN (B$) - 1) + MID$ (A$,C,1): HTAB 1: PRINT B$;: NEX
T C: GOTO 300
330 NORMAL : HOME : END
340 REM -----
350 REM S/P AFFICHAGE ECRAN
360 REM -----
370 HOME : VTAB 1: HTAB 8
380 PRINT "CREATION TABLE DE MASQUES"
390 HTAB 8: FOR E = 1 TO 25: PRINT "-";: NEXT
400 VTAB 5: HTAB 1: GOSUB 490
410 D = PEEK (6) + PEEK (7) * 256 - 8448 + 2 * N
420 VTAB 8: HTAB 5: PRINT "LONGUEUR DE LA TABLE : ";D
430 VTAB 10: HTAB 5: PRINT "NOMBRE DE MASQUES : ";N
440 VTAB 12: GOSUB 490: VTAB 22: GOSUB 490
450 VTAB 22: FOR E = 1 TO 40: PRINT "_";: NEXT
460 PRINT : PRINT " <ESC>: NON VALIDE ... <RETURN>: VALIDE";
470 VTAB 16: HTAB 6
480 RETURN
490 FOR E = 1 TO 40: PRINT "_";: NEXT : RETURN
500 REM -----
510 REM TRAITEMENT ERREUR
520 REM -----
530 HOME : VTAB 10: HTAB 12: FLASH : PRINT "INTROUWABLE !": NORMAL : FOR E
= 1 TO 10: CALL - 198: NEXT : GOTO 30
540 REM -----
550 REM S/P ATTENTE TOUCHE
560 REM -----
570 IF PEEK ( - 16384) < 128 THEN 570
580 GET A$: RETURN
590 REM -----
600 REM S/P EFFACE LIGN. 16 A 21
610 REM -----
620 POKE 34,15: POKE 35,21: CALL - 936
630 POKE34,0: POKE35,24: RETURN

```



Faites connaître **TREMPIN MICRO** à ceux de vos amis qui possèdent un Apple (et n'oubliez pas qu'un numéro de la revue ne coûte pas plus cher que quelques mauvaises photocopies !).



# ● ECR.IN

## Assemblage par ProCODE

	0	BAS	EQU	\$6	; Adresse de...
	1	HAUT	EQU	\$7	; ...la table
	2	LIGNE	EQU	\$8	
	3	SVA	EQU	\$9	; Sauvegarde A
	4	DBLE	EQU	\$1A	; 0 si même caractère
	5	NBR	EQU	\$1B	; Nbr de caracteres identiques
	6	BASL	EQU	\$28	; Adresse première colonne
	7	BASCALC	EQU	\$FBC1	; Calcule l'adresse de
	8	MOVE	EQU	\$FE2C	; Déplacement mémoire
	9	*			* Base BASL d'après A
	10		ORG	\$300	
0300:	A9 01	11	LDA	£\$1	; *
0302:	85 1A	12	STA	DBLE	; *
0304:	A9 00	13	LDA	£\$0	; * Initialisations
0306:	85 08	14	STA	LIGNE	; *
0308:	85 1B	15	STA	NBR	; *
030A:	85 09	16	STA	SVA	; *
030C:	20 C1 FB	17	DEBUT	JSR	BASCALC
030F:	A0 00	18	LDY	£\$0	; Début de boucle Y
0311:	B1 28	19	BCLY	LDA	(BASL),Y ; On va chercher le caractère
0313:	A6 1A	20	LDX	DBLE	; Est-on en "double" ?
0315:	D0 1B	21	BNE	SUITE2	; Non alors...
0317:	C5 09	22	CMP	SVA	; Oui -> toujours le même ?
0319:	D0 05	23	BNE	SUITE1	; Non alors...
031B:	E6 1B	24	INC	NBR	; Oui -> on incrémente NBR
031D:	4C 3D 03	25	JMP	SUITE4	
0320:	48	26	SUITE1	PHA	; On mémorise A sur la pile
0321:	A9 01	27	LDA	£\$1	; On "marque"
0323:	85 1A	28	STA	DBLE	; DBLE -> plus en "double"
0325:	A5 1B	29	LDA	NBR	; et on écrit le...
0327:	20 54 03	30	JSR	ECRIT	; ...NBR à la suite
032A:	A9 00	31	LDA	£\$0	; On initialise...
032C:	85 1B	32	STA	NBR	; ...le NBR
032E:	68	33	PLA		; On récupère A
032F:	4C 3A 03	34	JMP	SUITE3	
0332:	C5 09	35	SUITE2	CMP	SVA ; Est-ce le même ?
0334:	D0 04	36	BNE	SUITE3	; Non, alors on poursuit
0336:	A2 00	37	LDX	£\$0	; Oui, alors on...
0338:	86 1A	38	STX	DBLE	; ..."marque" DBLE -> 0
033A:	20 54 03	39	SUITE3	JSR	ECRIT ; On écrit le caractère
033D:	C8	40	SUITE4	INY	; Un caractère de plus
033E:	C0 28	41	CPY	£\$28	; Est-on à 40 ?
0340:	D0 CF	42	BNE	BCLY	; Non -> on tourne
0342:	E6 08	43	INC	LIGNE	; Oui -> + une ligne
0344:	A5 08	44	LDA	LIGNE	
0346:	C9 18	45	CMP	£\$18	; 24 lignes ?
0348:	D0 C2	46	BNE	DEBUT	; Non -> on tourne
034A:	A5 1A	47	LDA	DBLE	; Est-on en "double" ?
034C:	D0 05	48	BNE	SUITE5	; Non -> terminé

SUITE PAGE 44

```

034E: A5 1B      49          LDA  NBR          ; Oui -> on écrit NBR
0350: 20 54 03   50          JSR  ECRIT
0353: 60         51  SUITE5     RTS
                    52  ***** S/P d'écriture en mémoire *****
0354: A2 00      53  ECRIT     LDX  £#0
0356: 81 06      54          STA  (BAS,X)
0358: 85 09      55          STA  SVA
035A: E6 06      56          INC  BAS
035C: D0 02      57          BNE  ECRIT1
035E: E6 07      58          INC  HAUT
0360: 60         59  ECRIT1     RTS
                    60  ***** S/P de transfert *****
0361: A0 00      61          LDY  £#0
0363: 20 2C FE   62          JSR  MOVE
0366: 60         63          RTS

```



BSAVE ECR.IN,A\$300,L\$67

Table des symboles ordre alphabetique

-BAS.....\$0006	-BASCALC...\$FBC1	-BASL.....\$0028	-BCLY.....\$0311
-DBLE.....\$001A	-DEBUT.....\$030C	-ECRIT.....\$0354	-ECRIT1....\$0360
-HAUT.....\$0007	-LIGNE.....\$0008	-MOVE.....\$FE2C	-NBR.....\$001B
-SUITE1....\$0320	-SUITE2....\$0332	-SUITE3....\$033A	-SUITE4....\$033D
-SUITE5....\$0353	-SVA.....\$0009		

## ● ECR.OUT

Assemblage par ProCODE

```

0 DEP      EQU  $6
1 ARR      EQU  $8
2 DBLE     EQU  $1A      ; 0 si un caractère se reproduit
3 SVA      EQU  $1B      ; Mémoire A
4 LIGNE    EQU  $1C      ; Ligne courante
5 NBR      EQU  $1E      ; Nbr de caractères identiques
6 SCROLL   EQU  $FC70    ; Monte l'écran d'une ligne
7          ORG  $300
0300: A9 07      8          LDA  £#7      ; Adresse...
0302: 85 09      9          STA  ARR+1    ; de la
0304: A9 D0     10         LDA  £#D0     ; ligne
0306: 85 08     11         STA  ARR      ; numéro 24
0308: A9 00     12         LDA  £#0
030A: 85 1C     13         STA  LIGNE   ; *
030C: 85 1B     14         STA  SVA      ; * Initialisations
030E: A9 01     15         LDA  £#1      ; *
0310: 85 1A     16         STA  DBLE    ; *
0312: 20 70 FC  17  DEBUT   JSR  SCROLL   ; On remonte d'une ligne
0315: A0 00     18         LDY  £#0
0317: A6 1A     19  BCLY    LDX  DBLE    ; Est-ce en double ?
0319: D0 0D     20         BNE  SUITE2   ; Non, alors un autre caractère
031B: C6 1E     21         DEC  NBR      ; Oui, alors un de moins
031D: D0 04     22         BNE  SUITE1   ; Si 0 -> fin de double

```



```

031F: A2 01      23          LDX  £#1          ; On "marque"
0321: 86 1A      24          STX  DBLE         ; DBLE
0323: A5 1B      25  SUITE1     LDA  SVA
0325: 4C 3E 03   26          JMP  SUITE4
0328: 20 50 03   27  SUITE2     JSR  LIT          ; On lit le caractère
032B: C5 1B      28          CMP  SVA         ; Le meme caractère ?
032D: D0 0F      29          BNE  SUITE4     ; Non, alors on tourne
032F: 20 50 03   30          JSR  LIT          ; Oui, alors on va chercher NBR
0332: C9 00      31          CMP  £#0        ; Est-il égal à zéro ?
0334: F0 06      32          BEQ  SUITE3     ; Oui, alors pas de troisième
0336: 85 1E      33          STA  NBR        ; Non, alors on le mémorise
0338: A9 00      34          LDA  £#0        ; On "marque"
033A: 95 1A      35          STA  DBLE       ; DBLE
033C: A5 1B      36  SUITE3     LDA  SVA
033E: 91 08      37  SUITE4     STA  (ARR),Y    ; On écrit le caractère
0340: 85 1B      38          STA  SVA        ; On mémorise le caractère
                39  *          * DETECTER LES "DOUBLES"
0342: C8         40          INY             ; Un caractère de plus
0343: C0 28      41          CPY  £#28       ; Est-on à 40 ?
0345: D0 D0      42          BNE  BCLY       ; Non, alors on remet ça!
0347: E6 1C      43          INC  LIGNE      ; Oui, alors une ligne...
0349: A5 1C      44          LDA  LIGNE      ; ...de plus
034B: C9 18      45          CMP  £#18       ; Est-on à 24 ?
034D: D0 C3      46          BNE  DEBUT      ; Non, alors on repart
034F: 60         47          RTS
                48  ***** S/P DE LECTURE DE CARACTERE *****
0350: A2 00      49  LIT        LDX  £#0
0352: A1 06      50          LDA  (DEP,X)    ; On lit le caractère
0354: E6 06      51          INC  DEP        ; On avance en mémoire
0356: D0 02      52          BNE  LIT1       ; Si on a 0, on incrémente...
0358: E6 07      53          INC  DEP+1      ; ...aussi la partie haute
035A: 60         54  LIT1      RTS

```

**BSAVE ECR.OUT,A\$300,L\$5B**

Table des symboles ordre alphabétique

-ARR.....\$0008	-BCLY.....\$0317	-DBLE.....\$001A	-DEBUT.....\$0312
-DEP.....\$0006	-LIGNE.....\$001C	-LIT.....\$0350	-LIT1.....\$035A
-NBR.....\$001E	-SCROLL....\$FC70	-SUITE1....\$0323	-SUITE2....\$0328
-SUITE3....\$033C	-SUITE4....\$033E	-SVA.....\$001B	

## ● ECR.LIRE

```

10  ONERR  GOTO 40
20  TEXT : HOME :D$ = CHR$ (4)
30  PRINT D$"BLOAD ECR.OUT"
40  HOME : VTAB 3: HTAB 5: INVERSE : PRINT "LECTURE D'UNE TABLE DE MASQUES":
    NORMAL
50  VTAB 10: HTAB 10: INPUT "NUMERO DU MASQUE ? ";A
55  AD = 8192: REM  CORRESPOND A  $2000, A DETERMINER SELON  L'ADRESSE DE C
    HARGEMENT
60  POKE 6, PEEK (AD - 2 + 2 * A): POKE 7, PEEK (AD - 1 + 2 * A): CALL 768
70  IF  PEEK ( - 16384) < 128 THEN 70
80  GET A$: GOTO 40

```

# TRACER UN CADRE

## dans une page graphique

**B**ASIC ou LANGAGE MACHINE ? La seconde méthode est un peu plus rapide (notamment pour effacer l'écran), mais le programme en Basic est tellement simple... que l'on hésite à aller plus loin.

Il tient en effet en 3 lignes : 40, 50 et 60 et donne un bon résultat.

Gageons qu'il se trouvera, parmi nos Lectrices et Lecteurs, quelque génie capable de rendre la routine en langage machine plus performante.

### PROGRAMME DE DÉMONSTRATION

```

10 TEXT : PRINT CHR$(21); CHR$(12); HOME : GOSUB 160
20 VTAB 12: PRINT "PROGRAMME EN BASIC (70 OCTETS)": PRINT : PRINT : LIST 4
   0,50
30 GOSUB 150
40 HGR : POKE 49234,0: HCOLOR= 2: HPLLOT 0,0: CALL - 3082
50 HCOLOR= 0
60 FOR I = 10 TO 180: HPLLOT 10,I TO 268,I: NEXT
70 GOSUB 150: HOME : TEXT
80 VTAB 12: PRINT "PROGRAMME EN ASSEMBLEUR (57 OCTETS)": GOSUB 150
90 CALL 768: GOSUB 150
100 POKE 49235,0: PRINT : VTAB 22: PRINT "(E)NCORE M(ENU DE DISQUETTE T(E
   RMINE)": GOSUB 150
110 R = PEEK ( - 16384) - 128: IF R = 69 THEN 10
120 IF R = 77 THEN PRINT : PRINT CHR$(4)"RUN MENU"
130 IF R < > 84 THEN GOSUB 150: GOTO 110
140 TEXT : HOME : END
150 CALL - 198: POKE - 16368,0: WAIT - 16384,128,127: POKE - 16384,0: C
   ALL - 198: RETURN
160 POKE 768,169: POKE 769,32: POKE 770,133: POKE 771,230: POKE 772,173: PO
   KE 773,84: POKE 774,192: POKE 775,173: POKE 776,82: POKE 777,192: POKE 7
   78,173: POKE 779,87: POKE 780,192: POKE 781,173: POKE 782,80
170 POKE 783,192: POKE 784,169: POKE 785,85: POKE 786,32: POKE 787,244: POK
   E 788,243: POKE 789,169: POKE 790,10: POKE 791,133: POKE 792,6: POKE 793
   ,162: POKE 794,0: POKE 795,32: POKE 796,236: POKE 797,246: POKE 798,162
180 POKE 799,10: POKE 800,160: POKE 801,0: POKE 802,165: POKE 803,6: POKE 8
   04,32: POKE 805,17: POKE 806,244: POKE 807,169: POKE 808,12: POKE 809,16
   2: POKE 810,1: POKE 811,164: POKE 812,6: POKE 813,32: POKE 814,58
190 POKE 815,245: POKE 816,230: POKE 817,6: POKE 818,165: POKE 819,6: POKE
   820,201: POKE 821,181: POKE 822,208: POKE 823,230: POKE 824,96
200 RETURN

```





# ROUTINE EN LANGAGE MACHINE (lignes de pokes)

## EFFACER ET COLORER HGR

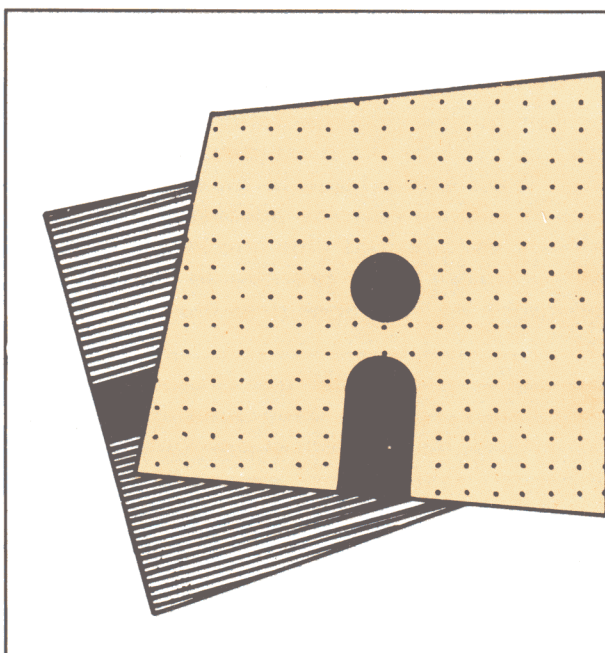
0300-	A9 20	LDA	£\$20	] \$E6 doit contenir \$20 quand on trace dans la page 1 et \$40 pour la page 2 ~ octet de poids fort de \$2000 ou \$4000 On commute la page 1... sans ses 4 lignes de texte... Puis le mode haute résolution... et le mode graphique. Le code du rouge est chargé BKGND donne à la page courante la couleur qui est dans A.
0302-	85 E6	STA	\$E6	
0304-	AD 54 C0	LDA	\$C054	
0307-	AD 52 C0	LDA	\$C052	
030A-	AD 57 C0	LDA	\$C057	
030D-	AD 50 C0	LDA	\$C050	
0310-	A9 55	LDA	£\$55	
0312-	20 F4 F3	JSR	\$F3F4	

## NE LAISSER QUE LE CADRE

0315-	A9 0A	LDA	£\$0A	\$0A est mis dans l'accumulateur et stocké à l'adresse \$6 (page zéro).
0317-	85 06	STA	\$06	Registre X à 0
0319-	A2 00	LDX	£\$00	Couleur fixée à 0
031B-	20 EC F6	JSR	\$F6EC	Partie basse abscisse
031E-	A2 0A	LDX	£\$0A	Partie haute abscisse
0320-	A0 00	LDY	£\$00	Ordonnée (pointeur \$6)
0322-	A5 06	LDA	\$06	HPOSN positionne le curseur sans tracer
0324-	20 11 F4	JSR	\$F411	Partie basse abscisse
0327-	A9 0C	LDA	£\$0C	Partie haute abscisse
0329-	A2 01	LDX	£\$01	Ordonnée (pointeur \$6)
032B-	A4 06	LDY	\$06	HLIN trace la ligne ainsi délimitée
032D-	20 3A F5	JSR	\$F53A	Plus 1 pour le pointeur...
0330-	E6 06	INC	\$06	... que l'on charge en mémoire
0332-	A5 06	LDA	\$06	Est-il égal à \$B5 ?
0334-	C9 B5	CMP	£\$B5	Si non, encore une ligne !
0336-	D0 E6	BNE	\$031E	Retour au Basic.
0338-	60	RTS		

*Début de la ligne*

*Fin de la ligne*



Offrez-vous les disquettes de programmes de

## TREMPLIN MICRO

Au choix : DOS 3.3. ou ProDOS.

(ProDOS ne fonctionne qu'après avoir lancé le système avec une disquette MASTER).

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

# Votre bibliothèque INFORMATIQUE

par Clément RENARD

## ● *La Micro en 100 questions (Cedic/Nathan)*

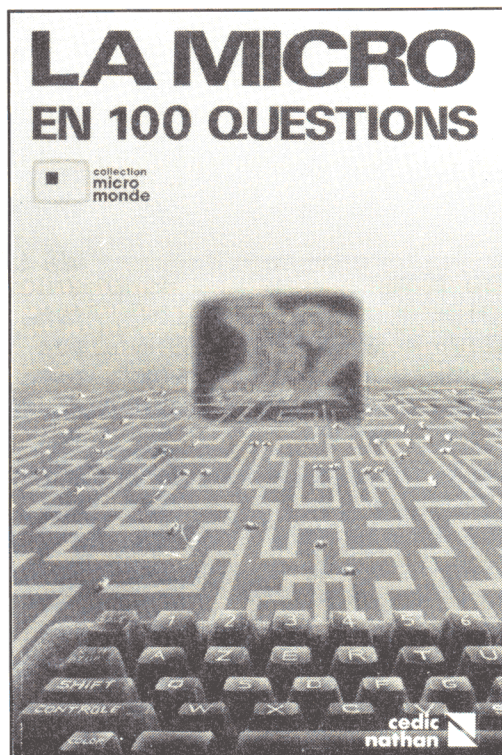
J'ignore si le guide réalisé par Bruno de Latour saura répondre aux questions que vous vous posez encore sur la micro-informatique, sujet d'actualité s'il en est.

Sur Apple, sans doute n'apprendrez-vous rien que vous ne sachiez déjà. Et d'ailleurs, que vous importe le passé de Stephen G. Wozniak ou celui de Steve Jobs... puisque le premier s'est remercié lui-même, tandis que le second se voyait détrôné par John Sculley ? Vous vous en fichez royalement, et davantage encore du fait que, en juin 1983 (quel jour... et à quelle heure ?), Madame Thatcher autorisait Mister Sinclair à devenir Sir Clive !

C'est pourtant cela, *La Micro en 100 questions*, mais aussi la réponse à "comment mesure-t-on la mémoire ?", "Comment reconnaître un bon clavier ?" ou "Qu'est-ce qu'un logiciel intégré ?".

Le choix des questions pourra vous paraître agaçant, orienté, original ou judicieux. Il est certain que la réponse à certaines d'entre elles vous laissera sur votre soif de savoir, tandis que d'autres apaiseront votre insatiable curiosité.

Personnellement, pour le même prix, j'aurais souhaité 500 questions... et autant de réponses, mais peut-être suis-je trop gourmand ?



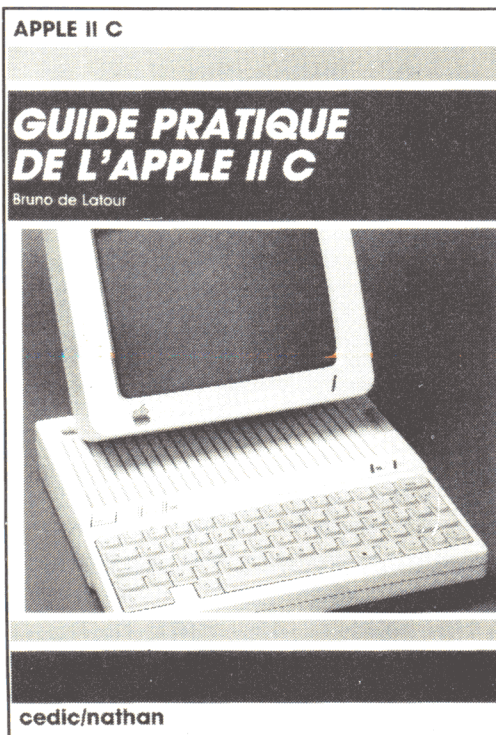
## ● *Guide pratique de l'Apple IIc (Cedic/Nathan)*

Si aucune documentation n'était fournie avec l'Apple IIc, il est sûr que ce recueil (compilation des Guide Apple, mais avec le style — clair et direct — de Bruno de Latour) se révélerait utile, voire indispensable.

Toutefois, sachant qu'une documentation, fut-elle aussi bien présentée que celle du IIc, n'est jamais lue avec une réelle attention, il est évident que le *Guide pratique de l'Apple IIc* pourra séduire certains possesseurs de cette machine. L'ennui, c'est que, pas plus que la doc Apple, il ne parle pas ! Il faudra donc prendre la peine (et le temps) d'en parcourir les 139 pages (très illustrées et non moins lisibles).

On saura alors copier et formater une disquette, copier les fichiers, etc. sous ProDOS, bien entendu. On lira aussi, non sans intérêt, les différences existant entre ProDOS et DOS 3.3.

Bref : une introduction (qui n'a d'ailleurs pas la prétention d'être autre chose) au portable d'Apple, à ses périphériques et à ses logiciels.



C.R





# DESSINE-MOI DES ÉTOILES

**C** E programme n'a pas la prétention de tracer toutes les étoiles possibles et imaginables, mais il vous présente une méthode à la fois simple et efficace. Essayez plutôt !

**POKE 230,32** permet de tracer dans la page 1

**HPLLOT 0,0 : CALL — 3082** efface la page 1 (si **HCOLOR = 0**, bien sûr)

**HGR** initialise la page 1 (déjà "nettoyée" par la ligne précédente)

**POKE 49234,0** supprime les 4 lignes de texte de la page 1

On utilise ici la même méthode que pour le tracé du cercle (voir *Tremplin Micro n°3*, page 47).

**POKE 49235,0** rétablit les 4 lignes de texte.

**ESCAPE** permet de visualiser la totalité de la page 1

Vous comprendrez facilement les autres commandes en tapant votre programme, puis en l'exécutant.

Peut-être constituera-t-il la base d'un jeu ?

```

10 TEXT : PRINT CHR$(12); CHR$(21) : HOME
20 HCOLOR = 0 : HPLLOT 0,0 : POKE 230,32 : CALL — 3082
30 HGR
40 PRINT : VTAB 22 : PRINT : CALL — 868 : INPUT "RAYON DE L'ÉTOILE = " : R$ : IF
  VAL(R$) > 96 AND VAL(R$) < 12 THEN R = VAL(R$)
50 IF R = 0 THEN 310
60 L$ = "CENTRE DE L'★ (" + STR$(0 + R) + "—" + STR$(279 — R) + "," + STR$(0
  + R) + "—" + STR$(191 — R) + ") = "
70 VTAB 21 : CALL — 958 : PRINT L$ : INPUT "" : XC$,YC$ : REM CTRL — 7 ENTRE ""
80 IF XC$ <> "" THEN XC = VAL(XC$)
90 IF XC = 0 THEN XC = R
100 IF YC$ <> "" THEN YC = VAL(YC$)
110 IF YC = 0 THEN YC = R
120 IF XC < 0 + R OR XC > 279 — R OR YC < 0 + R OR YC > 191 — R THEN CALL — 198
  : GOTO 60
130 PRINT : VTAB 21 : PRINT L$:XC$,"";YC$ : CALL — 868
140 POKE 49234,0
150 S = 5 : PI = 3.1415926 : PB = PI * 2 : CE = PB/S
160 HCOLOR = 3 : V = 0
  
```

## COORDONNÉES DES POINTES

```

170 FOR A = 0 TO PB STEP CE
180 B = A — .313
190 V = V + 1 : H(V) = R * COS(B) + XC : V(V) = R * SIN(B) + YC
200 NEXT
  
```

## TRACÉ PROPREMENT DIT

```

210 HPLLOT H(1),V(1) TO H(3),V(3) TO H(5),V(5) TO H(2),V(2) TO H(4),V(4) TO H(1),V(1)

220 POKE 49235,0
230 VTAB 22 : PRINT : CALL — 958 : CALL — 198 : PRINT "(E) EFFACER (C) OMPLETER
  (D) OUBLE (A) UTRE " : GET R$ : POKE — 16368,0
240 IF R$ = CHR$(27) THEN POKE 49234,0 : GOSUB 300 : GOTO 220
250 IF R$ = "E" THEN CALL — 3086 : REM EFFACER
260 IF R$ = "C" THEN 40
270 IF R$ = "D" THEN XC = XC + 3 : YC = YC + 2 : GOTO 120
280 IF R$ = "A" THEN 310
290 GOTO 230
300 CALL — 198 : POKE — 16368,0 : WAIT — 16384,128,127 : POKE — 16368,0 : RETURN
310 VTAB 22 : PRINT : CALL — 958 : PRINT "(F) IN (M) ENU DISQUETTE (S) AVE PAGE1 —>
  " : GET R$ : POKE — 16368,0
320 IF R$ = CHR$(27) THEN POKE 49234,0 : GOSUB 300 : GOTO 220
330 IF R$ = "M" THEN PRINT : PRINT CHR$(4) "RUN MENU"
340 IF R$ = "S" THEN PRINT : PRINT CHR$(4) "BSAVE ETO, A$2000, L$2000"
350 HOME : IF R$ = "F" THEN TEXT : END
360 GOTO 230
  
```

# ÉTOILE ET CADRE

LANGAGE MACHINE

ES quelques lignes qui suivent vous montreront, mieux que de longues explications, comment dessiner (instantanément) une étoile dont on connaît les paramètres, et comment tracer un cadre. Rappelons que TAX passe la valeur de l'Accumulateur dans le registre X et que TAY joue le même rôle, mais au profit du registre Y. La valeur de A n'est pas modifiée. A noter que le contenu de A n'est pas davantage modifié par l'exécution de HLIN et HFIND.

## ÉTOILE

300 :	20	E2	F3	JSR	\$F3E2
303 :	A2	03		LDX	£\$03
305 :	20	EC	F6	JSR	\$F6EC
308 :	A2	AF		LDX	£\$AF
30A :	A0	00		LDY	£\$00
30C :	A9	3E		LDA	£\$3E
30E :	20	57	F4	JSR	\$F457
311 :	A9	25		LDA	£\$25
313 :	A2	00		LDX	£\$00
315 :	A0	A2		LDY	£\$A2
317 :	20	3B	03	JSR	\$033B
31A :	A9	5A		LDA	£\$5A
31C :	A2	00		LDX	£\$00
31E :	A0	04		LDY	£\$04
320 :	20	3B	03	JSR	\$033B
323 :	A9	8E		LDA	£\$8E
325 :	A2	00		LDX	£\$00
327 :	A0	A2		LDY	£\$A2
329 :	20	3B	03	JSR	\$033B
32C :	A9	04		LDA	£\$04
32E :	A2	00		LDX	£\$00
330 :	A0	3E		LDY	£\$3E
332 :	20	3B	03	JSR	\$033B
335 :	A9	AF		LDA	£\$AF
337 :	A2	00		LDX	£\$00
339 :	A0	3E		LDY	£\$3E
33B :	20	3A	F5	JSR	\$F53A
33E :	20	CB	F5	JSR	\$F5CB
341 :	60			RTS	

## CADRE

342 :	A9	00		LDA	£\$00
344 :	AA			TAX	
345 :	A8			TAY	
346 :	20	57	F4	JSR	\$F457
349 :	A9	17		LDA	£\$17
34B :	A2	01		LDX	£\$01
34D :	A0	00		LDY	£\$00
34F :	20	3B	03	JSR	\$033B
352 :	A2	01		LDX	£\$01
354 :	A0	BF		LDY	£\$BF
356 :	20	3B	03	JSR	\$033B
359 :	A9	00		LDA	£\$00
35B :	AA			TAX	
35C :	A0	BF		LDY	£\$BF
35E :	20	3B	03	JSR	\$033B
361 :	AA			TAX	
362 :	A8			TAY	
363 :	4C	3B	03	JMP	\$033B

Fait passer en mode HGR  
 Pour fixer la couleur (blanc)  
 SETHCOL met la couleur à la valeur X  
 Partie basse abscisse  
 Partie haute abscisse  
 Ordonnée du point  
**HPLLOT 175,62**

HPLLOT positionne le curseur et trace le point  
 Partie basse abscisse  
 Partie haute abscisse  
 Ordonnée du point  
**HPLLOT 37,162**

Saut pour le tracé  
 Partie basse abscisse  
 Partie haute abscisse  
 Ordonnée du point  
**HPLLOT 90,6**

Saut pour le tracé  
 Partie basse abscisse  
 Partie haute abscisse  
 Ordonnée du point  
**HPLLOT 142,162**

Saut pour le tracé  
 Partie basse abscisse  
 Partie haute abscisse  
 Ordonnée du point  
**HPLLOT 4,62**

Saut pour le tracé  
 Partie basse abscisse  
 Partie haute abscisse  
 Ordonnée du point  
**HPLLOT 175,62**

HLIN trace la ligne  
 HFIND réinitialise  
**TRACÉ HPLLOT**

## PROGRAMME DEMO PAGE SUIVANTE

Partie basse abscisse  
 Partie haute abscisse  
 Ordonnée du point  
**HPLLOT 0,0**

HPLLOT positionne le curseur et trace le point  
 Partie haute abscisse  
 Partie basse abscisse  
 Ordonnée du point  
**HPLLOT 279,0**  
**279 = \$117**

Saut pour le tracé  
 Partie basse abscisse  
 Ordonnée du point  
**HPLLOT 279,191**  
**(A n'a pas varié)**

Saut pour le tracé  
 Partie haute abscisse  
 Partie basse abscisse  
 Ordonnée du point  
**HPLLOT 0,191**

Saut pour le tracé  
 Partie basse abscisse  
 Partie haute abscisse  
 Dernier saut pour le tracé  
**HPLLOT 0,0**  
**(A n'a pas varié)**



## PROGRAMME DE DÉMONSTRATION

(ÉTOILE ET CADRE)

```
10 TEXT : PRINT CHR$(12); CHR$(21); HOME
20 HCOLOR= 0: HPLLOT 0,0: POKE 230,32: CALL - 3082
30 PRINT "DEMONSTRATION AVEC ROUTINE LM
-----"
40 PRINT : PRINT : PRINT "67 OCTETS (CADRE) & 35 OCTETS (CADRE)"
50 PRINT CHR$(4)"BLOAD ETOILE.LM"
60 VTAB 22: PRINT "ENFONCEZ UNE TOUCHE ";: GET R$
70 CALL 768
80 POKE 49234,0: REM SUPPRIME LES 4 LIGNES DE TEXTE
90 CALL - 198: GET R$
100 POKE 49235,0: REM RETABLIT LE TEXTE
110 HOME : VTAB 22: PRINT "MEME ROUTINE AVEC CADRE LANGAGE MACHINE"
120 CALL 768: POKE 49234,0: CALL 834:S = S + 1: IF S = 1 THEN FOR I = 1 T
O 2000: NEXT
130 IF S < 12 THEN 120
140 POKE 49235,0
150 HOME : VTAB 22: PRINT "(M)ENU DE DISQUETTE (E)TOILE BASIC (F)IN"
160 POKE - 16368,0: WAIT - 16384,128,127: POKE - 16368,0
170 R = PEEK ( - 16384): IF R = 77 THEN PRINT : PRINT CHR$(4)"RUN MENU"
180 IF R = 69 THEN PRINT : PRINT CHR$(4)"RUN ETOILE"
190 IF R = 70 THEN TEXT : HOME : END
200 GOTO 160
```

INITIATION

DOS

ProDOS

Langage machine

## DÉPLACEMENT et COMPARAISON DES ZONES DE MÉMOIRE

### • Comment déplacer la zone (\$300.350 en \$800.850) ?

**Remarque importante** : on ne déplace pas une zone de la mémoire, mais on la RECOPIE. De plus, l'adresse du début de destination ne doit pas être comprise entre les deux adresses (début et fin) de la zone à recopier..

Passez d'abord en mode moniteur par un CALL—151 (suivi de RETURN). Ensuite, tapez :  
\* 800 < 300.350M (puis RETURN) M = MOVE (routine qui se charge de tout).

### • Comment vérifier que les zones \$300.350 et \$800.850 sont identiques ?

Même procédure que précédemment, mais en faisant appel à la routine VERIFY :

\* 800 < 300.350V (puis RETURN)

Si la comparaison est exacte, l'astérisque va s'afficher à la ligne suivante. Dans le cas contraire, VERIFY va afficher (n'est-ce pas merveilleux ?) :

1. L'adresse de la zone-source où existe une différence
2. La valeur hexadécimale écrite à cette adresse
3. La valeur hexadécimale lue à l'adresse correspondante, dans la zone à vérifier (zone de destination).

# RAM.COPIE

Ou comment transférer vos programmes-disquette sur /RAM.

**Q** UAND vous utilisez ProDOS (*Professional Disk Operating System*), vous disposez d'un disque virtuel de 61 K (mémoire de la carte 80 colonnes). Si vous demandez à votre utilitaire Système de vous afficher la liste des volumes, il vous signalera que /RAM est dans le lecteur 2, slot (ou port) 3.

Nous avons déjà vu qu'il est très facile d'accéder à ce volume, soit par PREFIX/RAM, soit par CATALOG/RAM. Si l'on a un programme en mémoire, il est non moins facile de le sauvegarder sur /RAM par SAVE/RAM/NOMPROGRAMME.

Bien entendu, il est possible de créer un ou plusieurs sous-catalogues (DIR) sur /RAM. C'est ce que vous permet, entre autres, notre RAM.COPIE.

**Limitation :** RAM.COPIE ne transfère que des fichiers précédés de la mention BAS. Son utilité est donc restreinte, mais cette routine a le mérite de montrer comment lire un DIRECTORY, et aussi comment tirer profit de commandes EXEC.

Pour transférer n'importe quel fichier sur /RAM, il est évidemment recommandé d'utiliser l'utilitaire de copie fourni avec ProDOS.

Dans un cas comme dans l'autre, on n'oubliera pas que le contenu du disque virtuel serait immédiatement perdu si une intempestive coupure de courant survenait...

## VOTRE PROGRAMME

```

10 TEXT : HOME :D$ = CHR$ (4): PRINT D$"PR$3"
20 DIM PR$(50)
30 FOR I = 1 TO 20:T$ = T$ + "----": NEXT
40 PRINT T$:TI$ = "COPIE D'UNE DISQUETTE SUR LE DISQUE
   VIRTUEL /RAM"
50 VTAB 3: PRINT TI$: PRINT T$
60 TR$ = "/RAM/PROG/":ST$ = "STARTUP.RAM"
70 BD = 118: REM NOMBRE DE BLOCS DISPONIBLES
80 PR = 50: REM MAXIMUM DE PROGRAMMES SUR /RAM
90 :
100 REM *****
110 REM **      PREFIX DE LA DISQUETTE      **
120 REM *****
130 :
140 PRINT D$"PREFIX"
150 INPUT P$: GOSUB 160: GOTO 220
160 VTAB 3: POKE 1403,80 - LEN (P$): INVERSE : PRINT
   P$: NORMAL : POKE 34,5: RETURN
170 :
180 REM *****
190 REM **      CATALOGUE DE DISQUETTE      **
200 REM *****
210 :
220 PRINT D$"CATALOG"
230 :
```

Vous pourrez certes copier 50 fichiers Basic, mais dans la limite des 118 blocs disponibles.

← Pourquoi **GOSUB 160** ?  
 Simplement parce que la ligne 160 est réutilisée un peu plus loin (290), ce qui évite une répétition.



Quand on parle de disquette entière, se souvenir que cet utilitaire ne copie que des fichiers Basic précédés de la mention **BAS**.

- 
- L1\$** = Titre du catalogue.
  - L2\$** = Ligne des titres du catalogue.
  - L3\$** = Ligne blanche.
  - L4\$** = Fichier.
- 

La variable **PR\$(P)** va contenir les titres des fichiers à effacer de **/RAM** (P étant leur nombre).

S'il existe un **STARTUP.RAM**, on en prend note.

```

240 REM *****
250 REM ** DISQUETTE ENTIERE OU DIR **
260 REM *****
270 :
280 VTAB 24: POKE 1403,0: INPUT "PRECISER LE TITRE DU
    SOUS-CATALOG (DIR) EVENTUEL ";TI$
290 IF TI$ < > "" THEN P$ = P$ + TI$ + "/": GOSUB 160
300 :
310 REM *****
320 REM ** CONTROLE DE /RAM **
330 REM *****
340 :
350 HOME
360 ONERR GOTO 450
370 PRINT D$"OPEN"TR$,TDIR": PRINT D$"READ"TR$: INPUT
    L1$: INPUT L2$: INPUT L3$
380 INVERSE : PRINT L1$: NORMAL
390 PRINT L2$: PRINT L3$
400 INPUT L4$: IF L4$ = "" THEN 420
410 P = P + 1:PR$(P) = MID$ (L4$,2,15): PRINT PR$(P):
    GOTO 400
420 PRINT D$"CLOSE"TR$
430 FOR I = 1 TO P: PRINT D$"DELETE"TR$;PR$(I): NEXT
440 GOTO 520
450 POKE 216,0: CALL - 3288
460 PRINT D$"CREATE"TR$: PRINT D$"CATALOG"TR$
470 :
480 REM *****
490 REM ** LECTURE CATALOGUE A COPIER **
500 REM *****
510 :
520 P = 0: REM NOMBRE DE PROGRAMMES A COPIER
530 BL = 0: REM NOMBRE DE BLOCS
535 PRINT D$"OPEN"P$,TDIR"
540 PRINT D$"READ"P$: INPUT L1$: INPUT L2$: INPUT L3$
550 INPUT L4$: IF L4$ = "" THEN 650
560 IF MID$ (L4$,18,3) < > "BAS" THEN 550
570 P = P + 1: IF P > PR THEN 650
580 BL = BL + VAL ( MID$ (L4$,23,6)): IF BL > BD THEN
    650
590 TI$ = MID$ (L4$,2,15)
600 FOR Z = 15 TO 1 STEP - 1: IF MID$ (TI$,Z,1) < >
    " " THEN 620
610 NEXT
620 PR$(P) = LEFT$ (TI$,Z)
630 IF PR$(P) = ST$ THEN ST = 1
640 GOTO 550
650 PRINT D$"CLOSE"P$
660 HOME : VTAB 13: PRINT "Il y a "; INVERSE : PRINT
    P;: NORMAL : PRINT " programmes à copier sur "TR$
670 :

```

SUITE ET FIN PAGE 54

## RAM.COPIE

(Suite et fin)

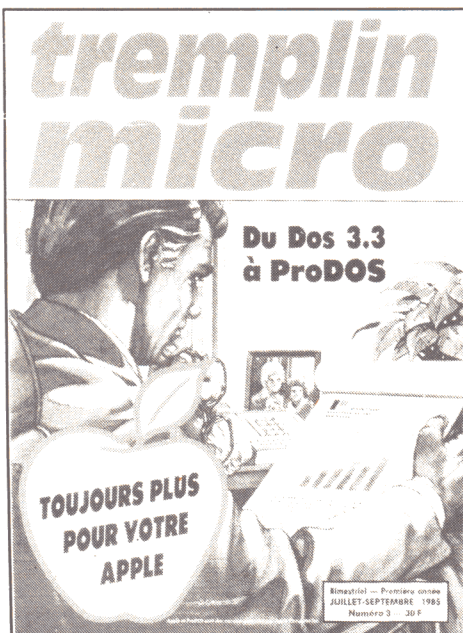
```
680 REM *****
690 REM **  CREATION DU FICHIER EXEC  **
700 REM *****
710 :
720 PRINT : PRINT "Création du fichier EXEC"
730 E$ = "EX"
740 PRINT D$"OPEN"E$: PRINT D$"CLOSE"E$: PRINT D$"DELE
TE"E$: PRINT D$"OPEN"E$: PRINT D$"WRITE"E$
750 FOR I = 1 TO P: PRINT "LOAD"P$;PR$(I): PRINT "SAVE
";TR$;PR$(I): NEXT
760 PRINT "HOME": PRINT "CATALOG/RAM/PROG"
770 IF ST = 1 THEN PRINT "--TR$;ST$
780 PRINT
790 PRINT D$"CLOSE"
800 :
810 REM *****
820 REM ** COPIE DES PROGRAMMES SUR /RAM **
830 REM *****
840 :
850 PRINT : PRINT "La copie des programmes est en cour
s": PRINT
860 PRINT D$"EXEC"E$
```

S'il existe un **STAR-TUP.RAM** (à vous de le créer !), il est lancé par l'instruction de ligne 770.

← **EXEC** se charge de tout...

## ERRATA

### TREMPIN MICRO N°3



Sûr : les programmes sont soigneusement revus et essayés (sur Apple IIe et Apple IIc, aussi bien avec DOS que ProDOS). Certain : notre distinguée correctrice fait tout ce qu'elle peut pour ne rien laisser passer... mais il y a les impondérables ! Bref, veuillez corriger les oublis ou erreurs qui suivent :

**ECRANS.LM** (page 6) : A la fin de la ligne 38D, supprimer RTS.

**ECRANS.BAS** (page 9) : Le GOSUB 1580 de la ligne 1090 est superflu (double emploi avec celui de la ligne 1100). Ces deux corrections ne troublent pas le fonctionnement du programme. Par contre, si vous possédez un Apple IIc, modifiez les lignes 180 et 190 de la page 11 (en bas et à droite).  
180 IF H < 0... et 190 IF H > 73...

**CRYPTOTEXTE** (page 14) : Les valeurs ASCII indiquées à partir de 31D sont inexactes (C1 C5, etc.). Lire : 4F 41 55 45 49 41 45 49 4F et 55. Ces codes sont poqués à partir du Basic et cette modification est donc sans influence sur le fonctionnement du programme.

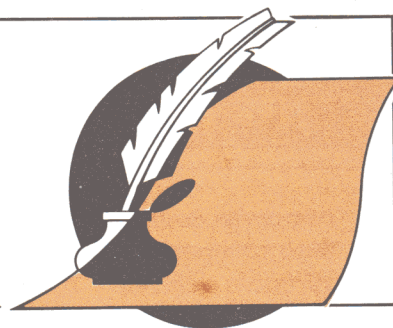
**BOUTABOUT** (page 30) : En 319, il faut écrire E9 03 (au lieu de E9 02). De plus avant de réunir deux sections d'un programme, supprimer éventuellement une ligne LOMEM ou HIMEM.

**ECRAN.LM** (page 37) : Il manque un 60 à la fin de la dernière ligne. Le résultat doit être 6796 et non 6955, nombre inventé par un mauvais plaisant.

————— Nous avons besoin de vacances ! —————



# Vous avez écrit à TREMPLIN MICRO



• *Lecteur assidu de votre revue **TREMP LIN MICRO**, je me passionne pour l'informatique, comme beaucoup de jeunes actuellement. Je possède un **Apple IIe** et je viens de terminer un programme de dessin en basic. J'ai toutefois quelques questions à vous poser : est-il possible de sauvegarder un dessin sur disquette, et si oui comment ? Peut-on sortir celui-ci sur une imprimante **IMAGEWRITER** ?*

*D'autre part, pourriez-vous m'envoyer tous les **PEEKs**, **POKEs** et **CHR\$** ainsi que leur signification, car ils ne sont répertoriés dans aucun de mes manuels et j'en aurais grandement besoin pour mes futurs programmes. Je vous en remercie d'avance... Veuillez agréer, cher Monsieur, l'expression de mes sentiments distingués.*

Pierre L. (59810 LESQUIN)

TM

Je suis surpris de votre première question. Comment pouvez-vous programmer en Basic sans savoir sauvegarder un dessin sur disquette ?

Il suffit de taper, en direct, un **BSAVE IMAGE, A\$2000, L\$1FF8** pour transférer votre dessin sur disquette (33 secteurs, sous le titre **IMAGE**, bien sûr). Pour la page 2, remplacez **A\$2000 (8192)** par **A\$4000 (16384)**.

Nous avons publié, dans le courrier du n°3 de *Tremplin Micro*, une petite routine permettant d'imprimer un dessin sur **IMAGEWRITER**, à partir de la routine **PICDMP** d'Apple (disquette **TOOL KIT**).

Si vous disposez d'une version plus récente de cet utilitaire, commencez par abrégier le programme binaire **GF**, comme indiqué ci-après, puis inspirez-vous de notre listing pour utiliser **IW1**.

```
10 PRINT CHR$(4) "BLOAD GF" : PRINT CHR$(4) "BSAVE IW1, A36864, L601"
11 END
12 REM Il suffira ensuite d'utiliser IW1
15 PRINT CHR$(4) "BLOAD IW1"
20 REM POUR X=0, impression NORMALE (blanc sur fond noir)
21 REM POUR X= 1, impression INVERSE (noir sur fond blanc)
```

22 REM POUR X=2, impression DOUBLE NORMALE

23 REM POUR X=3, impression DOUBLE INVERSE

24 X=1 : REM X=X+4 pour HGR2

30 POKE 6,1 : POKE 7,X

40 CALL 36864 : REM IMPRESSION DE LA PAGE 1

Quant aux **PEEKs** et **POKEs**, continuez de lire *Tremplin Micro* !

• *Pouvez-vous me faire connaître comment rendre compatible un **APPLE IIc** et une imprimante à matrice de points par impact, Thomson PR.90.080, fabriquée par **SEIKOSHA**.*

*Le IIc étant équipé d'un port SÉRIE, j'ai dû faire l'acquisition d'une interface parallèle et malgré cela, tout ne se passe pas comme on pourrait l'espérer.*

*En utilisant le programme "APPLE WORKS" par exemple, je suis arrivé, en configurant l'imprimante, à obtenir une frappe normale en ce qui concerne les majuscules et les minuscules. Par contre, il ne m'a été possible d'obtenir : ni les caractères accentués, ni le mode graphique, ni le mode caractères double largeur.*

*Dans un programme Basic l'impression d'une page écran ressemble à une série de "hiéroglyphes" sans aucune signification.*

*TREMP LIN MICRO, (ou l'un de ses Lecteurs), pourrait sûrement me sortir de cette impasse.*

*D'avance merci.*

J. COLATRELLA (64600 ANGLET).

TM

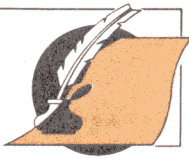
C'est un S.O.S., mais je suis presque sûr que l'un de nos Lecteurs (ou l'une de nos Lectrices) a déjà résolu ce problème...

• *Hormis le fait que votre revue m'enthousiasme de A à Z, car elle est simple à comprendre, et ne s'intéresse qu'aux Apple, j'ai deux questions à vous poser :*

1. *J'ai appris dans votre n°1 comment transformer son curseur en tiret clignotant, ou par*

SUITE DU COURRIER (PAGE 56)

## Vous avez écrit à TREMLIN MICRO



n'importe quoi d'autre par un poke 2043,X. Mais comment faire pour le transformer en un véritable curseur tîret clignotant ? (Avec DELETE pour effacer, avec insertion, etc.).

2. Où trouver une liste complète de tous les POKES faciles et utiles ?

Merci d'avance pour vos avis éclairés.

Remi MICHEL (13114 PUYLOUBIER)

**TM** Essayez le micro-programme ci-après. Vous pourrez l'appeler par BRUN CURSEUR (ou BLOAD CURSEUR, puis CALL 768). Vous constaterez que, si ça n'insère pas, ça clignote... et c'est bien un tîret ! Gageons que vous aurez à cœur de le développer pour en faire une routine plus complète !

300	:	A9	0B	85	38	A9	03	85	39
308	:	4C	EA	03	91	28	B1	28	48
310	:	A9	DF	91	28	68	91	28	2C
318	:	00	C0	10	F1	AD	00	C0	2C
320	:	10	C0	48	8A	48	A2	20	A9
328	:	08	20	A8	FC	AD	30	C0	CA
330	:	D0	F5	68	AA	68	60		

\* BSAVE CURSEUR, A\$300, L\$36

On trouve, dans la collection du PSI, éparpillés dans les ouvrages consacrés à l'Apple, bon nombre de PEEKS et de POKES. Tremplin Micro vous rappelle les adresses de quelques-uns dans chacun de ses numéros.

• Avant toute chose, bravo pour votre revue qui est une véritable mine de renseignements pratiques dans laquelle il fait bon puiser. Malheureusement, j'ai un problème et ici, à Nice, personne n'a pu me renseigner ! Voici de quoi il s'agit :

Je possède un Apple II europlus + carte langage avec une imprimante OKI data microline 82A + carte graphique. Des logiciels tels que Zoom Graphic fonctionnent parfaitement ainsi que Koala. L'impression graphique est irréprochable, par contre, d'autres tels que The Print Shop, Graphor Castor, ne sortent pas en graphique simplement parce que dans le choix de l'option pour l'imprimante l'OKI n'apparaît pas.

Ma question : pouvez-vous me fournir le listing des caractères de contrôle à inclure dans le programme ou le listing d'appel de sous-programmes en lan-

gage machine faisant le transcodage (les pokes) de manière à ce que je puisse transformer certains de mes programmes ?

Vous remerciant vivement par avance de votre collaboration, je vous souhaite longue vie et beaucoup de numéros de Tremplin Micro et surtout, si vous le pouvez, du concret, pas de pub. Sincèrement vôtre.

Jacques P. (06000 NICE)

**TM** Un autre S.O.S. (nous en avons reçu plusieurs concernant la même imprimante). Quand on n'utilise pas une configuration Apple complète, cela devient horriblement complexe. Qui connaît la réponse ?  
Merci pour notre Lecteur !

• Je dispose de l'imprimante LX-80 (EPSON) avec son interface série/parallèle pour les graphiques, mais je ne trouve pas dans T.M. et (ou) ses disquettes comment exécuter de la copie d'écran, de dessins ou autres à partir de mon APPLE IIc.

Je vous serais très reconnaissant si vous pouviez m'aider.

Claude P. (84300 CAVAILLON)...

et plusieurs autres lettres allant dans le même sens.

**TM** Nous avons posé la question à l'importateur, et celui-ci nous a promis de mettre ce problème à l'étude. Il est souhaitable que la marque EPSON fournisse un logiciel capable de permettre, sur l'Apple IIc, une bonne copie des pages graphiques. Ou alors... adieu la compatibilité !

• Je voudrais que tu m'envoies personnellement la liste des effets de chaque PEEK, POKE, CALL, CHR\$ et que tu m'expliques comment composer soi-même de la musique avec la routine FAITSON.LM (du n°3) parce que, quand j'essaie de modifier les chiffres, cela ne marche pas.

Olivier B. (06200 NICE)

**TM** La routine Faitson (parue dans notre n°3) n'avait pas la prétention de t'aider à composer de la musique. Tu vas trouver, dans le présent numéro de Tremplin Micro, des programmes qui vont te combler d'aise !

• Mes chiffres mini et maxi de clientèle sortent des bornes fixées par le programme Histogramme (n°1 de T.M.). Comment modifier ces bornes ?

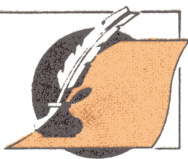
Jean A. (31390 CARBONNE)

**TM** Exemple, avec le rapport 1/20 :

780 M = A/20 190 Y = 140 — B \* (A/2)



## Vous avez écrit à TREMPLIN MICRO



• Je commencerai cette lettre par des félicitations pour cette revue attrayante qui propose des programmes intéressants (c'est rare !), fonctionnant du premier coup. Personnellement, je possède un Apple IIc et m'intéresse beaucoup au langage machine et au fonctionnement général de celui-ci. Malheureusement, la documentation, surtout française, est rare et chère quand elle existe. Et, comme vous l'avez dit dans T.M. n°3, Apple France nous ignore ou se fout de nous. C'est pourquoi c'est à vous que je m'adresse maintenant en espérant plus de compréhension. En effet, je désirerais connaître l'emplacement en mémoire des systèmes d'exploitation DOS 3.3. et ProDOS ainsi que, si possible, leurs routines (surtout l'équivalent en ProDOS des RWTS du DOS 3.3.). Je voudrais aussi pouvoir utiliser les fonctions DRAW et XDRAW à partir du langage machine. Quelles mémoires sont utilisées pour initialiser une forme ? Pourriez-vous aussi indiquer dans la revue de petites routines simples avec des commentaires (beaucoup) pour l'utilisation efficace et facile de la mémoire auxiliaire de nombreux Apple. J'aimerais aussi savoir ce qui se passe, en détail, lors de la mise sous tension de l'ordinateur et comment il charge les systèmes d'exploitation.

Avec mes remerciements.

Jean-Luc M. (54600 VILLERS-LES-NANCY).

**TM** Je publie intégralement votre lettre parce qu'elle en résume des dizaines d'autres. Il n'est évidemment pas possible de satisfaire immédiatement toutes vos curiosités, non pas pour "faire durer le plaisir" (comme le prétend l'un de nos correspondants), mais parce que les réponses exigent de nombreuses et longues explications. Le DOS 3.3. occupe l'espace-mémoire qui va de \$9600 à \$BFFF ou, si vous préférez, de 38400 à 49151 (10752 octets).

ProDOS est beaucoup plus gourmand puisqu'il utilise aussi la carte langage (MEV1).

La lecture des catalogues est relativement simple sous ProDOS (voir le programme COPIE.RAM de ce numéro).

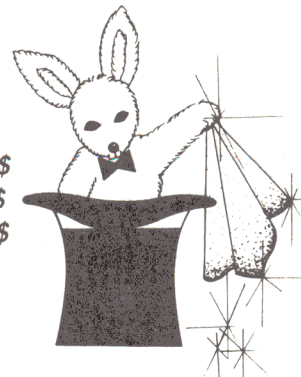
Nous aborderons progressivement toutes les questions que vous évoquez dans *Tremplin Micro*.

Nous avons déjà indiqué (n°3) comment utiliser la mémoire auxiliaire et la carte langage.

• A l'attention de Nestor :

J'ai un Apple IIe + Duodisk + monitor monochrome + Imagewriter. Depuis un an, je bosse dur. Comment sauver sur disque ce qui apparaît à l'écran résultant d'un programme du style :

```
10 A$ = "BIBI"
20 B$ = "HELLO"
30 C$ = "ALORS"
100 INPUT X
110 IF X = 1 THEN PRINT A$
120 IF X = 2 THEN PRINT B$
130 IF X = 3 THEN PRINT C$
140 IF X = 0 THEN 200
150 GOTO 100
200 PRINT "OK"
210 END
```



```
BIBI
BIBI
BIBI
HELLO
HELLO
BIBI
BIBI
OK
```

Je lance le programme par un RUN et je réponds aux INPUT par : 1/1/1 2/2 1/1/0.

Il apparaît à l'écran (enfin à peu près) ceci : voir le schéma ci-contre. Admettons que ce résultat me plaise, et que je veuille le sauvegarder sur disquette, comment faire ?

Alain B. (33880 CAMBES)

**TM** Ou bien vous désirez sauver l'écran tel qu'il est... ce qui exige un BSAVE ECRAN, A1024, L1016 (on le rappelle par BLOAD ECRAN). Bien sûr, dans le corps d'un programme, les commandes doivent être précédées par un PRINT CHR\$(4) : (exemple : PRINT CHR\$(4); "BSAVE ECRAN, A1024, L1016").

Ou bien vous désirez mémoriser vos réponses et il est indispensable de passer par une variable... et un fichier, mais c'est une autre histoire.

NESTOR

• Comment accéder directement à un mot, dans une ligne de DATA, sans être obligé de passer par RESTORE... et une lecture complète ?

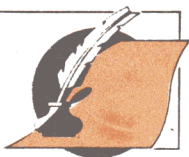
Antoine L. (16000 ANGOULÊME)

**TM** L'exemple ci-après ne prétend pas résoudre votre problème, mais vous montre comment sont mémorisées vos données, et comment il est possible d'y accéder directement... quand on en connaît le numéro d'ordre, bien sûr.

```
10 TEXT : HOME
20 DATA TOTO, TATA, LILI, DADA
25 DATA LULU, ZOZO, ERNEST, FIN
30 FOR I = 1 TO 8
```

(Suite page 58)

## Vous avez écrit à TREMPLIN MICRO



**TM** 35 PRINT I " —> " ;  
 40 AD(I) = PEEK (125) + PEEK (126) \* 256 :  
 PRINT AD(I) ; " " ;  
 50 READ A\$ : PRINT A\$ ; " — LIGNE " ;  
 60 PRINT PEEK (123) + PEEK (124) \* 256  
 70 NEXT  
 75 POKE 34,10 : PRINT  
 80 INPUT "NUMÉRO " ; N : IF N < 1 OR N > 8  
 THEN 120  
 90 POKE 125, AD(N) — INT (AD(N) / 256) \*  
 256 : POKE 126, INT (AD(N) / 256)  
 100 READ A\$ : PRINT " "N" — " A\$  
 110 GOTO 80  
 120 END

• Pourriez-vous me dire comment "chaîner" différents programmes, sur une même disquette (ProDOS) en gardant intactes, les variables ? PRINT CHR\$(4) ; "RUN PROG" n'est pas suffisant. Faut-il créer un fichier qui conserve ces données, puis qui les rappelle ? Merci de votre réponse.

Georges V. (92210 St-CLOUD)

**TM** A essayer sous ProDOS, bien sûr :

### DEMO1

```
10 TEXT : HOME
20 DIM A$(50)
30 FOR I = 1 TO 50 : A$(I) = STR$(I) : NEXT
40 PRINT CHR$(4) ; "CHAIN DEMO2, à 100"
50 HOME : PRINT "NOUS VOICI REVENUS A
  DEMO1"
```

### DEMO2

```
100 FOR I = 1 TO 50 : PRINT A$(I) , : NEXT
110 PRINT CHR$(4) ; "CHAIN DEMO1, à 50"
```

• Aucun problème pour sauver mes écrans 40 colonnes sur disquette, mais impossible de réussir la même opération avec les écrans en 80 colonnes. Avez-vous une solution ?

Anonyme

**TM** Essayez la petite routine ci-après :

```
10 D$ = CHR$(4) : PRINT D$"PRÉ3" : PRINT :
  HOME
20 FOR J = 1 TO 20 : A$ = A$ + "AAAA" :
  NEXT : FOR J = 0 TO 22
30 PRINT A$ ; NEXT
40 PRINT LEFT$(A$,79) ; : POKE 2039,193 :
  VTAB 1 : PRINT ""
```

**TM** 50 VTAB 1 : POKE 49237,0 : PRINT CHR\$(4)  
 "BSAVE ECRAN1,A1024,L1016" : REM  
 MEM AUX  
 60 VTAB 1 : POKE 49236,0 : PRINT CHR\$(4)  
 "BSAVE ECRAN2,A1024,L1016" : REM  
 MEM NORMALE  
 70 HOME : REM AVEC PRODOS, AJOUTER UN  
 POKE 48984,0 QUI EVITE LA MENTION  
 "NO BUFFERS AVAILABLE"  
 80 VTAB 1 : POKE 49237,0 : PRINT CHR\$(4)  
 "BLOAD ECRAN1" : REM MEM AUX  
 90 VTAB 1 : POKE 49236,0 : PRINT CHR\$(4)  
 "BLOAD ECRAN2" : REM MEM NORMALE

• Comment faire pour afficher des minuscules en inverse sur APPLE IIe ?

Comment changer le curseur de l'APPLE en souligné clignotant ?

Et enfin, je ne comprends pas très bien le fonctionnement de HIMEM et LOMEM ?

Jean M. (32000 AUCH)

**TM** Si vous essayez le mini-programme ci-après, vous obtiendrez, immédiatement avant le "GET R\$", un étrange T/4/ — TOTO... en mode inverse. Par contre, à la suite de la ligne 50, et à condition d'avoir une carte 80 colonnes, votre Apple chéri vous affichera bel et bien, toujours en mode inverse : Toto — TOTO. **Moralité** : il faut activer la carte 80 colonnes au début du programme, puis passer en 40 colonnes par PRINT CHR\$(17)... pour obtenir les minuscules en mode inverse.

```
10 TEXT : HOME
20 INVERSE : PRINT "Toto — TOTO" : NOR-
  MAL
30 VTAB 22 : GET R$ : PRINT
40 PRÉ 3 : PRINT CHR$(17)
50 INVERSE : PRINT "Toto — TOTO" :
  NORMAL.
```

Votre deuxième question fait l'objet d'une réponse, page 55. LOMEM vous permet de réserver un espace-mémoire à la suite d'un programme. **Exemple** : Menu va de l'adresse 2049 à l'adresse 4500, mais vous désirez réserver 1024 octets pour un stockage d'écran. Il suffit de fixer la valeur de LOMEM à 4500 + 1024 soit LOMEM : 5524. Pour connaître la valeur de LOMEM, chargez votre programme et tapez :

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

On peut protéger les pages graphiques en fixant LOMEM au-dessus (16384, pour la page 1, 24576 pour la page 2). HIMEM protège au contraire une routine placée en haut de la mémoire. On connaît sa valeur en tapant :

```
PRINT PEEK(116) * 256 + PEEK(115)
```



## Votre bibliothèque INFORMATIQUE



### DIÉTÉTIQUE SUR APPLE\*

Quand un médecin et un informaticien — Jean-Michel Lichtenberger et Jean-Michel Lego — se rencontrent, de quoi parlent-ils ? Par exemple de diététique, sujet actuel s'il en est !

Mais ils ne se contentent pas de parlotes. Passer aux actes cor-

par Guy-HACHETTE

respond mieux à leur caractère, d'où un livre que je vous conseille : *Diététique sur Apple II*.

Vous y apprendrez sûrement comment calculer votre surface corporelle, découvrir votre métabolisme énergétique et régler votre consommation sur vos dépenses, mais aussi à... programmer.

Car cet ouvrage vous livre non seulement les secrets d'une alimentation équilibrée, avec des tas de tableaux dignes de la collection *Que sais-je ?*, mais encore tous les programmes nécessaires à leur mise en œuvre.

Les débutants pourront certes se contenter de les copier, mais ils en tireront alors le meilleur profit...

\* Editions du P.S.I. (224 pages, 120 F)

### APPLE MODEMS ET SERVEURS\*

La communication est à l'ordre du jour, mais reste un peu mystérieuse pour bon nombre de gens,

y compris lorsque ceux-ci, disposant déjà d'un Apple, sont pourtant bien armés pour échanger des informations à l'intérieur d'un réseau local, utiliser un modem, bref, se familiariser avec les outils télématiques.

Plusieurs programmes leur feront assimiler les conventions de communication nécessaires à la conception du logiciel-utilisateur en Basic, en Pascal... et même en assembleur 6502.

\* Editions du P.S.I. (224 pages, 130 F)



## Adresses-écran avec DEF FN

Ce n'est qu'un exemple, parmi tant d'autres, d'utilisation de l'instruction DEF FN.

```
10 TEXT : NORMAL : PRINT CHR$(12); CHR$(21): HOME
20 DEF FN AD(X) = PEEK(X) + 256 * PEEK(X + 1)
30 FOR I = 23 TO 0 STEP -1: VTAB I + 1: HTAB 1: PRINT I;
  TAB(4) FN AD(40):: NEXT
40 POKE 32,10: POKE 33,30: PRINT
50 HOME : INVERSE : PRINT " ADRESSES-ECRAN AVEC (DEF
  FN) ": NORMAL
60 PRINT : LIST 10,50
70 FOR I = 2010 TO 2039: A = ASC(MID$("ENFONCEZ UNE TOU-
  CHE QUELCONQUE",I - 2009,1)): IF A = 32 THEN A = 96
80 POKE I,A - 64: NEXT
90 POKE - 16368,0: WAIT - 16384,128,127: POKE - 16368,0
100 PRINT : VTAB 19: PRINT "(1) MENU DISQUETTE (2) FIN": GET
  R$: POKE - 16368,0
110 IF R$ = "1" THEN PRINT : PRINT CHR$(4)"RUN MENU"
120 IF R$ <> "2" THEN 100
130 TEXT : HOME
```

# INDEX des articles déjà parus

& Goto (p. 46).....	N° 4	Disquettes à votre nom (p. 6).....	N° 1
Goto acrobatique par le truchement de l'Ampersand		Personnalisez vos disquettes en DOS 3.3	
Additions binaires (p. 19).....	N° 2	Disséquer un nombre (p. 38).....	N° 2
Jeu didactique pour apprendre à composer en binaire		Double listage en 80 colonnes (p. 11).....	N° 2
Adresse et longueur d'un programme (p. 36).....	N° 4	Basic et langage machine	
Adresses-écran (Fiche 4) (p. 41).....	N° 4	Découvrez l'assembleur avec Madeleine	
Affichage accéléré en langage		Hodé (p. 15).....	N° 3
machine (p. 63).....	N° 1	Désassemblage pratique (p. 18).....	N° 4
Affichage par poke (Fiche 4) (p. 41).....	N° 4	Pour désassembler, sur votre écran, autant d'instructions que vous le désirez	
Afficher un texte, empiler... dépiler (p. 38)...	N° 4	Echange (swap) de variables (p. 35).....	N° 4
Assembleur (initiation)		Pour accélérer vos programmes de tri	
Amortissement (p. 38).....	N° 1	Ecrans condensés (p. 36).....	N° 3
Avec ProCODE (p. 52).....	N° 1	Mémoriser... l'essentiel et économiser les octets	
Boucles simples en langage machine (p. 50)..	N° 1	Ecrire en page 2 (HGR2) (p. 25).....	N° 1
Boutabout (p. 30).....	N° 3	Ecriture sur pages graphiques (p. 54).....	N° 3
Routine LM permettant de mettre bout à bout deux programmes en Basic		Programme de Robert Faverge, Professeur au Lycée Thiers de Marseille	
Cadres express (p. 55).....	N° 4	Effacement d'écran par la gauche (p. 62).....	N° 1
Suite de TM1 (page 64)		Est-ce une anagramme ? (p. 33).....	N° 4
Calls (p. 31).....	N° 4	Un problème à traiter dans un programme de jeu ou d'enseignement	
Ils vous rendront service...		Evidemment qu'elle vous intéresse... la	
Carte langage (Fiche 3) (p. 41).....	N° 3	mémoire de votre Apple ! (p. 42).....	N° 1
Catalogue thématique (p. 19).....	N° 4	Fichiers programmes (Fiche 1) (p. 34).....	N° 1
Le catalogue complet de votre disquette, dans l'ordre alphabétique(DOS 3.3)		Fichiers EXEC (Fiche 1) (p. 34).....	N° 1
Charabia à décrypter (p. 10).....	N° 2	Fonte spéciale (p. 22).....	N° 4
Jeu dont la solution a paru dans le numéro 3		Vous pouvez la copier	
Chiffres en dents de scie (p. 3).....	N° 4	Gribouille (p. 3).....	N° 2
Des graphes (Utilitaire en Basic et langage machine)		Présentation de l'original traitement de texte de Madeleine Hodé	
Chiffres HGR (p. 12).....	N° 4	Générateur de programmes (p. 33).....	N° 3
Comment utiliser les chiffres d'une table de formes		Utilitaire créant automatiquement un programme de saisie de données	
Copie d'écran avec carte 80 colonnes (p. 40)..	N° 1	Hexadécimal ou décimal, et vice versa (p. 44)	N° 1
Cryptotexte (p. 13).....	N° 3	HTAB en 80 colonnes (p. 48).....	N° 1
Pour coder ou décoder à volonté (solution du charabia du numéro 2)		Imprimantes et graphisme (p. 56).....	N° 2
Créer des sous-volumes (p. 36).....	N° 4	Imprimez des listages formatés de vos programmes (p. 27).....	N° 2
Créer des écrans sur 80 colonnes (p. 3).....	N° 3	Éditez correctement vos meilleurs programmes	
Utilitaire dont vous ferez un usage intensif		Inversion de HGR1 ou HGR2 (p. 37).....	N° 1
Créez vos écrans en vous amusant (p. 17).....	N° 1	INPUT par la routine \$D52C (INLIN) (Fiche 4) (p. 42).....	N° 4
Raccourcissez vos programmes en mémorisant, sur la disquette, les écrans de présentation		Jeu (erreur) (p. 45).....	N° 1
Datez vos programmes en ProDOS (p. 31).....	N° 3	L'envers ne vaut pas l'endroit (p. 8).....	N° 1
De bons livres, et en français (p. 61).....	N° 1	Jeu simple, et améliorable	
De nouvelles perspectives pour vos histogrammes (p. 30).....	N° 1	La fenêtre (Fiche 1) (p. 33).....	N° 1
Un programme utilitaire indispensable		La page zéro de l'Apple IIe (Fiche 1) (p. 33)....	N° 1
Des octets en déplacement (p. 26).....	N° 1	La personne X (p. 17).....	N° 3
Utilisez mieux la mémoire auxiliaire de votre carte 80 colonnes		Jeu de logique élémentaire	
Dessine-moi un cercle (p. 47).....	N° 3		
Deux trucs pour votre Apple (p. 40).....	N° 2		



# INDEX des articles déjà parus

Le curseur (Fiche 1) (p. 33).....	N° 1	Prefix (p. 36).....	N° 4
Les caractères «souris» de l'Apple IIc (p. 34)..	N° 2	Le mot de passe pour accéder à une disquette ProDOS	
Les fichiers aléatoires (p. 15).....	N° 2	Print using (p. 48).....	N° 1
Enregistrements de taille fixe (initiation)		Programme en EXEC (Fiche 1) (p. 34).....	N° 1
Les fruits de la passion (p. 11).....	N° 4	Programmer ? Bien sûr que cela en vaut la	
Graphisme (pour figurer les touches Pomme ouverte et		peine ! (p. 39).....	N° 1
Pomme fermée)		ProDOS et la carte langage (p. 43).....	N° 3
Les nombres décimaux et votre micro (p. 16).	N° 1	ProDOS et vous (Ram, Catalog, Run et	
Les problèmes d'imprimante (p. 51).....	N° 4	Brun) (p. 44).....	N° 3
Les variables dans la mémoire de		Que va-t-il se passer ? (p. 54).....	N° 1
l'Apple (p. 46).....	N° 3	Quelques notes pour votre Apple (p. 51).....	N° 3
Initiation à l'assembleur		Saisie avec tri immédiat (p. 49).....	N° 1
Lignes DATA ou lignes de Pokes (p. 55).....	N° 3	Saisie d'un texte, suivie de son affi-	
List avec l'ampersand (p. 48).....	N° 1	chage (p. 45).....	N° 3
Listage et édition en langage machine (p. 14)	N° 1	Initiation à l'assembleur	
Utilitaire fort pratique		Saisie LM assistée (p. 48).....	N° 3
Listage-contrôle d'un fichier «T» (p. 44).....	N° 2	Utilitaire destiné à vous simplifier la vie	
Logic-grilles (p. 57).....	N° 1	Soustraction en langage machine (p. 43).....	N° 2
Jeu visuel		La suite de l'initiation à l'assembleur	
Logogrille (p. 50).....	N° 4	Spécial Bonjour (p. 3).....	N° 1
La phrase cachée (récré)		Tracer un cadre en langage machine (p. 64)...	N° 1
Ma chère vieille pomme (p. 41).....	N° 1	Transfert d'un écran de 80 colonnes en	
Majuscules en inverse (p. 32).....	N° 3	mémoire auxiliaire (Fiche 2) (p. 41).....	N° 2
Une routine permettant d'afficher une ligne dont seules les		Une fiche indispensable	
majuscules sont en inverse, par Y. Koenig		Transfert d'écrans texte (Fiche 3) (p. 42).....	N° 3
Onerr Goto (p. 38).....	N° 2	Un accessoire utile : le magnétophone (p. 31)	N° 2
Ouest-Est (p. 38).....	N° 3	Un certain jour de la semaine (p. 12).....	N° 1
Petit jeu géographique en mode Text		Quel jour êtes-vous né(e) ?	
Ouvrez donc une fenêtre ! (p. 43).....	N° 4	Un drive pour 1 500 F ? Banco ! (p. 46).....	N° 2
Apprenez à ouvrir (et à refermer) une fenêtre... en un éclair !		Pourquoi payer votre matériel au prix fort ?	
Ouvrez l'œil ! (p. 52).....	N° 3	Un moyen de savoir si un fichier existe (p. 36)	N° 1
Jeu visuel ultra-court		Une petite révision d'anglais (p. 28).....	N° 1
Pair ou impair avec quelques octets (p. 35)....	N° 2	La première leçon (à vous d'écrire les suivantes !)	
Une petite étude des nombres aléatoires... pour le plaisir		Utilisation de la page 2 (TEXT) pour stocker un	
Partez en voyage avec Max (p. 12).....	N° 3	écran (p. 48).....	N° 4
Présentation du logiciel Max the Globe Trotter		Valeur ASCII après GET (p. 48).....	N° 1
Petits trucs d'assembleur (p. 37).....	N° 4	Visublocs (p. 21).....	N° 3
La numération hexadécimale		Un gros morceau... pour une nouvelle approche d'une repré-	
Plus loin avec l'assembleur (p. 39).....	N° 2	sentation graphique des chiffres	
Plus loin avec la fonte GRAPH.CAR (p. 13).....	N° 4	Vos chiffres valent bien un fromage ! (p. 49)	N° 2
Comment lire (et même modifier) une fonte de caractères		La suite logique des histogrammes, avec une certaine	
telle que GRAPH.CAR		compatibilité	
Poke... d'accord, mais à quelle adresse ?		Vos menus à la carte (p. 5).....	N° 2
(p. 17).....	N° 4	En Basic et langage machine... pour un affichage ultra-rapide	
Une petite routine pour connaître l'adresse d'un point particu-		Vous êtes au courant, mais cela vous coûte	
lier de votre écran		combien ? (p. 46).....	N° 1
Pokes utiles (p. 5).....	N° 1	Petit utilitaire de gestion personnelle	
Pour oublier le jeu du pendu (p. 32).....	N° 2	Vous avez emprunté ? (p. 27).....	N° 1
Une solution : le jeu de l'homme libre !		Vous dites addition ? (p. 51).....	N° 1
Pour 7 ronds ! (p. 15).....	N° 4	Initiation au langage machine	
Un jeu graphique essentiellement visuel			

La reliure-écrivin de TREMPLIN MICRO (elle peut contenir 8 numéros).

## Bulletin de commande et d'abonnement

Nom ..... Prénom .....

Adresse complète .....

Code postal      Ville .....

LES DISQUETTES FONCTIONNENT SOUS DOS ET ProDOS  
(à condition de posséder une version Apple de ce SYSTEME D'EXPLOITATION)

<input type="checkbox"/>	1	Abonnement à 6 numéros (un an) : FRANCE	180 F
<input type="checkbox"/>	1	DISQUETTE de TREMPLIN MICRO N°1 (Ile et Ilc)	105 F
<input type="checkbox"/>	1	DISQUETTE de TREMPLIN MICRO N°2 (Ile et Ilc)	105 F
<input type="checkbox"/>	1	DISQUETTE de TREMPLIN MICRO N°3 (Ile et Ilc)	105 F
<input type="checkbox"/>	1	DISQUETTE de TREMPLIN MICRO N°4 (Ile et Ilc)	105 F
<input type="checkbox"/>	1	DISQUETTE de TREMPLIN MICRO N°5 (Ile et Ilc)	105 F
<input type="checkbox"/>	1	NUMÉRO 1 de TREMPLIN MICRO	30 F
<input type="checkbox"/>	1	NUMÉRO 2 de TREMPLIN MICRO	30 F
<input type="checkbox"/>	1	NUMÉRO 3 de TREMPLIN MICRO	30 F
<input type="checkbox"/>	1	NUMÉRO 4 de TREMPLIN MICRO	30 F
<input type="checkbox"/>	1	NUMÉRO 5 de TREMPLIN MICRO	30 F
<input type="checkbox"/>	1	RELIURE-ECRIN	35 F


\* Cochez la case de votre choix.

Participation aux frais d'envoi (gratuit pour les abonnés) + 10 F

Numéro d'abonné ou de client : ..... Total à payer ..... F

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

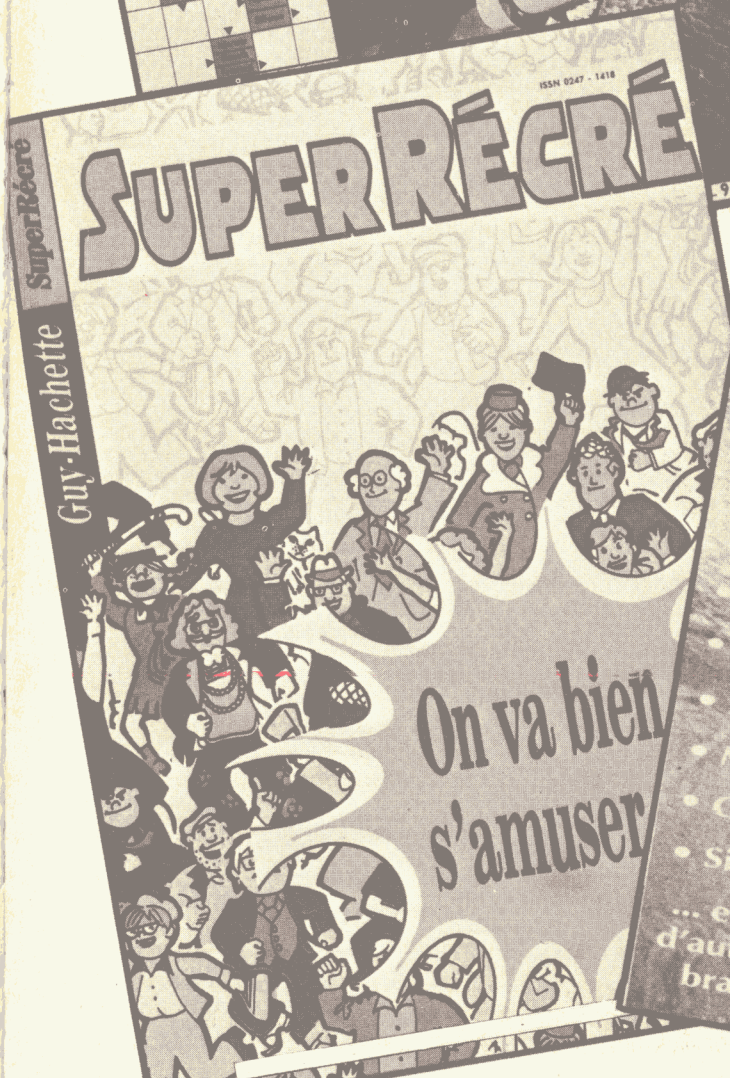
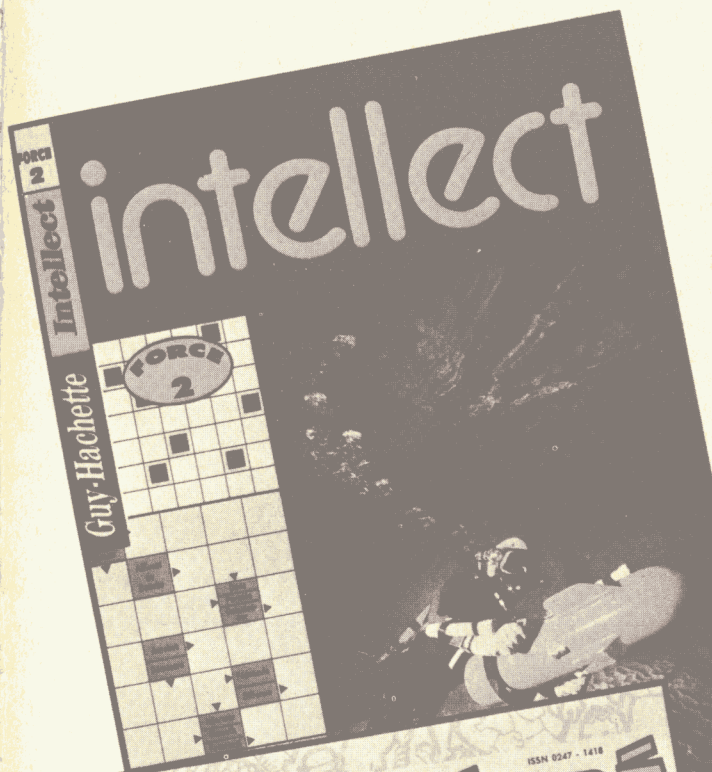
Mode de règlement choisi :  Chèque  Mandat-lettre  Carte Bleue

	N° de votre Carte Bleue <input type="text"/>	<input type="text"/>
	Date d'expiration <input type="text"/>	Signature (obligatoire)
	Montant à débiter <input type="text"/>	F

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



# Guy-Hachette pense à vos loisirs !



**TREMLIN MICRO** — Bimestriel — C'est une publication des Editions JIBENA, 4, rue de la Cour-des-Noues, 75020 PARIS — S.A. au capital de 3 600 000 F — Imprimé par CITÉ-PRESS/PARIS — Dépôt légal à la date de parution — Inscription à la Commission Paritaire des Publications et Agences de Presse : en cours — Directeur de la Publication : Guy-Clément COGNÉ — Diffusion N.M.P.P.



# Chasseur d'Images

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

The cover of the magazine 'Chasseur d'Images' is shown at an angle. It features a portrait of a smiling woman with blonde hair. In the top left corner of the cover, there is a photograph of a Nikon F 301 camera. Below the camera, the text reads 'TESTS Nikon F 301 Pentax P 30'. At the bottom of the cover, it says 'EXCLUSIF Tous les 50 mm au banc d'essai'.

**Chasseur  
d'Images**

**TESTS**  
Nikon F 301  
Pentax P 30

**EXCLUSIF**  
Tous les 50 mm  
au banc d'essai

N° 78 - Décembre 1985 - Dixième année - Canada 3,95 \$ - Espagne 450 PTA  
Maroc 20 DH - Suisse 6 FS - Belgique 120 FB - FRANCE 16 F

**Chez votre  
marchand de journaux !**