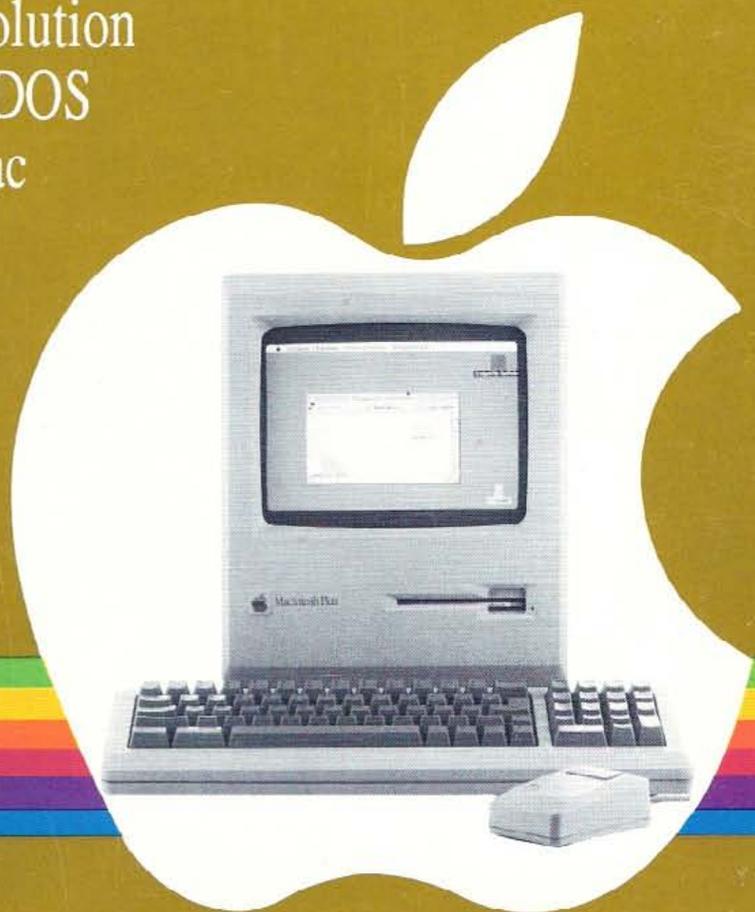


La revue francophone indépendante pour les utilisateurs des
Apple][+, //e, //c™ et Macintosh™

ppom's

- 🍏 Les routines en assembleur sous ProDOS
- 🍏 Composition de numéros de téléphone
- 🍏 Compacteur de programmes Basic
- 🍏 Hyperparallélépipèdes et fractaux
- 🍏 Un Print Using en assembleur
- 🍏 Copie d'écran basse résolution
- 🍏 "Startup" Basic sous ProDOS
- 🍏 Transfert Apple // vers Mac
- 🍏 De la formule à la courbe
- 🍏 Des titres dans le Basic
- 🍏 Des Bugs sous ProDOS
- 🍏 Rubrique MacAstuces
- 🍏 Mini-Éditeur Basic
- 🍏 Jeu de mots
- 🍏 Six lettres



NUMERO 22 . PRIX 40 F

M2366-22-40 F

ISSN : 0294-6068

DES LIVRES POUR VOTRE APPLE

MAITRISEZ LA PROGRAMMATION

Basic + 80 routines pour Apple*

par Michel Martin
144 pages - 95,00FF
Pour "Muscler" votre Apple, créer vos propres jeux d'animation et maîtriser la technique du dessin animé.

Techniques de programmation sur Apple II*

par René Belle
168 pages - 95,00 FF
Pour créer des logiciels plus puissants et plus structurés grâce à des astuces illustrées de nombreux programmes.

L'Apple et ses fichiers*

par Jacques Boisgontier
176 pages - 95,00 FF
Les commandes du Système d'Exploitation Disque, les instructions du Basic Applesoft et des fichiers séquentiels et à accès direct puis leur utilisation à l'aide de programmes divers.

REALISEZ VOS APPLICATIONS

Destination aventure sur Apple*

par Delton T. Horn
246 pages - 140,00 FF
Utilisez les instructions, les explications et les astuces de programmation de ces 4 jeux d'aventure en Basic pour créer les vôtres.

La programmation des jeux d'arcades*

par Jean-Luc Fischer
300 pages - 140,00 FF
Pour générer sur votre Apple des effets spéciaux époustoufflants grâce à des programmes en Basic et en Assembleur.

Apple, logique et systèmes experts*

par René Descamps
224 pages - 120,00 FF
Pour étudier toutes les étapes logiques de la pensée humaine et créer un micro système expert en Basic.



Apple, modems et serveurs*

par Alain Mariatte
224 pages - 120,00 FF
Apprenez à vous servir d'un modem, à utiliser un logiciel de communication, à connaître les principaux réseaux et vous familiariser avec les outils télématiques.

Diététique sur Apple*

par J.M. Jégo et J.M. Lichtenberger
224 pages - 120,00 FF
Pour calculer votre surface corporelle et découvrir votre métabolisme énergétique grâce à votre Apple.

EXPLOITER LE SYSTEME ET L'ASSEMBLEUR

Assembleur de l'Apple***

par N. Bréaud - Pouliquen
240 pages - 120,00 FF
Pour apprendre à obtenir rapidité d'exécution et économie de mémoire, voici une initiation à l'assembleur (6502 - 65C02) illustrée de nombreux programmes.

ProDos sur Apple IIe et IIc*

par Francis Verscheure
104 pages - 85,00 FF
Découvrez ProDos pour gérer des catalogues et des fichiers si vous voulez profiter de ce système performant.

Clefs pour l'Apple II**

144 pages - 105,00 FF
Clefs pour l'Apple IIc et IIe 65C02**

172 pages - 130,00 FF
par N. Bréaud - Pouliquen
Pour accéder rapidement à la syntaxe des commandes, aux codes caractères, aux codes machine... et des trucs utiles lorsque vous programmez.

Les ressources de l'Apple IIc***

par N. Bréaud Pouliquen
108 pages - 85,00 FF
Pour découvrir les avantages du IIc, utiliser le logiciel de base du système et programmer astucieusement la souris.

APPLE MODEMS ET SERVEURS



* sur Apple II plus, IIe et IIc

** sur Apple II plus et IIe

*** sur Apple IIc et IIe 65C02

EGALEMENT CHEZ VOTRE LIBRAIRE ET EN BOUTIQUE SPECIALISEE

Je commande le(s) livre(s) :

.....à.....,00FF
.....à.....,00FF
.....à.....,00FF
.....à.....,00FF

Nom.....Prénom.....

Adresse.....

Code postal.....Ville.....

.....paiement par chèque joint
.....paiement par Carte Bleue Visa

Date d'expiration.....

N° de la Carte.....

Signature.....

Envoyer ce bon accompagné de votre règlement à :
P.S.I DIFFUSION B.P 86
77402 Lagny / Marne Cedex Tel : 60 06 44 35
En Belgique à P.S.I. BENELUX
17 rue du Doyné, 1180 Bruxelles
Tel : (2) 345 39 03
En Suisse, à P.S.I. SUISSE
Route Neuve 1, 1701 Fribourg
Tel : (037) 23 18 28
Au Canada, à PROLOGUE INC.
2975 rue Sarréton, Ville St Laurent
Québec H4R1E6 - Tel : (514) 332 58 60



Composition de numéros de téléphone

Bernard Hoyez
Jean-Luc Nail



Page 6

Adaptation à ProDOS des routines en assembleur

Alexandre Avrane



Page 9

Compacteur de programme Basic

Sylvie Gallet



Page 17

Copie d'écran GR

Bernard France



Page 23

De la formule à la courbe

Jean-Louis Chauvin



Page 27

'Startup' Basic sous ProDOS

Pascal Cantot



Page 33

Des titres en Basic

Sylvie Gallet



Page 37

Jeu de mots

Marianne Sutz



Page 44

Transfert II -> Mac



Page 44

Hyperparallélipèdes et Fractaux

Jean-Luc Bazanegue



Page 45

MacAstuces

Page 50

Mini-Éditeur Basic

Guillaume Pot



Page 51

Six Lettres

Mandfred Brede



Page 57

Des Bugs sous ProDOS

Alexandre Avrane

Page 63

Print Using

Patrice Neveu



Page 66

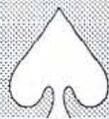
Editorial 5
Bibliographie 69
Courrier des Lecteurs 72

Les annonceurs :

Apple : pages 38 et 39. Computic : page 55. PSI : page 2. Infomac : page 75. Ordinateur Individuel : Page 76.

Editions MEV - 64/70 rue des Chantiers - 78000 Versailles. Directeur de la publication : Hervé Thiriez

Des atouts pour votre Apple :



**Editeur
plein écran**

Apple][+,
//e et //c

EPE

Listez vos programmes Basic en avant et en arrière. Modifiez, insérez, effacez des caractères en plein écran sans relire les lignes. Recherchez toute chaîne de caractères. Choisissez vous-même les codes de contrôle d'EPE. Modifiez EPE : le fichier source est sur la disquette.



Vous avez un traitement de texte et une ImageWriter, alors...

P-Format][

augmentera considérablement les possibilités d'impression de votre logiciel. Vous pouvez obtenir une justification droite, gauche, totale, centrée, ou encore page paire à droite et impaire à gauche de façon automatique. A noter que la justification n'est pas perturbée par les caractères du type 'é' ou 'ö'. Le style d'impression peut passer de la simple à la double largeur, de gras ou souligné à normal, en proportionnel et au sein d'une même ligne sans perte de justification. P-Format][c'est aussi : un mode mathématique et un mode graphique grâce à la police programmée.



**DISK
MANAGER,
LE DOS
EN KIT**

][+
//e
//c

Accompagné d'une cinquantaine de pages de documentation, Disk Manager permet de recréer les commandes du Dos, redéfinir l'organisation d'une disquette, grâce à un jeu d'instructions qui en fait un langage simple d'accès à la disquette. Il offre également un programme d'édition à l'aide de commandes évoluées. Quatre utilitaires figurent aussi sur la disquettes : Utili-disque (réparation du disquette détruite, vérification...), Ultra-copie (pour un backup très rapide), Edicat (édition du catalogue, ...) et Multi-disque.

Apple][+
Apple //e



**Coup de
cœur pour
le BASICIUM**
le Basic enrichi...

Cet utilitaire vous permet de gérer des masques ou écrans 40 ou 80 colonnes, saisir toutes les variables d'un écran en une seule commande, recopier l'écran 40 ou 80 colonnes sur l'imprimante, faire un nettoyage mémoire (garbage collection) très rapide, gérer des messages, ne saisir que l'une des variables affectées à un écran, etc.

Pour Apple][+ ou Apple //e

**Bon de
Commande**

Abonnements à Pom's pour 6 numéros :

- abonnement à la revue seule 200,00 F
- abonnement revue et disquettes Apple][. . . 480,00 F
- abonnement revue et disquettes Macintosh . . 600,00 F
- Editeur plein écran EPE 150,00 F
- P-Format][. 200,00 F
- DISK MANAGER 450,00 F
- BASICIUM 150,00 F

Nom

Prénom

Adresse

Envoyez ce bon et votre règlement à EDITIONS MEV - 64, rue des Chantiers - 78000 Versailles.
Supplément avion hors CEE : 15,00 F par numéro et/ou disquette.

Il n'est pas facile d'être honnête et de maintenir certains standards dans un monde où le souci d'intégrité est souvent pris pour de la naïveté. Comme vous avez pu le constater, Pom's est une revue comportant fort peu de publicité. Cela tient en bonne partie à notre refus, malgré l'insistance de nombreux de nos clients, de promettre des articles sur leurs produits en échange de contrats de publicité, un mode d'échange pudiquement baptisé "publicité rédactionnelle".

Pom's préfère garder sa liberté d'expression, au risque de perdre certains annonceurs. Nous faisons le pari que les lecteurs ne se laisseront pas abuser par ces faux articles, qui pourraient les encourager à acheter des produits de qualité insuffisante.

Le monde de l'Apple II, comme nous le montre bien Jean-Michel Gourévitch depuis plusieurs numéros, est en constante évolution. Dans combien de temps disposerons-nous d'un Apple II aussi puissant que le Macintosh ? Probablement plus très longtemps ; il n'en reste pas moins que – pour des raisons de compatibilité avec le vieil Apple II – on n'arrivera pas tout de suite avec ce matériel à la qualité ergonomique du Macintosh.

Ceci dit, ces derniers temps, le monde du Macintosh bouge encore plus vite que celui de l'Apple II, avec en particulier l'arrivée du MacPlus. Depuis quelques mois, le prix des extensions 512K pour le Mac baisse sans cesse : nous sommes passés en un an de pratiquement 12.000 F TTC à moins de 3.000 F TTC aujourd'hui. Quand on voit le prix que coûte encore aujourd'hui l'extension à 1 Mo, le possesseur d'un Mac aura intérêt à faire transformer son Mac en MacPlus (extension à 1 Mo, nouvelles ROMs, passage du lecteur intégré à 800K) plutôt que de s'offrir la seule extension mémoire.

Autre signe des temps, après s'être longtemps enfermé dans sa politique de standard à part de l'industrie, Apple met en relief sa capacité à travailler en réseau avec le monde IBM.

Nous continuons à recevoir des lettres de possesseurs d'Apple II s'inquiétant de savoir si nous n'allons pas abandonner progressivement celui-ci pour le Macintosh. Je pense que, depuis le temps que nous proposons le cahier Macintosh, ces lecteurs peuvent être rassurés : nous n'avons pas réduit la pagination Apple II – comme promis – et toute extension du cahier Mac se fera sous la forme d'une augmentation de la pagination.

Remarque importante : nous continuons à recevoir du courrier à l'ancienne adresse, mais celui-ci ne suit plus systématiquement. Pour être sûr que vos lettres nous parviennent, utilisez donc l'adresse : 64/70 rue des Chantiers - 78000 Versailles.

Hervé Thiriez

Photo de couverture : Macintosh Plus - Clavier AZERTY avec pavé numérique incorporé, 1 Moctets de RAM bientôt extensible à 4 Moctets, 128 Ko de ROM, routines QuickDraw réécrites pour plus de rapidité, lecteur de disquettes double face double densité 800 Ko. Prix hors TVA : Macintosh Plus : 25900 F (le Mac 512 Ko coûte désormais 22900 F, le Mac 128 Ko est abandonné) - lecteur externe 800 Ko (pour Mac Plus ou Mac standard) : 3900 F - mise à niveau Mac Plus pour Mac 128 Ko : 6500 F - mise à niveau Mac Plus pour Mac 512 Ko : 4500 F.

Ont collaboré à ce numéro : Alexandre Avrane, Jean-Luc Bazanegue, Manfred Brede, Pascal Cantot, Jean-Louis Chauvin, Alexandre Duback, Bruno Fénard, Bernard France, Sylvie Gallet, Jean-Michel Gourévitch, Olivier Herz, Bernard Hoyez, Gérard Michel, Jean-Luc Nail, Patrice Neveu, Christian Piard, Guillaume Pot, Marianne Sutz

Directeur de la publication, rédacteur en chef : Hervé Thiriez.

Rédacteurs : Alexandre Avrane, Olivier Herz.

Siège social : Editions MEV - 49, rue Lamartine - 78000 Versailles. Tél. : (1) 39.51.24.43.

Publicité : Editions MEV.

Diffusion : N.M.P.P.

Impression : Rosay - 47, avenue de Paris - 94300 Vincennes. Tél. : (1) 43.28.18.63.

Pom's est une revue indépendante non rattachée à Apple Computer, Inc. ni à Apple Computer France S.A.R.L. Apple, le logo Apple, Mac et le logo Macintosh sont des marques déposées d'Apple Computer, Inc.

Composition automatique de numéros de téléphone

Bernard Hoyez, Jean-Luc Nail

Ne plus se souvenir d'un numéro, ou le doigt qui dérape sur le clavier du téléphone, cela vous est déjà arrivé, n'est-ce pas ? La numérotation à huit chiffres n'a pas arrangé les choses. Combien de temps et de taxes de base cela vous a-t-il coûté ? Une solution à ce problème est de remplacer votre vieux téléphone à cadran par un superbe appareil à mémoires ou alors... de demander à votre vieil Apple de faire ce travail à votre place. Désormais, Roméo, faites apparaître le doux nom de Juliette sur l'écran de votre ordinateur favori et vous entendrez aussitôt la voix de votre chère et tendre amie.

Le composeur automatique de numéro nécessite :

- une interface entre l'Apple et la ligne téléphonique ;
- un logiciel de composition de numéro.

Principe de la composition d'un numéro téléphonique

Un appel téléphonique se décompose en quatre phases :

- la prise de ligne ;
- la composition du numéro sur le clavier ou le cadran ;
- l'échange audio ;
- le raccrochage.

La prise de ligne s'effectue lorsque l'on décroche le combiné : un interrupteur se ferme et court-circuite la ligne.

La composition de numéros consiste à envoyer sur la ligne une succession de signaux représentant chacun un chiffre. Un chiffre est lui-même une succession d'impulsions et à chaque chiffre correspond un certain nombre d'impulsions élémentaires : une impulsion pour le chiffre 1, deux impulsions pour le chiffre 2, ..., 10 impulsions pour le chiffre 0. Une impulsion comporte une ouverture de ligne d'une durée de 66 millisecondes suivie d'une fermeture de ligne de 33 millisecondes.

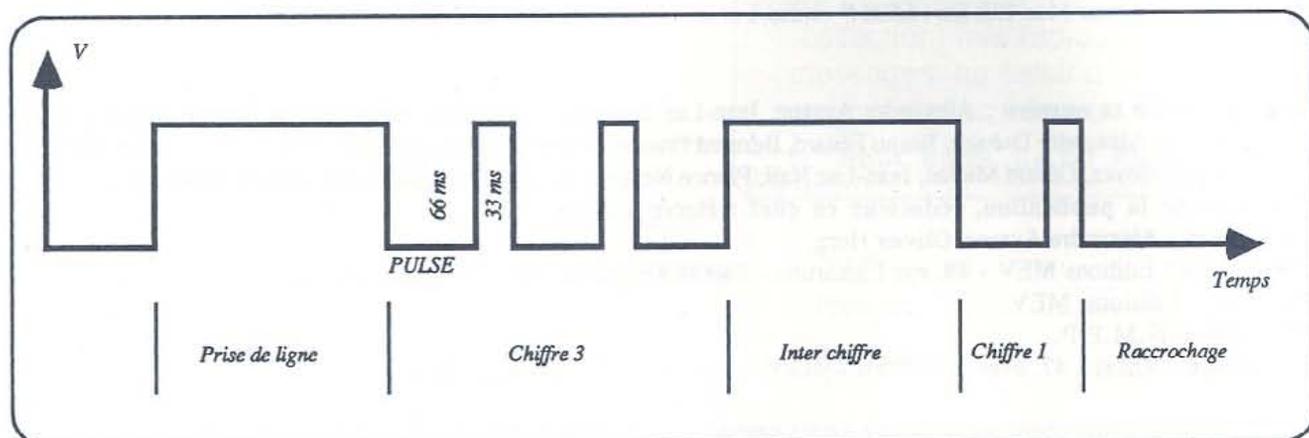
Le raccrochage s'opère lorsqu'on repose le combiné, ce qui a pour effet d'ouvrir à nouveau le circuit. La forme du signal envoyé sur la ligne a l'aspect représenté dans l'encadré.

L'interface : la prise de jeu !

Le principe en est simple, il s'agit de simuler un second téléphone en parallèle pour lequel la prise de ligne, la composition et le raccrochage seront gérés par l'ordinateur. Une fois la communication établie par ce "téléphone virtuel", on reprend la ligne par le téléphone réel.

Le téléphone virtuel agit comme un interrupteur, dont la commande peut être réalisée par une sortie logique de l'Apple. Inutile de concevoir une carte spécialisée car ces sorties logiques existent sur la prise de jeu. Elles sont au nombre de quatre et appelées AN0, AN1, AN2 et AN3. Elles ne sont accessibles qu'à l'intérieur de l'Apple au niveau du connecteur 16 broches.

Sur l'Apple IIe, on a négligé de reprendre ces sorties sur le connecteur externe 9 broches, c'est dommage. Une seule sortie est nécessaire, nous avons choisi AN0 situé sur la broche numéro 15.



Pour assurer une isolation parfaite entre l'Apple et la ligne téléphonique, l'interrupteur est celui d'un relais. Nous avons choisi un mini-relais 5 V de type CLARE PRME (ou équivalent). Pour commander ce relais, il faut amplifier le signal fourni par AN0, grâce à un transistor et au + 5 V disponible sur la broche 1 du connecteur de jeu.

L'ensemble des composants prend place sur une petite plate-forme enfichée dans le connecteur de jeu par l'intermédiaire d'un support de circuit intégré.

Le programme

Une partie de la mémoire de l'Apple est réservée aux entrées/sorties, notamment pour ce qui nous concerne aux sorties logiques ou "annonciateurs" de la prise de jeu. Pour l'annonciateur AN0, il s'agit des mémoires d'adresses \$C059 et \$C058. Le simple fait d'écrire à ces adresses met respectivement la broche AN0 à l'état 1 (3,5 V) ou à l'état 0 (0,3 V).

Cette écriture peut se faire de deux manières, soit en Basic, soit en langage-machine :

- en BASIC, on pourra faire un POKE 49241,0 (état 1) ou un

POKE 49240,0 (état 0) ;

- en langage d'assemblage, on écrira STA \$C059 (état 1) ou STA \$C058 (état 0).

Le logiciel consiste donc à gérer la commutation de ces deux adresses à des intervalles de temps précis et brefs. Lors de la composition d'un chiffre, entre deux impulsions successives, le passage à l'état haut dure 33 millisecondes. Un tel laps de temps est trop court pour être maîtrisé dans un programme Basic, d'où la nécessité d'écrire en langage-machine la partie du programme effectuant la composition des chiffres.

Le programme 'COMPOSEUR'

Ce programme fait appel à une routine de temporisation du moniteur appelée WAIT, implantée en \$FCA8. Elle permet de régler précisément les durées des passages à l'état haut et à l'état bas. La temporisation, en microsecondes, est une fonction de la valeur entrée dans l'accumulateur :

$$t = (26 + 27 * A + 5 * A * A) / 2$$

Ainsi, pour une durée de 66 ms, on entrera 159 dans A avant d'appeler la routine WAIT.

Le cas de la composition du chiffre 0 est traité à part, car il

correspond à la génération de 10 impulsions (et non de 0).

Le code ASCII du chiffre à composer, stocké en mémoire 7 par le programme Basic, est chargé dans l'accumulateur. Si le code ASCII n'est pas celui d'un chiffre, on passe au chiffre suivant en positionnant un drapeau en mémoire 8. Le AND #\$0F permet la transformation du code ASCII en chiffre de 0 à 9. Le cas particulier du 0 est ensuite traité. La routine principale BOUCLE génère alors un nombre d'impulsions (PULSE) différent selon le chiffre stocké dans le registre X.

Le programme 'AGENDA'

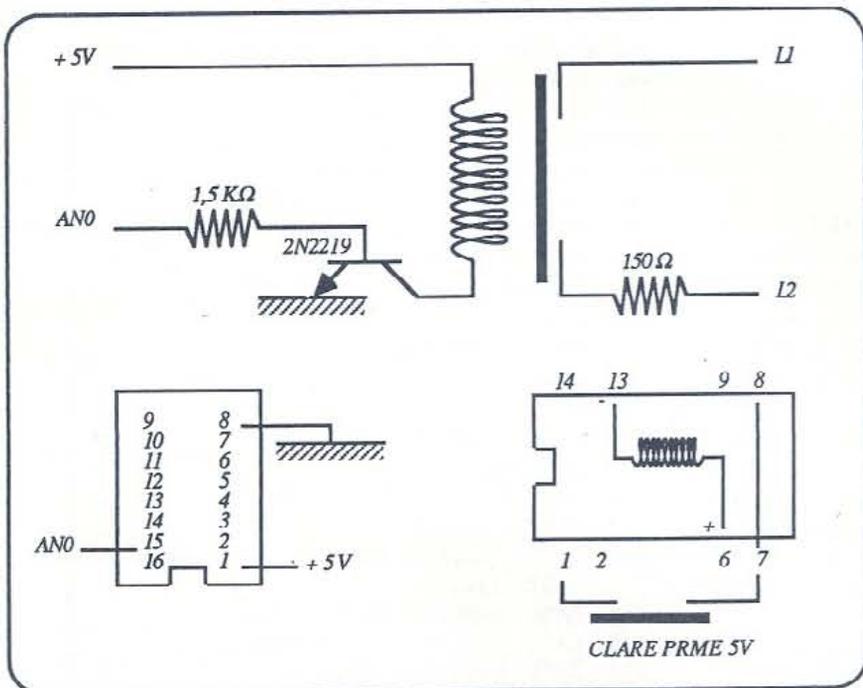
Ce programme Basic établit la correspondance entre le nom d'un abonné et son numéro de téléphone, puis il effectue la composition de ce numéro. Le choix de l'abonné se fait en visionnant l'agenda au moyen des flèches gauche et droite.

Pour ne pas compliquer le programme d'un gestionnaire de fichier, les noms d'abonnés et leur numéro sont intégrés à la fin du programme sous forme de DATA. Il suffira d'ajouter de nouvelles lignes de DATA pour entrer de nouveaux correspondants. Le programme place ces données dans le tableau T\$. Rien ne vous empêche, si vous possédez un agenda sur disquette dont vous connaissez la structure, de le récupérer dans le tableau T\$. Comme la liste de vos correspondants est susceptible de changer très souvent, on a évité de remplir le tableau T\$ au moyen d'une boucle FOR...NEXT que l'on aurait dû ajuster chaque fois. En cas de fin de DATA, la routine de traitement d'erreur interrompt le remplissage du tableau.

Une fois le choix validé par la frappe sur la touche <RETURN> :

A On effectue la prise de ligne (ligne 380).

B Le code ASCII de chaque chiffre du numéro téléphonique est stocké en mémoire 7 et un appel à la routine 'Composeur' est fait, ceci autant de fois que le numéro comporte de chiffres. Entre chaque chiffre



composé est laissé un temps de relaxation.

C Dès que le numéro est composé, vous pouvez décrocher le combiné téléphonique. Le "téléphone virtuel" reste branché en parallèle pendant quelques secondes, le délai restant est affiché sur l'écran.

D Le "téléphone virtuel" est alors coupé (ligne 550), vous restez alors seul en communication. Allô... Juliette... ?

N.D.L.R. : un tel montage n'entre pas dans le cadre de ce que vous autorise l'administration des Postes. Vous testerez ceci, bien entendu, sur une installation indépendante du réseau.



Programme COMPOSEUR.S (Assembleur Merlin)

```

1 *****
2 *                               *
3 *           Composeur          *
4 *                               *
5 *****
6
7 * 8 novembre 85
8
9
10 ;ADRESSES
11
12 SETANO = $C059
13 CLRANO = $C058
14 WAIT   = $FCA8
15 CHIFFRE = 7
16 ERREUR = 8
17
18         ORG $300
19
20 ENTREE = *
21
22         LDA CHIFFRE
23         CMP #*0"-1
24         BCC ERROR
25         CMP #*9"+1
26         BCS ERROR
27
28         AND #*1111

```

```

29         BNE COMPOSE
30         LDA #10
31 COMPOSE TAX
32
33 BOUCLE = *
34         JSR PULSE
35         DEX
36         BNE BOUCLE
37         RTS
38
39 ERROR = *
40         LDA #$FF
41         STA ERREUR
42         RTS
43
44 PULSE LDA #159
45         STA CLRANO
46         JSR WAIT
47         LDA #108
48         STA SETANO
49         JSR WAIT
50         RTS

```

Récapitulation COMPOSEUR

```

0300- A5 07 C9 AF 90 12 C9 BA
0308- B0 0E 29 0F D0 02 A9 0A
0310- AA 20 1D 03 CA D0 FA 60
0318- A9 FF 85 08 60 A9 9E 8D
0320- 58 C0 20 A8 FC A9 6C 8D
0328- 59 C0 20 A8 FC 60

```

Programme AGENDA

```

10 REM ***COMPOSEUR DE NUMERO***
20 REM *           par           *
30 REM *       Bernard HOYEZ    *
40 REM * et Jean-Luc NAIL      *
50 REM *****
60 REM ==>ENTRER LES NOMS ET LES NUMEROS EN D
   ATA A LA SUITE DES AUTRES
70 REM *****
80 HOME
90 ONERR GOTO 570
100 PRINT "          COMPOSITION AUTOMATIQUE DE"
110 PRINT "          NUMERO TELEPHONIQUE"
120 PRINT CHR$(4);"BLOADCOMPOSEUR"
130 DIM T$(100,2)
140 N = 0
150 N = N + 1
160 FOR J = 1 TO 2
170 READ T$(N,J)
180 NEXT J
190 GOTO 150
200 VTAB (8)
210 PRINT "Utilisez les flèches "; INVERSE
   : PRINT ">"; NORMAL : PRINT " et "; I
   NVERSE : PRINT "<-"; NORMAL
220 PRINT "pour sélectionner le nom de l'abo
   nné"
230 PRINT : PRINT "Validez avec la touche RE
   TURN"
240 I = 0
250 GET A$
260 A% = ASC (A$)
270 IF A% < > 13 AND A% < > 21 AND A% < >
   8 THEN 260

```

```

280 IF A% = 13 THEN 360
290 IF A% = 21 THEN I = I + 1
300 IF A% = 8 THEN I = I - 1
310 IF I > N - 1 THEN I = N - 1
320 IF I < 1 THEN I = 1
330 VTAB 15
340 CALL - 868: PRINT T$(I,1),T$(I,2)
350 COTO 250
360 X$ = T$(I,2)
370 REM **COMPOSITION DU NUMERO**
380 POKE 49241,0: REM PRISE DE LIGNE
390 FOR I = 1 TO 700: NEXT I
400 HOME
410 PRINT : PRINT "Je compose le numéro ";
420 FOR I = 1 TO LEN (X$)
430 A = ASC ( MID$ (X$,I,1) ) + 120
440 POKE 7,A
450 CALL 768
460 PRINT CHR$(A);
470 FOR T = 1 TO 600: NEXT T
480 NEXT I
490 HTAB 1: VTAB 5: PRINT "Décrochez le comb
   iné, maintenant..."
500 VTAB 8: PRINT ".....avant"
510 FOR I = 400 TO 1 STEP - 1
520 VTAB (15): HTAB (11)
530 PRINT INT (I / 50);" secondes"
540 NEXT I
550 POKE 49240,0
560 GOTO 580
570 IF PEEK (222) = 42 THEN 200
580 HOME : RESTORE
590 PRINT "...Pret pour un autre appel ...":
   GOTO 140
600 DATA RENSEIGNEMENTS,3612
620 DATA AUTRES RENSEIGNEMENTS,3611
630 DATA JULIETTE,12345678

```

Adaptation à ProDOS des routines en assembleur

Alexandre Avrane

Il existe actuellement plusieurs centaines d'utilitaires en assembleur qui facilitent la programmation en Applesoft : tris, contrôle de saisie, formatage de nombres, etc. Malheureusement ces routines, conçues pour fonctionner correctement sous DOS 3.3, ont été assemblées à des adresses fixes et ne tournent plus dans un environnement ProDOS ; l'objectif de cet article est donc de répondre aux questions des lecteurs qui désirent conserver l'acquis d'une importante logithèque d'utilitaires, en les transposant pour un environnement ProDOS.

Cependant, il s'agit plus de fournir des conseils que de proposer un module tout chaud et prêt-à-exécuter, car une telle conversion n'est pas une opération automatique qui peut facilement se ramener à un algorithme unique. N'oublions pas qu'il ne s'agit pas de recopier un fichier sous un format différent, mais également de le modifier pour qu'il soit adapté à son nouvel environnement.

Certains programmes publiés par Pom's serviront d'exemple et de référence... auto-publicité exige! Enfin, la démarche à appliquer sera volontairement détaillée car elle s'adresse également aux lecteurs qui connaissent peu l'assembleur ; certaines recommandations pourraient donc paraître superflues.

Avant de se précipiter sur son clavier, il n'est pas inutile de se poser quelques questions sur le programme à adapter :

- est-il utile ?
- est-il adaptable ?
- où se loge-t-il ?
- comment est-il utilisé ?
- comment se protège-t-il ?

Le programme est-il utile ?

Une routine peut être fort utile pour surmonter certains handicaps du DOS 3.3, mais se révéler en revanche de peu d'attrait avec ProDOS.

C'est le cas des programmes destinés à accélérer les temps de chargement de fichiers ; ceux qui relogent le DOS sur la carte langage, ou tous les utilitaires tournant dans un environnement DOS 3.3 + Integer Basic.

Idem pour le programme FRE16 de nettoyage de la mémoire (Pom's 2), désormais intégré dans la commande FRE de ProDOS.

Le programme est-il adaptable ?

Une routine qui supposait comme principe essentiel de fonctionnement la présence du DOS 3.3 ne pourra pas être adaptée à ProDOS sans une révision générale.

En effet, les appels du File Manager ou RWTS d'une part, du ProDOS MLI d'autre part, ont des structures totalement différentes. En pratique, l'examen du source du programme donne une sérieuse indication de sa dépendance vis à vis du DOS 3.3 : toute référence à une étiquette externe située dans la plage d'adresses \$9600-\$BDFE constitue normalement une forte présomption.

Ainsi, le HELLO complet de Pom's 6 ne peut pas être adapté en quelques minutes à ProDOS ; il serait préférable de le réécrire complètement. De même pour les programmes de francisation du DOS (Pom's 2 et 8) : les formats sont différents sous ProDOS et, de plus, les adresses absolues des commandes et libellés d'erreur varient pour chaque version.

Où le programme se loge-t-il ?

Sous DOS 3.3, les programmes en assembleur pouvaient se loger pratiquement n'importe où ! Sous ProDOS, certains emplacements sont à éliminer, en particulier :

- le buffer d'entrée (\$200-\$2FF),
- les adresses \$9A00-\$9CFF (anciennement les buffers du DOS 3.3),
- la carte langage, dans son intégralité.

La seule exception concerne la routine d'accès à une carte horloge, pour laquelle un petit emplacement est réservé, moyennant l'adhésion à un protocole rigoureux d'interface avec ProDOS.

Trois emplacements sont généralement classiques pour placer les routines :

- La page 3 (adresses \$300-\$3CF) est laissée disponible. Dans ce cas, vous n'aurez probablement aucun problème.
- En revanche, les emplacements situés sous les buffers du DOS 3.3 (limite supérieure en \$9600), ainsi que ceux situés un peu plus haut et pour lesquels on déplaçait les buffers du DOS sous la routine, vont nécessiter un déplacement du programme. Sinon on obtient généralement, tôt ou tard, le message 'NO BUFFERS AVAILABLE' de ProDOS. Nous verrons plus loin comment reloger un programme en assembleur.
- Enfin, le dernier emplacement favori - cacher la routine au sein d'un programme Applesoft (pour les détails, lire Pom's 12 page 35) - est totalement transparent par rapport au système d'exploitation utilisé, et ne pose aucun problème.

Comment le programme est-il utilisé ?

Toute routine en assembleur est appelée soit directement, soit indirectement.

La méthode directe consiste à demander son exécution par un CALL vers une adresse déterminée. Si le programme a été relogé, il ne faudra donc pas oublier d'indiquer la nouvelle adresse au programme appelant.

L'autre méthode utilise indifféremment l'ampersand (&), l'instruction USR de l'Applesoft ou le Control-Y du moniteur. Ces vecteurs sont initialisés soit par le programme appelant, soit par le programme appelé (et relogé) ; ici aussi, il faudra modifier les valeurs d'initialisation (si possible en créant un chaînage car plusieurs routines peuvent utiliser le même vecteur).

Comment le programme est-il protégé ?

Dans l'environnement ProDOS, les pages d'adresses mémoire (blocs de 256 octets) sont marquées libres ou occupées. ProDOS utilise les pages libres pour s'allouer dynamiquement les buffers nécessaires aux accès disque.

En particulier, ProDOS refusera de charger un programme assembleur vers un emplacement occupé et, inversement, pourrait l'écraser si celui-ci n'a pas indiqué les pages qu'il utilise.

Le programme RAM.BITMAP de Guy d'HERBEMONT permet d'illustrer ce processus. Il affiche la carte des pages mémoire disponibles (pages \$00 à \$BF) en indiquant pour chacune si elle est occupée ou libre.

Après le chargement de ProDOS + Basic.System, on obtient :

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
\$00	P	P	°	°	P	P	P	P	°	°	°	°	°	°	°	°
\$10	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
\$20	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
\$30	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
\$40	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
\$50	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
\$60	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
\$70	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
\$80	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
\$90	°	°	°	°	°	°	°	°	°	°	P	P	P	P	P	P
\$A0	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
\$B0	P	P	P	P	P	P	P	P	P	P	°	°	°	°	P	P

P = protégé ° = libre

On note en particulier:

- Les pages 0, 1 (pile du 6502), 4 à 7 (écran texte) sont protégées en permanence. Un programme assembleur peut, cependant, les utiliser; en particulier les adresses en page zéro qui sont disponibles sous DOS 3.3 le restent sous ProDOS.

- La page 2 (buffer d'entrée) est libre car ProDOS l'utilise (il recopie la commande en cours pour en extraire les paramètres, et y formate également le chemin d'accès complet à un fichier).

- La page 3 est également libre ; si vous y placez une routine en assembleur, ProDOS pourra donc l'écraser par le chargement d'une autre routine.

- Himem (\$73-74) pointe vers \$9600. ProDOS en conclut que les 4 pages suivantes (\$96 à \$99) correspondent à son buffer de brouillon général. Ces pages ne doivent pas être utilisées.

- Les pages suivantes sont occupées (les pages \$BA à \$BD servent de zones de travail au Basic.System).

Notez que l'on obtient le message "NO BUFFERS AVAILABLE" si on tente de charger un fichier directement sur une page protégée. Or ProDOS gère dynamiquement l'espace mémoire de l'Apple pour s'allouer des buffers de fichiers (à la différence du DOS 3.3 qui, en fonction du MAXFILES, se réservait définitivement de l'espace).

Trouve-moi une place où me loger

Observons la séquence à l'ouverture d'un fichier :

- ProDOS abaisse Himem de 1Ko (\$400 octets) et déplace les variables chaînes de caractères de l'Applesoft vers le bas (déplacement physique des chaînes et mise à jour de leur pointeur) ;

- Son buffer brouillon débute donc maintenant en \$9200, le buffer du fichier ouvert est placé \$400 octets plus haut, en \$9600 (jusqu'en \$99FF).

Cette séquence est répétée pour chaque nouveau fichier ouvert, jusqu'à concurrence de 8 fichiers, ce qui porte Himem à \$7600. Pour chaque fermeture, elle s'exécute à nouveau mais en sens inverse.

Il n'est pas possible de placer nos programmes assembleur juste en dessous de \$7600 (quelle perte de place!), car ceux-ci seraient écrasés sans pitié par les chaînes de caractères de l'Applesoft déplacées par ProDOS.

En revanche, une solution beaucoup plus élégante existe : le placer entre ProDOS et ses buffers.

Supposons que notre programme soit assemblé de \$94A6 jusqu'en \$95FF. En comptant en nombre de pages, il en occupe donc 2. Si nous le plaçons en \$98A6, il occupera donc l'espace jusqu'en \$99FF. Simultanément, abaissons Himem de 2 pages en \$9400. Comme ProDOS détermine la position de son buffer brouillon en fonction de Himem, il en conclura que celui-ci se situe en \$9400-\$97FF. Par la suite, les ouvertures de fichiers pourront s'effectuer sans problème vers le bas.

Deux problèmes restent à résoudre:

- d'une part, notre programme ne peut pas être chargé directement à son adresse définitive car il serait alors chargé dans son propre buffer, ce que ProDOS refuse. Il faut donc prévoir une petite routine qui le charge à une adresse plus basse, puis le reloge.

- d'autre part, son chargement ne pourrait s'effectuer que si aucun autre fichier n'est ouvert, et les chaînes de caractères qu'aurait pû créer un programme Applesoft seront détruites. Heureusement, une routine publique de ProDOS permet - nous le verrons - de gérer ce point.

Déplacement des programmes

Résumons-nous : les petites routines en assembleur, celles qui se logent sur la page 3 ou se cachent dans un programme basic, ne posent pas de problèmes majeurs.

En revanche, les gros programmes - donc les plus difficiles à recréer soi-même -, qui se logent autour des buffers du DOS 3.3 devront être placés à une autre adresse.

Malheureusement, quiconque a observé un module en assembleur sait que l'on ne peut pas impunément les déplacer à travers la mémoire. Le résultat, quasi-assuré, est un magnifique plantage. Sculs, quelques programmes sont écrits sous une forme dite relogeable... mais ils sont bien rares car ils nécessitent soit des acrobaties de programmation pour éliminer les JMP, JSR et autres instructions à adresse fixe, soit la nécessité d'utiliser un assembleur précis et une routine spéciale de chargement (l'EDASM et le module RLOAD d'Apple par exemple).

Il est donc nécessaire, dans la très grande majorité des cas, de réassembler soi-même la routine vers sa nouvelle adresse d'implantation. Comme ProDOS ne supporte que les allocations de mémoire par frontière de page, il faut se simplifier la vie en fixant la nouvelle adresse selon la formule :

[nouvelle adresse] =
\$9A00 - ((partie haute de l'adresse de fin) + 1 - [partie haute de l'adresse de début]) * \$100
+ [partie basse de l'adresse de début]

Ainsi, un programme situé en \$94A6-\$9596 se loge en :

$\$9A00 - (\$95 - \$94 + 1) * \$100 + \$A6 = \$98A6$

Notons que l'on pourrait envisager de faire cohabiter simultanément plusieurs programmes assembleur sur des pages contiguës. Mais il faudrait alors les réassembler après leur chargement (puisque l'on ne peut connaître à l'avance leur adresse finale d'implantation), ce qui sort des limites de cet article.

Pour réassembler notre programme, deux méthodes sont alors possible :

1) La méthode dite "appel à Murphy" qui consiste, lorsque l'on ne dispose pas d'assembleur, à reloger la routine à la main. Il faut prendre en compte trois aspects :

```
DEBUT      JMP (VECTEUR1) ;  
           instruction à trois octets :  
           modifier le dernier  
VECTEUR1 DA ROUTINE1 ;  
           table d'adresse :  
           modifier le 1er octet  
ROUTINE1 LDA #>DEBUT ;  
           valeur - dépendante d'une  
           adresse - à modifier
```

Comme l'indique son nom, cette méthode, conduite sur de gros modules, amène à coup sûr à un plantage. En désespoir de cause, consultez la routine RELOCATE de Pom's 1.

2) Plus raisonnablement, il faut prendre un assembleur, charger le fichier source, modifier simplement (dans la plupart des cas) la directive ORG qui fixe l'adresse de début du module objet, réassembler et sauver sur disque. Vous n'avez pas l'assembleur adéquat ? La plupart des assembleurs permettent de sauver les sources sous le format des fichiers TEXT classiques ; de plus Pom's a publié :

- les équivalences de directives entre assembleurs (Pom's 9) ;
- une routine de conversion Big Mac/Lisa 2.5 (Pom's 10).

Routine d'initialisation

Vous avez donc sauvé sur disque le nouveau module objet au format ProDOS. Il reste à réaliser la routine de chargement et d'initialisation de ce module. Nous avons pu déterminer plus haut l'algorithme à utiliser :

1) Chargement à partir du disque vers la zone mémoire pointée par Lomem (\$6D-6E) : ainsi nous minimisons le risque d'écraser des variables de l'Applesoft.

2) Appel de la routine GETBUFR du Basic.System (version 1.1 et suivantes) avec l'accumulateur en

entrée contenant le nombre de pages désirées. Si la retenue est à 1 en sortie : il n'y a pas de place disponible et on arrête. Sinon, la place demandée a été réservée et Himem a été mis à jour par le Basic.System (en corollaire, le programme ne doit pas lui-même manipuler Himem: éliminez donc toute référence aux adresses \$73 et \$74).

3) Recopie du module vers sa destination finale.

4) Eventuellement, mise à jour de la carte des pages RAM de ProDOS afin que le programme ne soit pas écrasé par inadvertance.

5) Enfin, appel éventuel du module (si celui-ci se lance habituellement par BRUN).

Qui dit algorithme, dit programme. Il s'appelle - bien sûr - DARWIN puisqu'il permet de s'adapter à un nouvel environnement...

Et maintenant la question à cent sous : où va-t-on le placer?

Il n'y a guère de choix puisque la plupart des emplacements possibles ne peuvent être supposés disponibles : notre programme d'initialisation devra se loger entre Lomem et Himem. En pratique nous allons assembler simultanément DARWIN et le programme à adapter, de manière à obtenir un unique module objet.

Méthode d'utilisation

Afin d'illustrer la façon d'utiliser DARWIN, nous allons utiliser en exemple un des programmes de tri publiés par Pom's :

TRITABLEAU (Pom's 11, p 47).

Un premier examen du source indique que ce programme n'est pas dépendant de l'environnement DOS 3.3 sous lequel il a été conçu à l'origine. Seule la présence de l'Applesoft est nécessaire.

L'assembleur du DOS Tool Kit, utilisé à l'origine, ne supporte pas la notion de fichiers source

chaînés avec laquelle un source peut être scindé en plusieurs fichiers distincts. Nous utiliserons en conséquence l'assembleur Big Mac.

Première étape : le fichier source est renommé T.TRITABLEAU afin de pouvoir être lu par Big Mac. Par un assemblage sous la version DOS 3.3, on remarque que le module occupe normalement les adresses \$9AA6 à \$9C3A. Profitons-en pour corriger une erreur du source TRITABLEAU publié dans Pom's : les lignes 30 et 32 doivent être interverties.

Deuxième étape : La ligne 28 du source, qui contient la directive OBJ, doit maintenant être supprimée : l'adresse de début d'assemblage sera déterminée par DARWIN. C'est la seule modification nécessaire pour TRITABLEAU. Le source est à

nouveau stocké sur disque sous forme d'un fichier TEXT.

Troisième étape : on charge sous Big Mac le source DARWIN.S ; la dernière ligne doit être modifiée pour contenir le nom du programme à adapter, en l'occurrence TRITABLEAU.

Quatrième étape : placer la disquette contenant le fichier T.TRITABLEAU dans le lecteur courant. Demandons un assemblage : aussitôt, l'utilisateur doit indiquer la valeur des variables DARWIN_D et DARWIN_F et VA_DEBUT ; DARWIN_D représente l'adresse de début sous DOS 3.3, en l'occurrence \$9AA6. DARWIN_F est l'adresse de fin, donc \$9C3A. VA_DEBUT indique si l'on doit ou non sauter au programme après son chargement ; comme TRITABLEAU est normalement chargé par un BRUN sous DOS

Programme TRITABLEAU.DEMO

```

1  REM TRITABLEAU.DEMO A. Avrane
10 D$ = CHR$(4)
15 PRINT D$"BRUNTRITABLEAU,A"256 * ( PEEK (110)
   + 5): REM laisse au moins $400 octets libre
   s
20 TEXT : HOME : PRINT D$"PR#3"
25 PRINT "DEMONSTRATION DE TRI RAPIDE SOUS PROD
   OS"
30 DIM A$(20),A(20)
35 INVERSE : PRINT "Indice- Element": NORMAL
40 FOR I = 1 TO 20
50 FOR J = 1 TO 12
60 A$(I) = A$(I) + CHR$(65 + 25 * RND(1))
70 NEXT :A(I) = I
80 GOSUB 900: NEXT
100 & T,0,A$(1),A$(20),A(1)
105 HTAB 20: VTAB 2: INVERSE : PRINT "Tri sur 1
   er octet": NORMAL
110 FOR I = 1 TO 20
120 HTAB 20: VTAB I + 2
140 GOSUB 900: NEXT
200 & S,12,0,A$(1),A$(20),A(1)
205 HTAB 40: VTAB 2: INVERSE : PRINT "Tri sur d
   ernier octet": NORMAL
210 FOR I = 1 TO 20
220 HTAB 40: VTAB I + 2
230 GOSUB 900: NEXT
240 END
900 PRINT RIGHT$( "0" + STR$(A(I)),2) "- "A$(
   I): RETURN

```

3.3, il faut donner la valeur 1 ; et la valeur 0 pour un BLOAD. La suite de l'assemblage charge en overlay le source T.TRITABLEAU et assemble les deux sources. Il ne reste plus qu'à sauver sur disque ce module objet sous le nom TRITABLEAU.

Cinquième étape : modifier l'appel à TRITABLEAU du programme Applesoft qui l'utilise. Le format d'appel devient :

```
PRINT CHR$(4) "BRUN
TRITABLEAU, A" 256 *
(PEEK(110) + 5)
```

et charge le module dans une zone libre, au dessus des variables numériques de l'Applesoft et d'un buffer de \$400 octets utilisé par GETBUFR.

Fonctionnement de DARWIN

L'essentiel a déjà été dit du fonctionnement de DARWIN. On peut cependant remarquer que celui-ci est écrit sous Big Mac et en utilise les directives KBD (saisie de la valeur d'une variable

par l'utilisateur) et PUT (lecture d'un fichier source). Ces directives sont transposables sans difficultés pour Lisa, EDASM ProDOS, etc.

La version actuelle suppose un appel au programme relogé afin de l'initialiser. Dans le cas contraire, il suffit de placer un RTS au lieu du JMP indirect final. DARWIN est bien entendu entièrement relogeable puisque l'on ignore son adresse d'implantation. Enfin, la carte des pages mémoire protégées n'est pas mise à jour : cela n'est nécessaire que si l'on craint une interaction avec d'autres programmes assembleur.

A l'exécution, le message PROGRAM TOO LARGE est généré si le Basic.System utilisé sous ProDOS est antérieur à la version 1.1, s'il est impossible de trouver suffisamment de place en mémoire vive pour reloger le programme, ou enfin si on tente d'exécuter DARWIN sous DOS 3.3.

Un petit programme de démonstration montre les tris d'un tableau de chaînes de caractères,

avec la clé en première ou dernière position de chaque chaîne. Reportez-vous à Pom's 11 pour les syntaxes d'appel à la routine.

La disquette Pom's contient les fichiers suivants :

DARWIN.S
source en format Big Mac
T.DARWIN
source en format TEXT pour conversion vers d'autres assembleurs
T.TRITABLEAU
source en format TEXT (appelé par l'assemblage de DARWIN.S)
TRITABLEAU
module exécutable sous ProDOS
TRITABLEAU.DEMO
programme Applesoft de démonstration (en 80 colonnes).



Programme DARWIN.S (Assembleur BigMac)

NB : Ce programme figure sur la disquette d'accompagnement également sous format TEXT. Il est nommé T.DARWIN

```
1 *****
2 *          DARWIN          *
3 *****
4
5 * Routine d'initialisation à ProDOS
6 * d'un programme assembleur DOS 3.3
7
8 * (C) 1985 Alexandre Avrane
9 * Modifié: 24/11/85
10 * Créé: 16/11/85
11
12 * Ce source est assemblé en conjonction
13 * avec le programme à adapter ProDOS
14
15 DARWIN_D KBD          ;adr début DOS 3.3
16 DARWIN_F KBD          ;adr fin DOS 3.3
17 VA_DEBUT KBD          ;saute en début de
                           pgm après
                           chargement?
18
19 * Calcul des adresses sous ProDOS
20 DARWINV1 - DARWIN_F/$100+1
```

```
21 DARWINV2 - DARWIN_D/$100
22 DARWINV3 - DARWINV1-DARWINV2
23 DARWINV4 - $100*DARWINV2
24 DARWINV5 - DARWIN_D-DARWINV4
25 DARWINV6 - DARWINV3*$100-DARWINV5
26
27 DARWIN_A - $9A00-DARWINV6 ;ad début ProDOS
28 DARWIN_L = DARWIN_F-DARWIN_D+1 ; lg ProDOS
29
30          ORG DARWIN_A-73 ;DARWIN - 72 octets
31
32 * -----
33 * 1 - Recherche de la place mémoire
34 * -----
35          LDA $RFFD          IVERSION
36          BEQ DARWIN10      Basic.System 1.0
37          LDA #0-DARWIN_A/$100+$9B
38          JSR $BEF5          GETBUFR
39          BCC DARWIN11
40 DARWIN10 LDA #14          PROGRAM TOO LARGE
41          JMP $BE09
42 DARWIN11 = *
43
44 * -----
45 * 2 - Copie vers l'adresse d'implantation
46 * -----
47 DARWIN20 JSR $FF58          contient un RTS
48          TSX
49          DEX
50          CLC
```

```

51      LDA $100,X      recherch sur la pile
52      ADC #DARWIN99-DARWIN20-2 ;adresse d
                    ébut routine
53      STA $3C          A1
54      LDA $101,X
55      ADC #0
56      STA $3C+1
57      LDA $3C          adresse de fin
58      ADC #<DARWIN_L
59      STA $3E          A2
60      LDA $3C+1
61      ADC #>DARWIN_L
62      STA $3E+1
63      LDA #<DARWIN_A  adresse d'arrivée
64      STA $42          A4
65      PHA
66      LDA #>DARWIN_A
67      STA $42+1
68      PHA
69      LDY #0
70      JSR $FE2C        déplace le bloc
                    par MOVE
71
72 * -----
73 * 3 - Appel du module et sortie
74 * -----
75      PLA
76      STA $42+1
77      PLA
78      STA $42
79      DO VA_DEBUT
80      JMP ($42)        saute en début de
                    programme...
81      ELSE
82      RTS              ...ou retour à
                    l'Applesoft
83      NOP
84      NOP
85      FIN
86 DARWIN99 = *
87 * Inclut le source du programme à adapter
                    (format TEXT)
88      PUT TRITABLEAU
89      END

```

Programme T.TRITABLEAU (Assembleur BigMac)

NB : Sauvegarder ce source sous format TEXT (Voir texte)

```

1 FLAG EQU $6          * Flag réel.entier
2 IND EQU $9           * Indice du tri
3 ECART EQU $7         * Ecart entre
                    articles
4 ECO EQU $8           * Ecart indice 0
                    et tri
5 START EQU $19        * Pointeur 1er
                    élément du tri
6 FIN EQU $1B          * Pointeur dernier
                    élément
7 ART1 EQU $FA         *Pointeur temporaire
8 ART2 EQU $FC         *Pointeur temporaire
9 CHA2 EQU $FE         *Pointeur temporaire
10 MOVE EQU $1E        *Pointeur temporaire
11 CSTART EQU $E0      *Pointeur 1er
                    élément COTableau
12 CART1 EQU $E2       * Pointeur COTableau
13 CART2 EQU $E4       * Pointeur COTableau
14 TECART EQU $1D      * Ecart 1er tableau
15 TECO EQU $E6        * Ecart élément 1er
                    tableau
16 CECART EQU $E7      * Ecart COTableau
17 PARTIEL EQU $E8
18 SARA EQU $9B        * Pointeur du
                    tableau
19 CHRGET EQU $B1
20 CHRGOT EQU $B7
21 FRMNUM EQU $DD67    * Evalue dans FAC
22 CHKCOM EQU $DEBE    * Teste ','
23 MOVFM EQU $EAF9     * Transfert M vers
                    FAC
24 SOUST EQU $EBB2     * FAC - ARG
25 GETADR EQU $E752    * Rend entier FAC
26 PTRGET EQU $DFE3
27 * ORG $9AA6 indiqué à l'assemblage de DARWIN
28 * -----Initialisation AMPERSAND-----
29      LDA #<ST
30      STA $3F7
31      LDA #>ST
32      STA $3F6
33      RTS
34 * -----Début du programme
35 ST CMP #$54          * 'T'
36 BEQ TRI
37 CMP #$53            * 'S'
38 BEQ SORT
39 JMP $DEC9           * SYNTAX ERROR
40 SORT JSR CHRGET
41 JSR CHKCOM
42 JSR FRMNUM
43 JSR GETADR
44 DEY
45 DEY
46 STY PARTIEL
47 JMP TS
48 TRI LDA #$FF
49 STA PARTIEL
50 JSR CHRGET
51 TS JSR CHKCOM
52 JSR FRMNUM
53 JSR GETADR          * Rend FAC entier
54 STY IND             * Indice du tri
55 JSR CHKCOM
56 JSR PTRGET
57 STA START
58 STA ART1
59 STY START+1
60 STY ART1+1
61 JSR CHKCOM
62 JSR PTRGET
63 STA FIN
64 STY FIN+1
65 JSR CALCUL          * Initialise
                    pointeurs
66 LDA ECO
67 STA TECO
68 LDA ECART
69 STA TECART
70 STX FLAG
71 JSR CHRGOT
72 CMP #$2C            * COTABLEAU ?
73 BNE DEBUT           * Non
74 JSR CHRGET
75 JSR PTRGET
76 STA CSTART
77 STA CART1
78 STY CSTART+1
79 STY CART1+1
80 JSR CALCUL
81 LDA ECART
82 STA CECART
83 * -----Début du tri-----

```

```

84 DEBUT LDA #0
85 STA IND * Flag d'échange
86 SUIVANT CLC * Calcul adresse
      2ème article
87 LDA TECART
88 ADC ART1
89 STA ART2
90 LDA #0
91 ADC ART1+1
92 STA ART2+1
93 CLC * Calcul adresse
2ème article
94 LDA CECART
95 ADC CART1
96 STA CART2
97 LDA #0
98 ADC CART1+1
99 STA CART2+1
100 LDA FLAG
101 CMP #5 * Réel ?
102 BEQ REEL * Oui
103 TAY
104 DEY
105 CMP #2 * Entier ?
106 BEQ ENTIER * OUI
107 LDA (ART1),Y
108 STA MOVE+1
109 LDA (ART2),Y
110 STA CHA2+1
111 DEY
112 LDA (ART1),Y
113 STA MOVE
114 LDA (ART2),Y
115 STA CHA2
116 LDY PARTIEL
117 COMP INY
118 LDA (CHA2),Y
119 CMP (MOVE),Y
120 BCC DEPL * CHA2 < CHA1
121 BNE RIEN * CHA2 > CHA1
122 CPY #15 * Profondeur du tri
123 BCC COMP
124 BCS RIEN
125 ENTIER SEC
126 LDA (ART2),Y
127 SBC (ART1),Y
128 DEY
129 LDA (ART2),Y
130 SBC (ART1),Y
131 BPL RIEN * ART2 = ou > ART1
132 BMI DEPL * ART2 < ART1
133 * -----Recommence à l'article 1-----
134 BOUCLE LDA CSTART * 1er article
135 STA CART1
136 LDA CSTART+1
137 STA CART1+1
138 LDA START * 1er article
139 STA ART1
140 LDA START+1
141 STA ART1+1
142 BNE DEBUT
143 * -----Article suivant-----
144 SUITE LDA CART2 * Article suivant
145 STA CART1
146 LDA CART2+1
147 STA CART1+1
148 LDA ART2 * Article suivant
149 STA ART1
150 LDA ART2+1
151 STA ART1+1
152 BNE SUIVANT
153 * -----Test de fin du tri-----
154 RIEN LDA ART2

```

```

155 CMP FIN
156 BNE SUITE
157 LDA ART2+1
158 CMP FIN+1
159 BNE SUITE
160 LDA IND
161 BNE BOUCLE
162 RTS
163 * -----
164 REEL LDA ART2
165 LDY ART2+1
166 JSR MOVFM * ART2 dans FAC
167 LDA ART1
168 LDY ART1+1
169 JSR SOUST * ART2 - ART1
170 BPL RIEN * ART2 = ou > ART1
171 * -----Déplacement des pointeurs-----
172 DEPL SEC * Calcul élément 0
      de ART1
173 LDA ART1
174 SBC TECO
175 STA ART1
176 LDA ART1+1
177 SBC #0
178 STA ART1+1
179 SEC * Calcul élément 0
      de ART2
180 LDA ART2
181 SBC TECO
182 STA MOVE

```

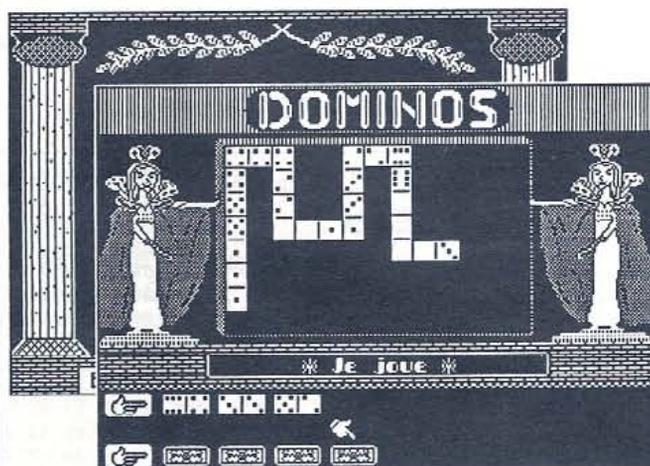
Pom's vous propose

"Dominos"

Thierry Haurie

Apple][+, IIe, IIc

Il est inutile de présenter le jeu de dominos; celui-ci bénéficie d'un graphisme très soigné (en couleur si vous disposez d'une carte "Chat Mauve") et les messages transmis par le programme sont, au choix, en Français, en Italien, en Allemand ou en Anglais.



80.00 F TTC franco
Bon de commande page 74

183	LDA	ART2+1		215	BPL	DIM1	* Oui
184	SBC	#0		216	LDX	#\$2	
185	STA	MOVE+1		217	LDY	#\$4	* Dimension ?
186	LDY	TECART		218	LDA	(SARA),Y	
187	DEY			219	CMP	#\$1	* Une seule ?
188	STA	IND	* Positionne flag	220	BEQ	D1	* Oui
189	SWAP	LDA	(ART1),Y	221	LDY	#\$8	
190	PHA			222	LDA	(SARA),Y	* Taille lère
191	LDA	(MOVE),Y				dimension	
192	STA	(ART1),Y	* échange 2 dans 1	223	D1	TAY	
193	PLA			224	LDA	#0	
194	STA	(MOVE),Y	* échange 1 dans 2	225	STA	ECART	
195	DEY			226	STA	ECO	
196	BPL	SWAP		227	STA	CECART	
197	LDY	CECART		228	CLC		* Calcul écart entre
198	BEQ	RIEN				articles	
199	DEY			229	D2	TXA	* 5 . 3 . 2
200	SWAP1	LDA	(CART1),Y	230	ADC	ECART	
201	PHA			231	STA	ECART	
202	LDA	(CART2),Y		232	DEY		
203	STA	(CART1),Y	* échange 2 dans 1	233	BNE	D2	
204	PLA			234	LDY	IND	
205	STA	(CART2),Y	* échange 1 dans 2	235	BEQ	D4	
206	DEY			236	CLC		* écart élément 0 et
207	BPL	SWAP1				tri	
208	BMI	RIEN		237	D3	TXA	* 5 . 3 . 2
209	*	-----	Sous programme recherche pointeurs	238	ADC	ECO	
210	CALCUL	LDX	#\$3	239	STA	ECO	
211		BIT	#\$11 * Chaine ?	240	DEY		
212		BMI	DIM1 * Oui	241	BNE	D3	
213		LDX	#\$5	242	D4	RTS	
214		BIT	#\$1 * Réel ?				

Programme RAM.BITMAP

```

60005 TEXT : HOME :Y = 48984: REM
      $BF58:BITMAP
60006 P = 0:L = 0: VTAB 9: HTAB 1:
      A$ = "0020406080A0": FOR I =
      1 TO 11 STEP 2: PRINT "$" M
      ID$ (A$, I, 2): NEXT : VTAB 1
60007 POKE 33,32: POKE 32,4: PRIN
      T "PAGES MEMOIRE PROTEGEES (
      PRODOS)": PRINT "sur 192 pag
      es (48K de $00 à $BF)": PRIN
      T : PRINT SPC( 16)"11111111
      111111110123456789ABCDEF0123

```

```

456789ABCDEF"
60010 FOR I = 0 TO 23:X = PEEK (
      Y + I): REM LES 24 OCTETS D
      E LA BIT-MAP
60020 FOR J = 1 TO 8: REM LES 8 B
      ITS DE CHAQUE OCTET
60030 IF X > = 128 THEN X = 2 *
      X - 256:P = P + 1: PRINT "P"
      ;: GOTO 60050: REM CAS BIT=1
60040 X = X * 2:L = L + 1: PRINT "
      .";: REM CAS BIT=0
60050 NEXT J: NEXT I: POKE 33,40:
      POKE 32,0: PRINT
60055 PRINT "P = protégée ; au to
      tal : "P: PRINT ". = libre
      ; au total : "L: PRINT

```

Récapitulation TRITABLEAU

4000- AD FD BF F0 07 A9 03 20	4088- E3 DF 85 19 85 FA 84 1A	4130- 94 A5 E4 85 E2 A5 E5 85
4008- F5 BE 90 05 A9 0E 4C 09	4090- 84 FB 20 BE DE 20 E3 DF	4138- E3 A5 FC 85 FA A5 FD 85
4010- BE 20 58 FF BA CA 18 BD	4098- 85 1B 84 1C 20 01 99 A5	4140- FB D0 86 A5 FC C5 1B D0
4018- 00 01 69 36 85 3C BD 01	40A0- 08 85 E6 A5 07 85 1D 86	4148- E8 A5 FD C5 1C D0 E2 A5
4020- 01 69 00 85 3D A5 3C 69	40A8- 06 20 B7 00 C9 2C D0 15	4150- 09 D0 CC 60 A5 FC A4 FD
4028- 95 85 3E A5 3D 69 01 85	40B0- 20 B1 00 20 E3 DF 85 E0	4158- 20 F9 EA A5 FA A4 FB 20
4030- 3F A9 A6 85 42 48 A9 97	40B8- 85 E2 84 E1 84 E3 20 01	4160- B2 EB 10 DF 38 A5 FA E5
4038- 85 43 48 A0 00 20 2C FE	40C0- 99 A5 07 85 E7 A9 00 85	4168- E6 85 FA A5 FB E9 00 85
4040- 68 85 43 68 85 42 6C 42	40C8- 09 18 A5 1D 65 FA 85 FC	4170- FB 38 A5 FC E5 E6 85 1E
4048- 00 A9 B1 8D F6 03 A9 97	40D0- A9 00 65 FB 85 FD 18 A5	4178- A5 FD E9 00 85 1F A4 1D
4050- 8D F7 03 60 C9 54 F0 1A	40D8- E7 65 E2 85 E4 A9 00 65	4180- 88 85 09 B1 FA 48 B1 1E
4058- C9 53 F0 03 4C C9 DE 20	40E0- E3 85 E5 A5 06 C9 05 F0	4188- 91 FA 68 91 1E 88 10 F3
4060- B1 00 20 BE DE 20 67 DD	40E8- 6B A8 88 C9 02 F0 22 B1	4190- A4 E7 F0 AF 88 B1 E2 48
4068- 20 52 E7 88 88 84 E8 4C	40F0- FA 85 1F B1 FC 85 FF 88	4198- B1 E4 91 E2 68 91 E4 88
4070- D6 97 A9 FF 85 E8 20 B1	40F8- B1 FA 85 1E B1 FC 85 FE	41A0- 10 F3 30 9F A2 03 24 11
4078- 00 20 BE DE 20 67 DD 20	4100- A4 E8 C8 B1 FE D1 1E 90	41A8- 30 08 A2 05 24 81 10 02
4080- 52 E7 84 09 20 BE DE 20	4108- 5B D0 38 C0 0F 90 F3 B0	41B0- A2 02 A0 04 B1 9B C9 01
	4110- 32 38 B1 FC F1 FA 88 B1	41B8- F0 04 A0 08 B1 9B A8 A9
	4118- FC F1 FA 10 26 30 45 A5	41C0- 00 85 07 85 08 85 E7 18
	4120- E0 85 E2 A5 E1 85 E3 A5	41C8- 8A 65 07 85 07 88 D0 F8
	4128- 19 85 FA A5 1A 85 FB D0	41D0- A4 09 F0 09 18 8A 65 08
		41D8- 85 08 88 D0 FB 60

Compacteur de programmes

Sylvie Gallet

Le programme proposé ici est écrit en langage machine pour des raisons évidentes de rapidité et de facilité d'emploi. Il occupe très peu de place en mémoire (moins de 3 pages) et il se charge avec RENUMBER car il utilise aussi l'ampersand et parce qu'ils semblent faire la paire !

Bien entendu, ce programme tient compte de tous les GOTO, GOSUB, ONERR GOTO et autres IF... THEN. Il en profite au passage pour supprimer les REMs, et raccourcir les noms des variables mais il ne raccourcit aucune instruction &, CALL ou DATA : en effet, la syntaxe de ces instructions est elle-même très variable !

Les lignes ainsi obtenues peuvent contenir jusqu'à \$F8 (248) caractères ou tokens ce qui déplaît fortement à RENUMBER et à P.L.E. Sur la plupart des programmes que j'ai compactés, j'ai noté un gain de place d'environ 10 à 15 % avec des résultats spectaculaires sur des programmes très documentés (Gescompte...)

Si une ligne ne contenant qu'une REM est adressée par un GOTO, GOSUB,... il conserve ligne avec seulement un ".". Il est possible qu'un deuxième compactage

donne un résultat différent du premier : en effet la longueur "utile" des lignes est déterminée avant de raccourcir les variables. Ceci peut se produire si les noms des variables sont très longs.

Un autre détail involontaire, il ne touche pas à une éventuelle ligne 65535, j'ai préféré laisser faire.

Utilisation

Pour le charger, faire "BRUN COMP+RENUM". Il se place en \$8B00, il descend HIMEM et vectorise l'ampersand. Pour compacter, faire simplement : "&COMPACTE".

Comment faire ?

Pour utiliser le compacteur, il suffit de faire :
"BRUN COMP+RENUM"

puis pour compacter :
&COMPACTE

Pour obtenir le fichier COMP+RENUM, vous devrez suivre les étapes suivantes :

- Si vous n'avez pas la disquette Pom's, saisir sous moniteur la récapitulation COMP, puis la sauvegarder (BSAVE COMP,A\$8B00,L\$2FF).
- Exécuter le programme FAIT EXEC : celui-ci crée un fichier EXEC qui va récupérer la partie binaire du programme RENUMBER de la disquette Maître DOS 3.3.
- Faites alors EXEC CREE COMP+RENUM.
- On vous demande successivement de mettre la disquette Maître contenant RENUMBER, puis la disquette contenant le fichier COMP, et enfin la disquette qui recevra le fichier final COMP+RENUM.

Fonctionnement

Il commence par faire un tableau de toutes les lignes appelées par GOTO, GOSUB et IF...THEN GOTO. Il procède ensuite au compactage des lignes après avoir exploré chaque ligne pour savoir si elle contient un IF, si elle est adressée par GOTO, et ses acolytes, et pour connaître sa longueur "utile" (c'est-à-dire sans les REMs).

Il termine en mettant à jour le pointeur de fin de programme.

Il faut noter que dans le cas où le programme n'est pas chargé comme à l'accoutumée en \$801 (2049), le fonctionnement du compacteur n'est pas perturbé.

Remarque : Il a fallu faire une petite rustine à CHRGET pour qu'elle ne saute pas d'office tous les \$20 qu'elle rencontre, \$20 se trouvant aussi bien dans les adresses de lignes que dans les numéros de lignes.

Le lecteur pourra certainement apporter les petites modifications nécessaires pour obtenir un programme encore plus court : lorsque la ligne adressée par un GOTO ou un GOSUB ne comprend qu'une REM, pourquoi ne pas se brancher sur la ligne suivante par exemple ?



Programme

FAIT EXEC

```
10 TEXT : HOME : SPEED= 255:D$ = CH
R$ (4)
100 VTAB 12: PRINT " CREATION DU FI
CHIER CREE COMP+RENUM"
120 NOM$ = "CREE COMP+RENUM": PRINT D
$"OPEN"NOM$: PRINT D$"WRITE"NOM$
130 PRINT "TEXT:HOME:SPEED=255"
140 PRINT "PRINT"; CHR$(34); "INSERE
Z LA DISQUETTE CONTENANT RENUMBER
"; CHR$(34); ":PRINT"; CHR$(34);
"PUIS PRESSEZ UNE TOUCHE"; CHR$(
34); ": WAIT -16384,128:POKE -1636
8,0"
160 PRINT "LOAD RENUMBER"
164 PRINT "POKE 2968,96: POKE 2792,1
69:POKE 2793,200:POKE 2800,169:PO
KE 2801,10:CALL 2791"
165 REM SORTIE DE RENUMBER
PAR RTS AU LIEU DE JMP $D64B (EN
$B98)
167 REM MODIF DE RENUMBER
168 REM $AF0:LDA $7A REMPLACE
PAR LDA #$0A ET $AE8:LDA $79 REM
PLACE PAR LDA #$C8
170 PRINT "PRINT"; CHR$(34); "INSERE
Z LA DISQUETTE CONTENANT COMP"; C
HR$(34); ":PRINT"; CHR$(34); "PUI
S PRESSEZ UNE TOUCHE"; CHR$(34);
": WAIT -16384,128:POKE-16368,0"
190 PRINT "BRUN COMP"
195 PRINT "PRINT"; CHR$(34); "INSERE
Z LA DISQUETTE QUI CONTIENDRA
COMP+RENUM"; CHR$(34); ":PRINT";
CHR$(34); "PUIS PRESSEZ UNE TOUC
HE"; CHR$(34); ": WAIT -16384,128
:POKE-16368,0"
235 PRINT "BSAVECOMP+RENUM,A$8B00,L$
B00"
240 PRINT D$"CLOSE"NOM$
```

Un exemple

Le programme 'Agenda', publié page 8 de ce numéro.

AVANT : 1311 octets APRES : 858 octets
(Gain 34 %)

```
10 HOME : ONERR GOTO 570
100 PRINT " COMPOSITION AUTOMATIQUE DE": PRINT "
NUMERO TELEPHONIQUE": PRINT CHR$(4);"BLOA
DCOMPOSEUR": DIM T$(100,2)
140 N = 0
150 N = N + 1: FOR J = 1 TO 2: READ T$(N,J): NEXT J: GOTO
150
200 VTAB (8): PRINT "Utilisez les flèches ": INVERSE : PRINT "
=>": NORMAL : PRINT " et ": INVERSE : PRINT "<=": NOR
MAL : PRINT "pour sélectionner le nom de l'abonné": PRINT
: PRINT "Validez avec la touche RETURN": I = 0
250 GET A$
260 A% = ASC (A$): IF A% < > 13 AND A% < > 21 AND A%
< > 8 THEN 260
280 IF A% = 13 THEN 360
290 IF A% = 21 THEN I = I + 1
300 IF A% = 8 THEN I = I - 1
310 IF I > N - 1 THEN I = N - 1
320 IF I < 1 THEN I = 1
330 VTAB 15: CALL - 868: PRINT T$(I,1),T$(I,2): GOTO 250
360 X$ = T$(I,2): POKE 49241,0: FOR I = 1 TO 700: NEXT I: HO
ME : PRINT : PRINT "Je compose le numéro ": FOR I = 1
TO LEN (X$):A = ASC ( MID$( X$,I,1)) + 128: POKE 7,A:
CALL 768: PRINT CHR$(A):: FOR T = 1 TO 600: NEXT T
: NEXT I: HTAB 1: VTAB 5: PRINT "Décrochez le combiné,
maintenant...": VTAB 8: PRINT ".....avant": FOR I = 400 TO
1 STEP - 1: VTAB (15): HTAB (11): PRINT INT (I / 50),"
secondes": NEXT I: POKE 49240,0: GOTO 580
570 IF PEEK (222) = 42 THEN 200
580 HOME : RESTORE : PRINT "...Pret pour un autre appel ...":
GOTO 140: DATA RENSEIGNEMENTS,3612: DATA AUTRES
RENSEIGNEMENTS,3611: DATA JULIETTE,12345678
```

NDLR : Une ligne telle que la n° 360 de ce programme compacté ne manquera pas de vous poser des problèmes si vous voulez la modifier maintenant. Pour cette raison, il est sage de ne pas se séparer du programme original.

Si vous avez malencontreusement perdu votre version non compactée, vous utiliserez avec intérêt le **DECRUNCHER** de Patrice Neveu que nous publierons prochainement.

Récapitulation

COMP

8B00- A9 4C 8D F5 03 A9 14 8D	8B58- FF A0 00 A9 EA 85 C0 85	8BE0- 4F 47 52 41 4D 4D 45 20
8B08- F6 03 85 73 A9 8B 8D F7	8B60- C1 20 B7 00 20 B1 00 C9	8BE8- 45 4E 20 4D 45 4D 4F 49
8B10- 03 85 74 60 C9 43 F0 03	8B68- 00 D0 0C A9 FF A0 00 91	8BF0- 52 45 20 2A 2A 0D 0D 00
8B18- 4C 00 8E A0 00 C0 07 B0	8B70- FE C8 91 FE 4C F8 9B 20	8BF8- A5 68 85 B9 85 EF A5 67
8B20- 0B 20 B1 00 D9 34 8B D0	8B78- B1 00 20 B1 00 20 B1 00	8C00- D0 02 C6 B9 18 69 F8 85
8B28- 08 C8 D0 F1 20 B1 00 F0	8B80- C9 00 D0 06 20 B1 00 4C	8C08- B8 D0 02 C6 EF 18 69 FF
8B30- 0A 4C C9 DE 4F 4D 50 41	8B88- 5B 8B C9 AB F0 0F C9 B0	8C10- 85 EE A9 00 85 EB A9 03
8B38- 43 54 45 A5 67 85 B8 A5	8B90- F0 0B C9 C4 D0 E7 20 B1	8C18- 85 EC 85 19 85 18 20 D8
8B40- 68 85 B9 A0 01 B1 67 D0	8B98- 00 B0 E5 90 03 20 B1 00	8C20- 8C 20 B1 00 85 1D 20 B1
8B48- 03 4C C1 8B A5 AF 18 69	8BA0- 20 0C DA 48 A5 50 A0 00	8C28- 00 85 1E 20 12 8D 20 49
8B50- 05 85 FE A5 B0 69 00 85	8BA8- 91 FE E6 FE D0 02 E6 FF	8C30- 8D A5 19 D0 41 A5 1C D0
	8BB0- A5 51 91 FE E6 FE D0 02	8C38- 3D A5 1A F0 70 A5 18 F0
	8BB8- E6 FF 68 C9 2C F0 DF D0	8C40- 4F C9 F8 B0 31 18 65 1A
	8BC0- BF A0 00 B9 D1 8B F0 06	8C48- B0 04 C9 F8 90 35 A5 1B
	8BC8- 20 5C DB C8 D0 F5 4C B3	8C50- D0 24 A9 F8 38 E5 18 85
	8BD0- FE 0D 0D 2A 2A 20 07 50	8C58- F9 20 69 8D A5 FC F0 16
	8BD8- 41 53 20 44 45 20 50 52	8C60- A2 00 86 06 86 07 A9 3A

Programme COMPACTEUR (Assembleur Lisa 2.5)

```

1 ;
2 *****
3 *
4 * COMPACTE UN *
5 * PROGRAMME BASIC *
6 * ET SUPPRIME LES *
7 * REMARQUES *
8 * S.GALLET 24-8-85 *
9 *
10 *****
11 ;
12 CHRGET EPZ $B1
13 CHRGOT EPZ $B7
14 TXTTAB EPZ $67
15 TXTPTR EPZ $B8
16 TEMP EPZ $FD
17 TABLEAU EPZ $FE
18 GOTO EPZ $AB
19 GOSUB EPZ $B0
20 THEN EPZ $C4
21 CAR.ECR EPZ $EE
22 L EPZ $F9
23 DEUXPT EPZ $FC
24 LS EPZ $EB
25 LCLONG EPZ $18
26 LCIF EPZ $19
27 LSLONG EPZ $1A
28 LSIF EPZ $1B
29 LSGOTO EPZ $1C
30 NUM.B EPZ $1D
31 NUM.H EPZ $1E
32 IF EPZ $AD
33 REM EPZ $B2
34 AMPRS EPZ $AF ;TOKEN DE &
35 CALL EPZ $8C
36 DATA EPZ $83
37 PRGEND EPZ $AF
38 LMAX EPZ $F8
39 ONERR EPZ $A5
40 LINNUM EPZ $50
41 HIMEM EPZ $73
42 TEMP1 EPZ $06
43 TEMP2 EPZ $07
44 AMPER EQU $03F5
45 BASIC EQU $FED3
46 LINGET EQU $DA0C

```

```

47 RENUMBER EQU $8E00
48 STXERR EQU $DEC9
49 OUTDO EQU $DB5C
50 ;
51 ;
52 ORG $8B00
53 OBJ $B00
54 ;
55 A:
56 ;
57 LDA #$4C
58 STA AMPER
59 LDA #TEST
60 STA AMPER+1
61 STA HIMEM
62 LDA /TEST
63 STA AMPER+2
64 STA HIMEM+1
65 RTS
66 ;
67 TEST:
68 CMP #'C'
69 BEQ >1
70 JMP RENUMBER
71 ^1 LDY $0
72 ^2 CPY $7
73 BCS >3
74 JSR CHRGET
75 CMP COMMANDE,Y
76 BNE >4
77 INY
78 BNE <2
79 ^3 JSR CHRGET
80 BEQ INIT
81 ^4 JMP STXERR
82 COMMANDE ASC 'OMPACTE'
83 ;
84 INIT:
85 LDA TXTTAB
86 STA TXTPTR
87 LDA TXTTAB+1
88 STA TXTPTR+1
89 LDY $1
90 LDA (TXTTAB),Y
91 BNE >0
92 JMP PASPROG
93 ^0 LDA PRGEND
94 CLC
95 ADC $5
96 STA TABLEAU

```

```

8C68- 20 97 8D E6 18 C6 1A 20
8C70- B1 00 C6 FC D0 F2 20 B7
8C78- 8C 20 04 8D A9 00 85 18
8C80- 4C 90 8C A0 00 A9 JA D1
8C88- EE F0 05 20 F2 8D E6 18
8C90- A5 1B 85 19 A2 00 86 06
8C98- 86 07 20 B1 00 C9 00 F0
8CA0- 13 C9 B2 F0 08 20 97 8D
8CA8- E6 18 4C 9A 8C 20 B1 00
8CB0- C9 00 D0 F9 4C 1E 8C A5
8CB8- 18 D0 05 A9 3A 20 F2 8D
8CC0- A9 00 20 F2 8D 20 F2 8D
8CC8- A5 EE 91 EB C8 A5 EF 91
8CD0- EB 85 EC A5 EE 85 EB 60
8CD8- 20 B1 00 20 B1 00 C9 00
8CE0- D0 21 20 B7 8C A9 00 20
8CE8- F2 8D 20 F2 8D 20 F2 8D

```

```

8CF0- A5 EE 85 AF A5 EF 85 B0
8CF8- A9 F0 85 C0 A9 EF 85 C1
8D00- 4C B3 FE 60 20 F2 8D A5
8D08- 1D 20 F2 8D A5 1E 20 F2
8D10- 8D 60 A5 AF 18 69 04 85
8D18- FE A5 B0 69 00 85 FF A0
8D20- 00 20 40 8D C5 1D F0 0C
8D28- 20 40 8D C9 F0 D0 F2 A9
8D30- 00 85 1C 60 20 40 8D C5
8D38- 1E D0 F0 A9 01 85 1C 60
8D40- E6 FE D0 02 E6 FF B1 FE
8D48- 60 A0 00 84 1B 84 1A C8
8D50- B1 B8 F0 11 C9 B2 F0 0D
8D58- C9 AD F0 04 C9 A5 D0 02
8D60- 84 1B C8 D0 EB 88 84 1A
8D68- 60 A0 00 84 FD 84 FC C8
8D70- C4 F9 B0 22 B1 B8 F0 1E

```

```

8D78- C9 22 D0 09 A5 FD 49 01
8D80- 85 FD 4C 8F 8D C9 3A D0
8D88- 06 A5 FD D0 02 84 FC C8
8D90- C4 F9 90 E0 F0 DE 60 85
8D98- FD C9 83 F0 08 C9 AF F0
8DA0- 04 C9 8C D0 07 A9 01 85
8DA8- 07 4C ED 8D C9 22 D0 09
8DB0- A5 06 49 01 85 06 4C ED
8DB8- 8D A5 06 D0 30 A5 FD C9
8DC0- 3A D0 07 A9 00 85 07 4C
8DC8- ED 8D A5 07 D0 1F A5 FD
8DD0- C9 30 90 19 C9 5B B0 15
8DD8- C9 3A B0 06 E0 00 F0 0D
8DE0- D0 04 C9 41 90 07 E0 02
8DE8- 90 05 C6 18 60 A2 FF E8
8DF0- A5 FD A0 00 E6 EE D0 02
8DF8- E6 EF 91 EE 60 CA CD

```

```

97     LDA PRGEND+1
98     ADC #50
99     STA TABLEAU+1
100    LDY #50
101 ;
102 ;CREE UN TABLEAU DE TOUS LES NUMEROS DE LIGNES
      ADRESSES
103 ;PAR GOTO, GOSUB ET THEN
104 ;
105 DEBUT:
106     LDA #SEA           ;UNE PETITE RUSTINE
107     STA $C0           ;SUR CHRGET
108     STA $C1
109     JSR CHRGOT
110     JSR CHRGET
111     CMP #50
112     BNE >0
113     LDA #5FF
114     LDY #50
115     STA (TABLEAU),Y
116     INY
117     STA (TABLEAU),Y
118     JMP COMPACTE
119 ;
120 ^0   JSR CHRCET
121     JSR CHRGET
122 ^1   JSR CHRGET
123 ^2   CMP #50
124 ^3   BNE >4
125     JSR CHRGET
126     JMP DEBUT
127 ^4   CMP #GOTO
128     BEQ >5
129     CMP #GOSUB
130     BEQ >5
131     CMP #THEN
132     BNE <1
133     JSR CHRGET
134     BCS <2
135     BCC >6
136 ^5   JSR CHRGET
137 ^6   JSR LINGET
138     PHA
139     LDA LINNUM
140     LDY #50
141     STA (TABLEAU),Y
142     INC TABLEAU
143     BNE >7
144     INC TABLEAU+1
145 ^7   LDA LINNUM+1
146     STA (TABLEAU),Y
147     INC TABLEAU
148     BNE >8
149     INC TABLEAU+1
150 ^8   PLA
151     CMP #' ',' '
152     BEQ <5
153     BNE <2
154 PASPROG LDY #50
155 ^0   LDA MSG,Y
156     BEQ >1
157     JSR OUTDO
158     INY
159     BNE <0
160 ^1   JMP BASIC
161 MSG  HEX 0D0D
162     ASC '*** '
163     HEX 07
164     ASC 'PAS DE PROGRAMME EN MEMOIRE ***'
165     HEX 0D0D00
166 ;
167 ;
168 COMPACTE:
169 ;-----
170 ;
171 ;LCLONG ET LCIF CONCERNENT LA LIGNE
172 ; EN COURS D'ECRIURE
173 ;LSLONG, LSGOTO ET L$IF CONCERNENT
174 ; LA LIGNE SUIVANTE
175 ;
176 ;
177 ;AU DEPART, SI LE PROGRAMME BASIC
178 ;EST IMPLANTE EN $801:
179 ; - CHRGET = $800
180 ; - CAR.ECR = $7FF
181 ;
182 ;
183     LDA TXTTAB+1
184     STA TXTPTR+1
185     STA CAR.ECR+1
186     LDA TXTTAB
187     BNE >0
188     DEC TXTPTR+1
189 ^0   CLC
190     ADC #5FF
191     STA TXTPTR
192     BNE >1
193     DEC CAR.ECR+1
194 ^1   CLC
195     ADC #5FF
196     STA CAR.ECR
197     LDA #50
198     STA LS
199     LDA #503
200     STA LS+1
201     STA LCIF
202     STA LCLONG
203 ;
204 ^0   JSR LITADR
205     JSR CHRGET
206     STA NUM.B
207     JSR CHRGET
208     STA NUM.H
209     JSR GOTO?
210     JSR LONG.IF
211 ;
212     LDA LCIF           ;SI LCIF<>0 OU SI
213     BNE >3           ;LSGOTO<>0, IL NE
214     LDA LSGOTO       ;FAUT PAS "COLLER"
215     BNE >3           ;LES 2 LIGNES
216 ;
217     LDA LSLONG       ;SI LSLONG = 0 ->
218     BEQ >7           ;RIEN A RECOPIER
219 ;
220     LDA LCLONG       ;SI LCLONG = 0 ->
221     BEQ >5           ;ON COPIE LS TELLE
222 ;                   QUE L$IF
223 ;
224     CMP #LMAX       ;SI LCLONG >= LMAX
225     BCS >3           ;SON COMPTE EST BON
226 ;                   !
227     CLC
228     ADC LSLONG
229     BCS >1
230     CMP #LMAX       ;SI LCLONG+LSLONG <
231 ;                   ;LMAX ON PEUT COLLE
232 ;                   R LC ET LS
233 ;ICI, ON DOIT SE CONTENIR DE COLLER UN MORCEAU
234 ;DE LS AU BOUT DE LC, A CONDITION QUE LS NE
235 ;CONTIENNE PAS DE IF (NI ONERR GOTO) ET QU'IL Y
236 ;

```

```

237 ^1   LDA LSIF
238     RNE >3
239     LDA #LMAX
240     SEC
241     SBC LCLONG
242     STA L
243     JSR CHERCHE
244     LDA DEUXPT
245     BEQ >3
246     LDX #50
247     STX TEMP1
248     STX TEMP2
249     LDA #' : '
250 ^2   JSR ECRIRE
251     INC LCLONG
252     DEC LSLONG
253     JSR CHRGET
254     DEC DEUXPT
255     BNE <2
256 ;
257 ; ENTREE ICI SI ON NE COMPACTE PAS
258 ; LES DEUX LIGNES
259 ;
260 ^3   JSR CHAINER
261     JSR COPIENUM
262     LDA #50
263     STA LCLONG
264     JMP >5
265 ;
266 ; ENTREE ICI SI ON COMPACTE LES
267 ; DEUX LIGNES
268 ;
269 ^4   LDY #50
270     LDA #' : '
271     CMP (CAR.ECR),Y
272     BEQ >5
273     JSR ECRIRE2
274     INC LCLONG
275 ;
276 ; ENTREE ICI SI LCLONG=0 ET
277 ; LSLONG<>0: ON RECOPIE LS DANS LC
278 ;
279 ^5   LDA LSIF
280     STA LCIF
281     LDX #50
282     STX TEMP1
283     STX TEMP2
284 ^6   JSR CHRGET
285     CMP #50
286     BEQ >8
287     CMP #REM
288     BEQ >7
289     JSR ECRIRE
290     INC LCLONG
291     JMP <6
292 ;
293 ; ENTREE ICI SI LSLONG=0, ON FINIT
294 ; DE LIRE LA LIGNE SANS RECOPIER
295 ;
296 ^7   JSR CHRGET
297     CMP #50
298     BNE <7
299 ^8   JMP <0
300 ;
301 ;
302 ;
303 ;SOUS PROGRAMMES
304 ;-----
305 ;
306 CHAINER:
307     LDA LCLONG
308     BNE >1
309     LDA #' : '
310     JSR ECRIRE2
311 ^1   LDA #50
312     JSR ECRIRE2
313     JSR ECRIRE2
314     LDA CAR.ECR
315     STA (LS),Y
316     INY
317     LDA CAR.ECR+1
318     STA (LS),Y
319     STA LS+1
320     LDA CAR.ECR
321     STA LS
322     RTS
323 ;
324 ;
325 ;
326 ;LIT LES 2 PREMIERS OCTETS DE LS
327 ;SI 00 00 -> FIN DU PROGRAMME BASIC
328 ;
329 LITADR:
330     JSR CHRGET
331     JSR CHRGET
332     CMP #50
333     BNE >1
334     JSR CHAINER
335     LDA #50
336     JSR ECRIRE2
337     JSR ECRIRE2
338     JSR ECRIRE2
339     LDA CAR.ECR
340     STA PRGEND
341     LDA CAR.ECR+1
342     STA PRGEND+1
343     LDA #$F0
344     STA $C0
345     LDA #$EF
346     STA $C1
347     JMP BASIC
348 ^1   RTS
349 ;
350 COPIENUM:
351     JSR ECRIRE2
352     LDA NUM.B
353     JSR ECRIRE2
354     LDA NUM.H
355     JSR ECRIRE2
356     RTS
357 ;
358 ;
359 ;CHERCHE SI LS EST APPELEE PAR GOTO, GOSUB OU TH
360 ;SI OUI -> LSGOTO=1
361 ;SI NON -> LSGOTO=0
362 ;
363 GOTO?:
364     LDA PRGEND
365     CLC
366     ADC #54
367     STA TABLEAU
368     LDA PRGEND+1
369     ADC #50
370     STA TABLEAU+1
371     LDY #50
372 ^1   JSR LIRETABL
373     CMP NUM.B
374     BEQ >3
375     JSR LIRETABL
376 ^2   CMP #$FF
377     RNE <1
378     LDA #50

```

```

379      STA LSGOTO
380      RTS
381 ^3   JSK LIRETABL
382      CMP NUM.H
383      BNE <2
384      LDA #S1
385      STA LSGOTO
386      RTS
387 ;
388 LIRETABL:
389      INC TABLEAU
390      BNE >4
391      INC TABLEAU+1
392 ^4   LDA (TABLEAU),Y
393      RTS
394 ;
395 ;
396 ;
397 ;CHERCHE LA LONGUEUR DE LS (LSLONG)
398 ;ET LA PRESENCE DE IF OU DE ONERR
399 ;SI NI IF, NI ONERR -> LSIF=0
400 ;SINON -> LSIF<>0
401 ;
402 LONG.IF:
403      LDY #S0
404      STY LSIF
405      STY LSLONG
406      INY
407 ^1   LDA (TXTPTR),Y
408      BEQ >4
409      CMP #REM
410      BEQ >4
411      CMP #IF
412      BEQ >2
413      CMP #ONERR
414      BNE >3
415 ^2   STY LSIF
416 ^3   INY
417      BNE <1
418 ^4   DEY
419      STY LSLONG
420      RTS
421 ;
422 ;
423 ;
424 ;POUR COPIER UN MORCEAU DE LS,
425 ;CHERCHE SI DANS LES L PREMIERS
426 ;CARACTERES IL Y A UN : NON
427 ;INCLUS DANS DES GUILLEMETS
428 ;SI OUI -> DEUXPT = LA DERNIERE
429 ;      POSITION TROUVEE
430 ;SI NON -> DEUXPT=0
431 ;
432 CHERCHE:
433      LDY #S0
434      STY TEMP
435      STY DEUXPT
436      INY
437      CPY L
438      BCS >4
439 ^1   LDA (TXTPTR),Y
440      BEQ >4
441      CMP #' "'
442      BNE >2
443      LDA TEMP
444      EOR #S01
445      STA TEMP
446      JMP >3
447 ^2   CMP #' ':'
448      BNE >3
449      LDA TEMP
450      BNE >3
451      STY DEUXPT
452 ^3   INY
453      CPY L
454      BCC <1
455      BEQ <1
456 ^4   RTS
457 ;
458 ;DANS 'ECRIRE':
459 ;SI TEMP1=1: ON EST EN TRAIN DE RECOPIER
460 ;UNE CHAINE SANS LA MODIFIER
461 ;SI TEMP1=0: TOUT DEPEND DE TEMP2
462 ;
463 ;SI TEMP2=1: ON EST EN TRAIN DE COPIER UNE
464 ;INSTRUCTION QUI CONTIENT &, CALL OU DATA
465 ;DANS CE CAS, ON COPIE TOUT JUSQU'AU ':'
466 ;SEPARATEUR
467 ;SI TEMP2=0: L'INSTRUCTION NE CONTIENT NI &
468 ;NI CALL NI DATA: LES TEST DETERMINENT ALORS
469 ;SI ON TRAITE UN NOM DE VARIABLE OU UNE
470 ;CONSTANTE
471 ;DANS LE CAS D'UNE VARIABLE X INDIQUE LE NOMBRE
472 ;DE CARACTERES DEJA ECRITS
473 ;X EST REMIS A 0 PAR TOUT CARACTERE NON ALPHANUME
474 ;RIQUE
475 ;TEMP2 EST REMIS A 0 PAR UN ':' SEPARATEUR OU PA
476 ;R UN 0 DE FIN DE LIGNE
477 ;
478 ;
479 ECRIRE:
480      STA TEMP
481      CMP #DATA
482      BEQ >0
483      CMP #AMPRS
484      BEQ >0
485      CMP #CALL
486      BNE >1
487 ^0   LDA #S1
488      STA TEMP2
489      JMP >7
490 ^1   CMP #' "'
491      BNE >2
492      LDA TEMP1
493      EOR #S1
494      STA TEMP1
495      JMP >7
496 ^2   LDA TEMP1
497      BNE >7
498      LDA TEMP
499      CMP #' ':'
500      BNE >3
501      LDA #S0
502      STA TEMP2
503      JMP >7
504 ^3   LDA TEMP2
505      BNE >7
506      LDA TEMP
507      CMP #'0'
508      BCC >7
509      CMP #'Z'+1
510      BCS >7
511      CMP #'9'+1
512      BCS >4
513      CPX #S0
514      BEQ >7
515      BNE >5
516 ^4   CMP #'A'
517      BCC >7
518 ^5   CPX #S2
519      BCC >6
520      DEC LCLONG
521      RTS
522 ^7   LDX #SFF
523 ^6   INX
524      LDA TEMP
525 ECRIRE2:
526      LDY #S0
527      INC CAR.ECR
528      BNE >9
529      INC CAR.ECR+1
530 ^9   STA (CAR.ECR),Y
531      RTS
532 ;
533 ;
534 B:
535 LONGUEUR EQU B-A
536 ;
537      END

```

Copie d'écran

Bernard
France

basse résolution

Les utilitaires de recopie graphique des écrans haute-résolution sont très nombreux, disponibles sur le marché commercial, ou publiés par Pom's. En revanche, les écrans graphiques basse résolution semblent être ignorés.

Je vous propose ici un petit programme qui permet de générer des impressions papier semblables à la haute-résolution à partir des graphiques basse-résolution de l'Apple. Ceci donne un aspect beaucoup plus agréable, en éliminant l'aspect terne des impressions traditionnelles.

Le programme DHGR utilise la propriété de l'ImageWriter et de la DMP de pouvoir créer facilement sa propre police de caractère, et n'est donc exploitable qu'avec ces imprimantes.

Utilisation

Le programme fonctionne sous DOS 3.3 ; il se loge à l'adresse \$300 (768 en décimal) sur une longueur de \$F0 octets. Comme il écrase certaines adresses du DOS contenues en \$3D0-\$3EF, il peut être nécessaire de rebooter après son utilisation.

HCGR fonctionne sur tous les types d'Apple // pourvus d'une ImageWriter connectée au slot 1. Il s'initialise par un simple "BLOAD HCGR" et est alors exécuté aussi souvent que nécessaire par "CALL 768".

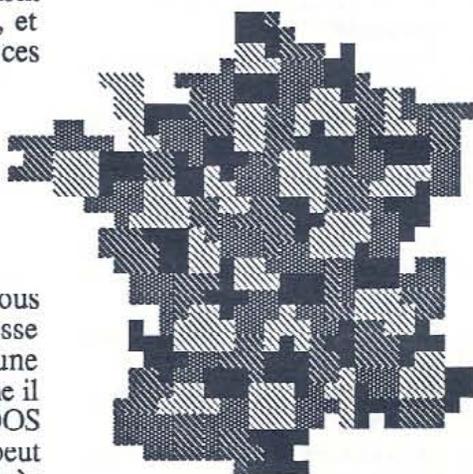
Les fichiers de la disquette Pom's sont :

HCGR, le code objet exécutable
HCGR.S, le source en Big Mac
HCGR.DEMO, la démonstration

Principe de fonctionnement

HCGR effectue l'impression de la page graphique basse résolution de l'Apple (accessible sous Applesoft par l'instruction "GR"). Cette page correspond à une grille de 40 x 48 pavés, chacun ayant une couleur parmi 16 disponibles.

En mémoire, les pavés sont regroupés deux à deux pour former un octet. Il serait possible de considérer chaque octet comme un code ASCII et d'imprimer le caractère correspondant. Le résultat ne serait cependant guère satisfaisant car l'écran graphique serait



alors imprimé sous forme d'une suite de lettres, de chiffres ou de caractères de ponctuation.

Afin d'améliorer cette présentation, HCGR utilise la possibilité de générer une police de caractères sur l'imprimante ImageWriter. Cela permet de définir des caractères en fonction de la couleur du pavé graphique : chaque caractère est imprimé en fonction d'une matrice 8 x 8 contenue dans la mémoire de

l'imprimante ; on n'a donc qu'à redéfinir ces matrices pour obtenir des caractères très clairs si le pavé graphique affiche une couleur claire, et le contraire dans le cas ... contraire.

Fonctionnement du programme

Le programme doit tout d'abord sélectionner l'imprimante (PR#1). Il faut ensuite créer une police de caractères personnalisés, ce qui est réalisé par la séquence suivante :

- initialisation (\$09,\$5A,\$0D) ;
- largeur des caractères fixée à 8 points d'aiguille (\$1B,\$2D) ;
- début de définition (\$1B,\$49) ;
- définition de chaque caractère : code ASCII suivi du code largeur (\$48 pour 8 points) et celui de la nouvelle matrice ;
- fin de définition (\$04) ;
- sélection des caractères 'Pica' (\$1B,\$4E) ;
- interligne de 16/144 (\$1B,\$54, \$31,\$36) ;
- options : simple largeur, impression bidirectionnelle, 5 sauts de lignes (\$0F,\$1B,\$3C, \$1F,\$35).

Il reste à examiner chaque pavé de la page graphique basse résolution implantée aux adresses de \$400 à \$7FF, de le convertir en caractère imprimable (un pavé blanc, donc de couleur \$F, est converti en \$3F correspondant au caractère "?" qui est imprimé comme un carré plein).

On utilise la routine SCREEN du moniteur (située en \$F871) qui renvoie, pour une ligne et une colonne données, la valeur du code couleur.

Enfin, on redonne la main au DOS 3.3 en effectuant un PR#0.

Programme HCGR.DEMO

```

10 HOME : PRINT "DEMONSTRATION
DE HCGR"
11 PRINT : PRINT "ALLUMEZ VOTR
E IMAGEWRITER"
12 PRINT CHR$(4); "BLOAD HCCR
,A768"
15 HOME : GR : POKE - 16302,0
: CALL - 1998
25 POKE 1004,62
30 POKE 1002,15: GOSUB 100
35 POKE 1002,14: COSUB 100
40 POKE 1002,15: GOSUB 300
45 POKE 1002,14: GOSUB 300
50 C(0) = 0:C(1) = 15:C(2) = 12
:C(3) = 9:C(4) = 0:C(5) = 5:
C(6) = 0
55 POKE 1002,15: GOSUB 500
60 C(0) = 0:C(1) = 15:C(2) = 15
:C(3) = 15:C(4) = 2:C(5) = 1
5:C(6) = 8
65 POKE 1002,15: GOSUB 500
90 TEXT : HOME : END
100 CALL - 1998
120 FOR I = 0 TO 19
140 COLOR= I
160 HLIN I,39 - I AT I
180 HLIN I,39 - I AT 39 - I
200 VLIN I,39 - I AT I
220 VLIN I,39 - I AT 39 - I
240 NEXT I
260 CALL 768
280 RETURN
300 CALL - 1998
320 FOR J = 0 TO 47
340 FOR I = 0 TO 39
360 COLOR= I + J
380 PLOT I,J
400 NEXT I
420 NEXT J
440 CALL 768
460 RETURN
500 REM CARTE DE FRANCE
520 CALL - 1998
540 RESTORE
560 FOR Y = 0 TO 47
580 FOR X = 0 TO 39
600 READ I
610 IF Y2 < Y1 THEN 580
620 COLOR= C(I)
640 PLOT X,Y
660 NEXT X
680 NEXT Y
700 CALL 768
720 RETURN
1000 DATA 0,0,0,0,0,0,1,1,1,1,0,
0,2,2,2,2,0,0,3,0,0,0,0,3,
0,0,0,1,0,0,5,5,5,0,0,0,
0,0,0
1001 DATA 0,0,0,0,0,0,1,0,0,1,0,
0,2,0,0,2,0,0,3,3,0,0,3,3,
0,0,1,1,0,0,5,0,0,0,0,0,0,
0,0,0
1002 DATA 0,0,0,0,0,0,1,0,0,1,0,
0,2,0,0,2,0,0,3,0,3,3,0,3,
0,0,0,0,0,0,5,0,0,0,0,0,0,
0,0,0
1003 DATA 0,0,0,0,0,0,1,1,1,1,0,
0,2,0,0,2,0,0,3,0,0,0,0,3,
0,0,0,0,0,0,5,5,5,5,0,0,0,

```

```

0,0,0
1004 DATA 0,0,0,0,0,0,1,0,0,0,0,
0,2,0,0,2,0,0,3,0,0,0,0,3,
0,0,0,0,0,0,0,0,0,5,0,0,0,
0,0,0
1005 DATA 0,0,0,0,0,0,1,0,0,0,0,
0,2,0,0,2,0,0,3,0,0,0,0,3,
0,0,0,0,0,0,0,0,0,5,0,0,0,
0,0,0
1006 DATA 0,0,0,0,0,0,1,0,0,0,0,
0,2,2,2,2,0,0,3,0,0,0,0,3,
0,0,0,0,0,0,5,5,5,0,0,0,
0,0,0
1007 DATA 0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0

```

```

1010 DATA 4,4,4,4,4,4,4,4,4,4,4,
4,4,4,4,4,4,4,4,5,5,1,3,3,
3,6,1,6,6,6,6,6,6,6,6,6,6,
6,6,6
1011 DATA 4,4,4,4,4,4,4,4,4,4,4,
4,4,4,4,4,4,4,4,2,5,5,5,2,2,
2,1,1,6,6,6,6,6,6,6,6,6,6,
6,6,6
1012 DATA 4,4,4,4,4,4,4,4,4,4,5,
5,5,4,4,2,2,2,2,1,1,1,2,2,
2,1,1,3,6,6,6,6,6,6,6,6,6,
6,6,6
1013 DATA 4,4,4,4,4,4,4,4,4,4,4,
5,5,4,4,4,3,2,2,1,1,1,2,2,
5,5,1,3,3,5,2,6,6,6,6,6,6,
6,6,6
1014 DATA 4,4,4,4,4,4,4,4,4,4,4,
5,5,1,1,1,3,3,3,5,5,3,2,5,
5,5,5,3,3,5,2,2,3,3,3,6,6,
6,6,6
1015 DATA 4,4,4,4,4,4,4,4,2,2,4,4,
4,5,1,2,2,2,3,3,5,5,3,3,5,
1,5,5,2,3,5,5,2,3,3,6,6,6,
6,6,6
1016 DATA 4,4,4,4,3,3,3,2,2,2,2,
4,5,5,2,2,2,1,1,5,5,3,3,1,
1,1,1,2,1,1,1,1,3,3,6,6,6,
6,6,6
1017 DATA 4,4,4,4,4,3,3,5,5,5,1,
1,1,3,3,1,2,1,1,1,2,3,5,5,
1,1,1,2,2,1,1,1,2,2,6,6,6,
6,6,6
1018 DATA 4,4,4,4,3,3,3,5,5,5,1,
1,1,3,3,1,1,5,1,2,2,2,5,5,
5,3,3,2,2,5,5,5,2,2,6,6,6,
6,6,6
1019 DATA 4,4,4,4,4,4,4,5,5,5,1,
2,2,5,5,1,1,5,5,2,2,2,5,5,
5,3,3,3,5,5,5,1,2,2,6,6,6,
6,6,6
1020 DATA 4,4,4,4,4,4,4,4,4,2,2,
2,2,5,5,5,3,3,5,5,3,3,1,1,
5,3,3,3,2,1,1,1,6,6,6,6,6,
6,6,6
1021 DATA 4,4,4,4,4,4,4,4,4,4,4,
2,5,5,5,5,3,3,2,3,3,3,3,1,
1,5,5,5,2,2,1,1,6,6,6,6,6,
6,6,6
1022 DATA 4,4,4,4,4,4,4,4,4,3,
3,3,1,1,1,5,3,2,2,2,3,3,1,
5,5,5,5,2,2,1,6,6,6,6,6,6,
6,6,6
1023 DATA 4,4,4,4,4,4,4,4,4,4,3,
3,3,3,1,1,5,5,2,2,2,3,2,2,
5,5,5,1,2,2,6,6,6,6,6,6,6,
6,6,6
1024 DATA 4,4,4,4,4,4,4,4,4,4,4,
3,3,3,1,1,5,5,1,5,5,2,2,2,
2,3,2,1,1,1,6,2,2,6,6,6,6,
6,6,6
1025 DATA 4,4,4,4,4,4,4,4,4,4,4,
4,2,2,2,3,3,3,1,5,5,5,1,1,
1,3,2,1,1,1,2,2,2,6,6,6,6,
6,6,6
1026 DATA 4,4,4,4,4,4,4,4,4,4,4,
4,2,2,2,3,3,1,1,1,2,2,1,1,
1,3,2,5,5,5,3,3,3,6,6,6,6,
6,6,6
1027 DATA 4,4,4,4,4,4,4,4,4,4,4,
4,4,4,2,3,5,5,1,2,2,2,5,2,
2,3,3,5,5,5,5,3,3,6,6,6,6,
6,6,6
1028 DATA 4,4,4,4,4,4,4,4,4,4,4,

```

La mise en page de ce programme

Bernard France nous pardonnera certainement d'avoir modifié légèrement les DATA de son source assembleur. Le programme de Marianne Sutz du précédent numéro nous a aidé à récupérer les 'motifs' du Macintosh. Pour obtenir les reproductions ci-contre, nous avons utilisé non l'ImageWriter mais la LaserWriter en suivant les étapes suivantes :

- Récupération des DATA sur le Macintosh,
- Transmission par la carte série sur l'Apple II,
- Insertion dans l'assembleur puis assemblage,
- 'Impression' non sur l'ImageWriter mais sur l'écran du Macintosh émulant l'imprimante, toujours par la carte série et avec quelques lignes de Basic,
- Sauvegarde de l'écran sur disquette,
- Récupération avec le programme MacPaint,
- Impression définitive par l'intermédiaire du programme MacDraw afin de choisir aisément le format.

Le résultat est la réplique fidèle (et même un peu plus) de ce que vous obtiendrez sur votre imprimante.

```

1008 DATA 4,4,4,4,4,4,4,4,4,4,4,
4,4,4,4,4,4,4,4,1,1,3,6,6,
6,6,6,6,6,6,6,6,6,6,6,6,6,
6,6,6
1009 DATA 4,4,4,4,4,4,4,4,4,4,4,
4,4,4,4,4,4,4,4,1,1,1,3,6,
6,6,6,6,6,6,6,6,6,6,6,6,6,
6,6,6

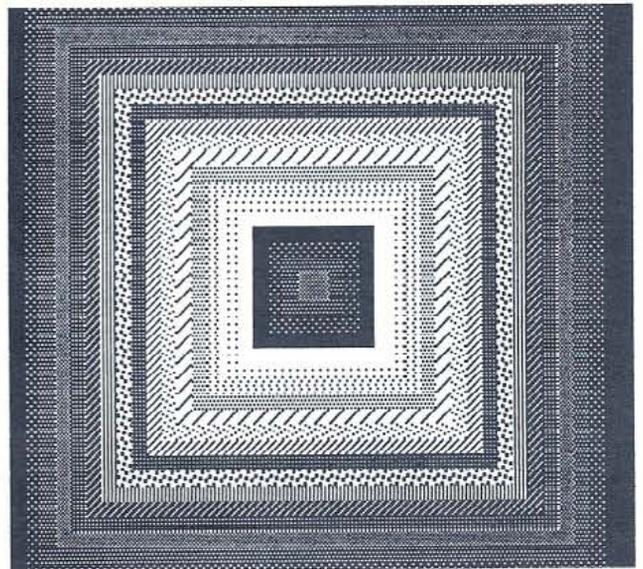
```


Programme HCGR.S (assembleur BigMac)

```

1 *****
2 * HCGR *
3 *****
4
5 * BERNARD FRANCE
6 * BIG MAC
7
8 COUT      =   $FDED
9 CSW       =   $36
10 DOSCSW   =   $9EBD
11 SCREEN   =   $F871
12          ORG  $300
13
14 ***** PR#1 *****
15
16          LDA  #$00
17          STA  CSW
18          LDA  #$C1
19          STA  CSW+1
20
21 ** DEFINITION NOUVEAUX CARACTERES
22
23          LDX  #$39          40 COL - 1
24 CAR      INX              CARACT SUIVT
25          LDA  $0300,X      CHARGE TABLE
26          JSR  COUT         ENVOI A IMPR
27          CPX  #$EE         FIN TABLE ?
28          BCC  CAR          NON
29
30 ** ANALYSE DE LA PAGE GRAPHIQUE
31
32          LDX  #$FF
33 LIG      INX              LIGN SUIVANTE
34          LDY  #$FF
35 COL     INY              COL SUIVANTE
36          TXA
37          JSR  SCREEN       RENV COULEUR
38          ORA  #$30         CONVERTI
39          JSR  COUT         ENVOI A IMPR
40          CPY  #$27         DERNIERE COL
41          BCC  COL          NON
42          LDA  #$0D         RETR CHARIOT
43          JSR  COUT
44          CPX  #$2F         DERNIERE LIGN
45          BCC  LIG          NON
46
47 ***** PR#0 *****
48
49          LDA  #<DOSCSW     RETOUR AU DOS
50                    3.3
51          STA  CSW
52          LDA  #>DOSCSW
53          STA  CSW+1
54          RTS              C'EST FINI!
55 ***** TABLE *****
56
57          DFB  $09,$5A,$0D INITIALISE
58          DFB  $1B,$2D      8 POINTS
59          DFB  $1B,$49      DEBUT DEF

```



```

60          DFB  $30,$48,$00,$00,$00,$00,
              $00,$00,$00,$00
              NOIR
61          DFB  $31,$48,$22,$00,$88,$00,
              $22,$00,$88,$00
              MAGENTA
62          DFB  $32,$48,$55,$00,$AA,$00,
              $55,$00,$AA,$00
63          DFB  $33,$48,$55,$AA,$55,$AA,
              $55,$AA,$55,$AA
64          DFB  $34,$48,$AA,$00,$AA,$00,
              $AA,$00,$AA,$00
65          DFB  $35,$48,$11,$22,$44,$88,
              $11,$22,$44,$88
66          DFB  $36,$48,$55,$55,$55,$55,
              $55,$55,$55,$55
67          DFB  $37,$48,$27,$3F,$F3,$72,
              $4E,$CF,$FC,$E4
68          DFB  $38,$48,$55,$00,$55,$00,
              $55,$00,$55,$00
69          DFB  $39,$48,$EE,$DD,$BB,$77,
              $EE,$DD,$BB,$77
70          DFB  $3A,$48,$F7,$DF,$FE,$FB,
              $BF,$EF,$FD,$7F
71          DFB  $3B,$48,$EF,$DF,$BF,$7F,
              $FE,$FD,$FB,$F7
72          DFB  $3C,$48,$55,$FF,$AA,$FF,
              $55,$FF,$AA,$FF
73          DFB  $3D,$48,$EE,$FF,$DD,$FF,
              $EE,$FF,$DD,$FF
74          DFB  $3E,$48,$77,$FF,$FF,$FF,
              $77,$FF,$FF,$FF
75          DFB  $3F,$48,$FF,$FF,$FF,$FF,
              $FF,$FF,$FF,$FF
76          DFB  $04          FIN POLICE
77          DFB  $1B,$4E      CARACT PICA
78          DFB  $1B,$27      POLICE PERSO
79          DFB  $1B,$54,$31,$36
80          DFB  $0F          SIMPL LARGEU
81          DFB  $1B,$3C      BIDIRECTIONN
82          DFB  $1F,$35      5 SAUTS LIGN
83          DFB  $0D          NON UTILISE
84          END

```

De la formule à la courbe

Jean-Louis Chauvin

En complément de l'article paru dans le n° 11 de Pom's, voici deux nouveaux programmes de tracé de courbes en conversationnel.

Le premier, baptisé *FORMULE1*, dessine toutes les fonctions du type $Y = f(X)$. Il permet aussi de tracer la courbe de la dérivée.

FORMULE2 trace les courbes définies soit par un couple d'équations paramétriques $X = f(T)$, $Y = g(T)$, soit par une équation polaire $R = f(T)$. Dans les deux cas, vous pouvez également :

- obtenir une famille de courbes en introduisant dans la formule un paramètre M ;
- cadrer la partie du plan (X Y) représentée ;
- faire apparaître ou non les axes de coordonnées ;
- tracer la courbe par points (pour les fonctions discontinues).

Avec les deux programmes, les formules apparaissent sur le graphique.

Quand vous lancez *FORMULE1*, laissez dans le lecteur une disquette non protégée (vous verrez plus loin pourquoi).

Complétez après 'Y =' la formule comme en Basic, sans autre variable que X (et M dans l'option avec 'paramètre'); vous pouvez utiliser les constantes PI et E (attention à la syntaxe : tapez $2 * PI$ et non $2 PI$). $X1$, $X2$, $Y1$ et $Y2$ délimitent la partie du plan représentée sur l'écran.

Si vous changez les valeurs proposées, prenez soin que $X2$ soit supérieur à $X1$ et que $Y2$ soit supérieur à $Y1$. Si vous avez demandé un paramètre M , vous devrez définir sa progression par les valeurs extrêmes $M1$ et $M2$ et par le pas qui doit être de même signe que $(M2-M1)$.

Avec *FORMULE2*, le processus est le même excepté que, dans les formules, la variable doit être T et non X .

En outre, vous aurez à introduire les bornes $T1$ et $T2$ du domaine de variation de T si les valeurs par défaut ne vous conviennent pas.

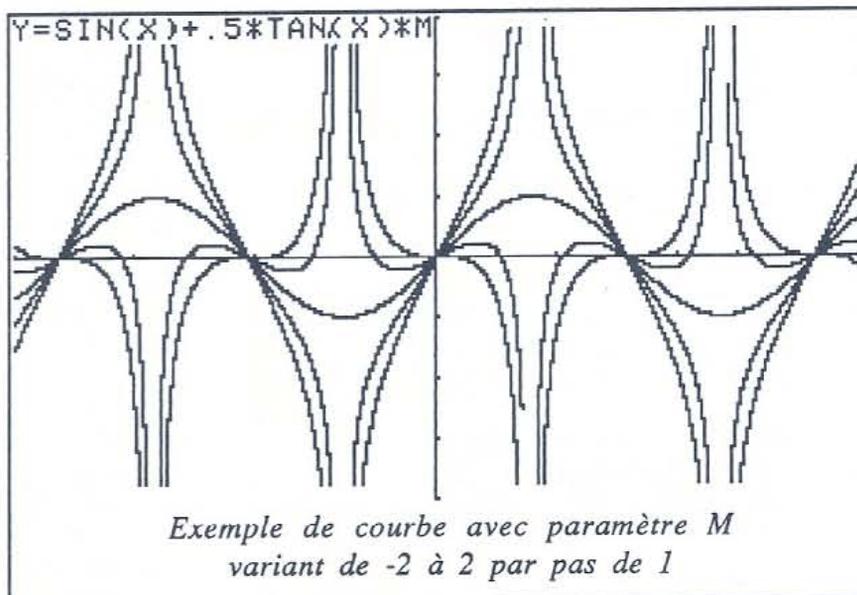
Le paramètre M vous permettra de réaliser des effets artistiques. Essayez par exemple :

$R = \text{COS}(4 * T) + M$
avec $M1 = -2$, $M2 = 2$ et $PAS = 1$.

- l'arrêt par **END** (ligne 290) déclenche l'exécution de la procédure.

Celle-ci, après avoir modifié le programme, le relance à la ligne 400 (**RUN** ligne 260). Ce **RUN** mettant les variables à zéro, les valeurs à conserver doivent être inscrites également dans des instructions par la procédure (lignes 225 à 250).

- après redémarrage, la procédure est effacée (ligne 410) et le déroulement se poursuit.



Le programme

Il est entièrement rédigé en Applesoft. Pour insérer une formule dans une instruction, il procède comme suit :

- la formule est introduite (ligne 90 de *FORMULE1*) comme variable chaîne $F\$$.
- le programme crée et enregistre sur la disquette (lignes 200 à 270) une procédure nommée **PROCEDURE** qui comporte entre autres commandes (ligne 220) la création à la ligne 560 de l'instruction représentée par la formule $Y=F\$$.

Pendant le tracé de la courbe, les valeurs non définies de la fonction ($1/0$, $\text{SQR}(-1)$...) sont éliminées par **ONERR GOTO 1700**, corrigé par la routine classique en langage machine de la ligne 35.

Par ailleurs, l'affichage de la formule en mode graphique est assuré par le sous-programme de la ligne 1500.

Celui-ci utilise une table de formes (fichier **ALPHA** sur la disquette) que le programme charge (ligne 37) à l'adresse 16384 ($\$4000$), immédiatement à la suite de la page graphique haute résolution n°1.



Programme FORMULE1

```

10 REM TRACE DE COURBES DE FONCTIO
  N
20 REM PAR J.L.CHAUVIN 1984
30 D$ = CHR$(4):FIRST = 1: SCALE
  = 1: ROT= 0:Q$ = CHR$(34):S =
  16384
35 POKE 768,104: POKE 769,168: POK
  E 770,104: POKE 771,166: POKE 7
  72,223: POKE 773,154: POKE 774,
  72: POKE 775,152: POKE 776,72:
  POKE 777,96
37 PRINT D$"BLOADALPHA,A"S: POKE 2
  32,0: POKE 233,64
40 POKE 34,0: TEXT : HOME : PRINT
  TAB(7)"COURBES DE FONCTIONS Y
  =F(X)
50 PRINT TAB(7)"=====
  =====
60 PRINT : PRINT "INTRODUCTION DE
  LA FORMULE
70 PRINT : INPUT "FORMULE AVEC PAR
  AMETRE 'M' (O)?" ; A$: IF A$ = "O
  " THEN PAR = 1
80 IF FIRST THEN PRINT : PRINT "R
  ESPECTEZ LA SYNTAXE DU BASIC":
  PRINT "VOUS POUVEZ UTILISER 'PI
  ' ET 'E'"
90 PRINT : PRINT "FORMULE:" ; INPU
  T " Y=" ; F$
100 PRINT : PRINT "VALEURS EXTREME
  S DE X(SINON:X1=-7,X2=7):": IN
  PUT "X1=" ; X1$: IF X1$ = "" THE
  N X1$ = "-7": HTAB 4: VTAB PE
  EK (37): PRINT X1$
105 INPUT "X2=" ; X2$: IF X2$ = "" T
  HEN X2$ = "7": HTAB 4: VTAB P
  EEK (37): PRINT X2$
110 PRINT : PRINT "VALEURS EXTREME
  S DE Y(SINON:Y1=-4,Y2=4):": IN
  PUT "Y1=" ; Y1$: IF Y1$ = "" THE
  N Y1$ = "-4": HTAB 4: VTAB PE
  EK (37): PRINT Y1$
115 INPUT "Y2=" ; Y2$: IF Y2$ = "" T
  HEN Y2$ = "4": HTAB 4: VTAB P
  EEK (37): PRINT Y2$
120 M$ = "0":MD$ = "0":PM$ = "0":
  IF PAR THEN PRINT : PRINT "VA
  LEURS DE M:" : INPUT "DE M=" ; M$
  : INPUT "A M=" ; MD$: INPUT "PAS
  =" ; PM$
130 HGR : HOME
200 REM CREATION PROCEDURE POUR IN
  TRODUIRE LES FONCTIONS DANS LE
  PROGRAMME *
210 PRINT D$;"OPENPROCEDURE": PRIN
  T D$;"WRITEPROCEDURE
220 PRINT "560Y="F$: PRINT "825Y="
  F$: PRINT "835Y="F$
225 PRINT "415F$="Q$;F$;Q$
230 PRINT "420X1="X1$":X2="X2$
240 PRINT "430Y1="Y1$":Y2="Y2$
250 PRINT "440M="M$":MD="MD$":PM="
  PM$
260 PRINT "RUN400
270 PRINT D$"CLOSEPROCEDURE
280 PRINT D$"EXECPROCEDURE
290 END : REM ARRET POUR DECLENCHE
  R EXEC
399 REM PREPARATION DONNEES **
400 PI = 4 * ATN (1):E = EXP (1)
  : ONERR GOTO 1700
410 D$ = CHR$(4):Q$ = CHR$(34)
  : PRINT D$"DELETEPROCEDURE
415 F$ = "EXP(X/2)"
420 X1 = - 7:X2 = 7
430 Y1 = - 4:Y2 = 4
440 M = 0:MD = 0:PM = 0
445 POKE 34,20: HOME : HCOLOR= 3
450 IF X1 > = X2 THEN PRINT "ERR
  EUR: X1>=X2": GET A$: GOTO 40
460 IF Y1 > = Y2 THEN PRINT "ERR
  EUR: Y1>=Y2": GET A$: GOTO 40
465 IF MD = M GOTO 480

```

Table de formes

ALPHA

Après avoir saisi ce fichier sous moniteur,
le sauvegarder par :

BSAVE ALPHA, A\$4000, L\$5C0

4000- 80 00 8A 01 A6 01 A8 01	4040- C6 02 D3 02 DD 02 EA 02	40C0- A2 05 A8 05 B8 05 00 00
4008- AF 01 B8 01 C5 01 D4 01	4048- F9 02 05 03 11 03 1D 03	40C8- 00 00 00 00 00 00 00 00
4010- DF 01 EC 01 EF 01 F8 01	4050- 29 03 33 03 3F 03 4C 03	40D0- 00 00 00 00 00 00 00 00
4018- 00 02 0F 02 18 02 1C 02	4058- 56 03 5F 03 6E 03 77 03	40D8- 00 00 00 00 00 00 00 00
4020- 21 02 24 02 2E 02 3E 02	4060- 85 03 93 03 A0 03 AA 03	40E0- 00 00 00 00 00 00 00 00
4028- 47 02 55 02 61 02 6C 02	4068- B7 03 C5 03 D2 03 DB 03	40E8- 00 00 00 00 00 00 00 00
4030- 79 02 85 02 92 02 9F 02	4070- E8 03 F5 03 03 04 14 04	40F0- 00 00 00 00 00 00 00 00
4038- AB 02 AE 02 B2 02 BF 02	4078- 1E 04 2C 04 31 04 3D 04	40F8- 00 00 00 00 00 00 00 00
	4080- 4C 04 53 04 59 04 61 04	4100- 00 00 01 00 00 00 00 00
	4088- 6E 04 79 04 81 04 8A 04	4108- 00 00 00 00 00 00 00 00
	4090- 95 04 9F 04 AC 04 B9 04	4110- 00 00 00 00 00 00 00 00
	4098- C1 04 CB 04 D8 04 E1 04	4118- 00 00 00 00 00 00 00 00
	40A0- EA 04 F3 04 FE 04 09 05	4120- 00 00 02 03 04 05 06 07
	40A8- 14 05 1C 05 27 05 31 05	4128- 08 09 0A 0B 0C 0D 0E 0F
	40B0- 3B 05 45 05 51 05 61 05	4130- 10 11 12 13 14 15 16 17
	40B8- 6E 05 79 05 87 05 93 05	4138- 18 19 1A 1B 1C 1D 1E 1F

```

470 IF (MD - M) * PM < 0 OR PM = 0
    THEN PRINT "ERREUR SUR M": G
    ET A$: GOTO 40
480 HOME : INPUT "COURBE PAR POINT
    S (O)?" ; A$: IF A$ < > "O" THE
    N CO = 1
490 PRINT : INPUT "TRACE DES AXES
    (O)?" ; A$: IF A$ = "O" THEN AX
    = 1
495 HOME
500 REM GRAPHIQUE **
510 T$ = "Y=" + F$: GOSUB 1500
520 RX = (X2 - X1) / 279: RY = (Y2
    - Y1) / 159: IF AX = 0 GOTO 54
    0
524 REM AXES ET GRADUATION
525 X0 = - X1 / RX: Y0 = 159 + Y1
    / RY: IF X0 < 0 OR X0 > 279 GO
    TO 532
530 HPLOT X0, 0 TO X0, 159: YP = INT
    (Y2): IF X0 > 278 GOTO 532
531 YY = (Y1 - YP) / RY + 159: HPL
    OT X0 + 1, YY: YP = YP - 1: IF Y
    P > = Y1 GOTO 531
532 IF Y0 < 0 OR Y0 > 159 GOTO 540
535 HPLOT 0, Y0 TO 279, Y0: XP = INT
    (X2): IF Y0 < 1 GOTO 540
537 XX = (XP - X1) / RX: HPLOT XX,
    Y0 - 1: XP = XP - 1: IF XP > =
    X1 GOTO 537
539 REM COURBE
540 XX = 0
550 X = X1 + XX * RX
560 Y = EXP (X / 2)
570 YY = (Y1 - Y) / RY + 159
580 IF YY < 0 OR YY > 159 THEN IN
    = 0: GOTO 620
590 IF IN * CO THEN HPLOT TO XX,
    YY: GOTO 610
600 HPLOT XX, YY
610 IN = 1
620 XX = XX + 1: IF XX < 280 GOTO
    550
625 IN = 0: IF PM = 0 GOTO 800
630 M = M + PM: IF (MD - M + PM) *
    PM > 0 GOTO 540: REM ITERAT
    ION
640 HOME : INPUT "AUTRE FORMULE (O
    )?" ; A$: IF A$ = "O" GOTO 40
650 END
800 REM DERIVEE **
810 HOME : INPUT "TRACE DE LA DERI
    VEE (O)?" ; A$: IF A$ < > "O" G
    OTO 640
815 XX = 0: HCOLOR= 2
820 X = X1 + XX * RX
825 Y = EXP (X / 2)
830 YD = Y: XD = X: X = X + .001
835 Y = EXP (X / 2)
840 DY = Y - YD: DX = X - XD
850 DER = DY / DX
860 DD = (Y1 - DER) / RY + 159
870 IF DD < 0 OR DD > 159 THEN IN
    = 0: GOTO 910
880 IF IN * CO THEN HPLOT TO XX,
    DD: GOTO 900
890 HPLOT XX, DD
900 IN = 1
910 XX = XX + 1: IF XX < 280 GOTO
    820
920 IN = 0: HCOLOR= 3: GOTO 640
1499 REM AFFICHAGE HGR D'UNE CHAIN
    E T$ **
1500 S = 16384: A = S + PEEK (S +
    2) + 256 * PEEK (S + 3): NB =
    PEEK (S): IC = 1: XB = PEEK
    (A - 6): YB = PEEK (A - 5)
1510 LO = LEN (T$): DX = PEEK (A
    - 8) + IC + 1
1520 FOR I = 1 TO LO
1530 AS = ASC ( MID$ (T$, I, 1))
1540 NF = PEEK (S + 2 * NB + 2 +
    AS)
1550 DRAW NF AT XA + XB, YA + YB
1560 XA = XA + DX: IF XA > 270 THE
    N XA = 0: YA = YA + 10

```

```

4140- 20 21 22 23 24 25 26 27
4148- 28 29 2A 2B 2C 2D 2E 2F
4150- 30 31 32 33 34 35 36 37
4158- 38 39 3A 3B 3C 3D 3E 3F
4160- 40 62 42 43 44 45 46 47
4168- 48 49 4A 4B 4C 4D 4E 4F
4170- 50 51 52 53 54 55 56 57
4178- 58 59 5A 5B 5C 5D 5E 5F
4180- 60 61 05 07 02 03 80 62
4188- BF 05 24 3C 2F 2D 2D 3E
4190- 3F 3F 2E 2D 2D 3E 3F 3F
4198- 2E 2D 2D 3E 3F 3F 2E 2D
41A0- 2D 3E 3F 3F 05 00 02 00
41A8- B6 1C 08 24 24 06 00 1B
41B0- 08 24 0D 36 1E 08 04 00
41B8- 3F 25 64 2D 3F 17 36 36
41C0- 1E 0D 2D 3D 00 24 34 2D
41C8- 3F BF 0E 2D 15 1E 3F 2F
41D0- 35 24 04 00 0C 0C DF 3C
41D8- B6 0A 17 4D 35 27 00 20
41E0- 1C 17 76 17 0D 0D 1E DF
41E8- 0E 6D 07 00 20 34 00 18
41F0- 08 B8 17 36 0E 0E 07 00
41F8- 92 0C 0C 24 1C 1C 05 00
4200- 24 34 1F 0D 0D 1E 3F 16
4208- 2D 15 1F 1F 0D 26 00 24
4210- 36 36 24 3F 2D 2D 07 00
4218- 32 1E 05 00 3F 2D 2D 07
4220- 00 92 04 00 1D 08 1D 08
4228- 1E 1E 1E 1E 05 00 1D 08
4230- 25 1C 3F 17 36 2E 37 0E
4238- 2D 1D 08 24 04 00 24 34
4240- 2F 36 36 3E 2D 07 00 1F
4248- 18 08 0C 2D 15 B6 18 BF
4250- 17 2E 2D 3D 00 DE 72 2D
4258- 1D 08 E4 2C 20 3F 3F 05
4260- 00 21 24 3E 1E 1E 2E 2D
4268- 3D 36 04 00 38 27 2C 2D
4270- B5 AB 36 1E 3F 1F 08 06
4278- 00 2F AD F6 3F 1F 08 24
4280- 64 0C 2D 07 00 17 36 24
4288- 1D 08 1D 08 1D 08 3C 3F
4290- 2F 00 2D 20 1C 3F 17 76
4298- 17 76 2D 1D 08 24 00 3F
42A0- 20 0C 2D 15 36 2F F6 1E
42A8- 3F 05 00 B0 04 00 B0 F6
42B0- 05 00 12 0E 1C 1C 1C 1D
42B8- 08 1D 08 1D 08 07 00 3A

```

```

1570 NEXT : RETURN
1700 REM TRAITEMENT ERREURS
1705 CALL 768
1710 ER = PEEK (222):LI = PEEK (
218) + PEEK (219) * 256
1720 IF (ER = 53 OR ER = 133) AND
LI = 560 THEN IN = 0: GOTO 62
0
1730 IF ER = 16 THEN PRINT "ERREU
R DE SYNTAXE LIGNE ";LI: GET
A$: GOTO 40
1740 IF (ER = 53 OR ER = 133) AND
(LI = 825 OR LI = 835) THEN I
N = 0: GOTO 910
1750 PRINT "ERREUR N("ER" LIGNE "L
I: PRINT CHR$(7): END

```



Programme FORMULE2

```

10 REM TRACE DE COURBES DE FONCTIO
N
20 REM PAR J.L.CHAUVIN 1984
30 D$ = CHR$(4):FIRST = 1: SCALE
= 1: ROT= 0:Q$ = CHR$(34):S =
16384
35 POKE 768,104: POKE 769,168: POK
E 770,104: POKE 771,166: POKE 7
72,223: POKE 773,154: POKE 774,
72: POKE 775,152: POKE 776,72:
POKE 777,96
37 PRINT D$"BLOAD ALPHA,A"S: POKE
232,0: POKE 233,64
40 POKE 34,0: TEXT : HOME : PRINT
TAB(11)"COURBES DE FONCTIONS
50 PRINT TAB(11)"=====
====
60 PRINT : PRINT "CHOISISSEZ ENTRE
:
70 PRINT : PRINT "1-COORDONNEES PA
RAMETRIQUES": PRINT TAB(18)"X

```

```

=F(T), Y=G(T)
80 PRINT : PRINT "2-COORDONNEES PO
LAIRES": PRINT TAB(18)"R=F(T)
90 PRINT : INPUT "VOTRE CHOIX:";CH
OIX$: IF CHOIX$ = "1" GOTO 120
100 IF CHOIX$ = "2" GOTO 200
110 PRINT "CHOIX ERRONE": GOTO 90
119 REM INTRODUCTION X=F(T), Y=G(T
) **
120 HOME : PRINT : PRINT "INTRODUC
TION DES FORMULES": GOSUB 130:
GOTO 150
130 PRINT : INPUT "FORMULE AVEC PA
RAMETRE 'M' (O)?";A$: IF A$ =
"O" THEN PAR = 1
140 IF FIRST THEN PRINT : PRINT "
RESPECTEZ LA SYNTAXE DU BASIC"
: PRINT "VOUS POUVEZ UTILISER
'PI' ET 'E'"
145 RETURN
150 PRINT : PRINT "FORMULES": INP
UT "X=";F$: INPUT "Y=";G$
160 PRINT : PRINT "VALEURS EXTREME
S DE X(SINON:X1=-7,X2=7)": IN
PUT "X1=";X1$: IF X1$ = "" THE
N X1$ = "-7": HTAB 4: VTAB PE
EK (37): PRINT X1$
170 INPUT "X2=";X2$: IF X2$ = "" T
HEN X2$ = "7": HTAB 4: VTAB P
EEK (37): PRINT X2$
180 PRINT : PRINT "VALEURS EXTREME
S DE Y(SINON:Y1=-4,Y2=4)": IN
PUT "Y1=";Y1$: IF Y1$ = "" THE
N Y1$ = "-4": HTAB 4: VTAB PE
EK (37): PRINT Y1$
190 INPUT "Y2=";Y2$: IF Y2$ = "" T
HEN Y2$ = "4": HTAB 4: VTAB P
EEK (37): PRINT Y2$
195 GOTO 240
199 REM INTRODUCTION R=F(T) **
200 HOME : PRINT "INTRODUCTION DE
LA FORMULE": GOSUB 130
210 PRINT : PRINT "FORMULE": INPU

```

```

42C0- 27 28 2D B5 3F 00 12 17
42C8- 0C 0C 0C 1F 08 1F 08 1F
42D0- 08 05 00 B6 1C 08 64 0C
42D8- 1C 3F 17 04 00 38 0C 70
42E0- 0E 17 15 3F BF 0E 2D 25
42E8- 04 00 3A 37 26 24 64 0C
42F0- 1E 08 0E 15 36 2F 36 04
42F8- 00 3F 24 2C 2D 15 36 36
4300- F6 3F 27 24 00 1B 24 0C
4308- 2D 15 96 F2 3F 1F 08 24
4310- 00 1B 24 2C 2D 15 36 36
4318- 1E 3F 27 24 00 AD 12 3F
4320- 3F 24 2C 27 24 2D 2D 07
4328- 00 3D 3F 36 26 24 24 2C
4330- 2D 3D 00 2A 35 3E 3F 1F
4338- 08 24 24 0C 2D 3D 00 2D

```

```

4340- 36 26 24 24 FC 1B 36 2E
4348- 37 36 04 00 24 3C 2D 37
4350- 36 36 3E 2D 07 00 98 72
4358- 2D 1D 08 24 24 34 00 3B
4360- 36 6E 09 1C 1C 1C 08 1D
4368- 08 0C DF 33 36 00 1B 24
4370- 34 36 36 2E 2D 3D 00 1F
4378- 36 26 24 24 34 4D 0C 25
4380- 36 36 36 04 00 1F 08 27
4388- 34 36 36 6E 09 24 2F 24
4390- 24 06 00 1B 36 0E 2D 1D
4398- 08 24 24 1C 3F 17 36 00
43A0- 2D 20 1C 3F 37 36 3D 36
43A8- 26 00 72 1D 08 24 E4 3F
43B0- 17 36 36 0E 6D 05 00 2D
43B8- 20 1C 3F 37 36 3D 36 26

```

```

43C0- 6C 15 15 04 00 3F 20 0C
43C8- 2D 15 D6 15 F6 3F 1F 08
43D0- 06 00 36 26 24 24 3C 2F
43D8- 2D 3D 00 1B 24 34 36 36
43E0- 0E 2D 1D 08 24 24 34 00
43E8- 1B 24 34 36 76 15 0C 1D
43F0- 08 24 24 06 00 1F 24 34
43F8- 36 36 26 2D 7D 35 24 24
4400- 24 06 00 1C 1F 08 34 96
4408- 32 64 0D 15 26 1C 08 18
4410- 1D 08 34 00 36 26 24 1C
4418- 1C 6C 09 F6 07 00 1E 1E
4420- 2E 2D 25 18 08 18 1D 08
4428- 3C 3F 2F 00 20 F0 0D 07
4430- 00 12 16 0C 2C 1C 08 38
4438- 3F 17 76 2D 00 1B 1C 08

```

```

T "R=";F$
220 PRINT : PRINT "VALEUR EXTREME
DE R (SINON R=4)": INPUT "R1="
;R1$: IF R1$ = "" THEN R1$ = "
4": HTAB 4: VTAB PEEK (37): P
RINT R1$
240 PRINT : PRINT "VARIATIONS DE T
(SINON:DE -PI A PI)": INPUT
"DE T1=";T1$: IF T1$ = "" THEN
T1$ = "-PI": HTAB 7: VTAB PE
EK (37): PRINT T1$
245 INPUT "A T2=";T2$: IF T2$ = ""
THEN T2$ = "PI": HTAB 6: VTAB
PEEK (37): PRINT T2$
250 M$ = "0":MD$ = "0":PM$ = "0":
IF PAR THEN PRINT : PRINT "VA
LEURS DE M:": INPUT "DE M=";M$
: INPUT "A M=";MD$: INPUT "PAS
=";PM$
255 PRINT : INPUT "VOULEZ-VOUS REP
RENDRE CES DONNEES (O)?" ;A$: I
F A$ = "O" GOTO 40
260 HGR : HOME : SCALE= 1: ROT= 0:
HCOLOR= 3
300 REM CREATION PROCEDURE POUR I
NTRODUIRE LES FONCTIONS DANS L
E PROGRAMME *
310 PRINT D$;"OPENPROCEDURE": PRIN
T D$;"WRITEPROCEDURE
320 IF CHOIX$ = "1" THEN PRINT "9
20X="F$":Y="G$: PRINT "460X$="
Q$ + F$ + Q$":Y$="Q$ + G$ + Q$
: GOTO 340
330 PRINT "920X=COS(T)*(F$)":Y=SI
N(T)*(F$)": PRINT "490R$="Q$
+ F$ + Q$: PRINT "480R1 = "R1
$: GOTO 350
340 PRINT "440X1="X1$":X2="X2$":Y1
="Y1$":Y2="Y2$
350 PRINT "420T1="T1$":T2="T2$":M=
"M$":MD="MD$":PM="PM$":CHOIX="
CHOIX$
360 PRINT "RUN400
370 PRINT D$"CLOSEPROCEDURE
380 PRINT D$"EXECPROCEDURE
390 END : REM ARRET POUR DECLENCH
ER EXEC
399 REM PREPARATION DONNEES **
400 PI = 4 * ATN (1):E = EXP (1)
: ONERR GOTO 1700
410 D$ = CHR$ (4):Q$ = CHR$ (34)
: PRINT D$"DELETEPROCEDURE
420 T1 = - PI:T2 = PI:M = 0:MD =
PI / 2:PM = PI / 12:CHOIX = 1
425 POKE 34,20: HOME
430 IF CHOIX = 2 GOTO 480
440 X1 = - 7:X2 = 7:Y1 = - 4:Y2
= 4
450 IF X1 = X2 OR Y1 = Y2 THEN PR
INT "ERREUR SUR X1,X2,Y1,Y2":
GET A$: GOTO 40
460 X$ = "2.5*COS(T)*SIN(M)":Y$ =
"4*SIN(T)"
470 GOTO 500
480 R1 = 4
490 R$ = "SINX"
495 IF R1 < = 0 THEN "ERREUR SUR
R1": GET A$: GOTO 40
500 IF MD = M GOTO 520
510 IF (MD - M) * PM < 0 OR PM = 0
THEN PRINT "ERREUR SUR M": G
ET A$: GOTO 40
520 HOME : INPUT "COURBE PAR POINT
S (O)?" ;A$: IF A$ < > "O" THE
N CO = 1
530 PRINT : INPUT "TRACE DES AXES
(O)?" ;A$: IF A$ = "O" THEN AX
= 1
600 REM GRAPHIQUE **
610 IF CHOIX = 1 THEN T$ = "X=" +
X$: GOSUB 1500:XA = 0:YA = YA
+ 10:T$ = "Y=" + Y$: GOSUB 150
0: GOTO 640
620 T$ = "R=" + R$: GOSUB 1500
630 X2 = 1.75 * R1:X1 = - X2:Y2 =
R1:Y1 = - R1

```

```

4440- 0C 2D B5 E2 3F 16 2D CD      44C0- 00 28 1C 08 16 36 36 1F      4540- 1D 08 24 06 00 1F 34 36
4448- F2 3F 2F 00 F0 17 0C 0D      44C8- 3F 30 00 29 B0 12 1F 08      4548- 2E 1D 08 34 0E 25 24 34
4450- 15 04 00 92 3A 2F 2D 3D      44D0- 1C 3F 36 24 24 24 06 00      4550- 00 3B B0 12 1D 08 0D 15
4458- 00 19 08 1F 08 1F 08 05      44D8- 24 3C 35 36 36 3E 2D 07      4558- 1F 08 1C 1D 08 1D 08 06
4460- 00 1B 08 2D 15 36 3E 3F      44E0- 00 36 DE 24 24 6D 35 36      4560- 00 1B 34 76 16 2D 1D 08
4468- 1F 08 0C 2D 05 00 28 15      44E8- 26 00 28 15 36 FE 1B 24      4568- 3C 2F 25 24 06 00 38 2F
4470- 36 1E 3F 27 24 2C 27 34      44F0- 24 2D 00 38 17 36 0E 2D      4570- 2D F5 17 17 17 2D 2D 07
4478- 00 2D 96 3F 3F 20 0C 05      44F8- 1D 08 24 1C 07 00 12 2D      4578- 00 60 0C D6 15 3E 3F 1C
4480- 00 BF 76 2D 25 24 2F 24      4500- 20 1C 3F 37 36 3D 36 04      4580- 08 17 36 0E 2D 3D 00 09
4488- 04 00 2A 25 1C 3F 17 2E      4508- 00 28 35 36 36 24 3F 3F      4588- 34 36 2F FE 3F 20 24 44
4490- 37 0E 2D 3D 00 3D 3F 35      4510- 20 0C 05 00 28 3D FF 2E      4590- 15 07 00 28 1C 08 15 16
4498- 36 24 24 64 AD 04 00 38      4518- 37 36 04 00 09 38 3F 17      4598- 3E 3F 1C 08 17 36 0E 2D
44A0- 17 76 16 2D 1D 08 3C 2F      4520- 0E 2D 15 1E 3F 2F 00 28      45A0- 3D 00 C3 18 08 4D 39 00
44A8- 25 E4 07 00 28 15 36 FE      4528- 3F 2F 24 36 36 76 2D 30      45A8- 1F 1C 08 0D 0D 1E 1F 0E
44B0- 1B 1E 08 24 24 3D 24 06      4530- 00 09 34 36 3E F0 3F 20      45B0- 0D 1E 1F 17 0D 0D 07 00
44B8- 00 24 B0 2F 36 36 2F 3D      4538- 24 06 00 1B 34 76 15 0C      45B8- 92 3A 2F 2D 2D 07 00 D0

```

```

640 RX = (X2 - X1) / 279:RY = (Y2
- Y1) / 159: IF AX = 0 GOTO 85
0
650 REM AXES ET GRADUATION
660 X0 = - X1 / RX:Y0 = 159 + Y1
/ RY: IF X0 < 0 OR X0 > 279 GO
TO 690
670 HPLOT X0,0 TO X0,159:YP = INT
(Y2): IF X0 > 278 GOTO 690
680 YY = (Y1 - YP) / RY + 159: HPL
OT X0 + 1,YY:YP = YP - 1: IF Y
P > = Y1 GOTO 680
690 IF Y0 < 0 OR Y0 > 159 GOTO 850
700 HPLOT 0,Y0 TO 279,Y0:XP = INT
(X2): IF Y0 < 1 GOTO 700
710 XX = (XP - X1) / RX: HPLLOT XX,
Y0 - 1:XP = XP - 1: IF XP > =
X1 GOTO 710
850 DT = PI / 120 * SGN (T2 - T1)
860 HOME
900 REM TRACE DE LA COURBE **
910 T = T1:IN = 0
920 X = 2.5 * COS (T) * SIN (M):
Y = 4 * SIN (T)
930 XX = (X - X1) / RX:YY = (Y1 -
Y) / RY + 159
940 IF XX < 0 OR XX > 279 OR YY <
0 OR YY > 159 THEN IN = 0: GOT
O 980
950 IF IN * CO THEN HPLOT TO XX,
YY: GOTO 970
960 HPLOT XX,YY
970 IN = 1
980 T = T + DT: IF (T - T1) * (T -
T2) < = 0 GOTO 920
990 IN = 0: IF PM = 0 GOTO 1010
1000 M = M + PM: IF (MD - M + PM)
* PM > 0 GOTO 850: REM ITERA
TION
1010 HOME : INPUT "AUTRE FORMULE (
O)?" :A$: IF A$ = "O" GOTO 40
1020 END
1499 REM AFFICHAGE HGR D'UNE CHAIN
E T$ **
1500 S = 16384:A = S + PEEK (S +
2) + 256 * PEEK (S + 3):NB =
PEEK (S):IC = 1:XB = PEEK
(A - 6):YB = PEEK (A - 5)
1510 LO = LEN (T$):DX = PEEK (A
- 8) + IC + 1
1520 FOR I = 1 TO LO
1530 AS = ASC ( MID$ (T$,I,1))
1540 NF = PEEK (S + 2 * NB + 2 +
AS)
1550 DRAW NF AT XA + XB,YA + YB
1560 XA = XA + DX: IF XA > 270 THE
N XA = 0:YA = YA + 10
1570 NEXT : RETURN
1700 REM TRAITEMENT ERREURS
1705 CALL 768
1710 ER = PEEK (222):LI = PEEK (
218) + PEEK (219) * 256
1715 IF ER = 16 THEN PRINT "ERREU
R DE SYNTAXE": GET A$: GOTO 4
0
1720 IF (ER = 53 OR ER = 133) AND
LI = 920 THEN IN = 0: GOTO 98
0
1760 PRINT "ERREUR N["ER" LIGNE "L
I: PRINT CHR$ (7): END

```

*Vous avez un Apple IIe avec Chat Mauve ou un IIc ?
Vous disposez de Pascal 1.2 ?*

Profitez de l'interpréteur **COGO**

Par Nicolas Montsarrat

Il s'agit d'un système graphique double-haute résolution écrit en Pascal. COGO vous permet de manipuler des graphiques grâce à un langage de description des objets - points, angles- et à l'emploi de fonctions primitives de manipulation très puissantes : cercle, tangente, intersections, parallèles, etc. Il est ainsi possible de tracer des grilles, des cercles, des segments de droite, des tangentes communes à deux cercles, de calculer des distances, des angles...

L'éditeur permet une saisie rapide du langage. Une instruction COGO peut-être exécutée dès la saisie pour faciliter la mise au point, ou au sein d'un programme.

Ce programme, destiné à résoudre des problèmes de géométrie plane, comporte des instructions de stockage sur fichier afin de permettre la reprise d'un calcul.

NB : Sur l'Apple IIc, l'affichage se fait en simple haute résolution.

150,00 F TTC, franco
Bon de commande page 74

'Startup' Basic sous ProDOS

Pascal Cantot

Voici un programme de **STARTUP**, presque entièrement en Basic Applesoft. Il occupe 17 blocs, mais le ProDOS le charge en quelques secondes. Pour une exécution plus rapide, il est possible de retirer les REMs (à l'aide d'un compacteur par exemple...).

STARTUP utilise les 80 colonnes. Vous ne devez donc l'utiliser qu'avec un //c ou un //e muni d'une carte 80 colonnes au standard d'Apple.

Une fois chargé, ce programme affiche le préfixe courant, la date, et éventuellement un titre correspondant à sa disquette de résidence (ce titre est fixé en data à la ligne 1205). Il va maintenant lire le répertoire principal du volume, puis affiche les premiers fichiers.

Le fichier courant apparaît en vidéo inversée. Pour changer ce pointeur vidéo, ou pour effectuer une opération sur le fichier courant, une trentaine de commandes sont disponibles :

Déplacement

- Les flèches ←↑⇒ et ↓ déplacent d'un fichier, **CTRL-W** et **CTRL-X** remonte ou redescend de 8 fichiers.
- **CTRL-A** et **CTRL-Z** ramène au début ou à la fin du catalogue actuel.
- **CTRL-F** place le pointeur en vidéo inverse sur le fichier dont le nom a été saisi (voir plus loin).

Sous-catalogue

- **CTRL-I** ouvre le fichier de

type DIR couramment pointé, lit ce sous-catalogue et en affiche les fichiers.

- **DEL** referme un sous-catalogue et réaffiche les fichiers du catalogue de niveau supérieur.

Utilitaires

- **ESC** ou **CTRL-C** permettent de sortir de **STARTUP** après confirmation.
- **?** affiche un écran d'aide récapitulant l'ensemble des commandes.
- **CTRL-D** permet de saisir la date sous la forme JJ-**MMM**-**AA**. Le mois doit être tapé en lettres majuscules (et en français). Cette date sera inscrite sur les pages globales de ProDOS et pourra donc être utilisée pendant le restant de la session.
- L'heure est réglée par les touches : **{}** pour les heures, **/** pour les minutes, (pour un clavier en mode US).

Opérations générales

- **/** permet d'introduire un nouveau préfixe au clavier.
- **1** ou **2** change le numéro de drive (le préfixe est automatiquement reconnu). C'est également un moyen pratique de revenir directement au catalogue principal d'un volume.
- **CTRL-P** imprime le répertoire (ou sous-répertoire) courant. L'interface doit être en slot 1. Aucun code de contrôle particulier n'est envoyé mais la ligne 620 du programme est à votre disposition...

Opérations sur les fichiers

- **RETURN** charge et exécute le fichier **SYS**, **BAS**, **TXT** ou **BIN** pointé.
- **CTRL-L** le charge seulement.
- ***** verrouille ou déverrouille le fichier.
- **CTRL-V** le vérifie
- **CTRL-E** l'efface (après confirmation). On peut effacer un fichier de type **DIR** mais le nombre de blocs disponibles ne sera plus correctement affiché.
- **CTRL-R** renomme le fichier.

En cas d'impossibilité d'exécuter la commande, quatre événements peuvent se produire :

- La commande est ignorée (cas d'un ^L sur un fichier **VAR** ou **TXT**)
- Il y a un message d'erreur et le programme continue normalement (cas d'une saisie de date incorrecte).
- Il y a un message d'erreur ProDOS et le programme redémarre à froid, par relecture du répertoire (cas d'un I/O **ERROR** ou d'un **WRITE PROTECTED**).
- Il y a une erreur Applesoft : une improbable (?) erreur de ma part...

Signalons que ce programme ignore le verrouillage des fichiers : **^D** efface un fichier verrouillé et **^R** laisse le verrouillage en place. Enfin, le programme utilise la commande **FRE** de ProDOS : vérifiez que votre version de ProDOS la supporte.



Programme **STARTUP**

```

1 REM *****
2 REM * STARTUP ProDOS *
3 REM *-----*
4 REM * P.Cantot, POM'S *
5 REM *****
10 DIM A$(100,5),ER$(19),CM$(20),NF(5),PX
    (5),COM$(36)
15 D$ = CHR$(4): PRINT D$"PR#3": PRINT D
    $"PREFIX": INPUT PR$:A$(0,0) = PR$
20 FOR I = 768 TO 828: READ J: POKE I,J:
    NEXT :PLAY = 768:ERR = 796:BIP = 807
25 VTAB 23: FOR I = 1 TO 10: PRINT "
    ";: NEXT : POKE 34,2: POKE 35,22
30 GOSUB 1075: FOR I = 0 TO 19: READ ER$(
    I): NEXT
35 DT = 49040:MO$ = "???JANFEVMARAVRMAIJUN
    JULAHOUSEPOCTNOVDEC": GOSUB 920
40 CR$ = CHR$(15) + "Return" + CHR$(14
    ):SP$ = "
45 ONERR GOTO 840
50 POKE 34,0: VTAB 1: POKE 1403,0: PRINT
    CHR$(15); CHR$(26);PR$; CHR$(14):
    PRINT : GOSUB 745
55 VTAB 12: PRINT : PRINT CHR$(15)A$(Y,
    X) CHR$(14)
60 GOSUB 950
65 VTAB 24: POKE 1403,0: PRINT CHR$(26)
    "Entrez commande ou <?> pour aide... "
    ;
70 GOSUB 260: PRINT D$"FRE"
75 IF KEY = 8 OR KEY = 11 THEN GOSUB 330
    : GOTO 65
80 IF KEY = 21 OR KEY = 10 THEN GOSUB 31
    0: GOTO 65
85 IF KEY = 127 AND X THEN X = X - 1:Y =
    P(X): GOSUB 795: GOTO 65
90 IF NOT NF(X) THEN 155
95 IF KEY = 9 AND MID$(A$(Y,X),18,3) =
    "DIR" AND X < 5 THEN 350
100 IF KEY = 42 THEN 380
105 IF KEY = 22 THEN PRINT D$"VERIFY" MI
    D$(A$(Y,X),2,15): CALL BIP: GOTO 65
110 IF KEY = 13 THEN 710
115 IF KEY = 12 THEN 685
120 IF KEY = 18 THEN 405
125 IF KEY = 5 THEN 455
130 IF KEY = 1 THEN Y = 1: GOSUB 805: GOT
    O 55
135 IF KEY = 26 THEN Y = NF(X): GOSUB 805
    : GOTO 55
140 IF KEY = 23 AND Y > 8 THEN Y = Y - 8:
    GOSUB 805: GOTO 55
145 IF KEY = 24 AND Y < NF(X) - 7 THEN Y
    = Y + 8: GOSUB 805: GOTO 55
150 IF KEY = 6 THEN 510
155 IF KEY = 47 THEN 560
160 IF KEY = 91 THEN MN = MN - 1: GOTO 10
    20
165 IF KEY = 93 THEN MN = MN + 1: GOTO 10
    20
170 IF KEY = 123 THEN HH = HH - 1: GOTO 1
    020
175 IF KEY = 125 THEN HH = HH + 1: GOTO 1
    020
180 IF KEY = 4 THEN 970
185 IF KEY = 63 THEN 1105
190 IF KEY = 49 THEN 590
195 IF KEY = 50 THEN 595
200 IF KEY = 27 THEN CALL BIP: CALL BIP:
    GOTO 895
205 IF KEY = 16 THEN 605
210 CALL PLAY,64,12: GOTO 65
215 END
220 REM --* AFFICHE BLOCKS LIBRES *--
225 VTAB 1: POKE 1403,49: PRINT CHR$(15
    )"BLOCKS LIBRES: "BF" SUR "BO + BF; C
    HR$(14)
230 RETURN
235 VTAB 24: PRINT " ou "CR$" " CHR$(29)
    ;
240 POKE 34,23: POKE 35,25: INPUT A$: POK
    E 34,4: POKE 35,22
245 IF A$ = "" THEN 65
250 RETURN
255 REM --* CURSEUR *--
260 W = PEEK(1403):KEY = 0
265 POKE 1403,W: PRINT " _";: GOSUB 285: I
    F KEY THEN 280
270 POKE 1403,W: PRINT " ";: GOSUB 285: I
    F NOT KEY THEN 265
275 IF PEEK(-16384) > 127 THEN 380
280 POKE 1403,W: PRINT " ";: RETURN
285 FOR I = 1 TO 40: IF PEEK(-16384)
    > 127 THEN KEY = PEEK(-16384) - 1
    28: POKE -16368,0:I = 40
290 NEXT : RETURN
295 IF PEEK(1063) < > 42 THEN POKE 10
    63,42: RETURN
300 POKE 1063,32: RETURN
305 REM --* DEPLACE CURSEUR *--
310 IF Y > = NF(X) THEN CALL PLAY,64,12
    : RETURN
315 VTAB 12: PRINT : NORMAL : PRINT A$(Y,
    X); CHR$(23): IF Y < NF(X) - 9 THEN
    VTAB 22: PRINT A$(Y + 10,X);: POKE 1
    403,0
320 VTAB 13: INVERSE :Y = Y + 1: PRINT A$
    (Y,X); CHR$(14)
325 RETURN
330 IF Y = 1 THEN CALL PLAY,64,12: RETUR
    N
335 VTAB 12: PRINT : NORMAL : PRINT A$(Y,
    X); CHR$(22): IF Y > 9 THEN VTAB 5:
    PRINT A$(Y - 9,X)
340 VTAB 13: INVERSE :Y = Y - 1: PRINT A$
    (Y,X); CHR$(14)
345 RETURN
350 A$ = MID$(A$(Y,X),2,15):P(X) = Y:X =
    X + 1
355 J = 15: FOR I = 1 TO 15: IF ASC(MID
    $(A$,I)) = 32 THEN J = I - 1:I = 15
360 NEXT :A$(0,X) = LEFT$(A$,J) + "/"
365 GOSUB 745
370 GOTO 65
375 REM --* LOCK/UNLOCK *--
380 A$ = "LOCK":I = 42: IF ASC(A$(Y,X))
    = 42 THEN A$ = "UNLOCK":I = 32
385 PRINT D$;A$; MID$(A$(Y,X),2,15)
390 A$(Y,X) = CHR$(I) + MID$(A$(Y,X),2
    )
395 GOTO 55
400 REM --* RENAME *--
405 VTAB 24:W$ = MID$(A$(Y,X),2,15): PO
    KE 1403,0
410 PRINT "Nouveau nom";
415 GOSUB 235
420 I = 0: IF ASC(A$(Y,X)) = 42 THEN I =
    1
425 PRINT D$"UNLOCK"W$:A$(Y,X) = " " + M
    ID$(A$(Y,X),2)
430 PRINT D$"RENAME"W$", "A$
435 A$(Y,X) = " " + LEFT$(A$ + SP$,15) +
    MID$(A$(Y,X),17)
440 IF I THEN PRINT D$"LOCK"A$:A$(Y,X) =
    "*" + MID$(A$(Y,X),2)
445 GOTO 55
450 REM --* DELETE *--

```

```

455 VTAB 24: POKE 1403,0: PRINT CHR$(29
)"Attention: le fichier sera DEFINITI
VEMENT efface! Continue (O/N) ? ";
460 CALL BIP: CALL BIP
465 GOSUB 260: IF KEY < > 79 AND KEY <
> 111 THEN 55
470 J = VAL ( MID$( A$(Y,X),21,9))
475 A$ = MID$( A$(Y,X),2,15): PRINT D$"UN
LOCK"A$: PRINT D$"DELETE"A$
480 FOR I = Y TO NF(X): A$(I,X) = A$(I +
1,X): NEXT
485 NF(X) = NF(X) - 1: IF Y > NF(X) THEN Y
= NF(X)
490 BO = BO - J: BF = BF + J: GOSUB 225
495 GOSUB 805
500 PRINT D$"FRE": GOTO 55
505 REM ==* RECHERCHE FICHER *==
510 VTAB 24: POKE 1403,0: PRINT "Nom du f
ichier";
515 GOSUB 235: PRINT D$"FRE"
520 I = 1: A$ = LEFT$( A$ + SP$,15)
525 IF A$ = MID$( A$(I,X),2,15) THEN 540
530 I = I + 1: IF I < = NF(X) THEN 525
535 CALL BIP: CALL BIP: GOTO 55
540 Y = I: CALL PLAY768,32,32
545 GOSUB 805
550 GOTO 55
555 REM ==* CHANGER PREFIXE *==
560 VTAB 24: POKE 1403,0: PRINT "Prefixe"
;
565 GOSUB 235: VTAB 5
570 PRINT D$"PREFIX"A$
575 VTAB 5: PRINT D$"FRE"
580 PRINT D$"PREFIX": INPUT PR$: X = 0
585 A$(0,0) = PR$: GOTO 50
590 PRINT D$"PREFIX,D1": GOTO 575
595 PRINT D$"PREFIX,D2": GOTO 575
600 REM ==* IMPRIME LE REPERTOIRE *==
605 VTAB 24: POKE 1403,0: PRINT CHR$(26
)"Verifiez l'imprimante et appuyez su
r "CR$" ";
610 GOSUB 260: IF KEY < > 13 THEN CALL
PLAY64,12: GOTO 610
615 PRINT D$"PR#1"
620 REM éventuellement codes ctrl de votr
e impr.ici
625 PRINT DT$" "TI$: PRINT
630 PRINT "PREFIX: "PR$: PRINT : PRINT T$
: PRINT
635 PRINT : FOR I = 1 TO NF(X): PRINT A$(
I,X): NEXT
640 PRINT : PRINT "BLOCKS LIBRES: "BF"
BLOCKS OCCUPES: "BO" TOTAL: "
BO + BF
645 PRINT D$"PR#3"
650 POKE 34,0: VTAB 1: PRINT CHR$(15) C
HR$(26); A$(0,0); CHR$(14)
655 PRINT PR$: GOSUB 920
660 INVERSE : PRINT T$" "": NORMAL
665 VTAB 23: FOR I = 1 TO 20: PRINT "
": NEXT : POKE 35,22
670 GOSUB 805
675 GOTO 55
680 REM ==* LOAD/EXECUTE FICHER *==
685 A$ = MID$( A$(Y,X),2,15): W$ = MID$(
A$(Y,X),18,3)
690 IF W$ = "BAS" THEN PRINT D$"PR#3": P
RINT D$"LOAD"A$
695 IF W$ < > "BIN" THEN 210
700 PRINT D$"PR#3": PRINT D$"BLOAD"A$
705 END
710 A$ = MID$( A$(Y,X),2,15)
715 CALL BIP
720 VTAB 24: POKE 1403,0: PRINT CHR$(26
)"Appuyez sur " CHR$(15) " C " CHR$(

```

```

14)" pour executer "A$: ";
725 GOSUB 260: IF KEY < > 67 THEN 210
730 PRINT D$"PR#3"
735 PRINT D$"A$
740 END
745 REM ==* LIT UN DIRECTORY *==
750 Y = 0: NF(X) = 999: PR$ = "": FOR I = 0
TO X: PR$ = PR$ + A$(0,I): NEXT
755 PRINT D$"OPEN"PR$", TDIR": PRINT D$"RE
AD"PR$
760 INPUT A$, T$, A$
765 INPUT A$: IF A$ = "" THEN INPUT "": A
$: GOTO 780
770 A$(Y + 1,X) = A$: Y = Y + 1: IF Y / 2 =
INT (Y / 2) THEN GOSUB 295
775 GOTO 765
780 PRINT D$"CLOSE"
785 BF = VAL ( MID$( A$,13)): BO = VAL (
MID$( A$,35)): GOSUB 225: PRINT
790 VTAB 2: PRINT : NF(X) = Y: POKE 1063,3
2: PRINT CHR$(15); T$" " CHR$(14):
POKE 34,4: PRINT : Y = 1
795 A$ = "": FOR I = 0 TO X: A$ = A$ + A$(
,I): NEXT : PRINT D$"PREFIX"A$
800 REM ==* AFFICHE UN REPERTOIRE *==
805 POKE 34,0: VTAB 2: POKE 1403,0: POKE
33,49: PRINT CHR$(29); : POKE 33,80:
FOR I = 0 TO X: PRINT A$(0,I); : NEXT
: GOSUB 920: PRINT : PRINT : POKE 34
,4
810 HOME : Z1 = Y - 8: IF Z1 < 1 THEN Z1 =
1
815 Z2 = Y + 9: IF Z2 > NF(X) THEN Z2 = NF
(X)
820 FOR I = Z2 TO Z1 STEP - 1: VTAB 13 +
(I - Y): POKE 1403,0: PRINT A$(I,X);
: NEXT : PRINT
825 INVERSE : VTAB 13: PRINT A$(Y,X): NOR
MAL
830 RETURN
835 REM ==* TRAITEMENT D'ERREUR *==
840 CALL ERR: W = PEEK (222): CALL BIP: C
ALL BIP
845 PRINT CHR$(4)"CLOSE"
850 IF W = 255 THEN 895
855 VTAB 24: POKE 1403,0: PRINT CHR$(26
); "ERREUR ";
860 IF W AND W < 22 THEN PRINT "ProDOS:
"ER$(W - 2); : GOTO 870
865 PRINT "APPLESOFT numero "W";
870 PRINT " , ligne " PEEK (218) + PEEK (
219) * 256;
875 PRINT ". Appuyez sur "CR$"... ";
880 GOSUB 260: IF KEY < > 13 THEN 880
885 RUN
890 PRINT D$"FRE"
895 VTAB 24: POKE 1403,0: PRINT CHR$(26
)"Quitter (O/N) ? ";
900 GOSUB 260: IF KEY < > 79 AND KEY <
> 111 THEN 55
905 PRINT D$"PR#3": POKE 783,80: CALL PLA
Y,0,0: POKE 783,81: PRINT D$"PREFIX"A
$(0,0)
910 END
915 REM ==* GESTION DATE *==
920 REM *** Lecture et affichage
925 W = 0: Z1 = PEEK (DT + 1): Z2 = PEEK (
DT)
930 IF INT (Z1 / 2) * 2 < > Z1 THEN Z1
= Z1 - 1: Z2 = Z2 + 128: W = 4
935 AA = INT (Z1 / 2): MM = INT (Z2 / 32)
+ W: JJ = Z2 - INT (Z2 / 32) * 32
940 IF JJ = 0 OR JJ > 31 OR AA > 99 OR MM
= 0 OR MM > 12 THEN DT$ = "Sans date
": GOTO 950

```

```

945 DT$ = RIGHT$ ("0" + STR$ (JJ),2) + "
    -" + MID$ (MO$,MM * 3 + 1,3) + "-" +
    RIGHT$ ("0" + STR$ (AA),2)
950 HH = PEEK (DT + 3):MN = PEEK (DT + 2
)
955 TI$ = RIGHT$ ("0" + STR$ (HH),2) + "
:" + RIGHT$ ("0" + STR$ (MN),2)
960 VTAB 2: POKE 1403,49: PRINT CHR$ (15
) " Date: "DT$" Time: "TI$; CHR$ (29
) CHR$ (14)
965 RETURN
970 REM *** Entree de la date ***
975 VTAB 24: POKE 1403,0: PRINT "Entrez d
ate (JJ-AAA)"::: GOSUB 235
980 W = VAL ( LEFT$ (A$,2)): IF W < 1 OR
W > 31 THEN GOTO 1055
985 JJ = W:W$ = MID$ (A$,4,3):W = 1: FOR
I = 1 TO 12: IF W$ = MID$ (MO$,I * 3
+ 1,3) THEN MM = I:W = 0
990 NEXT : IF W THEN 1055
995 AA = VAL ( MID$ (A$,8,2))
1000 Z2 = MM * 32:Z1 = AA * 2: IF Z2 > 255
THEN Z2 = Z2 - 256:Z1 = Z1 + 1
1005 Z2 = Z2 + JJ
1010 POKE DT,Z2: POKE DT + 1,Z1
1015 GOSUB 920: GOTO 65
1020 REM *** Regler heure ***
1025 IF MN > 59 THEN HH = HH + 1:MN = 0
1030 IF MN < 0 THEN MN = 59:HH = HH - 1
1035 IF HH > 23 THEN HH = 0
1040 IF HH < 0 THEN HH = 23
1045 POKE DT + 3,HH: POKE DT + 2,MN
1050 GOSUB 955: GOTO 65
1055 VTAB 24: POKE 1403,0: PRINT CHR$ (2
6)"ERREUR: date incorrecte! ";
1060 CALL BIP: CALL BIP: GOSUB 260
1065 GOTO 65
1070 REM ==* GNERIQUE ==*
1075 VTAB 6
1080 READ A$: IF A$ = "à" THEN 1095
1085 POKE 36,40 - LEN (A$) / 2: PRINT A$
: CALL PLAY,8,16: PRINT : PRINT
1090 FOR I = 1 TO 200: NEXT : GOTO 1080
1095 RETURN
1100 REM ==* AFFICHE COMMANDES ==*
1105 IF COM THEN 1130
1110 I = 0
1115 READ A$,W$: IF A$ = "à" THEN 1125
1120 COM$(I) = A$ + LEFT$ (".....",8 -
LEN (A$)) + W$:I = I + 1: GOTO 1115
1125 COM = I
1130 POKE 34,4: POKE 35,22: HOME
1135 POKE 1403,26: PRINT "--* LISTE DES C
OMMANDES ==*": PRINT
1140 FOR I = 0 TO COM STEP 2
1145 PRINT COM$(I)::: POKE 1403,40: PRINT
COM$(I + 1)
1150 NEXT
1155 VTAB 24: PRINT "Appuyez sur une touc
he... "; CHR$ (29)::: GOSUB 260
1160 GOSUB 805
1165 GOTO 55
1170 REM
1175 DATA 32,190,222,32,70,231,134,81,44,
48,192,136,208,4,198,81,240,8,202,208
,246,166,80,76,8,3,96,0
1180 DATA 104,168,104,166,223,154,72,152,
72,96,0
1185 DATA 162,32,169,2,32,168,252,141,48,
192,169,36,32,168,252,141,48,192,202,
208,237,96
1190 REM
1195 DATA STARTUP ProDOS,Par Pascal Canto
t pour POM'S
1200 REM (Nom de la disquette) vv
1205 DATA DISQUETTE DE DEMONSTRATION
1210 DATA à
1215 REM
1220 DATA RANGE ERROR,NO DEVICE CONNECTED
,WRITE PROTECTED
1225 DATA END OF DATA,PATH NOT FOUND,PATH
NOT FOUND
1230 DATA I/O ERROR,DISK FULL,FILE LOCKED
1235 DATA INVALID OPTION,NO BUFFERS AVAIL
ABLE
1240 DATA FILE TYPE MISMATCH,PROGRAM TOO
LARGE
1245 DATA NOT DIRECT COMMAND,SYNTAX ERROR
,DIRECTORY FULL
1250 DATA FILE NOT OPEN,DUPLICATE FILENAM
E
1255 DATA FILEBUSY,FILE(S) STILL OPEN
1260 REM
1265 DATA ESC, Quitter, Fleches, defilement
directory
1270 DATA ^W ^X,Deplacent de 8 fichiers,^
A ^Z,Debut/Fin du repertoire
1275 DATA Tab,Ouvre le fichier DIR,Del,Fe
rme le fichier DIR
1280 DATA Return,Execute,^L,Charge (BAS e
t BIN)
1285 DATA ^D,change Date,eè[],reglent heu
res/minutes
1290 DATA /,Change prefixe,1/2,change dri
ve
1295 DATA *,Lock/Unlock,^E,Delete
1300 DATA ^R,Rename,^V,Verify
1305 DATA ^F,Cherche fichier,^P,Imprime r
epertoire
1310 DATA à,à

```

Accompagné d'une cinquantaine de pages de documentation, Sisk Manager permet de recréer les commandes du Dos, redéfinir l'organisation d'une disquette, grâce à un jeu d'instructions qui en fait un langage simple d'accès à la disquette. Il offre également un programme simple d'édition à l'aide de commandes évoluées. 4 utilitaires figurent aussi sur la disquette :

Utili-disque : reconstruction d'une disquette détruite, Vérification, Plan d'occupation
Ultra-copie : pour un backup particulièrement rapide
Edicar : Edition du catalogue, classement des fichiers, Titres...
Multi-disque : pour le classement de tous vos programmes (tri instantané).

Sisk Manager, le Dos en Kit

de Dan Steevey

Ce petit programme permet d'insérer des titres dans un programme Basic à l'aide de REMs sans que l'on voie ni le numéro de ligne, ni le mot REM.

Il s'utilise de la manière suivante :

- Lancement par **BRUN TITRES.OBJ**
- Chaque fois que l'on veut faire un titre : **&T <RETURN>**, nnnn (une touche), nnnn étant le numéro de ligne, l'appui d'une touche faisant apparaître "REM" puis le texte et **<RETURN>**. Il ne se passe alors rien, en apparence... L'appui d'une touche non numérique après **&T <RETURN>** fait sortir du mode titre.

Fonctionnement

Ce programme :

- revectorise l'ampersand,
- déroute CHRGET (\$B1) et teste si on entre bien un numéro de ligne :
 - si oui, il écrit dans le buffer d'entrée 'REM', et un nombre de contrôle-H et d'espaces qui dépend de la longueur du numéro de ligne.
 - si non, il rend la main à Applesoft.



Récapitulation TITRES.OBJ

```

0300- A9 4C 8D F5 03 A9 10 8D
0308- F6 03 A9 03 8D F7 03 60
0310- C9 54 F0 03 4C C9 DE 20
0318- B1 00 D0 79 A9 01 8D 96
0320- 03 A5 38 85 FE A5 39 85
0328- FF A9 34 85 38 A9 03 85
0330- 39 4C EA 03 20 1B FD 48
0338- AD 96 03 D0 02 68 60 68
0340- C9 B0 30 08 C9 BA 10 04
0348- 9D 00 02 60 E0 00 F0 3E
0350- A9 D2 9D 00 02 20 5C DB
0358- E8 A9 C5 9D 00 02 20 5C
0360- DB E8 A9 CD 9D 00 02 20
0368- 5C DB E8 8A 18 69 03 85
0370- FC A9 88 9D 00 02 E8 C6
0378- FC D0 F8 8A 18 69 03 4A
0380- 85 FC A9 A0 9D 00 02 E8
0388- C6 FC D0 F8 A9 81 48 A9
0390- 00 8D 96 03 68 60 00

```

~~1 0 REM Des titres~~
~~2 0 REM dans le~~
~~3 0 REM Basic,~~
~~4 0 REM Sylvie Gallet~~

Programme TITRES.SCE (Assembleur Lisa 2.5)

```

1 ;
2 * TITRES DANS UN
3 *
4 * PROGRAMME
5 ;
6 ;
7 HOOK EPZ $38
8 AUX EPZ $FE
9 OUTDO EQU $DB5C
10 KEYIN EQU $FD1B
11 AMPER EQU $3F6
12 X.REG EPZ $FC
13 STXERR EQU $DEC9
14 CHRGET EPZ $B1
15 ;
16 ORG $300
17 OBJ $300
18 ;
19 INIT:
20 LDA #$4C
21 STA AMPER-1
22 LDA #DEB.TITR
23 STA AMPER
24 LDA /DEB.TITR
25 STA AMPER+1
26 RTS
27 ;
28 DEB.TITR:
29 CMP #'T'
30 BEQ TITRE
31 JMP STXERR
32 TITRE:
33 JSR CHRGET
34 BNE RTS
35 LDA #$01
36 STA PASSE
37 LDA HOOK
38 STA AUX
39 LDA HOOK+1
40 STA AUX+1
41 LDA #TESTS
42 STA HOOK
43 LDA /TESTS
44 STA HOOK+1
45 JMP $3EA
46 ;
47 TESTS:
48 JSR KEYIN
49 PHA
50 LDA PASSE
51 BNE SUITE.PL
52 PLA
53 RTS
54 ;
55 SUITE.PL:
56 PLA
57 SUITE:
58 CMP #$B0
59 BMI ECR.REM
60 CMP #$BA
61 BPL ECR.REM
62 STA $200,X
63 RTS
64 ;
65 ECR.REM:
66 CPX #$0
67 BEQ FIN
68 LDA #"H"
69 STA $200,X
70 JSR OUTDO
71 INX
72 LDA #"E"
73 STA $200,X
74 JSR OUTDO
75 INX
76 LDA #"M"
77 STA $200,X
78 JSR OUTDO
79 INX
80 TXA
81 CLC
82 ADC #$3
83 STA X.REG
84 LDA #$88
85 ECR.CTRH:
86 STA $200,X
87 INX
88 DEC X.REG
89 BNE ECR.CTRH
90 TXA
91 CLC
92 ADC #$3
93 LSR
94 STA X.REG
95 LDA #$A0
96 ECR.ESP:
97 STA $200,X
98 INX
99 DEC X.REG
100 BNE ECR.ESP
101 LDA #$81
102 FIN:
103 PHA
104 LDA #$0
105 STA PASSE
106 PLA
107 RTS:
108 RTS
109 PASSE BRK
110 LONGUEUR EQU PASSE+1-INIT
111 END

```

Tant qu'à faire 5 choses à la fois,



Que ceux qui aiment travailler en faisant deux ou trois choses à la fois ne changent rien, au contraire, avec Apple et Jazz ils peuvent faire mieux.

Jazz de Lotus, c'est un programme créé pour Macintosh 512 Ko, équipé d'un lecteur externe qui permet de devenir un parfait jongleur professionnel.

Cinq programmes en concert, c'est-à-dire un tableur, un grapheur, un gestionnaire de fichier, un traitement de texte et un programme de communication réunis en un seul programme. Ou comment être à cinq sur la même souris.

Pouvoir gérer cinq applications à partir d'un seul écran, modifier une donnée dans une application et qu'elle se modifie automatiquement dans les autres, pouvoir sauter d'un graphe à un traitement de texte sans attendre les secondes qui durent une éternité pour changer de programme, c'est bien...

Quand on s'aperçoit que ces cinq programmes sont individuellement excellents, c'est une révolution.

Jazz, c'est l'outil idéal d'un directeur de service.

Prenons un exemple qui exige beaucoup de souplesse, de rapidité, et de doigté, la direction des services secrets :

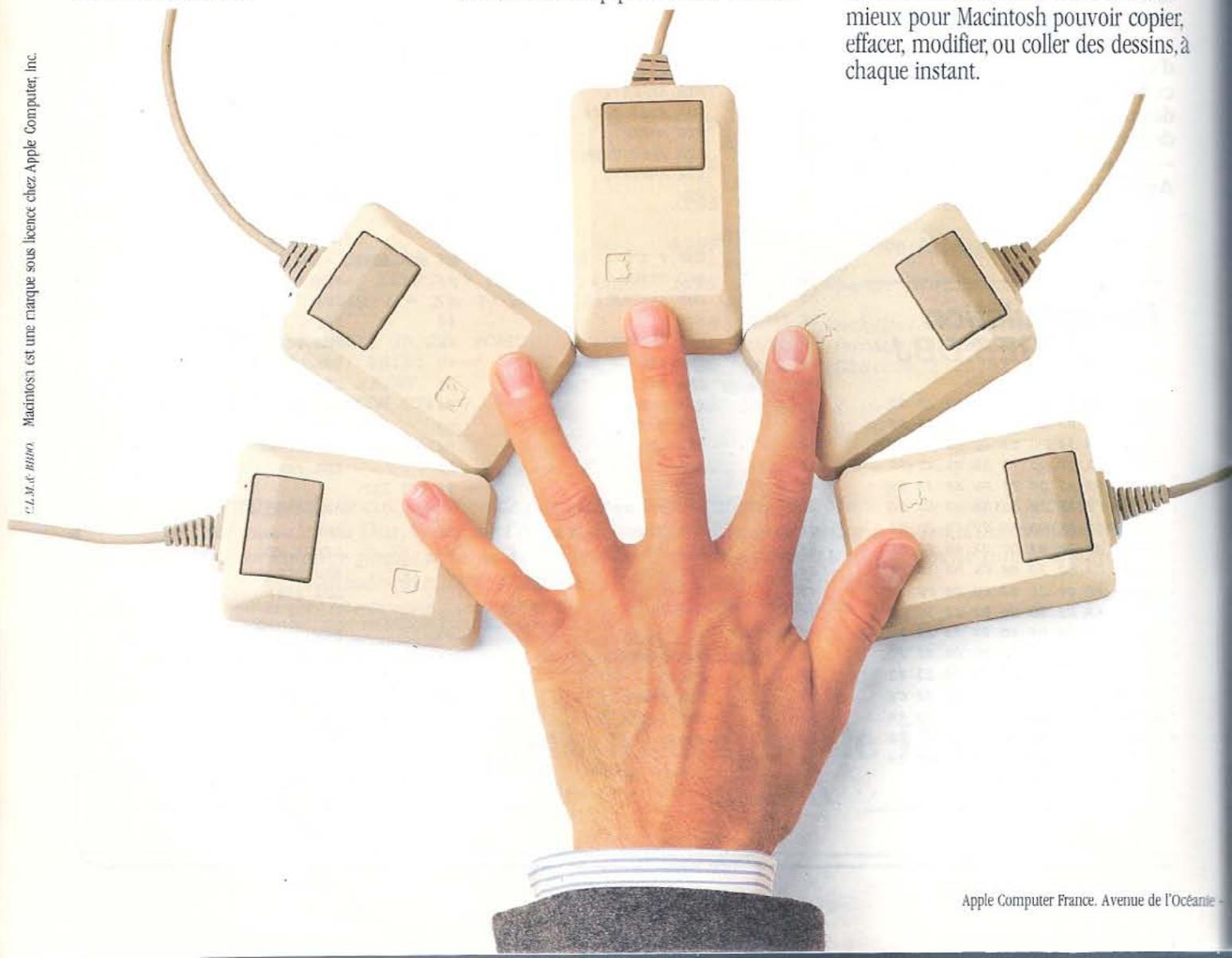
- Avec un tableur capable d'afficher 8192 lignes sur 256 colonnes, la gestion devient beaucoup plus facile. Si un fait

nouveau apparaît, rien n'est plus facile que d'étudier deux ou trois hypothèses afin de ne pas être pris de court.

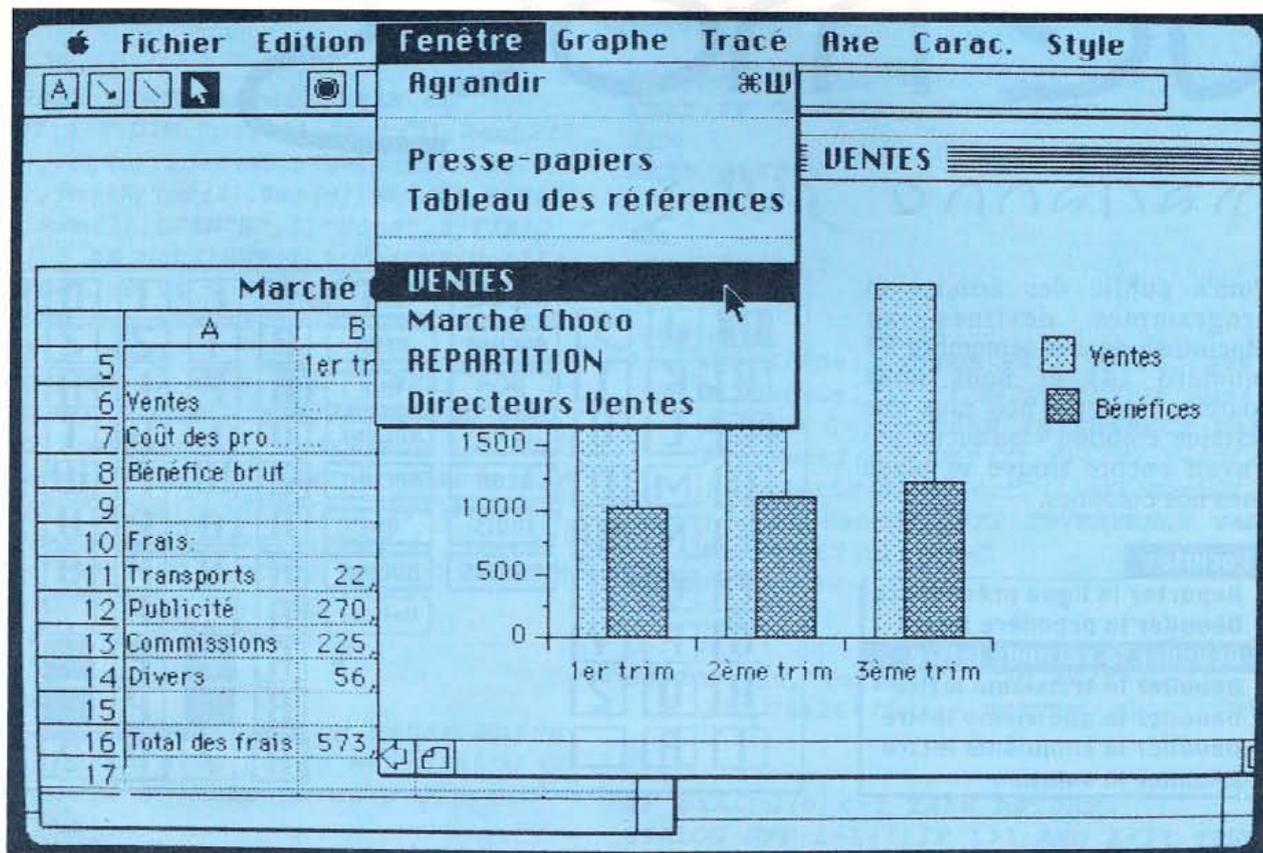
- Quand un projet est fin prêt, il faut bien le présenter à ses supérieurs et parfois même beaucoup plus haut; un grapheur permet de transformer toutes les données numériques obscures en graphiques lumineux.



- Si le projet est accepté, un traitement de texte est nécessaire pour que chaque agent concerné soit au courant dans ses moindres détails. Evidemment quand on travaille dans un tel service il vaut mieux pour Macintosh pouvoir copier, effacer, modifier, ou coller des dessins, à chaque instant.



autant les faire en même temps.



Apple, le logo Apple sont les marques déposées d'Apple Computer, Inc.
Jazz est une marque déposée de Lotus Development Corporation.

Feuille de travail Jazz.



Grâce à son gestionnaire de fichier, Jazz fait la fusion automatique entre un mémo et une liste de correspondants à sélectionner selon un critère spécial. Et avec un modem full duplex 1200 bauds en un instant votre correspondant reçoit toutes les informations précises à l'autre bout du monde. Avec Jazz l'entreprise est rentable, rondement menée, précise et discrète, inutile d'avoir 35 collaborateurs pour rédiger le projet.

Jazz c'est cinq programmes liés entre eux si parfaitement que l'agent double est enfin enterré, place à l'agent quintuple.



Apple

Jeu de mots

MARIANNE SUTZ

Pom's publie des articles et programmes destinés au Macintosh depuis septembre 84 (numéro 14) et nous nous sommes aperçu - non sans une certaine émotion - qu'aucun jeu n'avait encore trouvé sa place dans nos colonnes.

Facilités

- Reporter la ligne précédente
- Dévoiler la première lettre
- Dévoiler la seconde lettre
- Dévoiler la troisième lettre
- Dévoiler la quatrième lettre
- Dévoiler la cinquième lettre
- Dévoiler la solution

A	J	S	AUCUNE	UNE	A	C	I	D	E
B	K	T	AUCUNE	DEUX	P	I	Z	Z	A
C	L	U	UNE	UNE	A	P	L	A	N
D	M	V	DEUX	AUCUNE	O	L	A	I	N
E	N	W	DEUX	AUCUNE	Z	W	P	W	N
F	O	H	TROIS	UNE	R	A	P	U	N
G	P	Y	QUATRE	AUCUNE	R	U	P	A	N
H	Q	Z		VALIDER	R		P		N
I	R				R		P		N
					R		P		N
					R	?	P	?	N

Pour palier cela, nous vous proposons un petit casse-tête directement dérivé du célèbre "Master Mind". Les couleurs ont été ici remplacées par les 26 lettres de l'alphabet et il ne s'agit plus de trouver des codes, mais des noms communs de cinq lettres choisies au hasard par le Macintosh.

Utilisation du programme

Le gros avantage du Macintosh, c'est qu'il n'est pas nécessaire - grâce aux menus déroulants - de passer plus de temps à l'écriture du mode d'emploi qu'à celle du programme. Le seul point qui ne soit pas évident concerne ici les réponses à vos essais : la colonne de réponse située au dessus du bouton "Valider" (voir la copie d'écran ci-contre) indique le nombre de lettres mal placées, alors que la colonne située plus à gauche indique le nombre de lettres bien placées, la somme des deux colonnes indiquant le nombre de lettres réellement présentes dans le mot recherché.

Si vous ne disposez pas de la disquette 22...

... il vous faudra, bien sûr, saisir le programme et, dans ce cas quelques rappels et indications ne sont peut être pas inutiles :

- les indentations indiquent la continuité de la ligne ;
- il faut prendre garde à ne pas remplacer les 0 (zéro) par la lettre O (ou l'inverse) ;
- lorsque des valeurs en hexadécimal apparaissent dans le programme, il ne faut pas oublier le "&" ou le "h", surtout dans les DATA.

Le programme ne fonctionne pas sans un fichier de mots ; sa création, avec le programme "Crefich" est donc nécessaire. Notons toutefois qu'il n'est pas indispensable d'avoir un très grand nombre de mots comme dans le fichier de la disquette 22 (2135).



Mot numéro 2136 :

SULKY

OK Quitter

105 Mots pour commencer la constitution du fichier

(ils sont extraits du fichier "Mots" accompagnant le programme sur la disquette 22, et constitué de 2135 noms communs)

- ABACA ACCRU ADAGE AGAPE AGORA
- AIGLE AJOUT ALGOL ALLEU ALYTE
- AMIDE AMURE ANIER AORTE APPAT
- ARENE ARMEE ARYLE ATELE AUBIN
- AUTEL AVOIR BACHE BAGNE BALS
- BARRE BARRE BATON BEDON BENET
- BERYL BIAIS BIFFE BILLE BLAIR
- BLUES BOGUE BONDE BOREE BOUEE
- BOXER BREME BROME BRUNI DULGE
- BUTOR CABRI CAGET CAYEU CALOT
- CANON CARAT CARME CASTE CELTE
- CESTE CHAUX CHIPS CHYME CIRSE
- CLICK COATI COGNE COLZA COPIE
- CORON COUAC CRABE CRAWL CREVE
- CROUP CUMIN CYGNE DAMAS DATTE
- DECCA GEGEL DELTA DESIR DIANE
- DINAR DJINN DONNE DOUVE DRILL
- DUITE EBATS ECOLE EDILE ELEIS
- EMPAN ENGIN EPAVE EPOUX ESTER
- ETHER EVADE EXODE FAIRE FANON
- FAUTE FELON FERTE FILET FIOLE

Programme "Jeu de mots"

Ce programme requiert la présence d'un fichier de mots. Si vous ne disposez pas de la disquette 22, ce fichier peut être constitué à l'aide du programme "CreFich".

```
ON BREAK GOSUB Arret: BREAK ON
DEFINT A-Z: DIM Rps(19), Rect(3), Rect2(3), Pat(3), Tget(45), Tget2(45), Tc(10,4), Tm(4), Tm2(4), Tma(4), Nb$(5), Rff(7), Rsm(7): OPEN "R", 1, "Mots", 5: FIELD 1,5 AS Mot$: GET 1,1: Nm=CVI(Mot$): FOR I=0 TO 7: READ Mc, Me$: MENU 5, I, Me, Me$: NEXT: FOR I=0 TO 5: READ Nb$(I): NEXT
FOR I=0 TO 19: READ Rps(I): NEXT: FOR I=0 TO 7: READ Rff(I): NEXT: FOR I=0 TO 7: READ Rsm(I): NEXT: FOR I=0 TO 3: Pat(I)=&hAA55: NEXT: A!=VARPTR(Rff(0)): A!: A!=VARPTR(Rsm(0)): FOR I=1 TO 5: A! I: NEXT: WINDOW 1,, (13,33)-(499,329), 4: GOSUB AffAlpha: GOSUB AffGril: GOSUB Init
IF LOF(1)=0 THEN Erreur

BattJeu:
ON DIALOG GOSUB Dial: ON MOUSE GOSUB SpZ: ON MENU GOSUB Men: DIALOG ON: MOUSE ON: MENU ON: MENU 5,0,1:

Batt:
K$=INKEY$: IF K$<>" " THEN GOSUB Touche
GOTO Batt

BattAutre:
ON DIALOG GOSUB DialAutre: DIALOG ON: MENU OFF: MOUSE OFF: MENU 5,0,0

Batta:
GOTO Batta

SpZ:
DIALOG OFF: MOUSE OFF: Drz=5: A!=VARPTR(Rps(3)): A! 468,260,280,4: IF Rps(0) THEN Drz=4
A! 48,281,21,21: IF Rps(0) THEN Drz=1
A! 88,281,61,21: IF Rps(0) THEN Drz=2
A! 128,281,101,21: IF Rps(0) THEN Drz=3
ON Drz GOSUB Z1,Z2,Z3,Z4,Z5: DIALOG ON: MOUSE ON: RETURN

Z1:
X1=21: X2=48: C=64: GOSUB SpZ2: RETURN
Z2:
X1=61: X2=88: C=73: GOSUB SpZ2: RETURN
Z3:
X1=101: X2=128: C=82: GOSUB SpZ2: IF Nc=91 THEN Nc=0
RETURN
```

```
Z4:
Dp=-1: C1=0: FOR X=0 TO 4: X2=40*X: A!=VARPTR(Rps(3)): A! 308+X2,260,280+X2,4: IF Rps(0)=0 THEN Z4b
PUT(281+X2,5+26*(L-1)), Tget, PSET: Tc(L,X)=Nc: But=BUTTON(1): IF But=1 THEN IF Nc<65 THEN BUTTON 1,0: GOTO Z4b
FOR I=0 TO 4: IF Tc(L,I)<65 THEN Dp=0
NEXT: IF Dp THEN BUTTON 1,1
Z4b:
NEXT: RETURN

Z5:
BEEP: RETURN

SpZ2:
A!=VARPTR(Rps(3)): FOR Y=21 TO 261 STEP 30: C=C+1: A! X2,Y+20,X1,Y: IF Rps(0) AND C<>Nc THEN INVERTRECT VARPTR(Rect2(0)): GET(X1,Y)-(X2,Y+20), Tget: Rect2(0)=Y: Rect2(1)=X1: Rect2(2)=Y+21: Rect2(3)=X2: INVERTRECT VARPTR(Rect2(0)): Nc=C
NEXT: RETURN

Bo:
BUTTON 1,0, "Valider", (210,4+26*(L-1))-(270,26+26*(L-1)): RETURN

Dial:
IF DIALOG(0)<>1 THEN RETURN
DIALOG OFF: L=L+1: IF L>1 AND L<11 THEN MENU 5,1,1
IF L<11 THEN GOSUB Bo ELSE BUTTON CLOSE 1: MENU 5,1,0
L2=26*(L-2): FOR I=0 TO 4: Tm2(I)=Tm(I): Tma(I)=Tc(L-1,I): NEXT: LINE(140,4+L2)-(200,24+L2),, B: LINE(210,4+L2)-(270,24+L2),, B: LINE(140,6+L2)-(200,6+L2),, B: LINE(210,6+L2)-(270,26+L2),, B: Bp=0: Mp=0: FOR I=0 TO 4: IF Tma(I)=Tm2(I) THEN Bp=Bp+1: Tma(I)=0: Tm2(I)=-1
NEXT: FOR I=0 TO 4: FOR J=0 TO 4: IF Tma(I)=Tm2(J) THEN Mp=Mp+1: Tma(I)=0: Tm2(J)=-1
NEXT: NEXT: MOVETO 170-WIDTH(Nb$(Bp))/2,20+L2: PRINT Nb$(Bp): MOVETO 240-WIDTH(Nb$(Mp))/2,20+L2: PRINT Nb$(Mp): IF Bp<>5 THEN Dial2
IF L<11 THEN BUTTON CLOSE 1
GOSUB TrouveO: GOTO Dials
Dial2:
IF L=11 THEN GOSUB TrouveN: GOTO Dials
FOR X=0 TO 4: X2=40*X: IF Tc(L,X)=0 THEN LINE(281+X*40,5+(L-1)*26)-(307+X*40,25+(L-1)*26),30,BF
NEXT: DIALOG ON: RETURN
Dials:
```

DIALOG ON:RETURN BattAutre

SpCM:

```
RANDOMIZE TIMER:GET 1,INT(RND*Nm)+2:
M$=Mot$:FOR I=0 TO 4:Tm(I)=ASC(MI
D$(M$,I+1,1)):NEXT:RETURN
```

AffAlpha:

```
TEXTFONT 0:TEXTSIZE 24:TEXTMODE 1:C=
64:FOR X=0 TO 2:X2=40*X:FOR Y=0 T
O 8:Y2=Y*30:LINE(22+X2,21+Y2)-(50+
X2,43+Y2),,BF:LINE(20+X2,20+Y2)-(4
8+X2,42+Y2),,B:LINE(21+X2,21+Y2)-(
47+X2,41+Y2),30,BF:C=C+1
IF C<91 THEN MOVETO (35+X2)-WIDTH(CH
R$(C))/2,40+Y2:PRINT CHR$(C);
NEXT:NEXT:GET(21,21)-(48,42),Tget:Rect
2(0)=21:Rect2(1)=21:Rect2(2)=42:Rec
t2(3)=48:INVERTRECT VARPTR(Rect2
(0)):RETURN
```

AffGril:

```
FOR X=0 TO 4:X2=40*X:FOR Y=0 TO 9:Y2=Y
*26:LINE(280+X2,4+Y2)-(308+X2,26+Y
2),,B:IF Y THEN Rect(0)=5+Y2:Rect(
1)=281+X2:Rect(2)=26+Y2:Rect(3)=308
+X2:FILLRECT VARPTR(Rect(0)),VAR
PTR(Pat(0))
NEXT:LINE(280+X2,268)-(308+X2,290),,B:
MOVETO 294+X2-WIDTH("?")/2,288:PR
INT"?":NEXT:LINE(280,264)-(468,2
64):RETURN
```

Init:

```
FOR I=0 TO 7:MENU 5,I,(I<2)+1:NEXT:L=
1:GOSUB Bo:Drok=0:Drz=0:Nc=65:FOR
X=0 TO 4:FOR Y=1 TO 10:Tc(Y,X)=0:M
EXT:NEXT:GOSUB SpCM:TEXTSIZE 12:
Nmenu=4:RETURN
```

Trouve0:

Dtr=-1:GOTO Trouve

TrouveN:

Dtr=0

Trouve:

GOSUB AffSol

```
IF L>5 THEN WINDOW 2,,(30,50)-(270,120
),2 ELSE WINDOW 2,,(30,240)-(270,
310),2
```

```
TEXTFONT 0:TEXTMODE 1:TEXTSIZE 12:I
F Dtr THEN MOVETO 10,15:PRINT"Br
avo !"
```

```
MOVETO 10,31:PRINT "le mot était";:IF
Dtr THEN PRINT "bien";
```

```
PRINT " " M$ "" :BUTTON 1,1,"Autre par
tie",(5,45)-(100,65):BUTTON 2,1,"Q
uitter",(140,45)-(235,65)
```

RETURN

DialAutre:

IF DIALOG(0)<>1 THEN RETURN

```
IF DIALOG(1)=2 THEN WINDOW CLOSE 2:W
INDOW CLOSE 1:SYSTEM
WINDOW CLOSE 2:CLS:BUTTON CLOSE 1:G
OSUB AffAlpha:GOSUB AffGril:GOSUB
init:RETURN BattJeu
```

Touche:

```
Nn=ASC(K$):IF Nn=32 THEN Numc=2:Numl=8
:C=Nn:GOTO Touche2
```

```
Nn=ASC(K$) AND &h5F:IF Nn<65 OR Nn>90
THEN RETURN ELSE C=Nn:Numc=INT((
Nn-65)/9):Numl=(Nn-65) MOD 9
```

Touche2:

IF C=Nc THEN RETURN

```
INVERTRECT VARPTR(Rect2(0)):GOSUB Ge
tCar:INVERTRECT VARPTR(Rect2(0))
:Nc=C:DIALOG ON:RETURN
```

GetCar:

```
Rect2(0)=21+Numl*30:Rect2(1)=21+Numc*40:
Rect2(2)=42+Numl*30:Rect2(3)=48+Num
c*40:GET(Rect2(1),Rect2(0))-(Rect2
(3)-1,Rect2(2)-1),Tget:RETURN
```

AffSol:

```
INVERTRECT VARPTR(Rect2(0)):FOR I=0
TO 4:Numc=INT((Tm(I)-65)/9):Numl=(
Tm(I)-65) MOD 9:GOSUB GetCar:LINE
(278+40*I,266)-(310+40*I,292),,BF:P
UT(281+40*I,269),Tget,PSET:NEXT:R
ETURN
```

Men:

```
MENU OFF:MENU:Nmenu=MENU(1):ON Nmenu
GOSUB MenRrp,MenL,MenL,MenL,MenL,M
enL,MenSol
```

```
IF Nmenu=7 THEN RETURN BattAutre ELSE
MENU ON:RETURN
```

MenRrp:

```
FOR X=0 TO 4:X2=40*X:Y2=(L-1)*26:Y1=Y2-
26:GET(281+X2,5+Y1)-(307+X2,25+Y1)
,Tget2:PUT(281+X2,5+Y2),Tget2,PSET
:Tc(L,X)=Tc(L-1,X):NEXT:BUTTON 1,
1:MENU 5,1,0:RETURN
```

MenL:

```
MENU 5,Nmenu,0:Nmenu=Nmenu-2:INVERTRE
CT VARPTR(Rect2(0)):Numc=INT((Tm(
Nmenu)-65)/9):Numl=(Tm(Nmenu)-65) M
OD 9:Rect(0)=21+Numl*30:Rect(1)=21+
Numc*40:Rect(2)=42+Numl*30:Rect(3)=
48+Numc*40:GET(Rect(1),Rect(0))-(R
ect(3)-1,Rect(2)-1),Tget2
```

```

LINE (278+40*Nmenu,266)-(310+40*Nmenu,29
2),,BF:PUT(281+40*Nmenu,269),Tget2
,PSET:X2=40*Nmenu:FOR Y=L-1 TO 9:
Tc(Y+1,Nmenu)=Tm(Nmenu):Y2=Y*26:PU
T(281+X2,5+Y2),Tget2,PSET:NEXT:Nm
enul=Nmenul-1:IF Nmenul=0 THEN FO
R I=2 TO 6:MENU 5,I,0:NEXT
INVERTRECT VARPTR(Rect2(0)):RETURN

```

```

MenSol:
GOSUB TrouveN:RETURN

```

```

Arret:
RETURN

```

```

Erreur:
BEEP:WINDOW 1,,(100,120)-(412,160),2:T
EXTFONT 0:TEXTMODE 1:TEXTSIZE 1
2:MOVETO 10,15:PRINT"Désolé, pas
de fichier de mots accessible..":BU
TTON 1,1,"OK",(240,20)-(300,40):ON
DIALOG GOSUB DialEr:DIALOG ON

```

Programme "CreFich"

```

ON BREAK GOSUB Arret:BREAK ON:DEFIN
T A-Z:DIM Rff(7),Rsm(7):FOR I=0 T
O 7:READ Rff(I):NEXT:FOR I=0 TO 7
:READ Rsm(I):NEXT
A!=-VARPTR(Rff(0)):A!:A!=-VARPTR(Rsm(0))
:FOR I=1 TO 2:A! I:NEXT:FOR I=4 T
O 6:A! I:NEXT:WINDOW 1,,(180,120)
-(332,210),4:GOSUB Text:OPEN"R",1
,"Mots",5:FIELD 1,5 AS Mot$
IF LOF(1) THEN GET 1,1:Nm=CVI(Mot$)+1
ELSE Nm=1:LSET Mot$=MKI$(Nm):PUT
1,1
MOVETO 10,20:PRINT"Mot numéro":GOSUB
AffN:EDIT FIELD 1,"",(40,30)-(110
,45):BUTTON 1,1,"OK",(10,60)-(71,8
0):BUTTON 2,1,"Quitter",(81,60)-(1
42,80):ON DIALOG GOSUB Dial:DIAL
OG ON
Batt:
GOTO Batt

```

```

Dial:
DIALOG OFF:W0=WINDOW(0):W1=WINDOW(1)
:D0=DIALOG(0):D1=DIALOG(1):IF W1=
2 THEN WINDOW CLOSE 2:WINDOW 1:
EDIT FIELD 1,M$(40,30)-(110,45):
DIALOG ON:RETURN
IF D0=1 THEN SPb ELSE IF D0=6 THEN SP
d
DIALOG ON:RETURN
SPb:

```

```

AttEr:
GOTO AttER

```

```

DialEr:
WINDOW CLOSE 1:MENU RESET:END

```

```

DATA 0,Facilités,0,Reporter la ligne pr
écédente,1,Dévoiler la première let
tre,1,Dévoiler la seconde lettre,1,
Dévoiler la troisième lettre,1,Dévo
iler la quatrième lettre,1,Dévoiler
la cinquième lettre,1,Dévoiler la
solution
DATA AUCUNE,UNE,DEUX,TROIS,QUATRE,CINQ
DATA 0,0,0,&h4E56,0,&h487A,&hFFF6,&hA97
2,&h4267,&h47FA,&hFFEE,&h2F13,&h486
E,8,&hA8AD,&h47FA,&hFFE0,&h369F,&h4
E5E,&h4E75
DATA &h42A7,&hA924,&h2E1F,&h6706,&h2F07
,&hA916,&h60F2,&h4E75
DATA &h4E56,0,&h3F2E,8,&hA936,&hA937,&h
4E5E,&h4E75

```

```

IF D1=2 THEN LSET Mot$=MKI$(Nm-1):PUT
1,1:CLOSE:WINDOW CLOSE 1:SYSTE
M

```

```

Spd:
M$=EDIT$(1):IF LEN(M$)<>5 THEN GOSUB
Erreur:DIALOG ON:RETURN
DP=0:FOR i=1 TO 5:L$=MID$(M$,i,1):IF L
$<"A" OR L$>"Z" THEN DP=-1
NEXT:IF DP THEN GOSUB Erreur:DIALOG
ON:RETURN
LSET Mot$=M$:Nm=Nm+1:PUT 1,Nm:GOSUB Af
fN:EDIT FIELD 1,"",(40,30)-(110,4
5):DIALOG ON:RETURN

```

```

AffN:
LINE(90,0)-(152,24),30,BF:MOVETO 90,20
:PRINT Nm":":RETURN

```

```

Erreur:
WINDOW 2,,(100,160)-(412,200),2:GOSUB
Text:MOVETO 10,15:PRINT"Mots de c
inq lettres majuscules SVP":BUTTON
1,1,"OK",(240,20)-(300,40):RETURN

```

```

Text:
TEXTFONT 0:TEXTMODE 1:TEXTSIZE 12:R
ETURN

```

```

Arret:
RETURN

```

```

DATA &h42A7,&hA924,&h2E1F,&h6706,&h2F07
,&hA916,&h60F2,&h4E75,&h4E56,0,&h3F
2E,8,&hA936,&hA937,&h4E5E,&h4E75

```

De nombreux lecteurs qui utilisent Apple // et Macintosh nous ont demandé un programme permettant de transmettre des données dans le sens Apple // → Mac. Rappelons que le programme paru dans le Pom's 14 assurait la transmission dans l'autre sens.

Mode d'emploi

Côté Apple // : Configurer la carte série comme indiqué ci-contre. Puis adresser les données vers la carte. Si vous transmettez un programme Basic, faites PR#1 (si la carte est en slot 1) puis LIST. Pour éviter de fâcheux retours chariot, faites préalablement POKE 33,33. Sous AppleWriter, dans le menu impression mettre la marge droite à 255, puis faire NP.

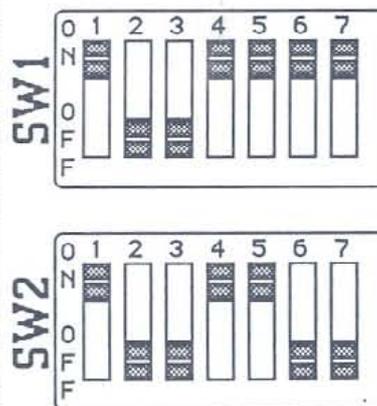
Côté Mac : Exécuter le programme Basic ci-contre, le câble étant connecté au port Modem. Ce programme étant réduit au strict nécessaire, il vous appartient dès la fin de la transmission d'arrêter le programme puis de faire en mode direct dans la fenêtre de commande : CLOSE.

Il nous a été possible de transmettre par exemple le programme de calcul de π du Pom's 21 et de l'exécuter avec succès moyennant de menues modifications de syntaxe. La transmission d'image HGR est également envisageable : au lieu d'écrire dans un fichier, il s'agit de stocker dans la mémoire-écran.

Programme de transfert Apple// -> Mac

```
DEFINT A-Z
CLS
TEXTFONT 4
TEXTSIZE 9
F$=FILES$(0,"Fichier de s
    auvegarde ?")
IF F$="" THEN END
IF F$="*" THEN END
WIDTH "SCRN:",80
OPEN "COM1:300,N,7,1" FOR
    INPUT AS 1
OPEN "O",2,F$
10 C$=INPUT$(1,1)
IF C$="(" THEN C$="é"
IF C$=")" THEN C$="è"
IF C$="@" THEN C$="à"
IF C$="\ " THEN C$="ç"
IF C$="|" THEN C$="ù"
PRINT C$;
PRINT #2,C$;
GOTO 10
```

Configuration de la carte Super Série

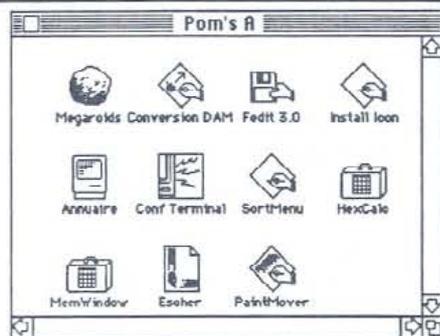


"Jumper Bloc" en position "MODEM"

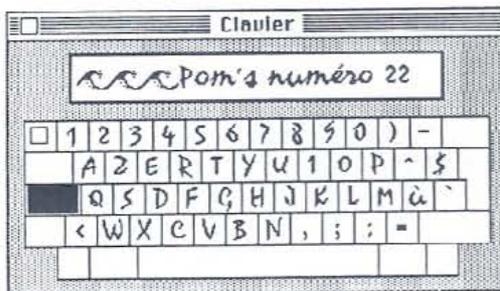
Disquette Mac 'A'

SORT MENU, un utilitaire de tri de menus,
PAINT MOVER : copier et manipuler les images MacPaint
ACCESSOIRES DE BUREAU 'MOCK' : mini-traitement de texte, grapheur et terminal
CONVERT DAM : pour passer de DA Mover à Font D/A Mover
FEDIT : un éditeur de blocs et de fichiers (en anglais)
CUBE : un jeu en accessoire
IDLE : pour préserver votre écran durant de longs traitements
MULTISCRAPE : accessoire permettant de disposer de plusieurs albums à la fois
C.CHECK : Pour vérifier les niveaux de parenthèses en langage 'C'
HEX CALC : l'indispensable calculatrice hexadécimale
MEGAROIDS : un jeu d'adresse
MEM WINDOW : permet de lister la mémoire du Mac

Disquette Mac 'A' : 80,00 F TTC Franco, Bon de Commande page 74



Disquette Mac 22



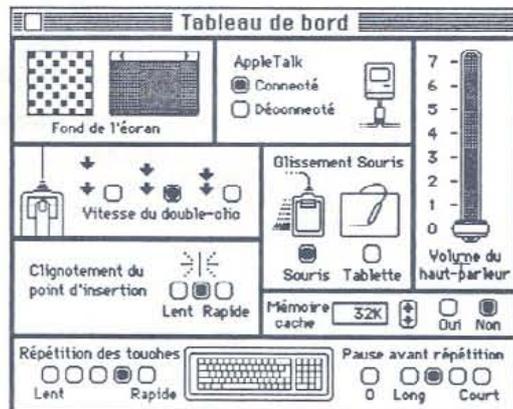
Vous trouverez sur la disquette, outre le nouveau fichier System et le nouveau Finder, le driver ImageWriter2, le driver LaserWriter pour Mac 128K et deux nouveaux accessoires de bureau : un sélecteur d'imprimante reconnaissant entre autres les deux types d'ImageWriter et, dans le tableau de bord, la possibilité de définir un cache-mémoire. Un cache-mémoire fait ce que faisait le Mac Booster d'ACI : mettre, dans une mémoire vive dont la taille est fixée par l'utilisateur, les éléments les plus souvent utilisés, cette fréquence d'utilisation étant appréciée par le programme de cache. Cela permet de nettement réduire les temps de réponse du Mac. L'accessoire "Clavier" autorise désormais l'essai des différentes polices présentes dans le système. Enfin, la disquette Pom's contient aussi une nouvelle version de "Font DA/Mover".

Vous trouverez sur la disquette d'accompagnement Macintosh de ce numéro, outre les programmes publiés dans le cahier Mac, le nouveau système correspondant à la version 5.1 du Finder. Les commentaires d'utilisation se trouveront directement sur la disquette. Le nouveau Finder comprend, de façon optionnelle, des mini-icônes et certains nouveaux gadgets pour le rangement des icônes. Ce qui fait vraiment son originalité, c'est le IIFS : Hierarchical File System. Voici enfin un système de gestion de fichiers réellement hiérarchisé.

Il s'activera dès que le support magnétique (disque ou disquette) aura plus de 400Ko de capacité.

Présentation

- ✓ par petite icône
- par icône
- par nom
- par date
- par taille
- par type



Hyperparallépipède et fractaux

Jean-Luc Bazaneque
et Christian Piard

“ Que faire avec l'écran du Macintosh ?” Séduit par sa finesse de définition, particulièrement lorsque l'on vient de l'Apple II, voici la question que l'on peut se poser. Le programme qui suit a été écrit en se pliant aux contraintes suivantes : utiliser pleinement la haute résolution, respecter les conventions de présentation 'Mac' et, si possible, être inutile... Dans un premier temps, le dessin d'une courbe qui pourrait évoluer en finesse a été le but de ce programme.

Tracé

✓Fractal

Hyperparallépipède

Voici quelques mots de théorie sur la courbe qu'étudia Von Koch. Considérons le segment de droite suivant :



Ce segment est divisé en trois parties égales. La partie centrale disparaît et est remplacée par les deux autres côtés d'un triangle équilatéral dont elle servait de base. Nous obtenons alors le fractal d'ordre 1 :



La figure ainsi obtenue comprend maintenant quatre segments de longueurs égales. Faisons subir le même sort à chacun d'eux, il en

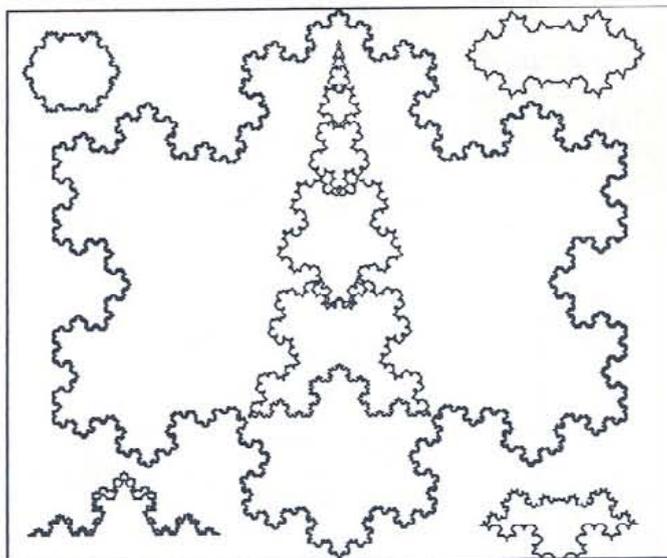


Figure :

- ⊙ triangle ←
- triangle →
- hexagone ←
- hexagone →

Complexité :

- 0
- 1
- 2
- ⊙ 3
- 4
- 5

Effacer

résultera évidemment le fractal d'ordre 2 :



Notons au passage que cette figure comprend seize segments. A chaque nouvelle étape dans la constructions de nos fractaux (fractals ?), le nombre de segments est multiplié par quatre. La longueur de chacun d'entre eux est égale au tiers de ceux du fractal d'ordre immédiatement inférieur.

Pour un fractal d'ordre n dont l'élément de base mesurait une unité, nous trouvons : 4^n côtés ayant chacun une longueur de $1/(3^n)$. La longueur développée est donc : $4^n/3^n$. Cette fonction, de type exponentiel, tend vers l'infini lorsque n tend

vers l'infini : le calcul de la longueur développée peut donc rapidement provoquer un *overflow* sans pour autant que le dessin prenne plus de place sur votre écran...

Un fractal d'ordre 5 ne comprendra pas moins de 1024 (1 K !) segments totalisant environ 4,213 unités de longueur.

Lorsque n tend vers l'infini, la courbe qui en résulte s'inscrit dans un espace à 1,26 dimension ! plus précisément :

$$\log(4)/\log(3),$$

autrement dit un nombre non entier... Le caractère ésotérique de la chose ne vous échappera pas ; le programme lui, ne s'attache qu'à son aspect esthétique.

Il vous propose de tracer ces figures en prenant pour base un triangle ou un hexagone, avec un développement interne ou externe. Il vous permet de superposer les graphes pour l'obtention de

dessins du plus bel effet, flocons de neige par exemple.

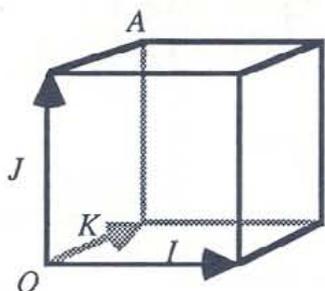
Tracé

Fractal

✓Hyperparallélépipède

Plus insaisissable (?) que le fractal : l'hyperparallélépipède. Le prétexte à nos dessins est maintenant la représentation en deux dimensions (notre écran) de parallélépipèdes en comprenant de 3 à 8. Le résultat, s'il est totalement incompréhensible, est souvent particulièrement élégant.

Si nous prenons le cube suivant :



nous voyons que quatre données

ont suffi à sa construction : l'origine O et les trois vecteurs I, J, K. Ainsi, la position de chaque sommet pourra s'obtenir par la simple combinaison des différents vecteurs : $S1 = O \rightarrow J + O \rightarrow K$ par exemple.

Vous allez pouvoir dessiner des parallélépipèdes en indiquant successivement l'origine puis les trois à huit vecteurs. Il seront représentés à l'écran en suivant les règles qui ont servi pour la construction du cube.

Ce premier curseur : 

vous permet de positionner l'origine. Elle est alors matérialisée ainsi :



Ce nouveau curseur : 

vous permet de définir les différents vecteurs. Ils se matérialiseront ainsi :



Voici l'origine et le premier

vecteur :



Un disposition comprenant une origine et 3 vecteurs vous donnera un volume bien réel, en-deçà de l'"espace-temps" ! Par la suite, en ajoutant des vecteurs, on dépasse cette limite et on se surprend à passer beaucoup de temps, dans un premier temps, à essayer de deviner le futur dessin puis à créer des formes esthétiques.

Pour le reste de la présentation, les menus Mac ne présentent aucun secret.

Le Basic qui suit utilise des routines en assembleur extraites des précédents numéros de Pom's, pour les traitements suivants :

- suppression des menus (cf Pom's 17),
- récupération des curseurs (cf Pom's 18),
- récupération des icônes (cf Pom's 18),
- récupération de la position de la souris (cf Pom's 19),
- fermeture de fenêtres (cf Pom's 19).



Tracés de fractaux et hyperparallélépipèdes

REM Initialisations - tableaux et constantes

```
BREAK ON:ON BREAK GOSUB BRK:HIDECUR
SOR:DEFINT A-Z:Unit1#=1/3:Unit2#=
1/2:Unit3#=2/3:Vect2#=SQR(3)/6:Pid
iv6#=-3.1415926#/6:DIM Lm(15),Pat(3
),Rect(3),Sr(19),Xn(256),Yn(256),V
ectX(8),VectY(8),Xvect(8),Yvect(8)
,Nn(8),Icône(18),TIcône(65)
FOR I=0 TO 3:Pat(I)=&hAA55:NEXT:Dtr=-1
```

REM Data pour implantation des routines en langage machine

REM Routine de fermeture des fenêtres - cf Pom's n° 19

```
DATA &h42A7,&hA924,&h2E1F,&h6706,&h2F07
,&hA916,&h60F2,&h4E75
```

REM Routine de suppression des menus -

cf Pom's n° 17

```
DATA &h4E56,0,&h3F2E,8,&hA936,&hA937,&h
4E5E,&h4E75
```

REM Routine de récupération des curseurs - cf Pom's n° 18

```
DATA &h4E56,0,&h42A7,&h3F2E,8,&hA9B9,&h
2057,&h2E90,&hA851,&h4E5E,&h4E75
```

REM Routine de détection de la position de la souris - cf Pom's n° 19

```
DATA 0,0,0,&h4E56,0,&h487A,&hFFF6,&hA97
2,&h4267,&h47FA,&hFFEE,&h2F13,&h48
6E,8,&hA8AD,&h47FA,&hFFE0,&h369F,&
h4E5E,&h4E75
```

REM Routine de récupération des icônes - cf Pom's n° 18

```
DATA &h4E56,0,&h4ZA7,&h3F2E,&hC,&hA9BB,
&h2057,&h2250,&h246E,8,&h24FC,32,3
2,&h701F,&h24D9,&h51C8,&hFFFC,&h4E
5E,&h4E75
```

```
CMS="A09F9DD783DCD08295868595D09682919E9
39F80989F9E95D0999E947E80959E94919
```

```

E8495D0809F8582D09CD78584999C99839
184958582D09485D0BD9193999E849F839
8":FOR I=1 TO LEN(CM$)-1 STEP 2:C
MZ$=CM2$+CHR$(VAL("&h"+MID$(CM$,
I,2)) XOR &hF0):NEXT

```

```

REM Lecture des Data, implantation des
routines en mémoire, fermeture des
fenêtres, suppression des menus
inutiles et configuration des
menus conservés

```

```

FOR I=0 TO 15:READ Lm(I):NEXT:FOR I=0
TO 10:READ Cur(I):NEXT:FOR I=0
TO 19:READ Sr(I):NEXT:FOR I=0 TO
18:READ Icone(I):NEXT:A!=VARPTR
(Lm(0)):A!
MENU 3,0,1,CHR$(20):MENU 3,1,1,CM2$:M
ENU 4,0,1,"Arrêt":MENU 4,1,1,"Qui
tter":MENU 5,0,0,"Tracé":MENU 5,1
,2,"Fractal":MENU 5,2,1,"Hyperpara
llélépipède":A!=VARPTR(Lm(8)):FOR
I=0 TO 3:A! I:NEXT:ERASE Lm:GOS
UB Ffr:ON MENU GOSUB Mfr

```

```

REM Boucle d'attente pour l'option
"Fractal"

```

```

Bfr2:
MENU ON:DIALOG ON:ON DIALOG GOSUB D
fr

```

```

Bfr:
IF MOUSE(0) THEN GOSUB Sfr
GOTO Bfr

```

```

REM On ignore "Commande-Majuscule- ."
BRK:
RETURN

```

```

REM Boucle d'attente pour l'option
"Hyperparallélépipède" et
modification de la forme du
 curseur si nécessaire

```

```

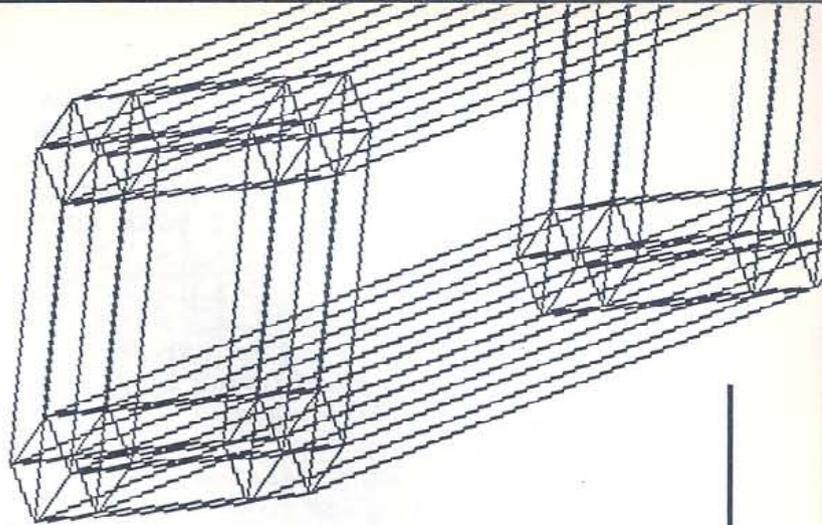
BHy2:
MENU ON:DIALOG ON
BHy3:
ON DIALOG GOSUB DHy
BHy:
IF MOUSE(0) THEN GOSUB SHy
IF BUTTON(1)=0 THEN BHy
A!=VARPTR(Sr(3)):A! 500,310,440,290:IF
Sr(0) THEN Dbo
Hbo:
IF Dcur THEN A!=VARPTR(Cur(0)):A! 2:Dc
ur=0
GOTO BHy
DBo:
IF NOT Dcur THEN INTCURSOR:Dcur=-1
GOTO BHy

```

```

REM Boucle d'attente d'une action sur
le bouton "Autre"

```



```

Att:
ON DIALOG GOSUB Att2:DIALOG ON
BAtt:
GOTO BAtt
Att2:
IF DIALOG(0)=1 THEN CLS:BUTTON 1,0,"
Valider", (440,290)-(500,310):A!=VA
RPTR(Cur(0)):A! 3:Drch=0:RETURN B
Hy3 ELSE RETURN

```

```

REM Traitement d'une action dans la
barre des menus

```

```

Mfr:
MENU:Me0=MENU(0):Me1=MENU(1):IF me0=3
THEN RETURN ELSE IF Me0=4 AND
Me1=1 THEN SYSTEM ELSE IF Me1=1
AND Dtr OR Me1=2 AND NOT Dtr THE
N RETURN

```

```

MENU 5,0,0:MENU OFF:DIALOG OFF:A!=VA
RPTR(Cur(0)):A! 4

```

```

IF Me1=1 THEN Dtr=-1:MENU 5,1,2:MENU
5,2,1:WINDOW CLOSE 1:GOSUB Ffr:
RETURN Bfr2

```

```

Dtr=0:MENU 5,1,1:MENU 5,2,2:WINDOW CL
OSE 1:WINDOW CLOSE 2:GOSUB FHy:
RETURN BHy2

```

```

REM Traitement d'une action sur le
bouton de la souris pour l'option
"Fractal"

```

```

Sfr:
IF WINDOW(0)-2 THEN RETURN ELSE Rect
Y=MOUSE(2):RectX=MOUSE(1)
WHILE MOUSE(0)=-1:Mo1=MOUSE(1):Mo2=MO
USE(2):Rect(0)=RectY:Rect(2)=Mo2:R
ect(1)=RectX:Rect(3)=Mo1
IF Mo2<RectY THEN SWAP Rect(0),Rect(2)
IF Mo1<RectX THEN SWAP Rect(1),Rect(3)
A!=VARPTR(Sr(3)):A! 382,315,1,1:IF Sr(
0) THEN FRAMERECT VARPTR(Rect(0
)):FRAMERECT VARPTR(Rect(0))
WEND:IF Sr(0)=0 THEN RETURN
X=(Rect(3)-Rect(1))/2:Y=(Rect(2)-Rect(0)
)/2:Cx=X+Rect(1):Cy=Y+Rect(0):IF X
<5 OR Y<5 THEN RETURN

```

```

A!=VARPTR(Cur(0)):A! 4:MENU 5,0,0:DIA
LOG OFF:ON Fig GOSUB Dtr,Htr,Dhe
,Hhe
INITCURSOR:WINDOW 2:MENU 5,0,1:DIAL
OG ON:IF BUTTON(11)=0 THEN BUTT
ON 11,1
RETURN

```

```

REM Traitement d'une action sur le
bouton de la souris pour l'option
"Hyperparallélépipède"

```

```

SHy:
WHILE MOUSE(0):WEND:RectY=MOUSE(2):R
ectX=MOUSE(1):IF Drch THEN SHy2
ELSE Drch=-1:Ind=-1:CIRCLE(RectX,
RectY),5:LINE(RectX-10,RectY)-(Rec
tX+10,RectY):LINE(RectX,RectY-10)-
(RectX,RectY+10):Xorg=RectX:Yorg=R
ectY:A!=VARPTR(Cur(0)):A! 2:Dcur=
0:RETURN

```

```

SHy2:
IF Ind<>7 THEN SHy3
DIALOG OFF:BUTTON 1,0:DIALOG OFF:BE
EP:WINDOW 2,,(140,150)-(372,203),
4:A!=VARPTR(Icône(0)):A! 1,VARPT
R(TIcône(0)):PUT(10,10),TIcône:TE
XTFONT 0:TEXTMODE 1:TEXTFACE 0:
TEXTSIZE 12:MOVETO 60,24:PRINT"
Pas plus de huit"
MOVETO 60,40:PRINT"vecteurs, S.V.P...";
BUTTON 1,1,"OK",(188,14)-(220,38)
:ON DIALOG GOSUB Bok:DIALOG ON

```

```

BBHy:
GOTO BBHy
BBHy2:
RETURN BHy2

```

```

REM traitement d'une action sur le
bouton "OK" et mise à jour de la
fenêtre (option
"Hyperparallélépipède")

```

```

Bok:
IF DIALOG(0)<>1 OR DIALOG(1)<>1 THEN
RETURN

```

```

DIALOG OFF:WINDOW CLOSE 2:WINDOW 1:
BUTTON 1,1:DIALOG ON:PENMODE 8

```

```

CIRCLE(Xorg,Yorg),5:LINE(Xorg-10,Yorg)
-(Xorg+10,Yorg):LINE(Xorg,Yorg-10)
-(Xorg,Yorg+10):FOR Z=0 TO 7:Vix=
VectX(Z)+Xorg:Viy=VectY(Z)+Yorg
LINE(Vix-2,Viy-2)-(Vix+2,Viy+2),,BF:LI
NE(Vix-5,Viy)-(Vix+5,Viy):LINE(Vi
x,Viy-5)-(Vix,Viy+5):MOVETO Xorg,
Yorg:LINETO VectX(Z)+Xorg,VectY(Z
)+Yorg:NEXT:PENMODE 10
RETURN BBHy2

```

```

REM suite traitement d'une action sur
le bouton de la souris pour
l'option "Hyperparallélépipède"

```

```

SHy3:
Ind=Ind+1:VectX(Ind)=RectX-Xorg:VectY(In
d)=RectY-Yorg:Nn(Ind)=0:LINE(RectX
-2,RectY-2)-(RectX+2,RectY+2),,BF:
LINE(RectX-5,RectY)-(RectX+5,RectY
):LINE(RectX,RectY-5)-(RectX,RectY
+5)
MOVETO Xorg,Yorg:LINETO RectX,RectY:I
F Ind=1 THEN BUTTON 1,1
RETURN

```

```

REM Traitement d'un événement de type
"Dialog" pour l'option "fractal"

```

```

Dfr:
DIALOG OFF:Dia=DIALOG(0):IF Dia<>1 T
HEN Dfr2
IF DIALOG(1)=11 THEN BUTTON 11,0:WIN
DOW 1:CLS:WINDOW 2:GOTO DfrF EL
SE IF DIALOG(1)>6 THEN BUTTON F
ig+6,1:BUTTON DIALOG(1),2:Fig=DI
ALOG(1)-6 ELSE BUTTON Com+1,1:BU
TTON DIALOG(1),2:Com=DIALOG(1)-1
:GOTO DfrF

```

```

Dfr2:
IF Dia=3 THEN Dia=DIALOG(3):WINDOW Di
a ELSE GOTO DfrF
IF Dia=1 THEN A!=VARPTR(Cur(0)):A! 2:
GOTO DfrF ELSE INITCURSOR

```

```

DfrF:
DIALOG ON:RETURN

```

```

REM Traitement d'un événement de type
"Dialog" pour l'option
"Hyperparallélépipède"

```

```

DHy:
DIALOG OFF:MENU 5,0,0:A!=VARPTR(Cur(
0)):A! 4:BUTTON CLOSE 1:CLS:Ind=
Ind+1:FOR I=0 TO 2^Ind-1:N=I:FOR
J=Ind-1 TO 0 STEP -1:Nn(J)=INT(N/
2^J):N=N-Nn(J)*2^J:NEXT:Xn(I)=Xorg
:Yn(I)=Yorg:FOR J=0 TO Ind-1:IF N
n(J)<>0 THEN Xn(I)=Xn(I)+VectX(J):
Yn(I)=Yn(I)+VectY(J)
NEXT:FOR J=0 TO Ind-1:IF Nn(J)=0 THEN
Xvect(J)=Xn(I)+VectX(J):Yvect(J)=
Yn(I)+VectY(J):LINE(Xn(I),Yn(I))-
(Xvect(J),Yvect(J))
NEXT:NEXT:BUTTON 1,1,"Autre",(440,290
)-(500,310):MENU 5,0,1:DIALOG ON
:INITCURSOR:RETURN Att

```

```

REM Tracé d'un fractal à l'intérieur
d'un triangle

```

```

Dtr:
FOR J=-5 TO 3 STEP 4:X1(Com)=Cx+X*cos(

```

```
J*PIdiv6#):Y1(Com)=Cy-Y*SIN(J*PIdiv6#):X2(Com)=Cx+X*COS((J+4)*PIdiv6#):Y2(Com)=Cy-Y*SIN((J+4)*PIdiv6#):GOSUB SpR:Com=Com-1:NEXT:RETURN
```

REM Tracé d'un fractal à l'extérieur d'un triangle

```
Htr:
FOR J=11 TO 3 STEP -4:X1(Com)=Cx+X*COS(J*PIdiv6#):Y1(Com)=Cy-Y*SIN(J*PIdiv6#):X2(Com)=Cx+X*COS((J-4)*PIdiv6#):Y2(Com)=Cy-Y*SIN((J-4)*PIdiv6#):GOSUB SpR:Com=Com-1:NEXT:RETURN
```

REM Tracé d'un fractal à l'intérieur d'un hexagone

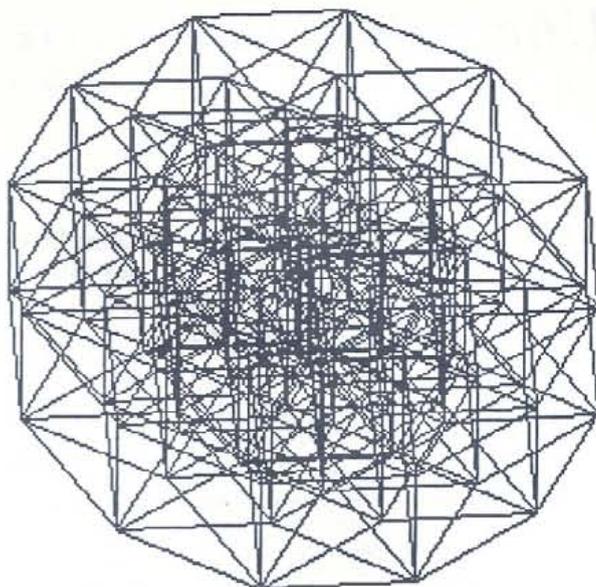
```
Dhe:
FOR J=-1 TO 9 STEP 2:X1(Com)=Cx+X*COS(J*PIdiv6#):Y1(Com)=Cy-Y*SIN(J*PIdiv6#):X2(Com)=Cx+X*COS((J+2)*PIdiv6#):Y2(Com)=Cy-Y*SIN((J+2)*PIdiv6#):GOSUB SpR:Com=Com-1:NEXT:RETURN
```

REM Tracé d'un fractal à l'extérieur d'un hexagone

```
Hhe:
FOR J=11 TO 1 STEP -2:X1(Com)=Cx+X*COS(J*PIdiv6#):Y1(Com)=Cy-Y*SIN(J*PIdiv6#):X2(Com)=Cx+X*COS((J-2)*PIdiv6#):Y2(Com)=Cy-Y*SIN((J-2)*PIdiv6#):GOSUB SpR:Com=Com-1:NEXT:RETURN
```

REM Sous-programme d'affichage fractal SpR:

```
IF Com=0 THEN LINE(X1(Com),Y1(Com))-(X2(Com),Y2(Com)):Com=Com+1:RETURN
DX(Com)=X2(Com)-X1(Com):DY(Com)=Y2(Com)-Y1(Com):X1(Com-1)=X1(Com):Y1(Com-1)=Y1(Com):X2(Com-1)=X1(Com)+Unit1#*DX(Com):Y2(Com-1)=Y1(Com)+Unit1#*DY(Com):Com=Com-1:GOSUB SpR
X1(Com-1)=X2(Com-1):Y1(Com-1)=Y2(Com-1):X2(Com-1)=X1(Com)+Unit2#*DX(Com)+Vect2#*DY(Com):Y2(Com-1)=Y1(Com)+Unit2#*DY(Com)-Vect2#*DX(Com):Com=Com-1:GOSUB SpR
X1(Com-1)=X2(Com-1):Y1(Com-1)=Y2(Com-1):X2(Com-1)=X1(Com)+Unit3#*DX(Com):Y2(Com-1)=Y1(Com)+Unit3#*DY(Com):Com=Com-1:GOSUB SpR
```



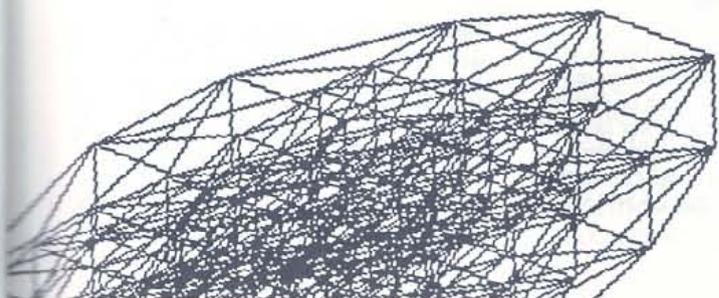
```
X1(Com-1)=X2(Com-1):Y1(Com-1)=Y2(Com-1):X2(Com-1)=X2(Com):Y2(Com-1)=Y2(Com):Com=Com-1:GOSUB SpR
Com=Com+1:RETURN
```

REM Affichage des fenêtres et initialisations nécessaires pour l'option "Fractal"

```
Ffr:
A!=VARPTR(Cur(0)):A! 4:SHOWCURSOR:Fig=1:Com=3:WINDOW 1,,(3,23)-(386,339),3:PENPAT VARPTR(Pat(0)):PENMODE 10
WINDOW 2,,(398,30)-(501,300),2:TEXTFONT 0:TEXTFACE 4:MOVETO 2,120:PRINT"Complexité ":"FOR I=0 TO 5:BUTTON I+1,ABS(I=3)+1,STR$(I),(2,132+I*16)-(50,147+I*16),3:NEXT
MOVETO 2,20:PRINT"Figure ":"BUTTON 7,2,"triangle <-",(2,32)-(100,47),3:BUTTON 8,1,"triangle ->",(2,48)-(100,63),3:BUTTON 9,1,"hexagone <-",(2,64)-(100,79),3:BUTTON 10,1,"hexagone ->",(2,80)-(100,95),3:BUTTON 11,0,"Effacer",(5,245)-(98,265)
MENU 5,0,1:INITCURSOR:RETURN
```

REM Affichage de la fenêtre et initialisations nécessaires pour l'option "Hyperparallélépipède"

```
FHy:
A!=VARPTR(Cur(0)):A! 4:SHOWCURSOR:WINDOW 1,,(3,23)-(509,339),3:BUTTON 1,0,"Valider",(440,290)-(500,310):MENU 5,0,1:A!=VARPTR(Cur(0)):A! 3:Drch=0:PENPAT VARPTR(Pat(0)):PENMODE 10:RETURN
```



Une rubrique ouverte aux lecteurs :

MacAstuces

Avec ce numéro débute cette série régulière. Nous serons heureux de vous y faire partager non seulement nos découvertes, mais aussi celles qui nous auront été proposées par les lecteurs de Pom's.

Réduire le système ?

Le fichier System comporte un certain nombre d'éléments qui ne sont pas tous indispensables pour un bon fonctionnement du Macintosh. C'est dans ce dossier que se trouvent entre autres les polices de caractères et les accessoires de bureau. Avec le système correspondant à la version 4.1 du Finder, que nous modifions ici, le fichier System occupe en général environ 130 K. Nous allons voir comment réduire ce système.

Remarques : attention à ne pas modifier le fichier système d'un programme du commerce sans en avoir fait une copie de sécurité. En effet, certains auteurs de programmes ont modifié le système et les techniques de réduction présentées ci-dessous pourraient alors avoir des conséquences destructrices. Il est parfaitement possible que certaines des réductions indiquées ci-dessous empêchent tel ou tel logiciel de tourner. Faites donc des essais avant de détruire un système original. La première étape pour gagner de la place, avant de réduire un système, consiste à remplacer le Finder par un MiniFinder, ce qui permet d'économiser d'un seul coup un cinquantaine de Ko. Il est même possible de fixer le démarrage sur un programme et d'éliminer Finder et MiniFinder : mais, dans ce cas, la sortie du programme entraîne une bombe.

Suppression des polices et accessoires

A l'aide du Font/DA Mover, vous pouvez facilement éliminer toutes les polices non indispensables. Ensuite, éliminez tous les accessoires sauf un ; gardez par exemple l'Album. S'il ne reste pas au moins un accessoire, le Finder "perd les pédales".

Avec la suppression de toutes les polices non indispensables et de tous les accessoires sauf l'Album, nous passons de 130000 à 53825 octets. A partir de maintenant, nous allons utiliser des techniques plus dangereuses : faites donc une copie de la disquette avant de continuer.

Remplacement de l'Album par un accessoire vide

Si nous voulons un système minimum, par exemple sur une disquette ne comportant que des utilitaires, nous pouvons remplacer l'Album par un accessoire "vide", qui ne fait rien. Nous utiliserons pour cela le programme Resource Editor.

Ouvrons le fichier System, ce qui nous fait voir l'ensemble des "ressources" constituant le System, toutes identifiées par un nom de trois ou quatre caractères. Ouvrons le DRVR, qui contient l'ensemble des accessoires, puis ouvrons l'Album. Nous voyons apparaître le code objet de cet accessoire, que nous remplaçons par un simple RTS (code \$4E75).

Suppression de l'imprimante

Si l'on n'a pas l'intention d'utiliser l'imprimante, on peut détruire le driver "Print" en le sélectionnant et en appuyant sur Effacer (Clear). Puis fermons DRVR et détruisons de la même façon PREC, le Printer REcord, et le fichier PAPA.

Réduction à une seule police

Ouvrons FONT et ne gardons plus que la police Chicago-12. Certes, cela rendra à l'écran les noms des disquettes et les listes sans icônes moins agréables à lire, mais cela gagnera encore de la place.

CURS

Nous pouvons éliminer les deux curseurs en forme de croix, ne gardant que la montre et le curseur d'insertion. Bien entendu, cela sera néfaste aux programmes qui, comme Multiplan, utilisent ces curseurs.

FKEY

Si nous n'avons pas besoin des clés de fonction permettant la sauvegarde ou la copie d'écran, avec la combinaison Commande, Majuscules et Blocage Majuscules, nous pouvons ouvrir FKEY. Comme nous pouvons le constater, seule les clés 3 et 4 sont programmées : fermons donc FKEY puis effaçons-le entièrement.

INIT

Si l'on n'a pas le clavier numérique, on peut éliminer le N°1, qui gère ce clavier. Si l'utilisation de la disquette sur laquelle on installe le système réduit ne requiert pas de clavier, on peut aussi éliminer le N°0 (ne le faisons pas ici). Le N°13 peut aussi être détruit.

PACK

Si l'on ne se sert pas de la calculatrice, on peut éliminer les packs 4 et 5, le premier servant aux calculs en virgule flottante et le second aux fonctions transcendentales. Attention là aussi : de nombreux programmes utilisent ces deux ressources.

PAT#

On peut éliminer les formes contenues dans PAT# si l'on n'a pas gardé le tableau de bord et si l'on ne compte pas utiliser MacPaint.

STR et STR#

Les fichiers des "strings" peuvent être totalement éliminés; ils comportent entre autres les noms de fichiers des icônes de l'imprimante (8192) et du calepin (15871 et 15872).

WDEF

Nous pouvons détruire le N°1, qui sert à définir des fenêtres à bord arrondi, telle que celle de la calculatrice.

Avec toutes ces modifications, on aboutit à un système ne faisant plus que 32143 octets, donc environ 40% plus petit.

Plus dangereux encore !

Enfin, on peut détruire les ressources qui permettent de reconstituer le bureau avec commande et option. Cela permet d'éliminer BNDL, FREF et MACS. Si le bureau doit être reconstruit, le processus sera très long et les icônes système n'apparaîtront pas. Dans la foulée, nous pouvons aussi détruire dans DITL les messages 6047, qui interviennent lors de problèmes de lecture des disques.

Nous sommes à présent descendus à 29019 octets. Nous joignons sur la disquette d'accompagnement le fichier MiniSYS correspondant. Il suffit de le copier sur une disquette et de le rebaptiser "System" pour qu'il fonctionne de façon tout à fait (?) normale.

En fait, on peut aller encore plus loin, et descendre aux environs de 20Ko ! Mais, avec le MiniFinder à la place du Finder, on se trouve déjà à un total raisonnable (moins de 35Ko) par rapport aux quelques 175Ko habituels.



Mini-Éditeur Basic

Guillaume Pot

C'est en utilisant l'assembleur Big Mac que j'ai pensé à réaliser ce programme. Il existe, en effet, dans cet assembleur de nombreuses fonctions d'édition dont on ne dispose pas lorsqu'on programme en Basic ; en particulier, les fonctions FIND (trouver une chaîne) et CHANGE (changer une chaîne). Ce programme a pour but d'assurer ces deux fonctions par le biais de l'ampersand.

Définition préliminaire

Pour bien comprendre les possibilités, mais aussi les limites, de cet éditeur il est nécessaire de définir, de façon très précise, ce qu'on appelle 'chaîne'.

Désormais, une chaîne sera une séquence de caractères contenue dans une ligne de programme et ne comportant pas d'espace, de virgule, de guillemets... Une chaîne ne peut représenter une instruction, un numéro de ligne... En fait, une chaîne sera une suite de caractères non codée par l'interpréteur Basic (pas de token). On pourra donc chercher, trouver et remplacer un nom de variable, un mot de commentaire, mais aussi un nombre référant un numéro de ligne (après THEN, GOTO ou GOSUB).

Utilisation du mini-éditeur

Le lancement du programme entraîne l'édition d'une page aide mémoire ; celle-ci contient tous les renseignements nécessaires. Cette page est accessible à tout moment par la commande & H (Help). Il suffit alors de charger le programme Basic sur lequel on effectuera des modifications.

Les deux fonctions de l'éditeur sont :

- l'édition de toutes les occurrences d'une chaîne dans le programme ;
- la substitution d'une chaîne à une autre.

Comment trouver une chaîne ?

La commande : & "chaîne assure l'édition de toutes les lignes du programme où la chaîne considérée apparaît ; si elle figure plusieurs fois dans une même ligne, la ligne sera éditée autant de fois (mais imprimée une seule fois). La barre d'espacement permet une édition pas à pas. Si la chaîne est une référence à une ligne, toutes les lignes contenant cette référence seront éditées, la ligne elle-même sera éditée si elle la contient.

Comment changer une chaîne ?

La commande & "chaîne1"chaîne2 permet de remplacer, à volonté, la chaîne1 par la chaîne2. L'éditeur affiche la ligne du programme qui contient la première occurrence, puis attend une touche. Plusieurs réponses sont possibles :

- O comme Oui change la chaîne, édite la nouvelle ligne, puis celle contenant l'occurrence suivante ;
- RETURN entraîne le changement automatique de l'ancienne chaîne par la nouvelle dans tout le programme. L'édition des nouvelles lignes est instantanée ;
- toute autre touche, à l'exception de CTRL-C (break), conserve la chaîne et entraîne l'édition de la ligne contenant l'occurrence suivante.

Quelques conseils

Comme avec tout autre éditeur, il faut faire preuve d'une grande prudence dans l'utilisation de la fonction & "chaîne1"chaîne2. Il est indispensable de garder une copie du programme original avant de procéder à un changement généralisé. Ne vous hâtez pas de presser la touche RETURN lors d'une substitution de chaîne ; il pourrait arriver que la chaîne à remplacer soit une sous-chaîne d'une autre. Par exemple : xx dans xx% ou encore 110 dans 1100. Pour changer le nom d'une variable (xx%, xx\$, xx), il faut, bien sûr, inclure la spécificité dans les chaînes 1 et 2. Par exemple, pour modifier le nom d'une variable dimensionnée, il est préférable de faire & "xx\$("yy\$(plutôt que & "xx"yy.

Analyse du programme

La seule difficulté technique du programme réside dans l'utilisation d'une partie de la boucle principale utilisée par l'interpréteur Basic (CMDLOOP). Cette partie (\$D46C - \$D52B) a été reprise pour éviter la sortie forcée, mais inopportune, dans l'interpréteur. Le programme procède de la façon suivante :

- 1) il stocke la nouvelle ligne, déjà codée, dans le buffer d'entrée (\$200) et sa longueur en \$0F ;
- 2) il a recours à la partie utilisée par l'interpréteur pour insérer la nouvelle ligne dans le programme Basic ;
- 3) puis, retour au programme pour l'édition de la nouvelle ligne.



Source MINIEDIT.S Assembleur Big Mac

```

1 *****
2 * chercher/changer une *
3 * chaine dans un programme *
4 * Basic. guillaume pot *
5 *****
6
7
8 LENGPT = $0F
9 WNDTOP = $22
10 CH = $24
11 CV = $25
12 INVFLG = $32
13
14 LINNUM = $50
15 INDEX = $5E
16 INDIK = $60
17 TXTTAB = $67
18 LOMEM = $69
19 FREBEG = $6D
20 FRETOP = $6F
21 HIMEM = $73
22
23 HIGHDS = $94
24 HIGHTR = $96
25 LOWTR = $9B
26 PRGEND = $AF
27 CHRGET = $B1
28 CHRGOT = $B7
29 TXTPTR = $B8
30
31 LINPT = $F8
32 ALLFLC = $F9
33 LENG1 = $FA
34 LENG2 = $FB
35 STRBEG = $FC
36 STREND = $FD
37 LINE = $FE
38
39 IN = $200
40 WARMDS = $3D0
41 INTDOS = $3EA
42 AMPERV = $3F6
43
44     ORG $8E00
45
46     LDA #<BEGIN
47     STA AMPERV
48     STA HIMEM
49     LDA #>BEGIN
50     STA AMPERV+1
51     STA HIMEM+1
52     BNE HELP
53
54 ** saisie commandes **
55
56 BEGIN   LDA #0
57         LDX #3
58 INIPRM  STA SLOT,X
59         STA ALLFLG,X
60         DEX
61         BPL INIPRM
62         STX LAST
63         STX LAST+1
64         STX STRBEG
65         JSR CHRGOT
66         BNE FNDCMD
67 BEGEND  JMP EXEC
68
69 GETCMD  JSR CHRGET
70         BEQ BEGEND
71 FNDCMD  LDX #4
72 FND1    CMP CMDCHR,X
73         BEQ TESTH
74         DEX
75         BPL FND1
76         JMP SYNTAX
77
78 TESTH  CPX #4      ;= 'H'
79         BNE NOTH
80 HELP   JSR HOME
81         LDA #0
82         JSR PRINT
83         JMP BASIC2
84
85 NOTH   CPX #3      ;= ''
86         BEQ GETSTR
87         TXA
88         PHA
89         JSR CHRGET
90         JSR FRMNUM
91         JSR GETADR
92         PLA
93         ASL
94         TAX
95         LDA LINNUM
96         STA SLOT,X
97         LDA LINNUM+1
98         STA SLOT+1,X
99         JSR CHRGOT
100        BEQ EXEC
101        CMP #','
102        BEQ GETCMD
103        JMP SYNTAX
104
105 GETSTR  LDX #0
106 GETSTR1 JSR TSTCHR
107         BEQ STRIEND
108         STA STRNG1,X
109         INX
110        BNE GETSTR1
111
112 STRIEND STX LENG1
113         LDA #0
114         STA STRNG1,X
115 XSTREND CPX #16
116         BCS STRERR
117         JSR CHRGOT
118         BEQ EXEC
119         CMP #','
120         BEQ GETCMD
121         LDX LENG2
122         BNE EXEC
123
124 GETSTR2 JSR TSTCHR
125         BEQ STR2END
126         STA STRNG2,X
127         INX
128         BNE GETSTR2
129
130 STR2END STX LENG2
131         LDA #0
132         STA STRNG2,X
133         BEQ XSTREND
134
135 TSTCHR  JSR CHRGET
136         BEQ RETURN
137         CMP #','
138         BEQ RETURN
139         CMP #','
140 RETURN  RTS
141
142 STRERR  JMP TOOLNG
143
144 ** execute commandes **
145
146 EXEC    LDA LENG2
147         BEQ EXEC1
148         LDA #0
149         STA SLOT
150 EXEC1   JSR PRINTER
151         JSR TITLE
152         LDY #1
153         LDX TXTTAB
154         LDA TXTTAB+1
155         BNE NXTADR
156
157 NXTLIN  LDY #0
158         LDA (LINE),Y
159         TAX
160         INY
161         LDA (LINE),Y
162 NXTADR  STX LINE
163         STA LINE+1
164         LDA (LINE),Y
165         BNE XFIRST
166
167 EXEND   LDA LENG1
168         BEQ EXIT
169         JSR MARGE
170         LDA #5      ;= END
171         JSR PRINT
172 EXIT    JSR SETVID
173         JSR INIT
174         JSR UP
175         JMP WARMDS
176
177 XFIRST  JSR GETNUM
178         CMP FIRST+1
179         BCC NXTLIN
180         BNE XLAST
181         CPX FIRST
182         BCC NXTLIN
183 XLAST   CMP LAST+1
184         BCC XLENG1
185         BNE EXEND
186         LDA LAST
187         CMP LINNUM
188         BCC EXEND
189 XLENG1  LDA LENG1
190         BEQ LISTING
191         STY STRBEG
192         BNE SEARCH
193
194 LISTING JSR LIST
195         JSR KEYSTT
196         LDA LENG1
197         BEQ NXTLIN
198         LDA SLOT
199         BNE NXTLIN
200 SRCH    JSR CROUT
201
202 *** chercher une chaine ***
203
204 SEARCH  LDX #0
205         INC STRBEG
206         LDY STRBEG
207 NXTCHR  LDA (LINE),Y
208         BMI SEARCH
209         BEQ NXTLIN

```

210	CMP	STRNG1,X	266	STA	LOMEM	322	LDA	TXTTAB		
211	BNE	SEARCH	267	STA	INDIX	323	LDY	TXTTAB+1		
212	INY		268	LDA	LOMEM+1	324	STA	INDEX		
213	INX		269	ADC	#\$FF	325	STY	INDEX+1		
214	CPX	LENG1	270	STA	LOMEM+1	326	CLC			
215	BNE	NXTCHR	271	SBC	LOWTR+1	327	HD4FE	LDY	#\$01	
216	STY	STREND	272	TAX		328	LDA	(INDEX),Y		
217	LDA	LENG2	273	SEC		329	BNE	HD50F		
218	BEQ	LISTING	274	LDA	LOWTR	330	LDA	LOMEM		
219			275	SBC	LOMEM	331	STA	PRGEND		
220	***	changer une chaine	276	TAY		332	LDA	LOMEM+1		
221			277	BCS	HD49F	333	STA	PRGEND+1		
222	LDA	ALLFLG	278	INX		334	JMP	DONE		
223	BNE	CHANGE	279	DEC	INDIX+1	335				
224	JSR	LIST	280	HD49F	CLC	336	HD50F	LDY	#\$04	
225	JSR	RDKEY	281	ADC	INDEX	337	HD511	INY		
226	CMP	#"O"	282	BCC	HD4A7	338	LDA	(INDEX),Y		
227	BEQ	CHANGE	283	DEC	INDEX+1	339	BNE	HD511		
228	CMP	#\$83	284	CLC		340	INY			
229	BEQ	EXIT	285	HD4A7	LDA	(INDEX),Y	341	TYA		
230	CMP	#\$8D	286	STA	(INDIX),Y	342	ADC	INDEX		
231	BNE	SRCH	287	INY		343	TAX			
232	STA	ALLFLG	288	BNE	HD4A7	344	LDY	#\$00		
233	CHANGE	JSR	NEWLIN	289	INC	INDEX+1	345	STA	(INDEX),Y	
234	BNE	CMDLOOP	290	INC	INDIX+1	346	LDA	INDEX+1		
235	DONE	LDA	STRBEG	291	DEX		347	ADC	#\$00	
236	ADC	LENG2	292	BNE	HD4A7	348	INY			
237	STA	STREND	293			349	STA	(INDEX),Y		
238	JSR	LIST	294	NOTFND	LDA	HIMEM	350	STX	INDEX	
239	JSR	KEYTST	295	LDY	HIMEM+1	351	STA	INDEX+1		
240	LDY	STREND	296	STA	FRETOP	352	BCC	HD4FE		
241	DEY		297	STY	FRETOP+1	353				
242	STY	STRBEG	298	LDA	LOMEM	354	**	routine	new line	**
243	BNE	SRCH	299	STA	HIGHTR	355				
244			300	ADC	LENGPT	356	NEWLIN	LDA	#0	
245	*insere	une nouvelle ligne*	301	STA	HIGHDS	357		STA	LENCPT	
246			302	LDY	LOMEM+1	358		LDY	#3	
247	CMDLOOP	LDX	PRGEND	303	STY	HIGHTR+1	359	NXTCAR	INY	
248	STX	LOMEM	304	BCC	HD4D1	360	CPY	STRBEG		
249	LDX	PRGEND+1	305	INY		361	BNE	GETCAR		
250	STX	LOMEM+1	306	HD4D1	STY	HIGHDS+1	362	LDY	#0	
251	JSR	GETNUM	307	JSR	BLTU	363	NEWSTR	LDA	STRNG2,Y	
252	JSR	FNDLI	308	LDA	LINNUM	364	JSR	INBUF		
253	BCC	NOTFND	309	LDY	LINNUM+1	365	INY			
254	LDY	#\$01	310	STA	IN-2	366	CPY	LENG2		
255	LDA	(LOWTR),Y	311	STY	IN-1	367	BNE	NEWSTR		
256	STA	INDEX+1	312	LDA	FREBEG	368	LDY	STREND		
257	LDA	LOMEM	313	LDY	FREBEG+1	369	GETCAR	LDA	(LINE),Y	
258	STA	INDEX	314	STA	LOMEM	370	JSR	INBUF		
259	LDA	LOWTR+1	315	STY	LOMEM+1	371	CMP	#0		
260	STA	INDIX+1	316	LDY	LENGPT	372	BNE	NXTCAR		
261	LDA	LOWTR	317	MOVLIN	LDA	IN-5,Y	373	CPX	#\$FF	
262	DEY		318	DEY		374	BCS	LINERR		
263	SBC	(LOWTR),Y	319	STA	(LOWTR),Y	375	TXA			
264	CLC		320	BNE	MOVLIN	376	ADC	#5		
265	ADC	LOMEM	321	JSR	CLEAR	377	STA	LENGPT		

Éditeur Plein Écran EPE

Le Pacha

Apple][+, IIe, IIc

Listez vos programmes Basic en avant et en arrière.

Modifiez, insérez, effacez des caractères en plein écran sans relire les lignes.

Recherchez toute chaîne de caractères.

Choisissez vous-même les codes de contrôle d'EPE.

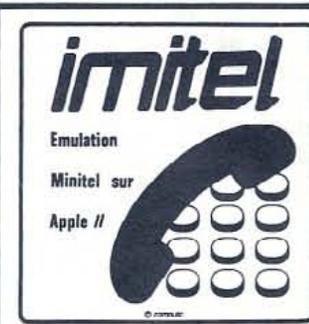
Modifiez EPE : le fichier source est sur la disquette.

150,00 F TTC franco (bon de commande page 74).

378	RTS	450	BPL	NXTCRC	522	LDA	LENG1
379		451	BMI	NXTCMD	523	BEQ	RET1
380	INBUF LDX LENGPT	452			524	JSK	LIGNE
381	STA IN,X	453	CMDOK	LDY #0	525	JSR	MARGE
382	INC LENGPT	454	CMDLEN	INY	526	LDA	#1 ;= FIND
383	RTS	455	LDA	(TXTPTR),Y	527	LDX	LENG2
384		456	BPL	CMDLEN	528	BEQ	TITLE1
385	LINERR JMP TOOLNG	457	INY		529	LDA	#2 ;= OLD
386		458	CLC		530	JSR	PRINT
387	*** routine list ***	459	TYA		531	JSR	MARGE
388		460	ADC	CH	532	LDA	#3 ;= NEW
389	GETNUM LDY #2	461	CMP	#\$26	533	JSR	PRINT
390	LDA (LINE),Y	462	BCC	CMDPRT	534	LDA	#4 ;= STR2
391	TAX	463	JSR	MARGE	535	TITLE1	JSR PRINT
392	STX LINNUM	464			536	JSR	LIGNE
393	INY	465	CMDPRT	LDA #SA0	537	LDX	CV
394	STY LINPT	466	JSR	COUT	538	INX	
395	LDA (LINE),Y	467	CMDPRT1	JSR CHRGET	539	STX	WNDTOP
396	STA LINNUM+1	468	TAX		540	JMP	CROUT
397	RTS	469	BMI	CMDPRT2	541		
398		470	ORA	#\$80	542	LIGNE	JSR CROUT
399	LIST JSR GETNUM	471	JSR	COUT	543	LDA	SLOT
400	INY	472	BNE	CMDPRT1	544	BNE	RET2
401	PHA	473	CMDPRT2	JSR COUT	545	LDX	#\$26
402	POSIT CMP BASH,Y	474	LDA	#\$20	546	LDA	#"-"
403	BCC POSIT1	475	BNE	INSPT	547	LIGNE1	JSR COUT
404	BNE NUMPRT	476			548	DEX	
405	TXA	477	OUTCHR	ORA #80	549	BNE	LIGNE1
406	CMP BASL,Y	478	JSR	COUT	550	RET2	RTS
407	BCS NUMPRT	479	LDA	CH	551		
408	POSIT1 LDA #SA0	480	CMP	#\$26	552	SLTERR	JMP ILLEG
409	JSR COUT	481	BCS	MARGE	553		
410	LDA LINNUM+1	482	RTS		554	PRINTER	LDA SLOT+1
411	DEY	483			555	BNE	SLTERR
412	BPL POSIT	484	** routines diverses **		556	IDA	SLOT
413		485			557	BEQ	RET2
414	NUMPRT PLA	486	KEYTST	LDA KEY	558	CMP	#9
415	JSR LINPRT	487	BPL	CONT	559	BCS	SLTERR
416	LDA #SA0	488	CMP	#\$83 ;= ^C	560	JSR	OUTPORT
417	JSR COUT	489	BNE	TSTSPC	561	JSR	INTDOS
418		490	BREAK	RIT STROBE	562	LDA	#6 ;= SETUP
419	LDY #504	491	JMP	EXIT	563	PRINT	ASL
420	LDA (LINE),Y	492			564	TAY	
421	BMI DECODE	493	TSTSPC	CMP #SA0	565	LDA	STRAD,Y
422	LDA #SA0	494	BNE	NOTSPC	566	STA	LINE
423	INSPRT LDY LINPT	495	BIT	STROBE	567	INY	
424	CHRPRT JSR OUTCHR	496	PAUSE	LDA KEY	568	LDA	STRAD,Y
425	INY	497	BPL	PAUSE	569	STA	LINE+1
426	CPY STRBEG	498	CMP	#\$83 ;= ^C	570	LDY	#0
427	BCC GETCHR	499	BEQ	BREAK	571	PRINT1	LDA (LINE),Y
428	BNE TSTEND	500	CMP	#\$A0	572	BEQ	RET2
429	JSR SETINV	501	BEQ	CONT	573	ORA	#\$80
430	TSTEND CPY STREND	502	NOTSPC	BIT STROBE	574	JSR	COUT
431	BNE GETCHR	503	CONT	CLC	575	INY	
432	JSR SETNRM	504	RTS		576	BNE	PRINT1
433		505			577	INC	LINE+1
434	GETCHR LDA (LINE),Y	506	MARGE	JSR CROUT	578	BNE	PRINT1
435	BNE DECODE	507		LDA INVFLG	579		
436	JMP CROUT	508		PHA	580		
437		509	BMI	MARGE1	581	STRAD	DA AIDE
438	DECODE BPL CHRPT	510	JSR	SETNRM	582	DA	TITRE1
439	STY LINPT	511	MARGE1	LDX #7	583	DA	TITRE2
440	AND #57F	512	JSR	PRBL2	584	DA	TITRE3
441	TAX	513		PLA	585	DA	STRNG2
442	LDA #>CMDLST	514	BMI	RET1	586	DA	FIN
443	STA TXTPTR+1	515	SETINV	LDA #S3F	587	DA	SETUP
444	LDA #<CMDLST	516	BNE	SETFLG	588		
445	STA TXTPTR	517	SETNRM	LDA #SFF	589	AIDE	HEX 8D
446	NXTCMD DEX	518	SETFLG	STA INVFLG			
447	BMI CMDOK	519	RET1	RTS			
448	NXTCRC JSR CHRGET	520					
449	TAY	521	TITLE	JSR HOME			

Note : afin de faciliter la lecture des instructions ASC, les espaces ont été remplacés par des 'puces' (*).

590	ASC	"*****"	610	ASC	".../..."	>1000,>10000
		*****	611	ASC	"*==CHANGER*UN	637 TITRE3 ASC "NOUVELLE*CHA
		*****"			E*CHAINE"	INE:.*"
591	HEX	8D	612	HEX	8D8D	638 HEX 00
592	ASC	"*...TROUVER*O	613	ASC	"VALEURS*PAR*D	639 TITRE2 ASC "ANCIENNE*"
		U*CHANGER*UNE*			EFAUT*:*&*SO,F	640 TITRE1 ASC "CHAINE:.*"
		CHAINE*****"			0,L65535"	641 STRNG1 HEX 00
593	HEX	8D	614	HEX	8D8D8D	642
594	ASC	"*...DANS*UN	615	ASC	"<<<*LES*AUTR	643 STRNG2 = STRNG1+16
		*PROGRAMME*BAS			ES*CLES*****>>	644
		IC*****"			>"	645
595	HEX	8D	616	HEX	8D8D	646 KEY = \$C000
596	ASC	"*...PAR*GUIL	617	ASC	"ONE*****CHAN	647 STROBE = \$C010
		LAUME**POT**			GE*UNE***ESPA	648
		*****"			CE**PAUSE"	649 CMDLST = \$D0CF
597	HEX	8D	618	HEX	8D	650 BLTU = \$D393
598	ASC	"*****"	619	ASC	"RETURN*==CHAN	651 FNDLIN = \$D61A
		*****"			GE*TOUT***CTR	652 CLEAR = \$D665
		*****"			L-C--STOP**"	653
599	HEX	8D8D8D	620	HEX	00	654 FRMNUM = \$DD67
600	ASC	"<<<*LES*CLES	621	ASC	"END-OF-PASS"	655 SYNTAX = \$DEC9
		*DE*L*AMPERSAN	622 FIN	ASC	"END-OF-PASS"	656 BASIC2 = \$E003
		D*...>>>"	623	HEX	8D00	657 ILLEG = \$E199
601	HEX	8D8D	624	HEX	89	658 TOOLNG = \$E5B2
602	ASC	"F#*==PREMIERE	625 SETUP	HEX	89	659 GETADR = \$E752
		*TIGNE*****S	626	ASC	"80N"	660 LINPRT = \$ED24
		#*...SLOT"	627	HEX	00	661
603	HEX	8D	628	ASC	'SFL"H'	662 PRBL2 = \$F94A
604	ASC	"I.#*==DERNIERE	629 CMDCHR	ASC	'SFL"H'	663 INIT = \$FB2F
		*LIGNE*****H	630	HEX	0000	664 UP = \$FC1A
		**AIDE"	631 SLOT	HEX	0000	665 HOME = \$FC58
605	HEX	8D8DA2	632 FIRST	HEX	0000	666 RDKEY = \$FD0C
606			633 LAST	HEX	FFFF	667 CROUT = \$FD8E
607	ASC	"..."	634	DFB	1,10,100,1000,	668 COUT = \$FDED
608	ASC	"*...TROUVE	635 BASL	DFB	10000	669 SETVID = \$FE93
		R*UNE*CHAINE"	636 BASH	DFB	>1,>10,>100,	670 OUTPORT = \$FE95
609	HEX	8DA2				



Transformez votre Apple //c en Minitel graphique et couleur avec sauvegarde des informations sur imprimante ou sur disquettes.

- Existe aussi en version texte (sans graphique)
- (nécessite un modem V23 ou un adaptateur LINK //c entre le Minitel et l'Apple //c.)

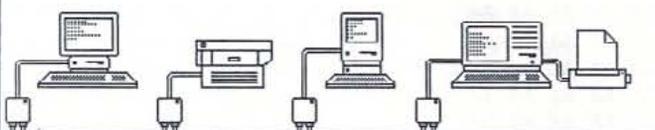
COPY II MAC™ 4.5

- DUPLIQUE LES LOGICIELS PROTÉGÉS *
 - CONTIENT DES UTILITAIRES PUISSANTS
 - Formatage et vérification de disque
 - Editeur de secteurs et fichiers
 - Réparation de disques endommagés
 - Installation de programmes protégés sur disque dur
 - Configuration : Macintosh 128 K ou 512 K
- Versions aussi disponibles pour IBM, Apple // et Commodore 64 !

Ces produits vous sont vendus pour permettre de réaliser des copies de sauvegarde uniquement.

XL/Serve et Mac Serve

Les premiers logiciels de réseaux pour Appletalk permettant le partage d'imprimantes et de disques durs.



Pour Macintosh 512 K, Macintosh XL, Macintosh Plus, Lisa.

COMPUTIC

143, Grande Rue
59100 ROUBAIX
Tél. 20.73.93.73

PRODUITS	PRIX TTC
IMITEL// - V 1.1 (version sans graphique) (Apple//c).....	495,00
IMITEL// - V 2.0 (version avec graphique) (Apple//c).....	895,00
Copy II Mac V 4.5	595,00
Copy II Plus V 5.5 (Apple//).....	595,00
Logiciel de réseau : XL/SERVE	2595,00
Logiciel de réseau : MAC SERVE (disponible le 20/02/86).....	3795,00
Link//c (interface minitel/Apple//c) prévu 02/86	495,00

Ces logiciels sont aussi disponibles chez votre Concessionnaire Agréé Apple.

Récapitulation MINIEDIT

Après la saisie de ce source sous moniteur,
vous le sauvegarderez par :

BSAVE MINIEDIT, A\$8E00, L\$647

8E00- A9 10 8D F6 03 85 73 A9
8E08- 8E 8D F7 03 85 74 D0 32
8E10- A9 00 A2 03 9D 13 94 95
8E18- F9 CA 10 F8 8E 17 94 8E
8E20- 18 94 86 FC 20 B7 00 D0
8E28- 08 4C BD 8E 20 B1 00 F0
8E30- F8 A2 04 DD 0E 94 F0 06
8E38- CA 10 F8 4C C9 DE E0 04
8E40- D0 0B 20 58 FC A9 00 20
8E48- C9 91 4C 03 E0 E0 03 F0
8E50- 24 8A 48 20 B1 00 20 67
8E58- DD 20 52 E7 68 0A AA A5
8E60- 50 9D 13 94 A5 51 9D 14
8E68- 94 20 B7 00 F0 4F C9 2C
8E70- F0 BA 4C C9 DE A2 00 20
8E78- AE 8E F0 06 9D 46 94 E8
8E80- D0 F5 86 FA A9 00 9D 46
8E88- 94 E0 10 B0 2D 20 B7 00
8E90- F0 2B C9 2C F0 96 A6 FB
8E98- D0 23 20 AE 8E F0 06 9D
8EA0- 56 94 E8 D0 F5 86 FB A9
8EA8- 00 9D 56 94 F0 DB 20 B1
8EB0- 00 F0 06 C9 2C F0 02 C9
8EB8- 2F 60 4C B2 E5 A5 FB F0
8EC0- 05 A9 00 8D 13 94 20 B3
8EC8- 91 20 6D 91 A0 01 A6 67
8ED0- A5 68 D0 08 A0 00 B1 FE
8ED8- AA C8 B1 FE 86 FE 85 FF
8EE0- B1 FE D0 18 A5 FA F0 08
8EE8- 20 51 91 A9 05 20 C9 91
8EF0- 20 93 FE 20 2F FB 20 1A
8EF8- FC 4C D0 03 20 7E 90 CD
8F00- 16 94 90 D0 D0 05 EC 15
8F08- 94 90 C9 CD 18 94 90 09
8F10- D0 D2 AD 17 94 C5 50 90
8F18- CB A5 FA F0 04 84 FC D0
8F20- 12 20 8D 90 20 29 91 A5
8F28- FA F0 A9 AD 13 94 D0 A4
8F30- 20 8E FD A2 00 E6 FC A4
8F38- FC B1 FE 30 F6 F0 95 DD
8F40- 46 94 D0 EF C8 E8 E4 FA
8F48- D0 EF 84 FD A5 FB F0 D1
8F50- A5 F9 D0 14 20 8D 90 20
8F58- 0C FD C9 CF F0 0A C9 83
8F60- F0 8E C9 8D D0 CA 85 F9
8F68- 20 46 90 D0 13 A5 FC 65
8F70- FB 85 FD 20 8D 90 20 29
8F78- 91 A4 FD 88 84 FC D0 B0
8F80- A6 AF 86 69 A6 B0 86 6A
8F88- 20 7E 90 20 1A D6 90 44
8F90- A0 01 B1 9B 85 5F A5 69
8F98- 85 5E A5 9C 85 61 A5 9B
8FA0- 88 F1 9B 18 65 69 85 69
8FA8- 85 60 A5 6A 69 FF 85 6A
8FB0- E5 9C AA 38 A5 9B E5 69
8FB8- A8 B0 03 E8 C6 61 18 65
8FC0- 5E 90 03 C6 5F 18 B1 5E
8FC8- 91 60 C8 D0 F9 E6 5F E6
8FD0- 61 CA D0 F2 A5 73 A4 74
8FD8- 85 6F 84 70 A5 69 85 96
8FE0- 65 0F 85 94 A4 6A 84 97
8FE8- 90 01 C8 84 95 20 93 D3

8FF0- A5 50 A4 51 8D FE 01 8C
8FF8- FF 01 A5 6D A4 6E 85 69
9000- 84 6A A4 0F B9 FB 01 88
9008- 91 9B D0 F8 20 65 D6 A5
9010- 67 A4 68 85 5E 84 5F 18
9018- A0 01 B1 5E D0 0B A5 69
9020- 85 AF A5 6A 85 B0 4C 6D
9028- 8F A0 04 C8 B1 5E D0 FB
9030- C8 98 65 5E AA A0 00 91
9038- 5E A5 5F 69 00 C8 91 5E
9040- 86 5F 85 F9 90 D2 A9 00
9048- 85 0F A0 03 C8 C4 FC D0
9050- 0F A0 00 B9 56 94 20 73
9058- 90 C8 C4 FB D0 F5 A4 FD
9060- B1 FE 20 73 90 C9 00 D0
9068- E3 E0 EF B0 0E 8A 69 05
9070- 85 0F 60 A6 0F 9D 00 02
9078- E6 0F 60 4C B2 E5 A0 02
9080- B1 FE AA 86 50 C8 84 F8
9088- B1 FE 85 51 60 20 7E 90
9090- C8 48 D9 1E 94 90 08 D0
9098- 10 8A D9 19 94 B0 0A A9
90A0- A0 20 ED FD A5 51 88 10
90A8- E9 68 20 24 ED A9 A0 20
90B0- ED FD A0 04 B1 FE 30 1F
90B8- A9 A0 A4 F8 20 1D 91 C8
90C0- C4 FC 90 UC D0 03 20 64
90C8- 91 C4 FD D0 03 20 68 91
90D0- B1 FE D0 03 4C 8E FD 10
90D8- E3 84 F8 29 7F AA A9 D0
90E0- 85 B9 A9 CF 85 B8 CA 30
90E8- 08 20 B1 00 A8 10 FA 30
90F0- F5 A0 00 C8 B1 B8 10 FB
90F8- C8 18 98 65 24 C9 26 90
9100- 03 20 51 91 A9 A0 20 ED
9108- FD 20 B1 00 AA 30 07 09
9110- 80 20 ED FD D0 F3 20 ED
9118- FD A9 20 D0 9D 09 80 20
9120- ED FD A5 24 C9 26 B0 29
9128- 60 AD 00 C0 10 21 C9 83
9130- D0 06 2C 10 C0 4C F0 8E
9138- C9 A0 D0 10 2C 10 C0 AD
9140- 00 C0 10 FB C9 83 F0 EA
9148- C9 A0 F0 03 2C 10 C0 18
9150- 60 20 8E FD A5 32 48 30
9158- 03 20 68 91 A2 07 20 4A
9160- F9 68 30 08 A9 3F D0 02
9168- A9 FF 85 32 60 20 58 FC
9170- A5 FA F0 F8 20 9D 91 20
9178- 51 91 A9 01 A6 FB F0 0F
9180- A9 02 20 C9 91 20 51 91
9188- A9 03 20 C9 91 A9 04 20
9190- C9 91 20 9D 91 A6 25 E8
9198- 86 22 4C 8E FD 20 8E FD
91A0- AD 13 94 D0 0A A2 26 A9
91A8- AD 20 ED FD CA D0 FA 60
91B0- 4C 99 E1 AD 14 94 D0 F8
91B8- AD 13 94 F0 F2 C9 09 B0
91C0- EF 20 95 FE 20 EA 03 A9
91C8- 06 0A A8 B9 E8 91 85 FE
91D0- C8 B9 E8 91 85 FF A0 00
91D8- B1 FE F0 D3 09 80 20 ED
91E0- FD C8 D0 F4 E6 FF D0 F0
91E8- F6 91 3E 94 35 94 23 94
91F0- 56 94 FC 93 09 94 8D AA
91F8- AA AA AA AA AA AA AA AA
9200- AA AA AA AA AA AA AA AA
9208- AA AA AA AA AA AA AA AA
9210- AA AA AA AA AA AA AA AA
9218- AA AA AA AA AA AA 8D AA

9220- AA A0 A0 D4 D2 CF D5 D6
9228- C5 D2 A0 CF D5 A0 C3 C8
9230- C1 CE C7 C5 D2 A0 D5 CE
9238- C5 A0 C3 C8 C1 C9 CE C5
9240- A0 A0 A0 A0 AA AA 8D AA
9248- AA A0 A0 A0 A0 C4 C1 CE
9250- D3 A0 D5 CE A0 D0 D2 CF
9258- C7 D2 C1 CD CD C5 A0 C2
9260- C1 D3 C9 C3 A0 A0 A0 A0
9268- A0 A0 A0 A0 AA AA 8D AA
9270- AA A0 A0 D0 C1 D2 A0 A0
9278- C7 D5 C9 CC CC C1 D5 CD
9280- C5 A0 A0 A0 D0 CF D4 A0
9288- A0 A0 A0 A0 A0 A0 A0 A0
9290- A0 A0 A0 A0 AA AA 8D AA
9298- AA AA AA AA AA AA AA AA
92A0- AA AA AA AA AA AA AA AA
92A8- AA AA AA AA AA AA AA AA
92B0- AA AA AA AA AA AA AA AA
92B8- AA AA AA AA AA AA 8D 8D
92C0- 8D BC BC BC A0 A0 CC C5
92C8- D3 A0 C3 CC C5 D3 A0 C4
92D0- C5 A0 C0 A0 C1 CD D0 C5
92D8- D2 D3 C1 CE C4 A0 A0 BE
92E0- BE BE 8D 8D C6 A3 A0 BD
92E8- A0 D0 D2 C5 CD C9 C5 D2
92F0- C5 A0 CC C9 C7 CE C5 A0
92F8- A0 A0 A0 A0 A0 A0 D3 A3
9300- A0 BD A0 D3 CC CF D4 8D
9308- CC A3 A0 BD A0 C4 C5 D2
9310- CE C9 C5 D2 C5 A0 CC C9
9318- C7 CE C5 A0 A0 A0 A0 A0
9320- A0 A0 C8 A0 A0 BD A0 C1
9328- C9 C4 C5 8D 8D A2 AE AE
9330- AE A0 A0 A0 A0 A0 BD A0
9338- D4 D2 CF D5 D6 C5 D2 A0
9340- D5 CE C5 A0 C3 C8 C1 C9
9348- CE C5 8D A2 AE AE AE AF
9350- AE AE AE A0 BD A0 C3 C8
9358- C1 CE C7 C5 D2 A0 D5 CE
9360- C5 A0 C3 C8 C1 C9 CE C5
9368- 8D 8D D6 C1 CC C5 D5 D2
9370- D3 A0 D0 C1 D2 A0 C4 C5
9378- C6 C1 D5 D4 A0 BA A0 A6
9380- A0 D3 B0 AC C6 B0 AC CC
9388- B6 B5 B5 B3 B5 8D 8D 8D
9390- BC BC BC A0 A0 CC C5 D3
9398- A0 C1 D5 D4 D2 C5 D3 A0
93A0- C3 CC C5 D3 A0 A0 A0 A0
93A8- A0 BE BE BE 8D 8D CF CE
93B0- C5 A0 A0 A0 A0 BD A0 C3
93B8- C8 C1 CE C7 C5 A0 D5 CE
93C0- C5 A0 A0 A0 A0 C5 D3 D0
93C8- C1 C3 C5 A0 A0 BD A0 D0
93D0- C1 D5 D3 C5 8D D2 C5 D4
93D8- D5 D2 CE A0 BD A0 C3 C8
93E0- C1 CE C7 C5 A0 D4 CF D5
93E8- D4 A0 A0 A0 A0 C3 D4 D2
93F0- CC A0 C3 A0 BD A0 D3 D4
93F8- CF D0 A0 00 C5 CE C4 AD
9400- CF C6 AD D0 C1 D3 D3 8D
9408- 00 89 B8 B0 CE 00 53 46
9410- 4C 22 48 00 00 00 00 FF
9418- FF 01 0A 64 E8 10 00 00
9420- 00 03 27 CE CF D5 D6 C5
9428- CC CC C5 A0 C3 C8 C1 C9
9430- CE C5 BA A0 00 C1 CE C3
9438- C9 C5 CE CE C5 A0 C3 C8
9440- C1 C9 CE C5 BA A0 00

S I X L E T T R E S

Vous connaissez tous le jeu du "Master Mind", mais avez-vous déjà joué aux "Six lettres" ? Ce divertissement, connu depuis une bonne cinquantaine d'années, utilise des règles et une présentation fort semblables.

Le jeu

Il consiste à retrouver des mots de six lettres présents dans les dictionnaires courants (nous ne ferons pas de publicité). Le programme affiche, pour chaque proposition, le nombre de lettres correctement placées.

Le détail des règles est affiché en début de programme. Il est également possible de modifier le fichier contenant le dictionnaire (un millier de mots). D'autre part, le programme gère un index afin que le joueur ait à trouver tous les mots du fichier avant de retomber sur un mot déjà tiré.

Le programme

Les fichiers utilisés sont :

- SIX LETTRES
menu général;
- SIX LETTRES.TEXT
fichier renfermant les mots à trouver;
- SIX LETTRES.JEU
le jeu proprement dit;
- SIX LETTRES.REGLES
affichage des règles;
- SIX LETTRES.MODIFIER
modification du fichier texte.

On accède au programme par :
RUN SIX LETTRES

La configuration nécessaire comprend un Apple II+, IIe ou IIc sous DOS 3.3 ou ProDOS.

Les lecteurs de Pom's qui ne disposent pas de la disquette d'accompagnement devront créer le fichier SIX LETTRES.TEXT, sous forme de fichier à accès direct de longueur 7, comportant

en enregistrement 0 le nombre de mots (999), en enregistrement 1 la position de l'index (normalement 999) suivi de 999 enregistrements de mots de six lettres suivis d'un retour-chariot.

Sur un Apple II+ sans ROM LC

Modification à apporter :

- 1) SIX LETTRES :
ajouter "100 CALL 768";
- 2) SIX LETTRES.JEU :
ajouter "590 CALL 768";
- 3) SIX LETTRES.MODIFIER :
enlever le RETURN de la ligne 590, et ajouter "5530 CALL 768: RETURN";
- 4) pour commencer le jeu :
BLOAD SHIFT (voir dans Pom's N° 14), puis RUN SIX LETTRES.



Fichier 'SIX LETTRES.TEXT' (voir texte)

0	999	13	SABBAT
1	664	14	CORAIL
2	AMICAL		...
3	PETIOT	992	ELIXIR
4	BRAYER	993	DECLIC
5	ABREGE	994	EXAMEN
6	PROCES	995	PINSON
7	INFLUX	996	CUISSE
8	MOITIE	997	ALIENE
9	BEGAYE	998	SEIGLE
10	THORAX	999	MANUEL
11	SURVOL	1000	YOUYOU
12	ASSAUT	1001	RECOIN

Routine Shift

Cette routine, extraite du Pom's n°14, permet aux lecteurs qui n'ont pas les minuscules à l'écran d'utiliser ce programme sans problème. Après saisie sous moniteur, vous la sauvegarderez par :

BSAVE SHIFT, A\$300, LS4B

```
0300- A9 0F 85 36 A9 03 85 37
0308- A9 00 85 48 4C EA 03 8D
0310- 4A 03 08 8A 48 AD 4A 03
0318- A2 07 DD 3A 03 F0 0C CA
0320- 10 F8 C9 E0 90 08 38 E9
0328- 20 D0 03 BD 42 03 8D 4A
0330- 03 68 AA 28 AD 4A 03 4C
0338- F0 FD C0 DC E0 FB FC FD
0340- FE FF C1 C3 A1 C5 D5 C5
0348- AD A1 00
```

Programme SIX LETTRES

```
10 REM le jeu desSIX LETTRES reali
se par Manfred Brede

20 REM six lettres menu principal
version 1, 23.05.85
110 E$ = CHR$(4) + "RUN SIX LETTRE
S."
120 A$ = "-----":A$ = A$ + A$:A$ = A
$ + A$:A$ = A$ + A$
130 POKE 32,0: POKE 33,40: POKE 34,0
: POKE 35,24: HOME
200 HTAB 9: INVERSE : PRINT " S I X
L E T T R E S ": NORMAL
210 VTAB 4: HTAB 14: PRINT "menu pri
ncipal": PRINT A$
220 VTAB 10: HTAB 8: PRINT "J joue
r aux six lettres"
230 VTAB 12: HTAB 8: PRINT "R regl
es du jeu"
240 VTAB 14: HTAB 8: PRINT "M modi
fier le fichier"
250 PRINT : PRINT : HTAB 7: PRINT "E
sc fin"
260 VTAB 23: PRINT A$;: VTAB 24: HTA
B 1: PRINT "appuyez sur la touch
e J, R, M ou Esc: ";
300 GET A$: IF A$ = CHR$(27) THEN
HOME : END
310 IF A$ = "J" OR A$ = "j" THEN E$
= E$ + "JEU": GOTO 350
320 IF A$ = "R" OR A$ = "r" THEN E$
= E$ + "REGLES": GOTO 350
330 IF A$ = "M" OR A$ = "m" THEN E$
= E$ + "MODIFIER": GOTO 350
340 GOTO 300
350 PRINT A$;: VTAB 1: HTAB 1: PRINT
: PRINT E$
```

Programme SIX LETTRES. REGLES

```
100 REM regles du jeu des six lettre
s version 1, 28.05.85
110 A$ = "-----":A$ = A$ + A$:A$ = A
$ + A$:A$ = A$ + A$
120 POKE 32,0: POKE 33,40: POKE 34,0
: POKE 35,24: HOME
130 HTAB 9: INVERSE : PRINT " S I X
L E T T R E S ": NORMAL : VTA
B 23: PRINT A$;: VTAB 24: HTAB 1
0: PRINT "appuyez sur une touche
";
140 POKE 34,2: POKE 35,21: HOME
1000 REM page 1
1010 T$ = "Regles du jeu des 'six le
ttres'": GOSUB 7000
1020 PRINT "Le but du jeu est de tro
uver un mot de": PRINT
1030 PRINT "six lettres : verbe, nom
commun, adjec-": PRINT
1040 PRINT "tif, participe etc.": PR
INT : PRINT
1050 PRINT "Les noms propres, les no
ms mis au femi-": PRINT
1060 PRINT "nin ou au pluriel et les
verbes conju-": PRINT
1070 PRINT "ques ne figurent pas da
ns les mots a": PRINT
1080 PRINT "trouver."
1090 GOSUB 8000
1500 REM page 2
1510 T$ = "Mecanisme": GOSUB 7000
1520 PRINT "Au debut du jeu, le prog
ramme choisit": PRINT
1530 PRINT "un mot que le joueur doi
t trouver. Ce": PRINT
1540 PRINT "dernier ecrit un mot de
six lettres au": PRINT
1550 PRINT "hasard.": PRINT : PRINT
1560 PRINT "Le programme affiche le
nombre de let-": PRINT
1570 PRINT "tres qui sont placees a
la bonne place": PRINT
1580 PRINT "dans le mot a trouver.":
1590 GOSUB 8000
2000 REM page 3
2010 T$ = "Exemple": GOSUB 7000: PRI
NT
2020 PRINT "mot a trouver : "; INVE
RSE : PRINT "CIGALE": NORMAL : P
RINT : PRINT A$
2030 PRINT "mot enonce : "; INVE
RSE : PRINT "EGALES";: NORMAL :
HTAB 30: PRINT "reponse : 0"
2040 PRINT "Aucune lettre n'occupe l
a meme place.": PRINT
2050 PRINT A$
2060 PRINT "mot enonce : "; INVE
RSE : PRINT "GITANE";: NORMAL :
HTAB 30: PRINT "reponse : 3"
2070 PRINT "Les lettres I, A et E oc
cupent la meme place que dans l
e mot CIGALE.":
2080 GOSUB 8000
2500 REM page 4
2510 T$ = "Nota": GOSUB 7000
2520 PRINT "La citation de mots par
le joueur admet": PRINT
2530 PRINT "toutes les formes gramma
tiques :": PRINT : PRINT
2540 PRINT "pluriel, feminin, verbe
conjugue etc.": PRINT
2550 GOSUB 8000
3000 REM page 5
3010 T$ = "Conduite du jeu": GOSUB 7
000
3020 PRINT "Ayant obtenu la reponse
a un mot enonce,"
3030 PRINT "le joueur l'exploite en
eliminant les": PRINT
3040 PRINT "lettres rejetees.": PRIN
T : PRINT
```

```

3050 PRINT "Ainsi un zero permet d'e
liminer defini-": PRINT
3060 PRINT "tivement toutes les lett
res du mot.": PRINT
3070 GOSUB 8000
3500 REM page 6
3510 PRINT : PRINT "Une reponse posi
tive guide le joueur": PRINT
3520 PRINT "dans le choix des mots s
uivants, selon": PRINT
3530 PRINT "les lettres qu'il suppos
e bonnes et que": PRINT
3540 PRINT "d'autres mots lui permet
tront de con-": PRINT
3550 PRINT "firmer ou d'exclure.": P
RINT : PRINT
3560 PRINT "Conclusion:": PRINT
3570 PRINT "La performance reside d
ans l'enonce": PRINT
3580 PRINT "d'un minimum de mots pou
r decouvrir le": PRINT
3590 PRINT "mot a trouver.":
3600 GOSUB 8000
4000 REM page 7
4010 T$ = "Bloc-notes": GOSUB 7000
4020 PRINT "Pour se rappeler les let
tres deja trou-": PRINT
4030 PRINT "vees, le joueur a la pos
sibilite de les": PRINT
4040 PRINT "noter en bas de l'ecran
a gauche.": PRINT : PRINT
4050 PRINT "L'accès au bloc-notes et
le retour au": PRINT
4060 PRINT "jeu se font au moyen de
la touche Esc."
4070 GOSUB 8000
4500 REM page 8
4510 T$ = "Touches permises": GOSUB
7000
4520 PRINT "A ... Z : ecrire une le
ttre": PRINT
4530 PRINT "-> : positionner s
ur la lettre": PRINT "
suivante": PRINT
4540 PRINT "<- : positionner s
ur la lettre": PRINT "
precedante": PRINT
4550 PRINT "->| : positionner s
ur la premiere": PRINT "
lettre du mot": PRINT 4560 PR
INT "Return : enoncer un mot":
PRINT
4570 GOSUB 8000
5000 REM page 9
5010 PRINT
5020 PRINT "Ctrl-A : afficher un a
nagramme du": PRINT "
mot a trouver": PRINT
5030 PRINT "Ctrl-L : afficher une
lettre du": PRINT " mo
t a trouver": PRINT
5040 PRINT "Ctrl-T : verification
des lettres": PRINT "
du bloc-notes": PRINT

```

```

5050 PRINT "Ctrl-F : fin du jeu":
PRINT
5060 PRINT "Esc : acces au bloc
-notes:": PRINT " reto
ur au jeu": PRINT
5070 PRINT "Del ou . : effacer une l
ettre du bloc-": PRINT "
note":
5080 GOSUB 8000
5500 REM page 10
5510 T$ = "Remarques": GOSUB 7000
5520 PRINT "Le nombre total des mots
de six lettres": PRINT
5530 PRINT "est affiche en haut a ga
uche de l'ecran.": PRINT
5540 PRINT "Le nombre en bas a droi
te de l'ecran": PRINT
5550 PRINT "vous informe sur le nomb
re des mots qui": PRINT
5560 PRINT "restent a trouver."
6000 REM fin
6010 VTAB 21: HTAB 1: PRINT "f i n
d e s r e q l e s":
6020 GOSUB 8000
6030 POKE 34,0: POKE 35,24: HOME
6040 PRINT CHR$(4);"RUN SIX LETTRE
S"
7000 REM afficher un titre
7010 T1% = LEN (T$):T2% = 1 + (40 -
T1%) / 2: VTAB 3: HTAB T2%: PRI
NT T$: HTAB T2%: PRINT LEFT$(A
$,T1%): PRINT : RETURN
8000 REM appuyer sur une touche
8010 POKE 49168,0: VTAB 21: HTAB 40:
GET B$: PRINT : HOME : RETURN

```

Programme SIX LETTRES.MODIFIER

```

100 REM modifications au fichier des
mots version 1, 23.5.85
120 E$ = "SIX LETTRES.TEXT":D$ = CH
R$(4):K$ = CHR$(7)
140 ER$ = D$ + "READ" + E$ + ",R":EW
$ = D$ + "WRITE" + E$ + ",R"
160 DIM B$(6),S$(6)
180 A$ = "-----":A$ = A$ + A$:A$ = A
$ + A$:A$ = A$ + A$
200 S$(0) = A$:S$(1) = "1 ajouter
un mot":S$(2) = "2 changer un
mot":S$(3) = "3 eliminer un mot"
:S$(4) = "4 verifier un mot":S
$(5) = "5 liste des mots":S$(6)
= "6 modifier l'index"
240 PRINT D$;"UNLOCK";E$: PRINT D$;"
OPEN";E$;"L7": PRINT ER$;"0": I
NPUT MAX%: PRINT ER$;"1": INPUT
N%

```

```

260 OUI$ = K$ + "le mot existe deja
    dans le fichier. ";NO$ = K$ + "l
    e mot n'existe pas dans le fichi
    er. "
300 POKE 32,0: POKE 33,40: POKE 34,0
    : POKE 35,24: GOSUB 5500: HOME
320 HTAB 9: INVERSE: PRINT " S I X
    L E T T R E S ": NORMAL
340 VTAB 4: PRINT "modifications"; S
    PC( 11);"<Esc pour finir>": VTAB
    5: PRINT S$(0): GOSUB 8000
360 VTAB 23: HTAB 1: PRINT S$(0);: P
    OKE 34,9: POKE 35,22
500 REM menu
520 S% = 0: HOME : FOR I = 1 TO 6: H
    TAB 12: PRINT S$(I): PRINT : NEX
    T
540 VTAB 24: PRINT "appuyez sur la t
    ouche 1 a 6 ou Esc ";: CALL - 9
    58
560 GOSUB 5000: IF A$ = CHR$( 27) T
    HEN 800
580 IF A$ < "1" OR A$ > "6" THEN 560
600 VTAB 24: HTAB 1: PRINT S$( VAL (
    A$));: CALL - 958: HOME
620 ON VAL (A$) GOSUB 1000,1500,200
    0,2500,3000,3500
640 GOTO 500
800 REM fin
820 GOSUB 5600: PRINT D$;"CLOSE";E$:
    PRINT D$;"LOCK";E$
840 POKE 34,0: POKE 35,24: HOME
900 PRINT D$;"RUN SIX LETTRES"
910 END
1000 REM ajouter
1020 S% = S% + 1
1040 VTAB 12: HTAB 1: PRINT "ecrivez
    le mot a ajouter ";: GOSUB 600
    0
1060 GOSUB 7000: REM chercher
1080 IF E% THEN GOSUB 5500: VTAB 15
    : HTAB 1: PRINT OUI$;: GOTO 1140
1100 MAX% = MAX% + 1: PRINT EW$;MAX%
    + 2: PRINT A$: PRINT EW$;"0": P
    RINT MAX%
1120 GOSUB 5500: GOSUB 8000: VTAB 15
    : HTAB 1: PRINT K$;"le mot a ete
    ajoute. ";
1140 GOSUB 5000: GOTO 1040
1500 REM changer
1520 S% = S% + 1
1540 VTAB 12: HTAB 1: PRINT "ecrivez
    le mot a changer ";: GOSUB 600
    0
1560 GOSUB 7000: REM chercher
1580 GOSUB 5500: IF E% = 0 THEN VTA
    B 15: HTAB 1: PRINT NO$;: GOTO 1
    740
1600 RC% = R$:AC$ = A$
1620 VTAB 14: HTAB 1: PRINT K$;"chan
    ger contre quel mot ";: GOSUB
    6000
1640 IF A$ = AC$ THEN 1720
1660 GOSUB 7000: REM chercher
1680 IF E% THEN GOSUB 5500: VTAB 17
    : HTAB 1: PRINT OUI$;: GOSUB 500
    0: GOTO 1620
1700 PRINT EW$;RC%: PRINT A$: GOSUB
    5500
1720 VTAB 17: HTAB 1: PRINT K$;"le m
    ot a ete change. ";
1740 GOSUB 5000: GOTO 1540
2000 REM eliminer
2020 S% = S% + 1
2040 VTAB 12: HTAB 1: IF MAX% < 2 TH
    EN CALL - 958: PRINT "impossib
    le, il y a moins de 3 mots": PRI
    NT : PRINT "dans le fichier ";:
    GOTO 2160
2060 PRINT "ecrivez le mot a elimine
    r ";: GOSUB 6000: GOSUB 7000: R
    EM chercher
2080 GOSUB 5500: IF E% = 0 THEN VTA
    B 15: HTAB 1: PRINT NO$;: GOSUB
    5000: GOTO 2040
2100 M% = MAX% + 2: PRINT ER$;M%: IN
    PUT AI$: PRINT EW$;M%: PRINT A$:
    PRINT EW$;R%: PRINT AI$
2120 MAX% = MAX% - 1: PRINT EW$;"0":
    PRINT MAX%: IF N% > MAX% THEN N
    % = MAX%: PRINT EW$;"1": PRINT N
    %
2140 GOSUB 5500: GOSUB 8000: VTAB 15
    : HTAB 1: PRINT K$;"le mot a ete
    elimine. ";
2160 GOSUB 5000: GOTO 2040
2500 REM verifier
2520 S% = S% + 1
2540 VTAB 12: HTAB 1: PRINT "ecrivez
    le mot a verifier ";: GOSUB 60
    00
2560 GOSUB 7000: REM chercher
2580 GOSUB 5500: VTAB 15: HTAB 1: IF
    E% THEN PRINT OUI$;: GOTO 2620
    : REM trouve
2600 PRINT NO$;: REM pas trouve
2620 GOSUB 5000: GOTO 2540
3000 REM liste
3020 S% = S% + 1
3040 M% = MAX% + 2:C% = 3:R% = 1:P%
    = (MAX% + 1) / 50:P% = (P% + 9)
    / 10:P% = P% * 50:PF% = MAX% / 5
    0:PF% = PF% * 50: PRINT
3060 I% = 1:I = 2
3080 IF C% < 33 THEN 3120
3100 C% = 3:R% = R% + 1: IF R% > 10
    THEN GOSUB 3320
3120 PRINT ER$;I: INPUT AI$: VTAB 9
    + R%: HTAB C%: PRINT AI$:C% = C%
    + 7:I = I + 1: IF I < = M% THE
    N 3080
3140 C% = 3: GOSUB 3320: IF I < = M
    % THEN 3120
3160 GOTO 3060
3300 REM nouvelle page
3320 S% = S% + 1
3340 GOSUB 5600: VTAB 21: HTAB 3: PR
    INT "acces rapide par les touche

```

```

s 0 a 9";: VTAB 24: HTAB 20: PRI
NT "(";I%;"-";I - 2;")";K$;:I% =
I - 1
3360 VTAB 24: HTAB 36: GOSUB 5000: I
F A$ < "0" OR A$ > "9" THEN 3420
3380 I = VAL (A$) * P%: IF I > PF%
THEN I = PF%
3400 I = I + 2:I% = I - 1
3420 VTAB 24: HTAB 20: CALL - 958:
HOME : PRINT :R% = 1
3440 S% = S% - 1: RETURN
3500 REM l'index
3520 S% = S% + 1
3540 VTAB 12: HTAB 1: PRINT "l'index
actuel : ";N% + 1;: CALL - 958
3560 VTAB 14: HTAB 1: PRINT "nouvel
index : ";
3580 N$ = "0":I% = LEN ( STR$ (MAX%
+ 1)):I = 1
3600 GOSUB 5000: IF A$ = CHR$ (13)
THEN 3900: REM Return
3620 IF A$ < "0" OR "9" < A$ THEN 36
00
3640 PRINT A$;:N$ = N$ + A$:I = I +
1: IF I < = I% THEN 3600
3900 I = VAL (N$): IF I < 2 OR MAX%
+ 1 < I THEN VTAB 16: HTAB 1:
PRINT K$;"l'index doit etre entr
e 2 et ";MAX% + 1;" ";: GOSUB 50
00: GOTO 3540
3920 N% = I - 1: PRINT : PRINT EW$;"
1": PRINT N%: GOSUB 5520: PRINT
K$: GOTO 3540
5000 REM appuyer sur une touche
5020 S% = S% + 1: POKE 49168,0: GET
A$: IF A$ < > CHR$ (27) THEN 5
060: REM ESC
5040 IF S% > 1 THEN POP :S% = S% -
1: GOTO 5040
5060 S% = S% - 1: RETURN
5500 REM clavier,ecran
5520 PRINT D$;"IN#0": PRINT D$;"PR#0
": RETURN
5600 VTAB 1: HTAB 1: PRINT : GOTO 55
20
6000 REM entrer un mot (sousprogramm
e)
6020 S% = S% + 1
6040 CALL - 958:I% = 1:C% = 1 + PE
EK (36): REM position horizontal
e du curseur
6060 I = 1
6080 HTAB C% + I: GOSUB 5000
6100 IF I% < 7 THEN 6180
6120 IF A$ = CHR$ (13) THEN 6500
6140 IF I < 7 THEN 6180
6160 GOTO 6280
6180 IF A$ > "Z" THEN A$ = CHR$ ( A
SC (A$) - 32)
6200 IF A$ > "Z" THEN 6080
6220 IF A$ < "A" THEN 6280
6240 B$(I) = A$:I = I + 1: IF I% < I
THEN I% = I: REM lettre
6260 GOTO 6440

```

```

6280 IF A$ < > CHR$ (8) THEN 6340:
REM <-
6300 IF I > 1 THEN I = I - 1: GOTO 6
440
6320 GOTO 6080
6340 IF A$ < > CHR$ (21) THEN 6400
: REM ->
6360 IF I < I% THEN I = I + 1: GOTO
6440
6380 GOTO 6080
6400 IF A$ < > CHR$ (9) THEN 6080:
REM ->|
6420 GOTO 6060
6440 PRINT A$;: GOTO 6080
6500 A$ = B$(1) + B$(2) + B$(3) + B$
(4) + B$(5) + B$(6): HTAB C% + 7
: PRINT :S% = S% - 1: RETURN
7000 REM chercher un mot (sousprogra
mme)
7020 E% = 0:M% = MAX% + 2:R% = 2
7040 PRINT ER$;R%: INPUT AI$: IF A$
< > AI$ THEN R% = R% + 1: IF R%
< = M% THEN 7040
7060 IF R% < = M% THEN E% = 1
7080 GOSUB 5500: RETURN
8000 REM nombre
8020 VTAB 7: PRINT "nombre de mots :
";MAX% + 1;" ";: RETURN

```

Programme SIX LETTRES.JEU

```

100 REM six lettres versi
on 3, 26.05.85
110 DIM A$(6),B$(6),T$(6):I = 0:A$(0
) = "":B$(0) = "":T$(0) = ""
120 C$ = ".....":C$ = C$ + " " +
C$:C$ = C$ + " " + C$
130 E$ = "SIX LETTRES.TEXT":D$ = CH
R$ (4):ER$ = D$ + "READ" + E$ +
",R":EW$ = D$ + "WRITE" + E$ +
",R"
140 PRINT D$;"UNLOCK";E$: PRINT D$;"
OPEN";E$;"L7": PRINT ER$;"0": I
NPUT MAX%: PRINT ER$;"1": INPUT
N%
150 POKE 32,0: POKE 33,40: POKE 34,0
: POKE 35,24: HOME : PRINT MAX%
+ 1;: HTAB 8: INVERSE : PRINT "
S I X L E T T R E S ": NORMA
L : POKE 34,2
500 REM choisir un mot
510 X% = RND (1) * N%
520 PRINT ER$;X% + 2: INPUT A$(0): P
RINT ER$;N% + 2: INPUT A$(1): PR
INT EW$;N% + 2: PRINT A$(0): PRI
NT EW$;X% + 2: PRINT A$(1)
530 N% = N% - 1: IF N% < 1 THEN N% =
MAX%
540 PRINT EW$;"1": PRINT N%: PRINT D
$;"IN#0": PRINT D$;"PR#0"

```

```

550 FOR I = 1 TO 6:A$(I) = MID$(A$(
    (0),I,1): NEXT
560 HOME : VTAB 3: FOR I = 1 TO 10:
    PRINT : PRINT C$: NEXT : VTAB 24
    : INVERSE : PRINT LEFT$(C$,6);
    : NORMAL
570 A$ = STR$(N% + 1): HTAB 39 -
    LEN(A$): PRINT A$;
580 FOR I = 1 TO 6:T$(I) = ".": NEXT
    : REM effacer le bloc-notes
1000 REM ecrire un mot
1010 H% = 0
1020 V% = 4
1030 FOR I = 1 TO 6:B$(I) = "": NEXT
    :L% = 1:I% = 1
1040 I = 1
1050 VTAB V%: HTAB H% + I: POKE 4916
    8,0: GET A$
1060 IF I% < 7 THEN 1100
1070 IF L% = 1 AND A$ = CHR$(13) T
    HEN 1500: REM Return
1080 IF I < 7 THEN 1100
1090 GOTO 1170
1100 IF A$ > "Z" THEN A$ = CHR$(A
    SC(A$) - 32)
1110 IF A$ < "A" OR A$ > "Z" THEN 11
    70
1120 ON L% GOTO 1130,1140: REM lettr
    e
1130 B$(I) = A$: GOTO 1150
1140 T$(I) = A$
1150 I = I + 1: IF I% < I THEN I% =
    I
1160 GOTO 1450
1170 IF A$ < > CHR$(8) THEN 1200:
    REM <-
1180 IF I > 1 THEN I = I - 1: GOTO 1
    450
1190 GOTO 1050
1200 IF A$ < > CHR$(21) THEN 1230
    : REM ->
1210 IF I < I% THEN I = I + 1: GOTO
    1450
1220 GOTO 1050
1230 IF A$ = CHR$(9) THEN 1040: RE
    M ->|
1240 IF A$ < > CHR$(27) THEN 1290
    : REM Esc
1250 ON L% GOTO 1260,1280
1260 LV% = V%: LH% = H%: LI% = I%: LI =
    I:L% = 2: REM debut bloc-notes
1270 V% = 24:H% = 0:I% = 7: INVERSE
    : GOTO 1040
1280 V% = LV%: H% = LH%: I% = LI%: I =
    LI:L% = 1: NORMAL : GOTO 1050: R
    EM fin de notes
1290 ON L% GOTO 1300,1410
1300 IF A$ = CHR$(6) THEN 1550: RE
    M Ctrl-F fin
1310 IF A$ < > CHR$(1) THEN 1350:
    REM Ctrl-A anagramme
1320 FOR I = 1 TO 6:B$(I) = A$(I): N
    EXT :LI% = 6 + 100 * RND(1)
1330 FOR I = 1 TO LI%:X% = 1 + 6 *
    RND(1):I% = 1 + 6 * RND(1):A$
    = B$(X%):B$(X%) = B$(I%):B$(I%)
    = A$: NEXT
1340 HTAB H% + 1: INVERSE : FOR I =
    1 TO 6: PRINT B$(I);: NEXT : NOR
    MAL : GOTO 1530
1350 IF A$ < > CHR$(12) THEN 1370
    : REM Ctrl-L donner une lettre
1360 HTAB H% + 1: PRINT LEFT$(C$,6
    );:I = 1 + INT(RND(1) * 6):
    HTAB H% + I: PRINT A$(I);: GOTO
    1530
1370 IF A$ < > CHR$(20) THEN 1050
    : REM Ctrl-T test
1380 FOR I = 1 TO 6: IF T$(I) < > "
    ." THEN IF T$(I) < > A$(I) THE
    N A$ = "FAUX": GOTO 1400
1390 NEXT :A$ = "EXACT"
1400 HTAB H% + 1: PRINT LEFT$(C$,6
    );: HTAB H% + 1: INVERSE : PRINT
    A$;: NORMAL : GOTO 1530
1410 IF I > 6 THEN 1430
1420 IF A$ < > "." AND A$ < > CHR
    $(95) THEN 1050: REM . ou Del-3
    2
1430 IF I < I% THEN A$ = ".":T$(I) =
    ".":I = I + 1: GOTO 1450
1440 GOTO 1050
1450 PRINT A$;: GOTO 1050
1500 REM resultat
1510 I% = 0: FOR I = 1 TO 6: IF A$(I
    ) = B$(I) THEN I% = I% + 1
1520 NEXT : VTAB V%: HTAB H% + 8: PR
    INT I%: IF I% = 6 THEN 1560
1530 V% = V% + 2: IF V% < 23 THEN 10
    30
1540 H% = H% + 10: IF H% < 32 THEN 1
    020
1550 VTAB 24: HTAB 1: INVERSE : PRIN
    T "*":: FOR I = 1 TO 6: PRINT A$
    (I);: NEXT : PRINT "*":: NORMAL
    : GOTO 1600
1560 VTAB 24: HTAB 1: INVERSE : PRIN
    T "*GAGNE*": NORMAL
1600 PRINT " un autre mot ? (O/N)
    ";
1610 POKE 49168,0: GET A$
1620 IF (A$ = "O") OR (A$ = "o") THE
    N PRINT A$;: GOTO 510
1630 IF (A$ = "N") OR (A$ = "n") THE
    N PRINT A$;: GOTO 2000
1640 GOTO 1610
2000 REM fin du jeu
2010 PRINT D$;"LOCK";E$: POKE 34,0:
    HOME
2020 PRINT D$;"RUN SIX LETTRES"

```

ILLI-PAYEUR-PANZER-000000 -
 APHE-GOUPIL-OFFSET-OLIENE-INGENU-CYPRES-G
 IGER-INHALE-GUIGNE-JAGUAR-CHRONO-GADGET-S
 ESTA-ABJECT-CONDOR-GRISOUS-PATHOS-DIFFUS-V
 INONE-NOTION-ISTHME-VEXANT-LEADER-MOELLE-J
 TAITUE-BAMBIN-MIASME-PROFIL-DIURNE-F
 QUINZE-UTERUS-I

Des bugs sous ProDOS

Alexandre Avrane

ProDOS est disponible depuis seulement deux ans, et pourtant, on ne dénombre pas moins d'une demi-douzaine de versions différentes. Cependant, la lecture des manuels fournis par Apple ne donne pas d'indications sur les différences entre ces diverses versions. Tout au plus peut-on supposer qu'il s'agit d'améliorations sur des points de détail, des mises à niveau de quelques fonctions annoncées dans la documentation, ou de corrections de quelques bugs...



Apple assure (théoriquement) la compatibilité d'un programme tournant sous une version antérieure de ProDOS. Cependant, il n'est pas inutile de connaître exactement les caractéristiques de la version que l'on utilise, ne serait-ce que pour utiliser pleinement les performances du système.

Il faut également distinguer entre les versions du noyau (kernel) de ProDOS (fichier ProDOS d'une disquette) et de l'interpréteur BASIC.SYSTEM.

La version du ProDOS Kernel s'affiche en général pendant le boot du système. En revanche, celle du BASIC.SYSTEM est plus difficile à évaluer. Pour

l'obtenir il faut l'exécuter sans aucun fichier 'STARTUP' sur la disquette. De plus, deux octets précisent ces numéros : KVERSION en \$BFFF pour le noyau, IVERSION en \$BFFD pour l'interpréteur.

Les versions que j'ai pu recenser sont :

	Date	Octet	identificateur

ProDOS :			
B.4	04/02/83	Kversion=	0
1.0	01/07/83	Kversion=	0
1.0.1	01/01/84	Kversion=	0
1.0.2	15/02/84	Kversion=	0
1.1	17/08/84	Kversion=	1
1.1.1	18/09/84	Kversion=	1

Basic.System :			
1.0	15/11/83	Iversion=	0
1.1	18/06/84	Iversion=	1

La version B.4 de ProDOS n'était qu'une version d'évaluation. D'autre part, nous ne nous occupons pas ici des autres interpréteurs système fournis par Apple, tels que FILER, CONVERT, EDASM, etc.

Commandes sous Applesoft

- Les instructions MON et NOMON, issues du DOS 3.3, étaient autorisées (pour assurer la compatibilité DOS 3.3) mais néanmoins ignorées. Depuis la version 1.1.1, la commande MON engendre bizarrement un SYNTAX ERROR.

- La commande FRE est enfin apparue avec la 1.1.1 ; pour rappcl, elle nettoie la mémoire vive beaucoup plus rapidement que son homologue de l'Applesoft.

- Une nouvelle commande, BYE, déconnecte le BASIC.SYSTEM (version 1.1) et permet d'enchaîner vers un autre interpréteur (par exemple FILER ou CONVERT).

- Les commandes CAT et CATALOG ont été partiellement francisées (patch par Apple Seedrin ?) et fournissent la mention "SANS DATE" au lieu de "NO DATE" lors de l'affichage des catalogues. En revanche, les mentions "BLOCKS FREE" et "BLOCKS USED" sont toujours données en anglais... J'ai découvert également (ce n'était pas indiqué dans le manuel) que Ctrl-C permet d'arrêter l'affichage d'un catalogue.

- La commande VERIFY qui, à la différence du DOS 3.3, se contente de vérifier l'existence d'un fichier, donne le copyright d'Apple lorsqu'elle est appelée sans paramètres.

- Les paramètres "@" (numéro de ligne Applesoft) et "V" (numéro de volume, ignorés mais néanmoins autorisés pour la compatibilité DOS 3.3) posent problème lorsqu'ils sont utilisés simultanément. Editez le programme suivant:

```
10 PRINT "10"  
20 PRINT "20"  
30 PRINT "30"
```

et sauvez-le sous le nom BUG1. Maintenant tapez la commande :

```
RUN BUG1,@20,V30
```

et l'exécution commence à la ligne 30 au lieu de la ligne 20. Il faudrait taper :

```
RUN BUG1,V30,@20
```

Ce bug, heureusement sans grand danger, n'est toujours pas corrigé.

• Un bug éphémère est apparu avec la version 1.0.2 de ProDOS lors de l'utilisation de la commande CHAIN. Dans un cas très particulier (une fois sur 256 en moyenne), les variables étaient mal transmises entre deux programmes Applesoft. Evitez donc d'utiliser la version 1.0.2. si vous appelez CHAIN.

Gestion de la mémoire vive

Hormis la commande FRE vue précédemment, les instructions HGR et HGR2 de l'Applesoft doivent entraîner, lors de leur exécution sous Basic.System, le transfert en mémoire haute de la partie du programme Applesoft qui serait écrasée par l'utilisation de la page graphique haute-résolution. L'instruction TEXT doit transférer le programme à son adresse initiale. Ceci n'est encore que théorique dans le manuel Apple, car aucune des versions de Basic.System ne supporte cette possibilité.

• Les vecteurs de la page 3 (\$3D0-\$3FF) sont différemment initialisés selon les versions de ProDOS :

\$3D0- JMP SBE00	démarrage à chaud de BASIC.SYSTEM
\$3D3- JMP SBE00	idem pour compatibilité avec DOS 3.3
\$3D6 à \$3EC	octets réservés pour les interpréteurs tournant s/s ProDOS
\$3ED- DA \$xxxx	adresse de transfert avec \$C314 (Apple IIe et IIc seulement)
\$3EF- JMP SFA59	gestion du BRK du 6502/65C02
\$3F2- DA \$FF59	ProDOS 1.0 ne gère pas le RESET et renvoie au moniteur
ou DA SBE00	ProDOS 1.1 récupère le RESET et fait un démarrage à chaud
\$3F4- DFB \$5A	octet de contrôle du RESET (ProDOS 1.0)
ou DFB \$1B	idem (ProDOS 1.1)
\$3F5- JMP SFF59	ProDOS 1.0 ne gère pas l'instruction &
ou JMP SBE03	ProDOS 1.1 envoie & vers le point d'entrée externe
\$3F8- JMP SBE00	vers ProDOS

\$3FR- JMP SFF59 vers le démarrage à froid du moniteur si interruption NMI
 \$3FE DA SBFED gestion des interruptions IRQ par ProDOS

Pages globales de ProDOS

Les routines GETBUFR et FREEBUFR, dont les points d'entrée respectifs sont \$BEF5 et \$BEF8 et qui permettent d'allouer et libérer de l'espace mémoire entre ProDOS et ses buffers (par exemple pour un programme en assembleur), sont implémentées depuis ProDOS 1.1.1 ; il n'est donc plus nécessaire de les simuler (comme dans la routine CMDLOAD de Pom's 20).



Pour vérifier leur existence, passez sous le moniteur et tapez BEF5L. Si vous obtenez :

```
$BEF5- SEC
        RTS
        NOP
$BEF8- SEC
        RTS
        NOP
```

alors ces deux routines ne sont pas implantées.

Gestion des interruptions

Les interruptions masquables (IRQ) étaient mal gérées avant ProDOS 1.1.1 ; il reste néanmoins un bug avec cette nouvelle version, car le MLI de ProDOS n'est pas réentrant : si une interruption intervient alors qu'un programme principal a appelé le MLI pour exécuter une commande, et que la routine qui

gère cette interruption appelle également le MLI, alors certains octets de données gérés par le MLI auront pu être écrasés lorsque le programme principal reprendra le contrôle. Il est possible de régler cette situation conflictuelle, espérons néanmoins qu'une version future de ProDOS éliminera ce bug.

En aparté, il faut signaler que les interruptions (causées par exemple par une souris) sont très mal gérées sur un IIe lorsque la carte 80 colonnes est active. Le kit de mise à niveau remédie à ce problème (techniquement la valeur de l'accumulateur est désormais stockée sur la pile au lieu d'être placée à l'adresse \$45).

Lecture du catalogue

Une des contraintes que nous infligait le DOS 3.3 était la difficulté à accéder au catalogue d'une disquette. Le Basic.-System, pour sa part, permet de lire un catalogue sans avoir à manier les PEEK, POKE et autres CALL. Le petit programme ci-dessous permet d'obtenir un tableau A\$ où chaque variable est formatée comme une ligne affichée par la commande CATALOG sur le préfixe courant :

```
100 REM Lecture1
110 D$ = CHR$(4)
120 PRINT D$"PREFIX" : INP
    UT P$ : REM obtient le
    préfixe
130 DIM A$(200) : REM prév
    oyons 200 fichiers
140 PRINT D$"OPEN"P$,TDIR
    " : REM on ouvre le di
    rectory
150 PRINT D$"READ"P$
160 ONERR GOTO 200 : REM o
    n s'arrêtera en END OF
    DATA
170 FOR I = 0 TO 200
180 INPUT A$(I) : REM lit
    une ligne
190 NEXT
200 PRINT D$"CLOSE"P$
```

Conséquence logique, on peut obtenir dans les variables N1 et N2 le nombre de blocs disponibles et occupés sur un volume par le programme suivant :

```

100 REM Lecture2
110 D$ = CHR$(4)
120 PRINT D$"PREFIX" : INP
    UT P$
130 PRINT D$"OPEN"P$, TDIR
    "
140 PRINT D$"READ"P$
150 FOR I = 0 TO 2 STEP 0
    : REM boucle à vide
160 INPUT A$ : REM lit une
    ligne
170 I = I + (NOT LEN (A$))
    : REM on attend la 2e
    ligne vide
180 NEXT
190 INPUT A$ : REM cette l
    igne contient les vale
    urs recherchées
200 PRINT D$"CLOSE"P$
210 DEF FN N (X) = VAL (MI
    D$ (A$, X, 5))
220 N1 = FN N (16) : REM e
    xtrait le nombre de bl
    ocs libres
230 N2 = FN N (37) : REM e
    xtrait le nombre de bl
    ocs occupés

```

Hélas, tout marche correctement tant que l'on accède à un disque physique. Mais si on veut observer le disque virtuel installé

par ProDOS sur les Apple de 128K (préfixe /RAM), le Basic.System arrête alors la lecture sur le dernier fichier : la dernière ligne du catalogue n'est donc plus accessible et il est nécessaire de lire directement la VTOC du disque virtuel pour en extraire les valeurs d'occupation.

Compilation des programmes Applesoft

Il s'agit d'une petite digression puisqu'aucun compilateur de programmes Applesoft n'est officiellement supporté par Apple. En effet, leur utilisation pose quelques problèmes sous ProDOS :

- Celui-ci ignore les commandes ProDOS qui ont pu être compilées (un PRINT CHR\$(4); "CATALOG" affiche le mot "CATALOG" à l'écran).
- Par ailleurs, si on lance l'exécution par un BRUN du module objet, ProDOS est déconnecté après l'appel ; il est donc préférable de faire un BLOAD suivi d'un CALL.
- Enfin, les mêmes problèmes existent indifféremment avec

les compilateurs 'Tasc' ou 'Expediter'. Un lecteur connaît-il les solutions ?

Trois remarques en guise de conclusion :

- La liste des bugs relevés ci-dessus ne se prétend pas exhaustive. Ils ne sont, de plus, guère dangereux (pas de comparaison avec le célèbre bug de l'APPEND du DOS 3.3 qui vous détruisait un fichier sans même vous en avertir !).
- ProDOS est un excellent système d'exploitation, compte tenu des contraintes techniques qui ont présidé à sa conception ; il est désormais le système officiel de l'Apple II : ProFile ou le nouvel UnoDisk, par exemple, ne peuvent être utilisés sous DOS 3.3.



- Il est préférable de toujours utiliser la dernière version disponible de ProDOS car, d'une part les bugs s'y font plus rares et, d'autre part des commandes nouvelles ou améliorées peuvent y être implantées.



Max : le moniteur étendu

Jacques Supernant

Apple II+, IIe, IIc

Ce moniteur autorise un contrôle de l'exécution des routines en langage machine

Un mode Trace et Pas à Pas très évolués et sélectifs sont complétés par un accès direct aux registres du 6502 (ou 65C02).

La gestion des fenêtres d'écran simplifie le mode trace. Une routine permet la recherche de suites d'octets.

Un mini-assembleur très souple fait partie de Max. Une ligne de commande peut devenir une boucle avec l'ordre JUMP

Les fichiers source sont sur la disquette.

Disquette et documentation : 150,00 F TTC franco. Bon de commande page74.

Un 'Print Using' en assembleur

Patrice Neveu

a tâche de ce programme est de combler une lacune de l'Applesoft, à savoir l'absence de Print Using. Il peut être en effet nécessaire voire indispensable d'aligner des nombres de sorte que les chiffres des unités, dizaines etc., soient les uns au-dessous des autres. Un article de Pom's écrit par Jean-Francois Duvivier nous avait déjà donné une solution grâce à une routine BASIC ; je vous propose aujourd'hui une routine similaire en langage machine. Celle-ci est plus rapide, moins encombrante, et relogeable dans n'importe quelle zone de mémoire utilisable.

Utilisation du programme

Son utilisation est simple. Il suffit de mettre le masque voulu dans la variable M\$, de ranger le réel à afficher dans la variable X, puis enfin de faire Y = USR(X). Cette dernière instruction a pour effet de passer X du format compacté en format 'FAC' (voir pour cela le Pom's numéro 12).

Le masque suit les mêmes règles que dans l'article du Pom's 4 à quelques différences près: Il peut

être composé de '#', de '.', et d'espaces (au début du masque seulement), mais pas de '^' car la notation scientifique n'est plus disponible.

- # définit un chiffre, et donne l'emplacement de la virgule.
- Aucun retour à la ligne ne se produit après un affichage.
- Les zéros en tête sont remplacés par des espaces.
- Des zéros sont ajoutés à la fin pour compléter le masque si besoin est.
- Si le nombre de décimales du réel dépasse celui du masque, le nombre est arrondi par défaut.
- Lorsqu'un signe négatif est utilisé, il prend une position #.
- Si le masque ne répond pas à la syntaxe précédemment définie ou si le nombre ne peut rentrer dedans, le message 'MASK ERROR' suivi du bip habituel, est généré. La main est alors rendue au niveau de l'Applesoft.

Le programme PRINT-USING DEMO illustre tout ceci.

Remarques

Dans sa version d'origine, le programme renvoie dans la

variable Y le nombre affiché. Pour supprimer cela et donc gagner quelques fractions de secondes, mettre un RTS (\$60=96) en \$8126 (33064).

Une autre petite modification peut être opérée : si on ne désire pas que le message d'erreur propre au programme s'affiche, il faut modifier les lignes 121,122,123 de cette façon :

```
121 ERR LDX MSKLEN Remplir le
                        masque d'espaces
122 JSR $F94A PRNBL2affiche
                        X espaces
123 RTS
```

Ce qui se traduit depuis le moniteur par :

806E : A6 D7 20 4A F9 60

Et en basic, il faudra insérer la ligne :

```
175 POKE 32878,166 : POKE 32879,
215 : POKE 32880,32 : POKE 3
2881,74 : POKE 32882,249 : P
OKE 32883,96
```

D'autre part, si l'on frappe A=USR(X), le résultat sera dans A, mais aussi dans Y. La variable X reste inchangée.

Le déroulement du programme est expliqué plus en détail sur le listing.



Récapitulation PRINTUSING

```
8000- A9 00 85 CF 85 CE 85 1D
8008- 20 34 ED 84 07 85 06 A0
8010- 00 B1 06 F0 10 C9 2E D0
8018- 02 84 CF 99 00 03 C8 C0
8020- 10 D0 EE A9 00 99 00 03
8028- 84 EB A5 69 85 08 A5 6A
8030- 85 09 A0 U0 B1 08 C9 4D
8038- D0 18 C8 B1 08 C9 80 D0
8040- 11 C8 B1 08 85 D7 C8 B1
8048- 08 85 18 C8 B1 08 85 19
8050- D0 35 A5 08 18 69 07 85
8058- 08 90 D7 E6 09 A5 6C C5
8060- 09 B0 0B D0 06 A5 6B C5
8068- 08 B0 03 18 90 C4 A0 80
```

```
8070- A9 7B 20 3A DB 20 3A FF
8078- 4C D0 03 8D CD C1 D3 CB
8080- A0 C5 D2 D2 CF D2 00 A4
8088- D7 88 B1 18 C9 20 D0 04
8090- E6 1D D0 0A C9 23 F0 06
8098- C9 2E D0 D2 84 CE 88 10
80A0- E9 AD 00 03 C9 2E F0 08
80A8- A5 CF D0 04 A5 EB 85 CF
80B0- A5 CE D0 04 A5 D7 85 CE
80B8- A5 CE 38 E5 1D C5 CF 90
80C0- AD A4 CE A9 20 99 31 03
80C8- 88 10 FA A5 CF D0 08 A4
80D0- CE 88 A9 30 99 31 03 A0
80D8- 1F A9 30 99 31 03 88 C4
80E0- CE D0 F8 A4 D7 A9 00 99
80E8- 31 03 A5 CF F0 10 A6 CE
80F0- A4 CF 88 CA B9 00 03 9D
```

```
80F8- 31 03 CA 88 10 F6 A6 CE
8100- A4 CF A9 2E 9D 31 03 B9
8108- 00 03 F0 0D 9D 31 03 E8
8110- C8 C4 EB B0 04 E4 D7 90
8118- EE A0 00 B9 31 03 09 80
8120- 20 ED FD C8 C4 D7 90 F3
8128- A5 69 85 08 A5 6A 85 09
8130- A0 00 B1 08 C9 59 D0 05
8138- C8 B1 08 F0 1B A5 08 18
8140- 69 07 85 08 90 EA E6 09
8148- A5 6C C5 09 B0 37 D0 E0
8150- A5 6B C5 08 B0 2F 90 D8
8158- C8 98 18 65 08 85 08 90
8160- 02 E6 09 A5 B8 85 FE A5
8168- B9 85 FF A9 03 85 B9 A9
8170- 30 85 B8 20 4A EC A4 09
8178- A6 08 20 2B EB A5 FE 85
8180- B8 A5 FF 85 B9 60
```

Programme PRINTUSING.S (Assembleur BigMac)

```

1
2
3 *****
4 * PRINTUSING 1.1 *
5 *
6 ** (C) *06/06/85**PATRICE*NEVEU**
7
8
9 STROUT = $DB3A SORTIE DE CHAINE DE CARACTERES
10 MOVWF = $EB2B COMPACTE FAC ET LE MET EN (Y,X)
11 FIN = $EC4A SAISI UN NOMBRE PAR CHRGET > FAC
12 FOUT = $ED34 FAC -> CHAINE DE CARACTERES
13 COUT = $FDED SORTIE DE CARACTERE
14 BELL = $FF3A BIP SONORE
15
16 CHAINE = $0300 RESULTAT DE FOUT
17 RESULT = $0331 NOMBRE PASSE PAR LE MASQUE. A AF
FICHER
18 BASIC = $03D0 RETOUR A L'APPLESOFT
19
20 ADR = $06
21 VARTAB = $0B
22 MASKADR = $10
23 SPACES = $1D
24 LOMEM = $69
25 ARYTAB = $6B
26 TXTPTR = $B8
27 MSKPOINT = $CE
28 CHNPOINT = $CF
29 MASKLEN = $D7
30 CHAINLEN = $EB
31 TXTPTR2 = $FE
32
33
34 *-----
35 * INITIALISE
36 *-----
37
38 LDA #00
39 STA CHNPOINT
40 STA MSKPOINT
41 STA SPACES
42
43 *-----
44 * CONVERTIR FAC EN CHAINE DE
45 * CARACTERE EN $300
46 *-----
47
48 JSR FOUT FAIT LA CONVERSION
49 STA ADR+1 SAUVE L'ADRES DU RESULTAT OBTENU
50 STA ADR
51
52 LDY #0
53 MANT LDA (ADR),Y ET TRANSFERE LE RESULTAT EN $300
54 BEQ MANTIS POUR UTILISATION ULTERIEURE
55 CMP #' C'EST FINI ?
56 BNE MANTI SI C'EST UN POINT, ALORS IL FAUT
57 STY CHNPOINT EN SAUVER LA POSITION...
58 MANTI STA CHAINE,Y VOILA QUI EST FAIT.
59 TNY
60 CPY #$10 PASSE AU PROCHAIN CARACTERE
61 BNE MANT ON EN TRANSFERE 16 MAXIMUM
62 LDA #00
63 MANTIS STA CHAINE,Y MARQUE FIN DE CHAINE PAR UN $00
64 STY CHAINLEN CONSERVE LONGUEUR DE LA CHAINE.
65
66 *-----
67 * VA CHERCHER LE MASQUE STOCKE
68 * DANS LA VARIABLE MS.
69 *-----
70
71 LDA LOMEM PARTIR DU DEBUT DES VARIABLES
72 STA VARTAB
73 LDA LOMEM+1
74 STA VARTAB+1
75
76 CHRCH LDY #0
77 LDA (VARTAB),Y
78 CMP #'M C'EST MS ?
79 BNE NXTVAR
80 TNY
81 LDA (VARTAB),Y
82 CMP #$90
83 BNE NXTVAR NON ALORS ON PASSE A PROCHAINE
84
85 * MS TROUVE.
86
87 TNY
88 LDA (VARTAB),Y

```

```

89 STA MASKLEN C'EST LA LONGUEUR DE LA CHAINE
90 TNY
91 LDA (VARTAB),Y
92 STA MASKADR ET SON ADRESSE EN MEMOIRE
93 TNY
94 LDA (VARTAB),Y
95 STA MASKADR+1
96 BNE VRFYMSK
97 *
98 * MS PAS TROUVE, SUIVANT...
99
100 NXTVAR LDA VARTAB
101 CLC
102 ADC #7
103 STA VARTAB
104 BCC CHRCH
105 INC VARTAB+1
106
107 LDA ARYTAB+1
108 CMP VARTAB+1
109 BCS ERR BOUT DE LA ZONE RESERVEE AUX
110 BNE CONTNXT VARIABLES SIMPLES ? SI OUI, ERRE
UR! SINON, ON PASSE A LA PROCHAI
NE VARIABLE
111 LDA ARYTAB
112 CMP VARTAB
113 BCS ERR
114
115 CONTNXT CLC
116 BCC CHRCH
117
118 * ERREUR, LE MASQUE N'EXISTE PAS
119 * OU N'EST PAS CONVENABLE
120
121 ERR LDY #>ERRMSG ON CHARGE L'ADRESSE DU MESSAGE D
'ERREUR
122 LDA #<ERRMSG
123 JSR STROUT PUIS ON SAUTE L'AFFICHER
124 JSR BELL AVEC BIP L'IGNOIBLE
125 JMP BASIC ET ON REND LA MAIN AU PROGRAMMEU
R DECU
126
127 ERRMSG HEX 8D
128 ASC "MASK ERROR"
129 HEX 00
130
131 *-----
132 * VERIFIE LA FORME DU MASQUE
133 *-----
134
135 VRFYMSK LDY MASKLEN PARTIR DE LA FIN
136 DEY -1 CAR LA POSITION ZERO COMPT
137 VRFYMSK1 LDA (MASKADR),Y PRENDRE UN CARACTERE DU MASQUE
138 CMP #' C'EST UN ESPACE ?
139 BNE VRFYMSK2
140 TNC SPACES
141 BNE NXTCHR
142 VRFYMSK2 CMP #'# C'EST UN # ?
143 BEQ NXTCHR
144 CMP #'# FEUT-ETRE UN . ?
145 BNE ERR RIEN DE TOUT CELA: LANCE L'INSUL
TE
146 STY MSKPOINT LA POSITION DU POINT DANS LE MAS
QUE
147 NXTCHR DEY PROCHAIN CARACTERE
148 BPL VRFYMSK1 ET VA JUSQU'AU BOUT !
149
150 *-----
151 * ATTENTION SI 0 < NOMBRE < 1
152 * SI ON A UN ENTIER, ALORS:
153 * CHNPOINT = CHAINLEN
154 *-----
155
156 LDA CHAINE SI LE PREMIER CARACTERE EST UN .
157 CMP #' ALORS ON NE PEUT PAS AVOIR UN EN
TIER...
158 BEQ MSKINT
159 LDA CHNPOINT LA POSITION DU POINT INDIQUE SI
ON A UN
160 BNE MSKINT ENTIER OU NON. (0-ENTIER, AUTRE-
REEL)
161 LDA CHAINLEN
162 STA CHNPOINT
163
164 *-----
165 * SI LE MASK EST ENTIER ALORS:
166 * MSKPOINT = MASKLEN
167 *-----
168
169 MSKINT LDA MSKPOINT OPERATION IDENTIQUE SUR LE MASQ
170 BNE VRFTAILLE
171 LDA MASKLEN
172 STA MASKPOINT
173
174 *-----
175 * CHNPOINT =< MSKPOINT-SPACES ?

```

```

176 *-----
177
178 VRFTAILLE LDA MSKPOINT      CECI POUR SAVOIR SI LA PARTIE EN
179 SEC                          TIERE DU NOMBRE PEUT ENTRER COMP
180 SBC SPACES                    LETEME NT DANS LE MASQUE.
181 CMP CHNPOINT
182 BCC ERR                        SI C'EST TROP ETROIT, IL FAUT LE
                                   DIRE
183
184 *-----
185 * ATTENTION SI 0 < NOMBRE < 1
186 * PARTIE AVANT LE POINT: ESPACES
187 *-----
188
189 LDY MSKPOINT                  ON MET DES ESPACES AVANT LE POIN
190 LDA #'                          T POUR QUE CA SOIT JOLI SI LE MA
191 CLRESULT STA RESULT, Y        SQUE EST TROP GRAND COTE ENTIER
192 DEY
193 BPL CLRESULT
194
195 LDA CHNPOINT                  SI LE PREMIER CARACTERE DU NOMB
196 BNE CLRESULT1                E EST UN POINT, ALORS CHNPOINT=0
197 LDY MSKPOINT                  ET IL FAUT METTRE UN ZERO AVANT
                                   LA VIRGULE
198 DEY
199 LDA #'0
200 STA RESULT, Y
201 CLRESULT1 EQU *
202
203 *-----
204 * PARTIE APRES LE POINT: ZEROS
205 *-----
206
207 LDY #SIF                       POUR QUE CA SOIT ENCORE PLUS JOL
208 LDA #'0                          I, ON MET DES ZERO A LA PARTIE D
209 ZERO STA RESULT, Y            ECIMALE AU CAS OU LE MASQUE SERA
                                   IT TROP GRAND
210 DEY
211 CPY MSKPOINT
212 BNE ZERO
213
214 *-----
215 * $00 EN FIN DE NOMBRE MASQUE
216 *-----
217
218 LDY MASKLEN
219 LDA #S00
220 STA RESULT, Y
221
222 *-----
223 * ATTENTION SI 0 < NOMBRE < 1
224 * COUPE LA CHAINE SELON LE MASQUE
225 * ET RANGE-LA DANS RESULT ($330)
226 *-----
227
228 LDA CHNPOINT                  0 < NOMBRE < 1 ?
229 BEQ FORM
230 LDX MSKPOINT                  IL FAUT SE POSITIONNER JUSTE AVA
                                   NT LE .
231
232 LDY CHNPOINT
233 DEY
234
235 FORMAT LDA CHAINE, Y          DECALE LA PARTIE ENTIERE EN PART
236 STA RESULT, X                ANT DU POINT
237 DEX
238 DEY
239 BPL FORMAT                    POUR TOUS LES CHIFFRES
240
241 FORM LDX MSKPOINT             ON INSCRIT LE POINT
242 LDY CHNPOINT
243 LDA #'
244 STA RESULT, X
245
246 FORMA LDA CHAINE, Y           ON EST AU BOUT DE LA PARTIE DECI
247 BEQ FORMEND                   MALE ?
248
249 STA RESULT, X                 NON ALORS CONTINUE DE DECALER A
                                   PARTIR
250
251 INX                            DU POINT ET VERS LA DROITE
252
253 CPY CHAINLEN                   IL RESTE DES CHIFFRES DANS LA CH
254
255                                AINE ?
256
257                                SI OUI, RESTE-T-IL DE LA PLACE D
258                                ANS LE MASQUE ?
259
260
261
262
263 FORMEND EQU *
264
265
266
267
268
269
270 *-----
271 * MET A JOUR LA VARIABLE Y=USR(X)
272 *-----
273
274 LDA LOMEM                      ON RECOMMENCE A CHERCHER UNE VAR
275 STA VARTAB                     IABLE DONC ON FAIT PAREIL QU'AU
                                   DEBUT...
276
277 LDA LOMEM+1
278 STA VARTAB+1
279
280 CHRCHY LDY #0
281 LDA (VARTAB), Y
282 CMP #'Y                          SAUF QUE MAINTENANT C'EST Y
283 BNE NXTY
284 INY
285 LDA (VARTAB), Y
286 BEQ YTROUVE
287
288 * Y PAS TROUVE, SUIVANT...
289
290 NXTY LDA VARTAB
291 CLC
292 ADC #7
293 STA VARTAB
294 BCC CHRCHY
295 INC VARTAB+1
296
297 LDA ARYTAB+1
298 CMP VARTAB+1
299 BCS RETOUR
300 BNE CHRCHY
301 LDA ARYTAB
302 CMP VARTAB
303 BCS RETOUR
304 BCC CHRCHY
305
306 YTROUVE INY
307 TYA
308 CLC
309 ADC VARTAB
310 STA VARTAB
311 BCC TRANSF
312 INC VARTAB+1
313
314 TRANSF LDA TXTPTR
315 STA TXTPTR2                      IL FAUT SAUVER LE TXTPTR ACTUEL
316 LDA TXTPTR+1                    CAR ON VA AVOIR A S'EN SERVIR PO
317 STA TXTPTR2+1                    UR REMETTRE UNE CHAINE DANS FAC
318 LDA #>RESULT
319 STA TXTPTR+1                      CETTE CHAINE, C'EST RESULT
320 STA TXTPTR-1                      C'EST A DIRE CE QUI SERA AFFICHE
                                   CAR TXTPTR POINTE SUR LE DERNIER
                                   CARAC.UTILISE. PRENDRE DONC LE P
                                   RECEDENT.
321
322 JSR FIN
323 LDY VARTAB+1
324
325 LDX VARTAB
326 JSR MOVME
327
328 LDA TXTPTR2
329 STA TXTPTR
330 LDA TXTPTR2+1
331 STA TXTPTR+1
332
333 RETOUR RTS

```

Programme PRINTUSING DEMO

```

110 PRINT CHRS (4) "BLAD PRINTUSING"
120 REM -----PROTEGE LA ROUTINE
130 HIGH: 32767
140 REM -----FINE LE VECTEUR USR
150 POKE 11,0: POKE 12,128
160 REM -----PAS DE VALEUR-RETOUR
170 POKE 33064,96
180 REM -----EXEMPLES
190 TEXT : HOME : INVERSE : PRINT " PRINT USING / DEMONSTRATION
    " : NORMAL : VTAB 5
200 PRINT "
210 VTAB 20: PRINT "LE 'MASK ERROR' SE PRODUIT CAR LE MASQUEEST TROP ETRO
    IT POUR LA PARTIE ENTIERE DU NOMBRE.": VTAB 6
220 FOR I = 1 TO 12 : READ X : PRINT X: HTAB 10
260 MS = " ###"Y = USR (X)
270 MS = " ##.##"Y = USR (X)
280 MS = " #####.###"Y = USR (X)
290 PRINT : NEXT
300 DATA -10,1.25,12.5,23.759812,.1,46.25,57.5,68.75,80,91.2513,-0.1,1234
    5.67890

```

Bibliographie

Alexandre Duback, Alexandre Avrane

Atlas de géographie historique de la France et de la Gaule, par Stéphane Sinclair - SEDES (88, bd. St. Germain, 75005 Paris) - 260 pages - 105 FF.

L'originalité de cet ouvrage, en dehors de son sujet, tient à ce qu'il comporte de très nombreuses cartes réalisées à partir du Macintosh et sorties en LaserWriter. Décidément, celle-ci gagne du terrain tous les jours !

Gérer le développement de l'entreprise (guide Multiplan), par W. R. Osgood, D. P. Curtin et F. Camelin - Cedic/Nathan - 172 pages - 240 FF - Traduction/Adaptation.

Contrôler et optimiser les décisions financières (guide Multiplan), par J. Alves, M. Selva, B. Laur, D. Rigolet et P. Derly - Cedic/Nathan - 186 pages - 240 FF - Traduction/Adaptation.

Dans une présentation claire, ces guides sont de bons outils de travail. Le premier est en quelque sorte un check-list pour la création, à l'aide de Multiplan, d'un dossier sur votre entreprise, en vue d'une demande de prêt ou d'une opération d'augmentation de capital. Le second est spécialisé dans la gestion financière. Ces deux livres sont intéressants par leur aspect pédagogique, dans la matière étudiée comme pour la constitution des tableaux.

Apple, modems et serveurs, par Alain Mariatte - Editions du PSI - 220 pages - 130 FF.

Après une introduction sur la révolution télématique, cet ouvrage comporte trois parties d'importance comparable : la technique des transmissions, les réseaux télématiques et les aspects logiciels.

Ce livre se lit aisément et comporte de nombreux exemples de connexion sur divers systèmes, ainsi que plusieurs petits programmes utilitaires. Un bon complément du livre de Jacques Bessières, la Téléinformatique Personnelle.

Apple ProDOS : Advanced Features for Programmers
Gary B. Little Prentice-Hall Publishing - 266 pages - \$23.35 (sans disquette)

Ce livre s'adresse aux programmeurs système désireux d'implanter leur création dans un environnement ProDOS. Il suppose donc un certain nombre d'acquis, en particulier de l'assembleur 6502 et du ProDOS Technical Reference Manuel (je trouve la lecture de ce dernier indispensable mais aussi insipide que les brochures IBM. Où est passé le style d'écriture d'Apple ?)

Il reprend les points fondamentaux des protocoles d'interface avec ProDOS : appel du MLI, gestion de la carte RAM, interpréteurs-système, routine d'interruptions, modules d'accès à une mémoire de masse ou une carte horloge, etc.

Moins descriptif que Beneath Apple ProDOS, il aborde surtout les aspects pratiques d'utilisation mais se montre, à l'usage, beaucoup plus utile.

Informatisez votre comptabilité et votre gestion, par Bernard Eben - Marabout Informatique - 256 pages (format livre de poche).

Cet ouvrage donne un certain nombre de conseils sur l'art et la manière de programmer la comptabilité d'une entreprise, avec un petit programme en Basic (où l'auteur gagnerait à découvrir l'existence de variables indicées) et un programme dBase II de 18 pages (plus intéressant). Il aurait été plus intéressant de construire le livre tout entier autour de la création d'un programme de compte personnalisé en dBase II.

Macintosh, l'ami du travail et du jeu, par Lon Poole - Editions Cedic Nathan - 448 pages - 195 FF - Traduction.

Un bon livre d'introduction au Macintosh : une première partie générale, une seconde partie avec des chapitres sur MacWrite, MacPaint et Multiplan, une troisième partie présentant 24 petites applications en 90 pages, et enfin un dossier de référence.

Applications graphiques pour Macintosh, par Andreas Pfeiffer - Cedic/Nathan - 281 pages - 150 FF.

Six parties : Macintosh et création graphique, Interfaces de digitalisation, Dessin de précision et CAO, Logiciels de mise en page, Générateurs de graphiques, De l'utilisation de belles images.

Livre sorti sur LaserWriter dans une présentation bien pensée et agréable. Le fond est à la hauteur de la forme. Un livre à conseiller à tous ceux que les applications graphiques intéressent.

Apple ProDOS Disk/File Handling de Graham Keeler - Prentice-Hall Publishing - 353 pages - \$17.95 (sans disquette)

Livre d'initiation aux techniques de manipulation de fichiers sous ProDOS, il s'adresse principalement aux débutants ou aux utilisateurs qui ne possèdent pas les manuels Apple de programmation sous ProDOS.

Les chapitres décrivant les commandes de ProDOS sont complétés par une série d'exercices en Basic Applesoft. Puis une introduction à la création d'une base de données est esquissée, suivie de quelques routines utilitaires.

Utiliser Multiplan, par Erwin Schneider - Editions d'Organisation - 260 pages - Traduction.

Hélas, encore une traduction effectuée par quelqu'un qui ne connaît pas Multiplan ou a bâclé son boulot ! Tous les exemples sont présentés à partir d'un Multiplan en version française, avec la syntaxe de la version américaine : par exemple, les arguments des fonctions sont séparés par des virgules, et non des points-virgules comme c'est le cas dans la version française.

La traduction mise à part, nous n'apprécions pas non plus la logique de l'ouvrage, qui consiste à présenter les liens externes avant même de montrer comment faire une somme...

"Ah ! le voilà lui". Devinez qui nous rentre au bercail sur l'air du père prodigue ? Eh oui, c'est bien lui : Steve Wozniak, le "papa" de l'Apple // qui revient dans le giron d'Apple à Cupertino, en commençant par racheter la valeur de 5 millions de dollars d'actions d'Apple. Il en convoite encore d'autres pour 15 millions de dollars. Au total, un million d'actions frappées de la pomme pour 20 millions de dollars. Une remarque au passage : Wozniak ne fait pas une mauvaise affaire : il en avait vendu 3 millions en février pour 70 millions de dollars. Un petit travail sur la calculatrice du Macintosh : il rachète 20 dollars des actions qu'il avait vendues 23,33 !

Pourquoi ce retour ? Officiellement, parce qu' "Apple va dans le bon sens". Il faut dire que depuis quelques temps, Steve Wozniak s'intéressait énormément aux produits d'Apple. Comme si l'Apple // lui manquait. Au point de se transformer en homme sandwich. Il ne se passait pas de semaine, sans qu'on ne voie sa photo dans une publicité pour proclamer, combien il admirait "de tout cœur, les produits d'Applied Engineering", ou s'exclamer que "la série des programmes Mouse (les produits de Version Soft, vendus aux Etats Unis par International Solutions) est l'évolution naturelle des logiciels pour Apple //, en apportant les avantages du Mac à l'Apple //".

En ce mois de janvier frileux, la date inscrite sur l'agenda de tous les Applemaniaques c'est celle de l'Assemblée des actionnaires qui apportera comme chaque année son lot de nouveautés. Cette année, c'est sur le Mac nouveau qu'on spéculait.

Un Macintosh plus puissant (1 Mégaoctet de mémoire en standard, extensible jusqu'à quatre, une nouvelle ROM de 128Ko, intégrant davantage d'éléments du système contenus jusqu'ici sur la disquette, un lecteur de disquettes double face de 800Ko, bref, la réponse aux souhaits de tous les utilisateurs de Macintosh). A dire vrai, on prévoyait même deux Macintosh, l'autre étant encore plus révolutionnaire : modulaire en trois éléments, avec un clavier plus complet, un écran séparé, et une unité centrale comportant une sortie pour disques durs au standard du marché.

Les possesseurs de Macintosh d'avant 1986 ne seraient pas laissés pour compte, puisqu'on leur proposerait une extension de mémoire à un Mégaoctet, accompagnée d'un lecteur de disquettes double face et de la nouvelle ROM. Les prix seraient intéressants, puisque l'extension de 128Ko à 1 Méga serait proposée au même prix que l'ancienne extension à 512Ko. A ce propos, les Macintosh de 128Ko ne seront vraisemblablement plus fabriqués. Le

512Ko devenant le bas de gamme Macintosh. Ce qui est sûr, c'est que le maître mot inscrit au fronton d'Apple, c'est désormais la compatibilité.

Compatibilité sur le Macintosh

Compatibilité interne d'abord : le Mac serait capable bientôt d'émuler parfaitement l'Apple //, et comme nous l'annoncions dans le précédent numéro de Pom's, l'Apple // rendrait la pareille à son cadet.

Compatibilité externe surtout : les Apple iraient en enfer et s'intégreraient dans l'univers impitoyable. Lequel ? Celui d'IBM, bien sûr. Tout en annonçant une augmentation de son budget de recherches de 50% (une manière de répondre à ceux qui, comme l'auteur de cette rubrique, doutent des capacités d'Apple, privé de Steve Jobs, à développer aujourd'hui l'ordinateur à sortir dans cinq ans), John Sculley, le PDG de la firme a annoncé en octobre "qu'Apple offrirait des produits compatibles avec les standards d'IBM".

Traduisons : il ne s'agit pas de construire des compatibles IBM sous la marque Apple (bien qu'à Cupertino, on s'intéresse, paraît-il, énormément à la seconde version, avec disque dur, de Mac Charlie, cette extension transformant le Mac en terminal compatible IBM, capable d'avaloir toutes les disquettes du PC, et de métamorphoser un superbe écran graphique intelligent en engin barbare affichant le sinistre A> du MS-DOS). Il s'agit d'intégrer les Apple à l'univers IBM, devenu que cela plaise ou non celui de la majorité des entreprises d'Outre Atlantique et de France. Cette intégration se réaliserait de deux façons :

1° par les fichiers. Apple et IBM seraient capables de travailler avec des programmes différents, mais de s'échanger les données, ce qui éviterait des doubles saisies. Ceci grâce à l'adoption par Apple de la norme des fichiers développée pour l'IBM et baptisée DCA.

2° par les réseaux : des Apple seraient branchés sur les réseaux comportant des IBM. Pour cela, Apple a annoncé son adhésion au standard de réseau en anneau à jeton dévoilé par IBM en octobre. Ce branchement serait aussi possible sur les réseaux d'autres constructeurs. A surveiller particulièrement le réseau EtherMac pour Macintosh, qui pourrait communiquer facilement avec le réseau EtherNet développé par le même constructeur pour les PC.

L'année 1986 apportera aussi aux amateurs d'Apple de bonnes nouvelles sous forme de baisses de prix. Celui de l'imprimante Laserwriter avait déjà baissé (sous la pression de la concurrence) de 1000 dollars aux Etats Unis à la fin 1985 pour passer à 6000

dollars. Les Apple //c étaient tombés au-dessous de 10000,00 Francs dès la fin de l'année pour une configuration "opérationnelle". On pouvait même trouver pour moins de 8000,00 Francs une configuration comprenant un Apple //e, un lecteur et un moniteur. Il est vraisemblable que ce n'est qu'un début. Et que des baisses de prix s'appliqueront encore à toute la gamme Apple. Pour le même prix, on aura peut être cette année en plus une imprimante. Qui s'en plaindrait ? En attendant, le panorama des

chaleureux supporters de l'Apple //.

Chute du prix des mémoires du Mac

En France aussi, la baisse du prix des mémoires a contaminé le Mac. En décembre, l'extension de 128 à 512Ko était ainsi couramment proposée à 2 500,00 Francs et le passage de 512 à 1,5 Mégas à moins de 8 000,00 F.

Micro-informations

Jean-Michel Gourévitch

Disques durs

Pour l'Apple // d'abord. Avec cette fois, un disque dur interne américain proposé par SCS. Il est contenu sur une carte s'installant sur un des slots et disponible en version 10 Mégas (\$1200s) et 20 Mégas (\$1600). Un pari : on n'a pas fini de voir se développer de tels disques durs "en carte" pour l'Apple // et leur prix baisser.

Et le Mac ?

En attendant le disque dur de 20 Mégas Apple, Micro Expansion, a proposé un disque dur de 5 Mégas pour le Mac, pour moins 12 000,00 F (qu'il est possible d'échanger ultérieurement contre un disque dur de plus grande capacité).

Mais, l'offensive a été ouverte sur le front des lecteurs double face. Premier à tirer, le français P-Ingénierie qui s'est avisé à la fin 1985 que le lecteur double face de 800Ko Unidisk pour l'Apple // pouvait parfaitement fonctionner sur le Mac (à condition de réaliser un bricolage très simple court-circuitant la sortie "de chaînage" de ce lecteur). Restait cependant un problème : comment formater des disquettes double-face ? Avec le nouveau Mac, ce sera facile, car les instructions sont dans la nouvelle ROM. Et en attendant ? P-Ingénierie a tout simplement écrit un logiciel de formatage des deux côtés de la disquette et décidé de vendre des disquettes double face Fuji pré-formatées. Pour les utilisateurs, le coût des disquettes était élevé, mais la modification intéressante : 800Ko, en utilisant un Unidisk, cela permet de caser un Switcher et plusieurs applications, ou d'avoir un gros système (avec accessoires de bureaux et polices

matériels et logiciels lancés à la fin 1985 pour les Apple s'avère particulièrement riche. Comme une manière de pied de nez aux concurrents du style Atari ou Amiga qui voudraient déjà hériter du marché.

Mémoire : toujours plus !

Il est bien loin le temps où 16 Kilo-octets suffisaient à tenir la comptabilité d'une banque. Désormais, les exigences de mémoire pour les utilisateurs de micro-ordinateurs se chiffrent en Mégaoctets. Reprenons les épisodes précédents : dans le numéro 21, Pom's annonçait 1,5 Mégaoctets casés sur une carte de l'Apple //e. Record battu par Applied Engineering avec 3 Mégas pour 1700 dollars. Qui dit mieux ? Cette course à la mémoire ne se fait pas sans dégâts. Pendant quelques temps, l'absence d'Apple avait laissé le champ libre à quelques pionniers comme Applied Engineering. Ceux-ci ont développé des cartes s'installant dans le slot auxiliaire de l'Apple //e. Or voici qu'Apple a récemment sorti sa carte d'expansion. Et que celle-ci s'installe dans un des slots normaux de l'Apple.

Conséquence : les programmes comme AppleWorks où l'on a installé un "patch" pour reconnaître la mémoire supplémentaire des fabricants de cartes ne reconnaissent pas celle d'Apple. Plus grave : les concepteurs de logiciels écriront de préférence des programmes pour la carte d'Apple, et ceux-ci ne pourront utiliser les cartes concurrentes. Si ce malentendu n'est pas réparé, il risque de faire du vilain chez les constructeurs d'extensions et cartes en tous genres qui sont pourtant les plus

de caractères). Reste que cette opération était forcément limitée dans le temps : les nouveaux Mac, présentés ce mois-ci auront en standard les 800Ko.

A noter d'ailleurs que plusieurs firmes d'Outre Atlantique se sont également lancées dans les lecteurs double face, c'est le cas de Mirror Technologies avec le Mirror Magnum de 800Ko, et de Dataspace Corporation avec un autre lecteur double face.

Des gadgets

La micro-informatique n'y échappe pas. Aux États-Unis, un constructeur vend ainsi une table spéciale pour le Mac : la MacTable de ScanCoFurn, spécialement construite pour abriter le Mac, l'Imagewriter, etc. Un fabricant a trouvé moyen de fournir un écran LCD pour l'Apple //c, plus lisible que celui d'Apple, c'est le C Vue de Roger Coats vendu avec une batterie pour alimenter l'Apple //c pour 600 dollars.

Plus intéressante l'Image Blue de Image Communications. Cet appareil (mi-scanner, mi-imprimante) permet simultanément de "scanner" (c'est à dire de transformer lignes ou caractères en données informatiques) un texte ou un dessin et de l'entrer dans un programme du Macintosh (4 pages graphiques peuvent être stockées sur une disquette), puis de l'imprimer. Prix : 1295 dollars. A noter que cette machine est également compatible avec l'IBM PC.

Enfin, le Soft Strip de Cauzin pourrait changer la vie de tous les amateurs qui recopient des programmes dans des revues spécialisées de micro-informatique. C'est un lecteur de code qui se branche sur un micro (et notamment sur l'Apple //) par l'entrée RS232. Les listings de programmes, graphiques, sons, etc, peuvent être imprimés sous la forme d'un ruban de code barre. Le lecteur lit ce code et le retranscrit dans l'ordinateur. Son prix n'est que de 199 dollars, et il suffit de 30 millisecondes pour lire un ruban. Les éditeurs américains étudient la possibilité de publier leurs programmes sous cette forme dans leurs revues : plus besoin d'acheter les disquettes, ni d'entrer les données au clavier... On rêve déjà de livres d'école comportant des "soft strips".

Logiciels : à l'Ouest du nouveau

Voici en effet une avalanche de nouveaux logiciels venus des États-Unis. Pour l'Apple // d'abord avec Catalyst 3.0. Ce logiciel permet d'installer des logiciels protégés sur un disque dur et de passer de l'un à l'autre. Le voici désormais distribué par Apple et livré en version souris et

graphique, utilisable avec le nouvel Unidisk de 3,5 pouces. Une impression : il ressemble bigrement au fameux MouseDesk de Version Soft. Serait-ce donc le même produit sous un autre nom ?

Voici ensuite, réservée aux initiés la version 6.0 de Locksmith d'Alpha Logic. Copie rapide en huit secondes seulement...

Voici enfin Fantavision de Broderbund. Un extraordinaire logiciel de création de dessins animés. La seule méthode pour donner de la vie à un dessin. On le dessine, et il s'anime, les ombres s'ajoutent, on le fait bouger sur l'écran. Fantavision permet d'importer des écrans haute résolution d'autres programmes.

En France aussi

En France, aussi, les créateurs de logiciels pour Apple // ont mis la main au clavier. L'un d'eux, MG Informatique, propose un additif s'installant sur Appleworks et permettant de l'utiliser avec une souris. Un autre, Priam, propose un traitement de texte baptisé Saphir intégrant des fonctions mathématiques et des calculs conditionnels, permettant de travailler simultanément sur deux documents, d'obtenir une double justification avec césure automatique et de visualiser le document final. Le tout pour moins de 2 000,00 F avec en prime mailing multicritère et glossaire : une manière d'exploiter. Enfin, VersionSoft, qui a fusionné avec Control X, propose une rafale de nouveaux logiciels : Version Tel, permettant de transformer en Minitel un Apple // équipé d'un modem (650,00 F), Arcane, pour prédire l'avenir (680,00 F) et pour les enfants des Puzzles et des Coloriages (360,00 F chaque).

Pour les professionnels du bâtiment, Architrion de Gimeor permet de créer des objets en trois dimensions. Dans chaque bloc, on peut créer des ouvertures et y insérer des huisseries, etc. Ce logiciel spécialement étudié pour les professionnels du bâtiment a été primé par l'Agence de l'Informatique et le Ministère de l'Urbanisme. Prix : 15 000,00 F. Ce n'est pas donné.

Nouveautés pour le Mac

Avec pour ceux qui utilisent le Macintosh comme outil de publication, de nouvelles versions des logiciels d'édition : Ready Set Go version 2.0 distribué par BIP, et Mac Editeur 2.0 distribué par Feeder. Tous permettent désormais d'importer des textes saisis avec un traitement de texte comme Macwrite ou Word.

Au rayon des utilitaires, Disk Ranger de Mainstay, permettant de réaliser facilement des étiquettes pour les disques, et de se constituer un catalogue (prix : 50 dollars). Et surtout Tempo, d'Affinity, un éditeur de macros, permettant d'automatiser des commandes, en prévoyant des pauses pour les entrées de textes, la création de zones de dialogue, des branchements logiques, etc. pour 99 dollars. Une seule commande à effectuer, et l'ordinateur prélève des informations dans un programme, les inclut dans un tableur, les compare aux mêmes résultats d'une période précédente, en fait un graphique et expédie le tout par modem à un analyste... De Vendée, arrive une comptabilité générale simple : Mélusine de Brocéliande.

Pour les professionnels de la chimie voici Mac Chemistry de Fortnum Software comportant des illustrations, un attirail de laboratoire permettant de réaliser à l'écran des titrages d'acides et bases, un utilitaire pour effectuer des annotations et une table périodique de références. Prix : 145 dollars.

Des jeux pour le Mac

Le Mac peut aussi être un objet futile. La preuve : MacInooga Choo Choo de Fortnum Software. Réinventer le train électrique sur micro, il fallait le faire non ? C'est fait. Avec ce programme permettant de dessiner un circuit ferré avec terrains, rails, aiguillages et d'y faire circuler des trains. N'est ce pas le journaliste Daniel Garric qui a dit que la micro informatique était le train électrique des adultes des années 1980 ? Prix 39 dollars. Wizardry était l'un des best sellers des jeux d'aventure pour l'Apple //. Le voici désormais pour le Mac : c'est Mac Wizardry de Sir Tech. Prix : 60 dollars. Strategic Conquest de PBI est un jeu de simulation de guerre comprenant 15 niveaux de difficultés et deux milliards de cartes du monde possible. Déjà vu de Mindscape est un jeu d'aventure mélangeant texte et graphique. Deluxe Music Construction Set d'Electronic Arts est l'un des plus sophistiqués des programmes permettant de faire de la musique sur le Mac. Quant à Enchanted Scepters de Silicon Beach Software, c'est un classique jeu d'aventure mélangeant texte et graphiques, mais y ajoutant des éléments sonores inédits. A noter enfin un remarquable jeu de boxe : Sierra Boxing Champion-ship de Sierra On Line. On peut jouer contre l'ordinateur en choisissant son adversaire parmi une kyrielle de grands boxeurs. On peut aussi les faire jouer entre eux, et organiser un championnat avec Cassius Clay. Le jeu a été réalisé par d'authentiques spécialistes, et les

boxeurs ont les caractéristiques de leurs vrais modèles.

Adresses

Applied Engineering
P.O. BOX 798 Carrollton, TX
75006

SCS
301 S. Washington Ave. P.O.
Box 2767, Titusville, FL 327081

P-Ingénierie
226 Bd Raspail, 75014 Paris

Mirror
Technologies 2209 Phelps Road
Hugo, Minnesota 55038

Data Space Corp
205 Riviera Drive, Markham
Ontario Canada L3R2L6

Micro Expansion
234 route de Genas 69003 Lyon

ScanCoFurn
Tél aux USA : 1.800 722 6263

Roger Coats
Tél aux USA : (619)274 1253

Image Communications
640 West Putnam Av. P.O. Box
4809 Greenwich CT 06836 0086

Cauzin Systems
835 South Main Street Waterbury
CT 06706

Alpha Logic Business
Systems
4119 North Union Road
Woodstock IL 60098

Broderbund
17 Paul Drive San Rafael CA
94903

MG Informatique
BP 1034 54521 Laou Cédex

Priam
53 rue de Paris 92100 Boulogne

Version Soft
93 rue Lauriston, 75016 Paris

BIP
13 rue Duc, 75018 Paris

Feeder
BP 781 13742 Vitrolles Cédex

Mainstay
28611B Canwood St Agoura
Hills CA 91301

Affinity
1050 Walnut Street Suite 425
Boulder CO 80302

Brocéliande Productions
BP 162 85105 Les Sables
d'Olonne

Fortnum Software
31W Sierra Madre Blvd Sierra
Madre CA 91024

Gimeor SA
39 quai d'Alsace 59500 Douai

Sir Tech
6 Main Street Ogdensburg N.Y.
13669

PBI
Software 1111 Triton Drive 2nd
Floor Foster City CA 94404

Mindscape
3444 Dundee Road Northbrook IL
60062

Electronic Arts
2755 Campus Drive San Mateo
CA 94403

Silicon Beach Software
P.O. Box 261430 San Diego CA
92126

Courrier des Lecteurs

Olivier Herz, Alexandre Avrane, Bruno Fénart

Voici deux petits programmes Basic pour recopier sur imprimante les écrans texte 40 ou 80 colonnes. Lors de la mise au point de programmes, ils peuvent rendre des services : il n'est alors pas nécessaire de se préoccuper immédiatement des éventuelles questions de compatibilité des routines en assembleur (HCT. OBJ déjà parue dans Pom's, par exemple).

Programme COPIE40

```
10 PRINT CHR$(4);"PR#1":
  FOR Y = 1 TO 24: FOR X =
  1 TO 40:A$ = CHR$( SCRN
  N( X - 1,2 * (Y - 1)) +
  (16 * SCRN( X - 1,2 * (
  Y - 1) + 1))) :B$ = B$ +
  A$: NE XT X: PRINT B$:B$
  = "" : NEXT Y: PRINT CHR
  $(4);"PR#0"
```

Programme COPIE80

```
10 PRINT CHR$(4);"PR#1":
  POKE - 16383,0: FOR Y =
  1 TO 24: FOR X = 1 TO 40
  : POKE - 16299,0: GOSUB
  20: POKE - 16300,0: GOS
  UB 20: NEXT X: PRINT B$:
  :B$ = "" : NEXT Y: POKE -
  16384,0: PRINT CHR$(4)
  ;"PR#3": END
20 A$ = CHR$( SCRN( X -
  1),2 * (Y - 1)) + (16 *
  SCRN( X - 1),2 * (Y - 1
  ) + 1))) :B$ = B$ + A$: R
  ETURN
```

J-M Roulon, Les Cayes sous Bois

A vendre 2 Apple III (128Ko et 256Ko) avec deux lecteurs chacun. Parfait état de marche et prix intéressant. Cause : achat Macintosh. Contacter M. Claude Lamoureux au (1) 46.87.26.12.

Message à Yves Martin

Vous nous avez annoncé votre déménagement. Si vous n'avez pas reçu notre lettre, prière de nous contacter.

Le programme 'Gestion de fenêtres' souffre de quelques défauts de jeunesse. Tout d'abord, deux erreurs typographiques :

&C NF, MG, MD, MH, MB (.CH) en mode création et

& LET NV, VT, HT, LN, NI, (,IL) pour les menus (signalé dans Pom's 21).

J'ai constaté que le caractère délimiteur défini lors de la création d'une fenêtre n'était pas sauvegardé. Je vous propose donc ci-dessous une petite modification au programme :

La méthode est la suivante :

```
]BLOAD WINDOW
]CALL -151
*8A00:00 89
*8E78:4C 48 89
*8F66:4C 23 89
*8F70:4C 23 89
*8784:4C 2C 89
*905D:4C 3A 89
```

Entrer le listing ci-dessous :

```
*8900:4C EE 89 C6 20...
```

Puis sauvegarder le nouveau code objet :

```
BSAVE WINDOW,A$8900,L$0D00
```

Patch WINDOW LAHY

```
8900- 4C EE 89 JMP $89EE
8903- C6 20 DEC $20
8905- E6 21 INC $21
8907- E6 21 INC $21
8909- C6 22 DEC $22
890B- E6 23 INC $23
890D- 60 RTS
890E- E6 20 INC $20
```

8910-	C6 21	DEC	\$21
8912-	C6 21	DEC	\$21
8914-	E6 22	INC	\$22
8916-	C6 23	DEC	\$23
8918-	60	RTS	
8919-	20 0E 89	JSR	\$890E
891C-	20 9E 8D	JSR	\$8D9E
891F-	60	RTS	
8920-	EA	NOP	
8921-	EA	NOP	
8922-	EA	NOP	
8923-	20 03 89	JSR	\$8903
8926-	20 F1 8E	JSR	\$8EF1
8929-	4C 73 8F	JMP	\$8F73
892C-	AD 4E 8A	LDA	\$8A4E
892F-	F0 03	BEQ	\$8934
8931-	20 0E 89	JSR	\$890E
8934-	20 9E 8D	JSR	\$8D9E
8937-	4C 87 8F	JMP	\$8F87
893A-	AD 4E 8A	LDA	\$8A4E
893D-	F0 03	BEQ	\$8942
893F-	20 0E 89	JSR	\$890E
8942-	20 9E 8D	JSR	\$8D9E
8945-	4C 60 90	JMP	\$9060
8948-	AD 53 8A	LDA	\$8A53
894B-	C9 03	CMP	#\$03
894D-	90 11	BCC	\$8960
894F-	38	SEC	
8950-	AD 55 8A	LDA	\$8A55
8953-	ED 54 8A	SBC	\$8A54
8956-	C9 03	CMP	#\$03
8958-	90 06	BCC	\$8960
895A-	AD 54 8A	LDA	\$8A54
895D-	4C 7B 8E	JMP	\$8E7B
8960-	4C B0 8E	JMP	\$8EB0

Claude LAHY, Villejuif

Icare...

En plus des précisions apportées dans le courrier des lecteurs du Pom's 19, je voudrais apporter les précisions suivantes :

• &PROUT x envoie le code ASCII x, x pouvant être supérieur à 128 contrairement à CHR\$(x)... Pratique pour commander l'imprimante.

Exemple :

```
&PROUT27,52,"TOTO":&PROUT2
7,53,"toto" donne sur une
Epson :
```

```
TOTOtoto
```

• *Le curseur vraiment bizarre de HGR PRINT peut-être corrigé par le patch suivant :*

```
8E51: E6 4E D0 11 E6 4F B1
26 49 FF 91 26 B1 28 AA A5
1F 91 28 86 1F A2 00 CA D0
FD EA EA EA 2C 00 C0 10 DE
```

Pascal Cantot, La Courneuve

La 36ème piste, suite...

Une précision sur l'article de B. Toméno du Pom's 21 :

La modification indiquée ne suffit pas pour que le DOS utilise les secteurs libérés sur la piste \$11. Heureusement, seul un petit patch du DOS est nécessaire. A l'aide de votre éditeur de secteur préféré, il convient de modifier l'octet \$92 du secteur 1 de la piste 2 : mettre \$A9 au lieu de \$69. Au niveau de l'assembleur, cela revient à remplacer en \$B291 :

```
CLC          par CLC
ADC  #$11    LDA  #$11
STA  $B3EB   STA  $B3EB
```

Le code original ajoute à \$11 la valeur du déplacement (+1 ou -1) pour passer à la piste suivante. Selon le sens, on commençait donc en piste \$12 ou \$10. Le patch permet de commencer en \$11.

Un second point : pourquoi se limiter à 36 pistes ? Mon Duo-disk en tolère très bien 40. Pour essayer, il suffit de faire :

```
CALL -151
BEFE:28
INIT HELLO
```

G. Jacquet, Chatillon la Palud

Encore un mot sur la saisie de date en Pascal (par Eric Pascual, Pom's 16 page 53) ! Pour l'utiliser avec Pascal 1.2, la constante ADRDATE doit être :

- -21252 avec la version 64Ko
- -18342 avec la version 128Ko

Reportez-vous à la lettre de Christian Bellot (Pom's 18, page

77), pour les autres détails d'implantation.

Voici par ailleurs un petit patch permettant aux possesseurs de compatibles Apple II (genre Franklin ou Basis 108) de charger ProDOS. En effet, celui-ci vérifie, au moment du chargement, le type d'Apple utilisé et, dans le cas qui nous intéresse ici, refuse de poursuivre le boot. Il faut rechercher les codes suivants, à l'aide d'un éditeur de secteurs, puis les modifier :

```
LDX $FBB3 (AE B3 FB) devient
LDX  #$EA, NOP  (A2 EA EA)
ADC  #$0B, BNE +5 (69 0B D0
03) devient ADC  #$0B, NOP,
NOP (69 0B EA EA)
```

Ce patch ne fonctionne pas sur certains logiciels (comme AppleWorks) qui tentent, indépendamment de ProDOS, de déterminer s'il s'agit d'un IIe ou IIc.

Hubert L'HOPITAL, 93300
Aubervilliers

Merci pour ces informations !

Je possède un Apple IIe avec une carte 80 colonnes et un lecteur de disquettes. J'ai essayé de convertir des programmes DOS 3.3 en ProDOS avec l'utilitaire CONVERT fourni par Apple. Malheureusement, il me faudrait deux lecteurs. Comment faire ?

Olivier Sonderer, 30490 Montfrin

Trois solutions possibles:

- La plus facile : un grand sourire à votre distributeur pour qu'il vous laisse convertir vos programmes dans sa boutique en une après-midi.

- La plus compliquée : vous chargez le programme de conversion paru dans Pom's 16, en sortez par la troisième option et vous obtenez ainsi le DOS 3.3 et ProDOS simultanément actifs en mémoire ; il ne reste qu'à charger et sauvegarder individuellement chacun de vos fichiers.

- La plus rapide: vous exécutez le programme d'initialisation d'une disquette mixte DOS 3.3/ProDOS (Pom's 18) ; la conversion est alors réalisée en trois étapes : alimentation de la disquette par FID, conversion des fichiers par CONVERT, puis transfert de ceux-ci par FILER vers la disquette définitive.

Est-il possible de sortir, sur écran ou imprimante, des fichiers source de programmes en assembleur quand on ne possède ni l'assembleur idoine, ni de traitement de texte ?

Philippe DEMERAUX, 43360
ARVANT

- L'affichage des sources d'assembleurs est possible sous ProDOS par la commande additionnelle TDUMP (Pom's 20).

- Sous DOS 3.3, les sources stockés sous forme de fichiers TEXT sont accessibles par le TDUMP/DOS 3.3 (Pom's 12).

- S'ils sont stockés sous forme de fichiers binaires (par exemple Big Mac ou Merlin), il est nécessaire de patcher le DOS pour qu'il autorise la lecture :

```
A7C9- AND  #$7F (29 7F)
devient LDA  #$00 (A9 00)
```

c'est à dire : POKE 42953,169 ;
POKE 42954,0

- En revanche les fichiers de Lisa, stockés sous un format très particulier du DOS 3.3, et de plus encodés (les instructions sont conservées sous forme de tokens, comme le fait l'Applesoft), sont alors inaccessibles.

Un simple PR#1 permet de sortir sur imprimante.

EDIGRAPH (Pom's 21)

Lorsque vous bootez sur la disquette Pom's n° 21, faites CTRL-C dès la mise en route pour que le 'LOMEM' du HELLO ne soit pas exécuté.

Ainsi, vous pouvez faire :
RUN EDIGRAPH
sans obtenir un cruel Out Of Memory Error...

Bon de commande

Disquettes

HAIFA source	(cf. Pom's n° 5)	à 55,00 F
H-BASIC	(cf. Pom's n° 8)	à 150,00 F
MUSIC	(cf. Pom's n° 10)	à 80,00 F
DISK-MANAGER	(cf. Pom's n° 11)	à 450,00 F
BASICIUM	(cf. Pom's n° 13)	à 150,00 F
E.P.E.	(cf. Pom's n° 15)	à 150,00 F
PASCAL	(cf. Pom's n° 15)	à 80,00 F
MAX (Moniteur étendu)	(cf. Pom's n° 18)	à 150,00 F
DOMINOS	(cf. Pom's n° 19)	à 80,00 F
MACASTUCES	(cf. Pom's n° 21)	à 200,00 F
P-FORMAT][, P-POLICE][.....	(cf. Pom's n° 21)	à 200,00 F
COGO	(cf. Pom's n° 21)	à 150,00 F

Recueils

N°1, recueil des revues 1 à 4	à 140,00 F
Disquettes d'accompagnement 1 à 4	à 150,00 F
N°2, recueil des revues 5 à 8	à 140,00 F
Disquettes d'accompagnement 5 à 8	à 190,00 F

Revue, disquettes

Revue 4 7 8	à 35,00 F
Revue 9 10 11 12 13 14 15 16 17 18 19 20 21 22	à 40,00 F
Disquettes Apple II, //e, //c		
1/2 3 4 5 6 7 8 9 10 11 12	à 55,00 F
13 14 15 16 17 18 19 20 21 22		
Disquettes Macintosh		
14/15/16 groupées	à 150,00 F
17 18 19 20 21 22	à 80,00 F
Mac 'A'	à 80,00 F

Abonnements

Pour 6 numéros à partir du n°

Abonnement à la revue seule	à 200,00 F
Abonnement revue + disquettes Apple II, //e, //c	à 480,00 F
Abonnement revue + disquettes Macintosh	à 600,00 F

Total TTC :

Supplément avion hors CEE : 15,00F par numéro et/ou disquette : _____

Montant du règlement : _____

Envoyez ce bon et votre règlement à : EDITIONS MEV, 64 rue des Chantiers 78000 VERSAILLES

Nom :

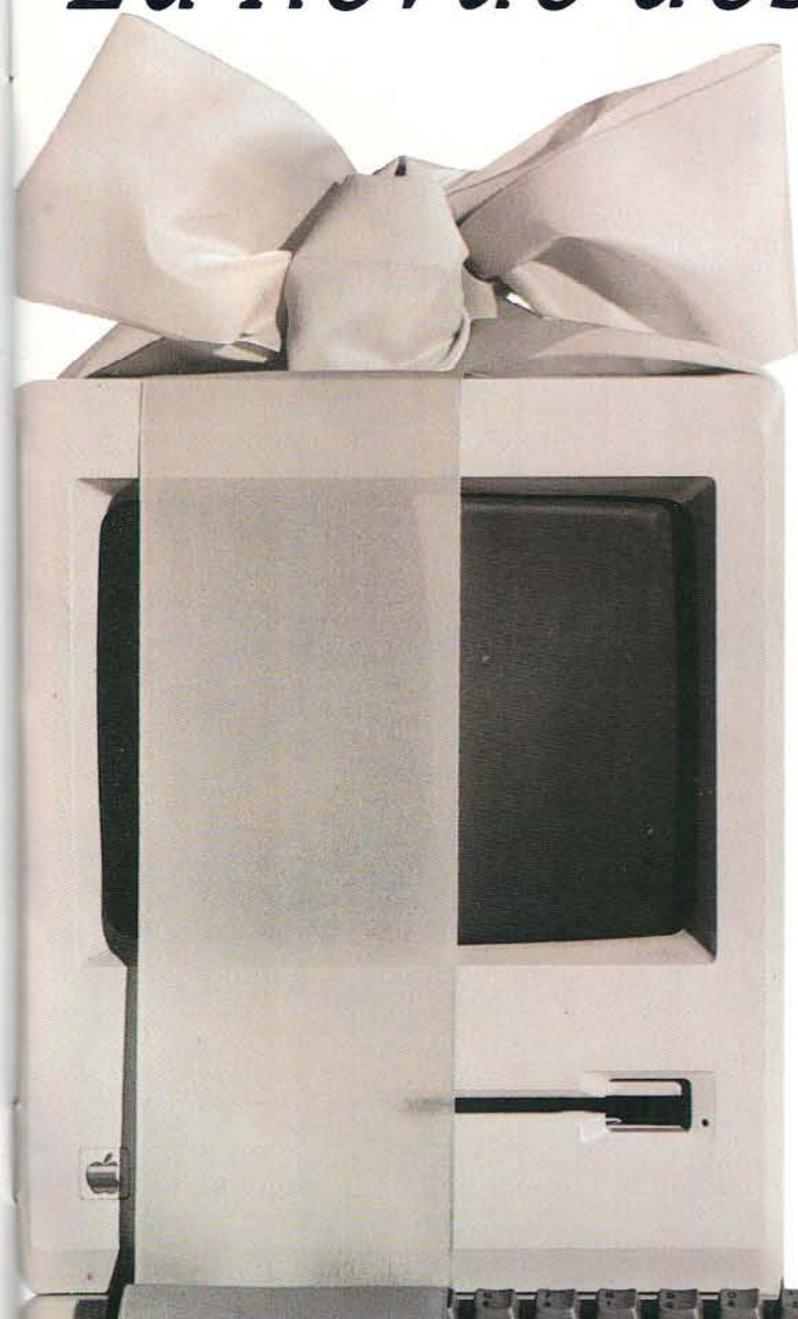
Adresse :

EXCEPTIONNEL!
le guide des 600
logiciels et périphériques

infomac

ISSN 0766-4915

La Revue des Macintosh



infomac
La Revue des Macintosh
5, PLACE DU COLONEL-FABIEN - 75491 PARIS CÉDEX 10 - TÉLÉPHONE : (1) 42.40.22.01

*- EXCLUSIF :
le nouveau Mac!
- MUSCLÉ :
les disques durs
- PRO :
la micro édition*

BON DE COMMANDE

à retourner à INFOMAG, 5 place du Colonel Fabien, 75491 Paris cedex 10

OUI je désire m'abonner pour 7 numéros à INFOMAG au prix de 215 F.
- Je désire recevoir, au prix de 35 F l'un : le numéro 1 le numéro 2 le numéro 3
Je vous adresse mon règlement par chèque bancaire à l'ordre de INFOMAG

NOM _____ PRENOM _____

ADRESSE _____

35^F DÉCEMBRE 85 N° 3

L'ORDINATEUR L'INDIVIDUEL

SPECIAL PROGRAMMES

42

ALICE • AMSTRAD • APPLE • ATARI • BBC
CANON X 07 • COMMODORE • EXL 100
IBM PC • MSX • ORIC • SINCLAIR QL
SPECTRUM • THOMSON M0 5 ET T0 7/70
TI 99/4A • TRS 80 • ZX 81

PROGRAMMES BASIC

inédits

**EN VENTE DANS
TOUS LES KIOSQUES**

JEUX, UTILITAIRES, EDUCATIFS
PLUS UN **TABLEAU COMPARATIF DES BASIC**
POUR ADAPTER LES PROGRAMMES
SUR TOUS LES MICRO-ORDINATEURS