

pom's

Sommaire

Editorial		3
Faites le ménage dans la mémoire	<i>Gérard Michel</i>	5
La leçon de calcul	<i>Annie Mauffrey</i>	9
Les utilitaires de documentation	<i>Hervé Thiriez</i>	11
Incursion dans les mystères du DOS	<i>J.-F. Duvivier</i>	17
Applications de graphique H.R.	<i>Guy Mathieu</i>	21
Les éditeurs de texte	<i>Hervé Thiriez</i>	27
Un exemple de Hard Copy	<i>Christian Guérin</i>	30
Le Pascal — Analyse du directory	<i>Michel Crimont</i>	35
3 secondes pour trier	<i>Nicole Bréaud</i>	41
Sprechen Sie DOS	<i>Christian Guérin</i>	47
Des vers dans la pomme...	<i>Alexandre Duback</i>	49
Survol de l'Apple III	<i>Bruno Rives</i>	51
Le jeu de la vie	<i>J.-F. Duvivier</i>	55
Les clubs ont la parole		62
Revue de presse		63
Courrier des lecteurs		65
Trucs et astuces		19-61-66



DEPARTEMENT SOFTWARE

PERSONAL SOFTWARE

Cette société, installée en pleine "Silicon Valley", à Sunnyvale en Californie, est celle qui a "inventé" l'un des best-sellers de programmes de microinformatique : le Visicalc.

Nous sommes fiers de vous présenter ici leurs nouveaux produits. Tous ces programmes nécessitent au moins un APPLE 48 K et une unité de floppy. Ils sont fournis avec une abondante documentation, hélas pour l'instant en langue américaine.

Contactez nous pour les versions disponibles pour CBM.

VISICALC

(nouvelle version)

1690 F TTC

L'un des programmes pour ordinateur personnel parmi les plus répandus au monde. Parce que VISICALC est aussi simple à utiliser pour manipuler des données chiffrées sous forme d'un tableau qu'une simple feuille de papier avec un crayon.

Pour créer un tableau, il suffit juste de taper le texte, les chiffres ainsi que les formules à l'endroit que vous désirez sur l'écran.

En posant la question "et si...", VISICALC vous donnera des réponses immédiates. Changez un seul chiffre, immédiatement, toutes les autres valeurs concernées seront changées et les résultats affichés sur l'écran ou sur votre imprimante.

Ainsi, vous pourrez visualiser instantanément le résultat de vos décisions, avant même d, les prendre. L'écran de VISICALC est une fenêtre dans un énorme plan de travail, de 63 colonnes sur 254 lignes. C'est une énorme feuille de papier électronique, qui peut effectuer un grand nombre de calculs, y compris les fonctions transcendentes, les "if then" ou les opérateurs booléens. Sur l'écran de votre ordinateur, vous pourrez voir VISICALC effectuer ses calculs et afficher en quelques secondes des résultats si complexes, que, sans lui, vous auriez mis des heures à obtenir.

VISICALC est aussi à l'aise à utiliser qu'une calculatrice de poche. Son éditeur d'écran vous permet de dupliquer vos chiffres et vos formules quasi-instantanément. Avec quelques touches de votre clavier, vous pourrez dupliquer, pour des calculs répétitifs, des formules par lignes ou par colonnes. Vous pourrez même observer deux portions différentes de votre "feuille électronique" simultanément sur l'écran, en partageant votre écran en deux, horizontalement ou verticalement. Observez vos données en début et en fin d'année, vous pourrez ainsi voir la dépendance d'une variable sur une ou plusieurs autres.

Et ensuite, vous pourrez faire un état imprimé. VISICALC peut aussi échanger ses données avec d'autres programmes de Personal Software. Prenez les comptes de vos clients de votre CCA/DMS, afin de projeter dans l'avenir votre chiffre d'affaires. Introduisez ces chiffres dans VISIPILOT pour produire des graphiques ou des histogrammes.

DESKTOP PLAN II

C'est un programme qui vous permet aisément d'automatiser et d'organiser toutes vos analyses de budget, ainsi que vos plannings budgétaires.

Il vous guide pas à pas, depuis le plan original de votre budget ou prévisions, jusqu'au résultat final, le rapport, d'une présentation impeccable.

Il peut également générer ses propres graphiques pour une meilleure interprétation. DESKTOP PLAN peut saisir les modèles financiers les plus complexes, ce qui le rend particulièrement utile pour consolider des budgets décentralisés et les combiner dans des modèles identiques pour les projeter dans un budget global.

Si vous n'avez jamais fait de modélisation financière, ce programme vous introduira à une méthode plus facile et plus efficace de planning financier ou de budgétisation que les laborieuses méthodes du passé.

1690 F TTC

Ancienne version : 880 F TTC

VISIPILOT

Un programme qui ajoute une dimension visuelle nouvelle à toutes vos analyses chiffrées, que ce soit de la finance, de la prévision ou de la planification.

VISIPILOT transforme les rangées et colonnes de chiffres, même les plus ardues à interpréter, en des graphiques ou histogrammes faciles à comprendre et cela devant vos yeux.

Vous obtenez ainsi une qualité de présentation de vos graphiques, qui améliorera votre information, et facilitera la compréhension.

Vous n'avez besoin d'aucune connaissance en programmation pour mettre en œuvre VISIPILOT. Un "menu" vous donne à choisir les options désirées, et vous n'avez plus qu'à entrer vos données. Votre APPLE fera tous les calculs nécessaires, et affichera votre graphique. Vous pourrez ensuite le sauvegarder sur disque pour une utilisation ultérieure, ou l'imprimer sur papier.

SHARP MZ 80

Programmes en Français

ECHecs 195 F TTC

Nécessite 20 K. Ce programme est recommandé aux débutants, vu la rapidité des réponses.

SUPER ECHecs 250 F TTC

Nouveau programme. Détrônara bien vite le précédent, à condition de disposer de 48 K. Utilise la notation internationale.

BLACK JACK 150 F TTC

C'est le jeu de carte bien connu, appelé aussi 21. Utilise le graphisme. De 1 à 8 joueurs. 48 K et Basic.

MUR DE BRIQUES 150 F TTC

Objectif : démolir un mur de brique. 1 à 9 joueurs sur 6 niveaux de difficulté. 32 K et Basic.

MORPION II 150 F TTC

Ce programme, mi Basic, mi assembleur, joue selon les règles traditionnelles. Le premier qui aligne 5 pions gagne. MZ 80 donne sa réponse en 0,2". Essayez de le battre. Depuis 20 K.

GUERRE DES ETOILES 195 F TTC

Jeu dérivé des fameux Startracks. Vous vous trouvez dans la galaxie, dans une aube de 8x8 quadrants. Votre mission est de détruire vos ennemis grâce à l'ordinateur de bord. 48 K et Basic.

Avec VISIPILOT, vous pourrez tracer plusieurs courbes à la fois, avec différentes séries d'informations, ombré plus ou moins les différents graphes, ou les colorier pour faciliter les comparaisons, par exemple des années ou des mois différents. Vous pourrez même surimpressionner plusieurs graphiques, par exemple vos stocks, et la courbe de vos ventes.

Ce programme permet de visualiser tous types de graphiques ou d'histogrammes, même circulaires. Il génère automatiquement les dimensions x et y, peut insérer des légendes, etc.

Bien entendu, il peut saisir aussi ses données depuis des programmes comme VISICALC, multipliant ainsi ses possibilités.

1690 F TTC

VISITREND / VISIPILOT

Si vous avez besoin d'analyses statistiques ou de prévisions, et que vous soyez découragés par la difficulté de les obtenir, alors, il vous faut VISITREND - VISIPILOT.

Ce programme répond aux besoins des chefs d'entreprise, des planificateurs, des investisseurs, analystes ou consultants. Vous pouvez obtenir des régressions linéaires pour connaître les rapports entre les variables, ou bien encore les moyennes variables, ou les lissages exponentiels, etc.

Où encore, écrivez des formules arbitraires pour calculer de nouvelles séries, basées sur celles existantes. Avec une simple commande, vous pouvez obtenir des statistiques descriptives sur n'importe quelles séries de données, par exemple leur variance, leur déviation standard, leur min ou leur max. Et avec la puissance graphique de ce programme, vous pourrez produire facilement graphiques ou histogrammes, ce qui pourra facilement vous permettre de comparer une série à une autre.

Ce programme a évidemment les mêmes possibilités graphiques que VISIPILOT. Comme tous les programmes de Personal Software, vous n'avez pas besoin de savoir programmer pour les mettre en œuvre, et ils sont compatibles avec VISICALC.

1950 F TTC

VISIDEX

Comment se souvenir de tout ? Ce programme est là pour cela. VISIDEX n'impose aucune restriction en ce qui concerne vos données : noms, adresses, dates importantes, rendez-vous, idées, notes, etc. et priorités, imports à payer (hélas), stocks et autres informations importantes.

Et vous pouvez retrouver exactement ce que vous voulez, car VISIDEX "pense" pour vous, vous donnant les moyens de retrouver n'importe quoi, plus vite.

C'est comme un fichier vide, vous tapez ce que vous voulez dans une fiche, et vous donnez à votre Apple les "clés" pour retrouver cette fiche : noms, phrases, chiffres, ou bien dates. Car vous décidez du classement de vos "fiches" et VISIDEX se souviendra pour vous de votre classement.

Vous donnez juste à votre ordinateur, une date ou un nom de client et VISIDEX vous visualise la "fiche" contenant ce nom ou cette information (par exemple, si vous introduisez le nom d'un client, vous aurez votre dernier rendez-vous, son planning d'organisation, ses hobbies, sa date d'anniversaire et les idées que vous pouvez lui proposer).

VISIDEX permet aussi d'imprimer vos informations par ordre alphabétique, numérique, ou encore par dates. Il permet de tenir votre agenda, et de vous alerter sur les dates, heures ou occasions importantes. Grâce à sa capacité de ne pas structurer par lui-même l'information, ce programme vous permet de faire vous-même votre classement.

Ainsi, votre secrétaire pourra introduire elle-même des données, et elles viendront se classer dans le bon ordre. Si vous craignez d'oublier quelque chose d'important, VISIDEX est fait pour vous !!!

1690 F TTC

CCA / DATA MANAGEMENT SYSTEM (CCA / DMS)

Mailings, fichiers clients, stocks, comptes fournisseurs et étiquettes adresses, tout ce type d'informations ainsi au jour-le-jour. CCA / DMS peut mieux le traiter.

Des tas de fiches sont remplacées par quelques disquettes ! Avec CCA / DMS, vous pouvez définir les informations que vous désirez saisir, la manière de les saisir ainsi que la manière de les présenter. Chaque enregistrement peut avoir jusqu'à 24 champs. C'est-à-dire que si vous introduisez le Nom, Adresse, Profession, Ville, Code Postal, il vous restera encore 18 informations différentes à saisir. Chaque fiche peut avoir 232 caractères de long. Vous pouvez trier vos fiches en alphabétique ou en numérique, en ascendant ou en descendant et jusqu'à dix critères à la fois ! Trier vos clients par chiffre d'affaire ou par nom, trier des adresses par code postal ou par ville !

Et CCA permet aussi de faire des opérations mathématiques simples, comme de multiplier des heures par un taux dans une même fiche. En mode d'impression, CCA peut imprimer des sous-totaux. Compatible avec VISICALC, VISIPILOT, etc. CCA / DMS est vraiment l'un des meilleurs programmes de fichiers, en séquentiel indexé, fait pour un microordinateur personnel.

990 F TTC

DEPARTEMENT LOCATION

ET MATERIEL D'EXPOSITION

La location est le meilleur moyen de choisir en connaissance de cause votre matériel. Un microordinateur est un achat coûteux.

Posez-vous les questions suivantes :

Aurai-je l'usage d'un microordinateur ? Celui que je choisis est-il le meilleur pour mes besoins ? Saurai-je m'en servir convenablement ? Etc...

Sans risque financier, grâce à la location, vous pourrez vous faire une opinion.

De plus, le montant de la location est entièrement déductible en cas d'achat

APPLE II 16 K	200 F TTC/jour
APPLE II 48 K	250 F TTC/jour
Moniteur vidéo N/B	50 F TTC/jour
DISK II avec contrôleur	110 F TTC/jour
CENTRONICS 779 avec interface	250 F TTC/jour
PET 2001	180 F TTC/jour
CBM 3032	230 F TTC/jour
VIDEOTEAM + TV coul. modifiée + écran 160 d'agon.	290 F TTC/jour

DEPARTEMENT OCCASIONS GARANTIES

Nos occasions pouvant très souvent se renouveler, nous vous recommandons de vous renseigner au préalable par téléphone.

LIBRAIRIE	Programmer en Pascal	66 F	Techniques d'Interface	124 F
	La découverte de l'Appaloft	55 F	Programmation du 2 80	98 F
	La pratique de l'Apple II (vol. 1)	55 F	Applications du 802	124 F
	Introduction aux microordinateurs	52 F	Le Basic par la pratique	68 F
	Programmation du 6502 (2 ^e édition)	98 F	Introduction au Basic	85 F
EN FRANÇAIS	Les microprocesseurs	98 F	Introduction au Pascal	98 F

29 rue de Clichy 75009 PARIS

Attention : les articles décrits sur ces pages ne sont pas forcément disponibles en magasin ; de plus, leurs caractéristiques techniques peuvent être modifiées sans préavis par le constructeur. Photos non contractuelles.

IC information : 285.24.55

Le recherche du meilleur prix d'achat par nos services peut nous amener à modifier à la hausse nos prix de vente, vous pouvez obtenir LA COTATION QUOTIDIENNE IC, en téléphonant à votre Centre IC.

Editorial

Contrairement à l'avis de certains sages et prophètes de mauvaise augure rencontrés au Sicob, le numéro 2 de Pom's voit le jour, et la série ne s'arrêtera pas là ! Au début du mois de décembre, Pom's s'est déjà diffusé à 1200 exemplaires, et des lettres nous parviennent tous les jours. Dans l'ensemble, la réaction des lecteurs a été très sympathique et constructive. Les seules critiques que nous ayons reçues à ce jour proviennent d'experts qui n'ont "rien appris" en lisant Pom's. Nous espérons qu'ils voudront bien partager leur savoir-faire en nous envoyant des contributions. A vos claviers ...

Nous n'avons pas pu faire tenir toutes les contributions dans ce numéro, et certaines nous sont parvenues trop tard pour que nous puissions les imprimer dans de bonnes conditions. Nous remettons donc au numéro 3 de Pom's l'article de Jean-Louis Meillaud sur l'apprentissage de l'assembleur. Rassurez-vous, il existe; nous l'avons enfin reçu. En attendant, vous pourrez démarrer votre initiation à l'assembleur avec "La pratique de l'Apple", tome III, sorti au Sicob. De même, l'article comparatif des cartes 80 colonnes devrait refaire surface au sommaire du prochain numéro. Vu les difficultés que nous avons pour tout sortir dans les délais, nous prenons la précaution de ne pas vous soumettre le futur sommaire, en vous laissant la surprise !

Nous terminons l'année 1981 avec l'apparition de l'Apple ///; il ne s'agit pas là d'un prématuré ! Nous vous le présentons sans tarder. Il y a aussi du nouveau dans notre disquette d'accompagnement : elle est maintenant bootable indifféremment en DOS 3.2 ou DOS 3.3. Par contre, elle n'est pas copiable par le programme de copie du Master; c'est une conséquence inévitable de la procédure utilisée cette fois-ci. Cela ne vous empêche pas de copier les fichiers qui vous intéressent sur d'autres disquettes.

Gilles Mauffrey a réalisé un joli programme permettant de copier le programme Pascal de la disquette Pom's numéro 2 sur une disquette Pascal. Il nous expliquera dans le numéro 3 (fin mars) comment il transpose un programme Pascal sur une disquette normale, et vice-versa. Comme vous pouvez le voir, Pom's est à la pointe du progrès.

Bonne lecture - GOTO 1982 avec tous nos vœux !

Hervé Thiriez

Directeur de la publication - rédacteur en chef : Hervé Thiriez - *Siège social* : Editions MEV - 49, rue Lamartine - 78000 Versailles - *Rédaction et abonnements* : 59, bd de Glatigny - 78000 Versailles - Tél. (3) 918.13.07 - *Régie publicitaire* : Force 7 - 41, rue de la Grange-aux-Belles - 75483 Paris Cedex 10 - Tél. (1) 238.66.10 - *Diffusion auprès des boutiques informatiques et libraires* : Editions du P&SI : 41-51, rue Jacquard - BP 86 - 77400 Lagny - Tél. (6) 007.59.31.
Ont collaboré à ce numéro : Claude Barbot, Nicole Bréaud, Michel Crimont, Alexandre Duback, Jean-François Duvivier, Christian Guérin, Guy Mathieu, Annie Mauffrey, Jean-Louis Meillaud, Gérard Michel, Bruno Rives, Hervé Thiriez.

Pour mieux choisir "votre" ordinateur et pour mieux l'utiliser.



Lisez

L'ORDINATEUR INDIVIDUEL

Vous y trouverez :

L'actualité et les tendances de l'informatique individuelle • des galops et des bancs d'essai des principaux matériels • des panoramas et des tests comparatifs • le point des grandes manifestations internationales • des articles d'initiation • des synthèses • des programmes • des interviews "exemplaires" • des conseils • des idées • des astuces.

L'ORDINATEUR
INDIVIDUEL

chez votre marchand de journaux

41 rue de la Grange aux Belles - 75010 Paris

Faites le ménage dans la mémoire

GERARD MICHEL

Si vous programmez en BASIC Applesoft des applications impliquant de nombreux traitements de chaînes de caractères, vous avez sans doute constaté que la zone de stockage de ces chaînes se trouve souvent trop vite saturée. En effet, si vous changez en cours de programme le contenu d'une chaîne A\$ (par exemple, A\$="NOM" devient A\$="PRENOM"), le nouveau contenu est stocké en mémoire sans élimination préalable de son prédécesseur. En conséquence, de nombreux caractères devenus inutiles viennent encombrer la mémoire et en provoquent bientôt la saturation.

Ainsi que l'explique le Manuel de Référence, Applesoft dispose d'une routine assurant automatiquement l'élimination des données inutiles lorsque l'accumulation des chaînes de caractères menace de 'déborder' sur d'autres zones de mémoire actives. Ce nettoyage peut également être imposé par le programme au moyen d'instructions du type "X=FRE(0)"; vous décidez alors de procéder à l'opération à l'issue d'une phase de traitement comportant de nombreuses manipulations de chaînes.

La routine de nettoyage est la même dans les deux cas et repose sur le principe suivant :

1. Explorer la zone de stockage, trouver la première chaîne 'active' et la placer juste sous HIMEM.
2. Explorer de nouveau la zone de stockage, trouver la deuxième chaîne 'active' et la placer juste sous la précédente ... et ainsi de suite.

Quiconque a eu recours à cette routine a pu constater la lenteur de l'opération. Heureusement, un article paru dans le numéro de Janvier 1981 de Call-A.P.P.L.E. ouvre des horizons nouveaux pour les 'victimes' du FRE(0), et il nous paraît indispensable d'en faire bénéficier tous ceux qui n'en auraient pas eu connaissance.

L'auteur y présente un programme en assembleur, destiné à remplacer la routine Applesoft. Ce nouveau programme ne trouve pas seulement une chaîne active unique à chaque exploration de la zone de stockage, mais en trouve un nombre N, défini par l'utilisateur, N devant avoir une valeur comprise entre 1 et 127. Bien sûr, plus N est grand, plus le nettoyage s'effectue rapidement. Toutefois, le traitement des chaînes impose de réserver une place mémoire égale à 6xN octets et située au-dessus de la routine elle-même. Le choix de la valeur de N doit donc résulter d'un arbitrage entre rapidité de traitement et 'encombrement'.

Dans la version du programme assembleur baptisée FRE(16), dont la liste est donnée ci-après, N est fixé à 16, mais nous vous présentons également, sous l'appellation FRE(64), le code hexadécimal pour N égal à 64. En tout état de cause, vous pourrez constater que le gain de temps résultant du passage de FRE(0) à FRE(16) est dans la majorité des cas beaucoup plus appréciable que celui résultant du passage de FRE(16) à FRE(64).

Par ailleurs, la routine telle qu'elle apparaît ici est destinée à la même utilisation que l'instruction "X=FRE(0)". En d'autres termes, il vous appartient d'effectuer un "CALL 37500" en un point du programme où l'opération de nettoyage

vous paraît nécessaire et d'en forcer ainsi l'exécution.

Le programme publié dans Call -A.P.P.L.E. se rapproche plutôt de l'hypothèse où le programme BASIC laisse à Applesoft l'initiative du nettoyage, dans la mesure où un test préalable est effectué sur la place mémoire encore libre entre la fin du stockage des chaînes et le début de la zone de stockage des pointeurs de tableaux. Ainsi, le nettoyage ne s'exécute que si la dimension de cette place libre tombe en-dessous de 1K octets.

Ce test permet donc de ne procéder au nettoyage que si celui-ci devient absolument indispensable. Par contre, il impose de très fréquents appels à la routine. En effet, si ces appels ne se répètent pas assez souvent, Applesoft risque de prendre l'initiative des opérations et de mettre à nouveau votre patience à rude épreuve.

En fait, cette version du programme se prête essentiellement à des traitements pour lesquels l'appel de la routine peut s'effectuer à l'intérieur d'une boucle assez 'courte' (algorithmes de tri ...). Si vous souhaitez l'utiliser dans un tel contexte, chargez le programme et modifiez-le en tapant en mode moniteur "927C: A5 70 38".

Les quelques petits programmes présentés plus bas vous permettront de comparer l'efficacité des différentes routines de nettoyage.

Remarque : quand le programme fait appel à une routine assembleur, le temps de traitement indiqué comprend le temps nécessaire au chargement de cette routine à partir de la disquette.

1) Exemple numéro 1

```
1 HIMEM:37490
10 D$=CHR(4):DIM Z$(500)
20 FOR I=1 TO 500:Z$(I)=STR$(I):PRINT I:IF INT(I/100)=I/100 THEN X=FRE(0)
30 NEXT:END
```

a) Exécution du programme tel quel.

Temps de traitement (du 'RETURN' après RUN jusqu'à réapparition finale du curseur) : environ 1 minute 30.

b) Exécution du programme avec FRE(16).

Rajout d'une ligne : 15 PRINT D\$"BLOAD FRE(16)"

Remplacement de "X=FRE(0)" par "CALL 37500"

Temps de traitement : environ 30 secondes.

2) Exemple numéro 2

```
1 HIMEM:37490
5 BL$="":FOR I=1 TO 5:BL$=BL$+BL$:NEXT
10 D$=CHR$(4):DIM Z$(1000),Y$(1000),T$(1000)
20 FOR I=1 TO 50:Z$=STR$(I)+BL$:Z$(I)=Z$:Y$(I)=Z$:T$(I)=Z$:PRINT I:IF INT(I/2)=I/2
THEN X=FRE(0)
30 NEXT:END
```

Modification	Temps de traitement
A Programme tel quel	plus de 5 minutes
B Rajout de: 15 PRINT D\$"BLOAD FRE(16) "X=FRE(0)" devient "CALL 37500"	40 secondes
C Idem mais avec FRE(64)	30 secondes
D Programme A mais sans "X=FRE(0)" en ligne 20 (nettoyage automatique d'Applesoft)	45 secondes
E Programme D mais avec: 15 Print D\$"BLOAD FRE(16).TEST" et ajout de "CALL 37500" en 20. On teste donc la place mémoire libre	12 secondes
F Idem, mais avec la version FRE(64)	9 secondes

FRE (16)

```

927C- 4C 86 92 E5
9280- 6E C9 04 90 01 60 20 04
9288- 94 A6 73 A5 74 86 6F 85
9290- 70 A9 55 A2 00 85 5E 86
9298- 5F C5 52 F0 05 20 1F 93
92A0- F0 F7 A9 07 85 8F A5 69
92A8- A6 6A 85 5E 86 5F E4 6C
92B0- D0 04 C5 6B F0 05 20 12
92B8- 93 F0 F3 85 94 86 95 A9
92C0- 03 85 8F A5 94 A6 95 E4
92C8- 6E D0 07 C5 6D D0 03 4C
92D0- BA 93 85 5E 86 5F A0 00
92D8- B1 5E AA C8 B1 5E 08 C8
92E0- B1 5E 65 94 85 94 C8 B1
92E8- 5E 65 95 85 95 28 10 D3
92F0- 8A 30 D0 C8 B1 5E A0 00
92F8- 0A 69 05 65 5E 85 5E 90
9300- 02 E6 5F A6 5F E4 95 D0
9308- 04 C5 94 F0 BA 20 1F 93
9310- F0 F3 B1 5E 30 05 C8 B1
9318- 5E 30 03 4C A9 93 C8 B1
9320- 5E F0 F8 85 18 C8 B1 5E
9328- 85 19 C5 6F C8 B1 5E 85
9330- 1A E5 70 B0 E6 A5 19 CD
9338- 14 94 A5 1A ED 15 94 90
9340- 68 A5 19 C5 6D A5 1A E5
9348- 6E 90 D0 A2 21 CA CA A5
9350- 19 DD 13 94 A5 1A FD 14
9358- 94 90 F2 86 1B A2 03 BD
9360- 13 94 9D 11 94 BD 14 94
9368- 9D 12 94 BD 34 94 9D 32
9370- 94 BD 33 94 9D 31 94 BD
9378- 54 94 9D 52 94 BD 53 94
9380- 9D 51 94 E4 1B E8 E8 90
9388- D6 A6 1B A5 19 9D 13 94
9390- A5 1A 9D 14 94 A5 18 9D
9398- 34 94 A5 5E 9D 53 94 A5
93A0- 5F 9D 54 94 A5 8F 9D 33
93A8- 94 A5 8F 18 65 5E 85 5E
93B0- 90 02 E6 5F A6 5F A0 00
93B8- 60 60 A2 1F BD 34 94 A8
93C0- F0 F7 38 A5 6F FD 34 94
93C8- 85 6F A5 70 E9 00 85 70

```

```

93D0- BD 13 94 85 1C BD 14 94
93D8- 85 1D 88 C0 FF F0 06 B1
93E0- 1C 91 6F 90 F5 BD 33 94
93E8- 29 04 4A A8 C8 BD 53 94
93F0- 85 1C BD 54 94 85 1D A5
93F8- 6F 91 1C A5 70 C8 91 1C
9400- CA CA 10 B8 A2 1F A9 00
9408- 9D 34 94 9D 14 94 CA 10
9410- F7 4C 91 92 FF FF FF FF
... 9474- EA

```

E.A.O.

Enseignement Assisté par Ordinateur pour Ordinateurs

MOPPE D'ANFICO - SILEX - ITT 2020 - APPLE II - APPLE II +

de B A S I C

- . APPLESOFT
- . PALSOPT
- . SILROM

COURS EN FRANCAIS

- APPLE II (+ Carte APPLESOFT)
- APPLE II +
- APPLE III
- MOPPE D'ANFICO (+ APPLESOFT en ROM)
- ITT 2020 (+ Carte PALSOPT)
- SILEX (de Léonard)

(Sur DISQUETTES 5 1/4 Pouches - DOS = DOS 3.3 - Mémoire = 48 K.)

1/ COURS 1 (BASIC) - En Français

- . TRES PROGRESSIF - Ne nécessite aucune connaissance préalable en informatique.
- . Pour débutants et non débutants. TOUT Y EST EXPLIQUÉ.
- . 20 Leçons - Environ 10 à 12 Heures de cours.
- . 80 Exercices commentés, expliqués, résolus, exécutés.
- . 140 Questions notées sur 20, par groupes. - Réponses aux questions.
- . GRAPHISME BASSE et HAUTE RESOLUTION.
- . Défilement automatique du Cours avec arrêts et reprises possibles en cours de leçon.
- . De nombreux exercices peuvent être réexécutés autant de fois que vous le souhaitez. Ainsi, vous pouvez obtenir les mêmes résultats ou des résultats différents en faisant varier les données d'entrée.

A LA FIN DE CE COURS, VOUS SAUREZ PROGRAMMER.

2/ COURS 2 (BASIC +) - En Français

- . Philosophie générale identique à celle du COURS 1 (BASIC).
- . 25 Leçons - 12 Heures de cours minimum.
- . 120 Exercices commentés, expliqués, résolus, exécutés.
- . 160 Questions notées sur 20, par groupes - Réponses aux questions.

3/ CONTRATS-LOCATION du COURS BASIC pour:

- . Etablissements d'Enseignement.
- . Etablissements de Formation payante.
- . Centre de Recherches, Laboratoires, Centre d'Essais,....
- . Comité d'Entreprises.

PRIX (T.V.A. comprise)

- . COURS 1 (BASIC) 690 FF.
- . COURS 2 (BASIC +) 690 FF.

Prix donnés à titre indicatif, pouvant être modifiés sans préavis.

ANDRÉ P. FINUT
5 Allée BUFFON
91000 EVRY-CORCORNNES

R E V E N D E U R S,
C O N T A C T E Z - N ' O U S .

Pour ne pas vous tromper



IMAGOL
informatique

vous apporte

ASSISTANCE et CONSEIL

pour la

MISE EN SERVICE

de votre

1^{er} MICRO-ORDINATEUR

- Réalisation de logiciels spécifiques
- Organisation de sessions d'initiation à l'ordinateur et de formation à l'utilisation des programmes sur APPLE II et Goupil
- Location de matériel courte ou moyenne durée

Configuration	Week-end	1 ^{re} Semaine	Semaine sup.
Apple II 48 K/Floppy/mon.	350 F.H.T.	700 F.H.T.	400 F.H.T.
Apple II 48 K/2 Floppy/mon.	450 F.H.T.	900 F.H.T.	500 F.H.T.
Apple II Pascal 2 Floppy	550 F.H.T.	1 100 F.H.T.	600 F.H.T.
Imprimantes	—	5 % du P. Vte	3 % du P. Vte

- Contrats d'entretien de matériel



- Aide au choix de logiciels standards

Demandez une démonstration de nos Logiciels Standards Interactifs « Série S » pour APPLE :

(Tous compatibles entre eux et avec VISICALC)

- SUPER-GENEFICHE version PRO (65 % de capacité supplémentaire sur un APPLE 64 K, et de nombreuses extensions nouvelles)
- S/FACTURE (pour la facturation) • S/ETIQUETTE (pour le mailing)
- S/COURRIER (pour le courrier personnalisé et le traitement de texte)
- MODIFICHE (utilitaire de transformation des fichiers)



IMAGOL
informatique
1 à 5, rue Gutenberg
75015 PARIS (M^o Javel)
tél. : (1) 577-59-39

JBLOAD FRE(64)

JCALL -151

*927C.9417

```

927C- 4C 86 92 E5
9280- 6E C9 04 90 01 60 20 04
9288- 94 A6 73 A5 74 86 6F 85
9290- 70 A9 55 A2 00 85 5E 86
9298- 5F C5 52 F0 05 20 1F 93
92A0- F0 F7 A9 07 85 8F A5 69
92A8- A6 6A 85 5E 86 5F E4 6C
92B0- D0 04 C5 6B F0 05 20 12
92B8- 93 F0 F3 85 94 86 95 A9
92C0- 03 85 8F A5 94 A6 95 E4
92C8- 6E D0 07 C5 6D D0 03 4C
92D0- BA 93 85 5E 86 5F A0 00
92D8- E1 5E AA C8 B1 5E 08 C8
92E0- B1 5E 65 94 85 94 C8 B1
92E8- 5E 65 95 85 95 28 10 D3
92F0- 8A 30 D0 C8 B1 5E A0 00
92F8- 0A 69 05 65 5E 85 5E 90
9300- 02 E6 5F A6 5F E4 95 D0
9308- 04 C5 94 F0 BA 20 1F 93
9310- F0 F3 B1 5E 30 05 C8 B1
9318- 5E 30 03 4C A9 93 C8 B1
9320- 5E F0 F8 85 18 C8 B1 5E
9328- 85 19 C5 6F C8 B1 5E 85
9330- 1A E5 70 B0 E6 A5 19 CD
9338- 14 94 A5 1A ED 15 94 90
9340- 68 A5 19 C5 6D A5 1A E5
9348- 6E 90 D0 A2 81 CA CA A5
9350- 19 DD 13 94 A5 1A FD 14
9358- 94 90 F2 86 1B A2 03 BD
9360- 13 94 9D 11 94 BD 14 94
9368- 9D 12 94 BD 94 94 9D 92
9370- 94 BD 93 94 9D 91 94 BD
9378- 14 95 9D 12 95 BD 13 95
9380- 9D 11 95 E4 1B E8 E8 90
9388- D6 A6 1B A5 19 9D 13 94
9390- A5 1A 9D 14 94 A5 18 9D
9398- 94 94 A5 5E 9D 13 95 A5
93A0- 5F 9D 14 95 A5 8F 9D 93
93A8- 94 A5 8F 18 65 5E 85 5E
93B0- 90 02 E6 5F A6 5F A0 00
93B8- 60 60 A2 79 BD 94 94 A8
93C0- F0 F7 38 A5 6F FD 94 94
93C8- 85 6F A5 70 E9 00 85 70
93D0- BD 13 94 85 1C BD 14 94
93D8- 85 1D 88 C0 FF F0 06 B1
93E0- 1C 91 6F 90 F5 ED 93 94
93E8- 29 04 4A A8 C8 BD 13 95
93F0- 85 1C BD 14 95 85 1D A5
93F8- 6F 91 1C A5 70 C8 91 1C
9400- CA CA 10 B8 A2 79 A9 00
9408- 9D 94 94 9D 14 94 CA 10
9410- F7 4C 91 92 FF FF FF FF
  
```

TABLE

9594- EA

La leçon de calcul

ANNIE MAUFFREY

L'objectif de ce programme est d'initier et d'entraîner vos enfants à la multiplication, en la présentant comme à l'école. Le programme demande d'abord s'il doit fonctionner en mode démonstration (initiation) ou si l'utilisateur fait lui-même la multiplication (entraînement). Dans les deux cas, l'utilisateur choisit le nombre de chiffres du multiplicateur et du multiplicande.

En mode initiation, toutes les étapes de la multiplication sont expliquées et c'est l'utilisateur qui est maître du déroulement dans le temps de l'opération. Il lui faut, en effet, appuyer sur une touche pour poursuivre après chaque calcul élémentaire. En mode entraînement, l'opération se déroule comme à la main. Seul le décalage d'une ligne à l'autre est automatique. Au bout d'un nombre de fautes (ici fixé à 4) le programme se branche automatiquement en mode démonstration pour terminer l'opération.

Je n'ai pas cherché à optimiser la programmation, le seul objectif étant de rendre moins fastidieux l'apprentissage de la multiplication. On pourrait étendre ce programme dans plusieurs directions, par exemple : multiplication de nombres décimaux, limitation du temps de réponse, ou même multiplication dans une autre base... En revanche, c'est volontairement qu'il n'y a pas possibilité de contrôle a posteriori, ni système de notation, du travail de l'enfant pour qu'il le fasse de lui-même, avec plaisir et sans crainte de jugement.

```
10 DIM A$(9),B$(9),S$(18)
20 REM *****
   *
   *      MENU
   *
   *****
30 HOME : PRINT "PREFEREZ-VOUS:"
   : HTAB 5: VTAB 5: PRINT "1-Q
   UE JE FASSE LA MULTIPLICATIO
   N?"
40 HTAB 5: VTAB 10: PRINT "2-LA
   FAIRE VOUS-MEME?"
50 VTAB 15: PRINT "REPONDEZ PAR
   LE NUMERO DE LA QUESTION": VTAB
   20: GET A$: PRINT A$:IND% =
   VAL (A$): IF ASC (A$) < 49
   OR ASC (A$) > 50 THEN 50
60 HOME : PRINT "LE MULTIPLICAND
   E ET LE MULTIPLICATEUR SON
   T DES NOMERES DE 1 A 9 CHIFF
   RES. A VOUS DE CHOISIR"
70 VTAB 10: PRINT "NOMBRE DE CHI
   FFRES DU MULTIPLICANDE: ";: GOSUB
   540:LA% = VAL (Z$)
80 VTAB 15: PRINT "NOMBRE DE CHI
   FFRES DU MULTIPLICATEUR: ";:
   GOSUB 540:LB% = VAL (Z$)
90 REM *****
   * CHOIX DES NOMERES *
   * ET *
   * ECRITURE SUR ECRAN *
   *****
100 FOR I = 0 TO 18:S$(I) = 0: NEXT
   :FAUT = 0: HOME :VS = 3:PH =
   25: FOR I = LA% TO 1 STEP -
   1
110 GOSUB 400: IF CH = 0 AND I =
   1 THEN 110
120 A$(I) = CH: GOSUB 420: NEXT I
130 VS = 4: VTAB VS: HTAB 22 - LB
   %: PRINT "X ";:PH = 25: FOR
   I = LB% TO 1 STEP - 1
140 GOSUB 400: IF CH = 0 AND I =
   1 THEN 140
150 B$(I) = CH: GOSUB 420: NEXT I
   :PV = 5: GOSUB 440
160 REM *****
   * CALCUL ET *
   * AFFICHAGE LIGNES *
   *****
170 FOR I = LB% TO 1 STEP - 1:P
   H = 25 - LB% + I:PV = PV + 1
   : IF B$(I) = 0 THEN PV = PV -
   1: GOTO 260
180 IF I < LB% THEN GOSUB 630
190 FOR J = LA% TO 1 STEP - 1
200 MULT% = A$(J) * B$(I):S$(I +
   J) = S$(I + J) + MULT%:MULT%
   = MULT% + RET%:RET% = MULT%
   / 10:CH = MULT% - RET% * 10
   :VS = PV:EN% = 0: IF IND% =
   1 THEN GOSUB 580: GOTO 220
210 GOSUB 480
```

```

220 GOSUB 420: IF RETZ = 0 THEN
250
230 CH = RETZ:VS = 1: IF J < > 1
    THEN ENZ = LBZ - I: GOSUB 4
    80: GOSUB 420:PH = PH + 1: GOTO
    250
240 VS = PV: GOSUB 480: GOSUB 420

250 NEXT J
260 GOSUB 650: IF INDZ = 1 AND I
    < > 1 THEN VTAB 21: SPEED=
    80: PRINT "JE DECALE D'UN CH
    IFFRE VERS LA GAUCHE": FOR L
    = 1 TO 100: NEXT L: SPEED=
    255
270 RETZ = 0: FOR K = 24 TO 24 -
    LAZ - LBZ STEP - 1: VTAB 1:
    HTAB K: PRINT " ": NEXT K: NEXT
    I: IF LBZ = 1 THEN 360
280 PV = PV + 1: GOSUB 440
290 REM *****
    * CALCUL ET *
    * AFFICHAGE SOMME *
    *****

300 PV = PV + 1:PH = 25:ENZ = 0
310 FOR I = LAZ + LBZ TO 1 STEP
    - 1:RETZ = SZ(I) / 10:CH =
    SZ(I) - RETZ * 10:SZ(I - 1) =
    SZ(I - 1) + RETZ:
320 VS = PV: IF I < > 1 OR SZ(I)
    < > 0 THEN GOSUB 480: GOSUB
    420
330 NEXT I: VTAB 21: IF INDZ = 1
    THEN SPEED= 80: PRINT "J'A
    I FAIT LA SOMME DES ";LBZ;"
    LIGNES OBTENUES": SPEED= 255

340 IF INDZ = 2 THEN HTAB 29: VTAB
    15: FLASH : PRINT "*****
    ": HTAB 29: VTAB 16: PRINT "
    * BRAVO *": HTAB 29: VTAB 17
    : PRINT "*****": NORMAL

350 IF INDZ = 1 THEN GOSUB 650:
    VTAB 20: HTAB 15: PRINT "A
    VOUS MAINTENANT"
360 GOSUB 650: VTAB 23: HTAB 10:
    PRINT "ENCORE UNE? ( O OU N
    )";: GET B$: IF B$ = "O" THEN
    30
370 IF B$ < > "N" THEN 360
380 END
390 REM *****
    * *
    * SOUS-ROUTINES *
    * *
    *****

400 REM ** CHOIX CHIFFRE **
410 CH = INT ( RND (1) * 10): RETURN
420 REM ** AFFICHAGE **

430 PH = PH - 1: HTAB PH + ENZ: VTAB
    VS: PRINT CH: RETURN
440 REM ** AFFICHAGE LIGNE **
450 VTAB PV: FOR I = 24 - LAZ -
    LBZ TO 24: HTAB I: PRINT "-"
    ;: NEXT I: RETURN
460 REM *****
    * TEST *
    *****

470 VTAB 1: HTAB PH: PRINT RETZ:
    RETURN
480 REM ***** TEST
490 IF INDZ = 1 THEN RETURN
500 HTAB PH + ENZ - 1: VTAB VS: GOSUB
    540
510 IF VAL (Z$) = CH THEN RETURN

520 FAUT = FAUT + 1: IF FAUT > 3 THEN
    INDZ = 1: VTAB 21: SPEED= 40
    : PRINT "VOUS AVEZ FAIT QUAT
    RE FAUTES ": VTAB 23: HTAB 7
    : PRINT "JE VAIS FINIR VOTRE
    MULTIPLICATION";: SPEED= 25
    5: RETURN
530 GOTO 500
540 REM *****
    * ENTREE ET *
    * ECRITURE RESULTAT*
    *****

550 GET Z$
560 IF ASC (Z$) < 48 OR ASC (Z
    $) > 57 THEN 550
570 PRINT Z$: RETURN
580 REM *****
    * EXPLICATIONS *
    *****

590 HTAB 29: VTAB 6: PRINT SPC(
    8): HTAB 29: VTAB 9: PRINT SPC(
    10)
600 W = AZ(J) * BZ(I): HTAB 29: VTAB
    6: PRINT BZ(I);"X";AZ(J);"=
    ";W: HTAB 29: VTAB 9: PRINT
    W;"+";MULTZ - W;"=";MULTZ
610 HTAB 29: VTAB 12: PRINT "JE
    POSE ";CH: HTAB 32: VTAB 15:
    PRINT "ET": HTAB 29: VTAB 1
    8: PRINT "JE RETIENS ";RETZ
620 GOSUB 650: VTAB 21: PRINT "P
    OUR QUE JE CONTINUE, "; VTAB
    23: HTAB 7: PRINT "APPUYER S
    UR UNE TOUCHE": GET A$: RETURN

630 REM ** TRACAGE DE ... **
640 FOR K = 1 TO LBZ - I: HTAB 2
    5 - K: VTAB PV: PRINT ".": NEXT
    K: RETURN
650 REM ** EFFACE BAS ECRAN **
660 FOR H = 1 TO 40: POKE 1615 +
    H,160: POKE 1871 + H,160: NEXT
    : RETURN

```

Banc d'essai d'utilitaires de documentation

HERVE THIRIEZ

1. Présentation générale.

Dans cet article, nous nous intéressons à des utilitaires de mise au point et documentation de programmes. Toutes les semaines, de nouveaux utilitaires arrivent sur le marché. Nous nous limiterons en règle générale à l'analyse de ceux ayant déjà atteint un certain niveau de diffusion ou, le cas échéant, d'un intérêt tel qu'il faut les faire connaître le plus rapidement possible.

Les trois utilitaires retenus pour cette analyse sont, dans l'ordre alphabétique, Apple Doc, Dakin, et le DOS Tool Kit, qui permettent tous trois la référence croisée de variables, c'est-à-dire la liste des numéros d'instructions dans lesquelles chaque variable est citée. Cet article fait suite à ceux parus dans POM'S No 1 au sujet des progiciels PLE et CRAE.

Les utilitaires ont été comparés à l'aide d'un petit programme servant à imprimer une par une des adresses sur des enveloppes. Dans le cas d'Apple Doc, la liste et les références croisées de variables et de numéros d'instruction seront fournies à titre d'illustration. Suivent les références croisées fournies par Dakin.

2. Apple Doc.

Ce progiciel possède quatre fonctions principales :

- . référence croisée des noms de variables ! sur tout
- . référence croisée des numéros d'instruction ! ou partie
- . liste de programme ! du programme
- . recherche et modification éventuelle de chaîne ! concerné

Apple Doc fonctionne aussi bien en DOS 3.2 qu'en 3.3. Il suffit pour passer à la version 3.3 de MUFFINiser la version antérieure.

Apple Doc se distingue par la possibilité de définir, et mémoriser éventuellement, une description en toutes lettres de la signification d'une variable ou d'une ligne. Celle-ci apparaît alors dans la référence croisée, comme nous pouvons le voir dans la liste de notre petit programme de démonstration.

Dans la référence croisée de variables, Apple Doc fait apparaître le nombre d'indices de chaque tableau, alors que Dakin se contente de faire la distinction entre variables indicées ou non.

L'intérêt principal d'Apple Doc réside dans le fait qu'il fonctionne sous la forme de fichiers EXEC. Il arrive souvent que l'on désire utiliser les fonctions d'un tel progiciel pendant la mise au point d'un programme; avec les fichiers EXEC, il est possible d'utiliser Apple Doc pendant que le programme à documenter réside en mémoire.

L'inconvénient majeur d'Apple Doc, en ce qui me concerne, est la mauvaise présen-

tation des résultats : les 80 colonnes de l'imprimante sont mal utilisées par la référence croisée; il n'y a pas un bon remplissage de la ligne. Mais ce n'est pas visible sur le court programme documenté à titre d'exemple. Le résultat est plus long et moins agréable à lire que celui obtenu avec Dakin, surtout si l'on n'utilise pas la possibilité d'associer une définition à chaque variable ou numéro de ligne.

Cet inconvénient peut être tourné en modifiant les programmes d'Apple Doc, mais il me semble plus équitable, lors de la comparaison de programmes, de comparer les versions sous lesquelles ces programmes sont commercialisés, et non telles que l'on peut les améliorer.

Remarque sur Apple Doc : Apple Doc écrase en partie les zones mémoire utilisées par le PLE et CRAE. Après avoir utilisé Apple Doc, il faut donc recharger ces deux utilitaires. Vous pouvez créer un fichier EXEC à cet effet, qu'il suffira d'appeler après avoir travaillé avec Apple Doc.

3. Dakin.

Le point commun entre Apple Doc et Dakin est l'existence de programmes destinés à établir les références croisées de lignes et de variables, et la liste d'un programme. A part ces fonctions, Dakin apporte de nombreux autres services, de qualité et d'intérêt très inégaux.

Il y a deux disquettes Dakin vendues ensemble dans la version DOS 3.2. La première est plus ancienne, ce qui explique l'existence du programme TEXT FILE COPY, alors que le COPIER (qui se trouve sur l'autre disquette) fait tout et mieux. Malheureusement, Dakin ne marche en DOS 3.3 ni sous BASICS, ni transformé par MUFFIN. Il vous reste, si vous passez au DOS 3.3, à acheter la version 3.3, ou à découvrir dans diverses revues ou par vous-même les modifications nécessaires.

Voici la description des diverses possibilités de Dakin, au-delà de la documentation de programme :

PEEKER : analyse du contenu d'enregistrements sélectionnés d'un fichier à accès direct. On peut se construire sans effort un petit programme Basic aussi puissant et nettement plus flexible d'utilisation.

CRUNCHER : le plus original et le plus utile des programmes du Dakin. Cet utilitaire sert à comprimer au maximum un programme en ôtant les remarques et en regroupant les lignes rattachables.

TEXT FILE COPY : inutile.

PROMPTER : exemple de routine d'input généralisé. De toute façon, chaque utilisateur a intérêt à construire la routine correspondant à ses besoins propres.

CALCULATOR : exemples de calculs à grande précision.

DISKETTE COPY : inutile.

SCREEN PRINTER : copie d'écran texte.

ARRAY EDITOR : éditeur de fichier séquentiel. Peut être intéressant, en particulier pour analyser et modifier un fichier EXEC.

COPIER : copie de fichier de quelque nature que ce soit. Il vaut mieux utiliser FID, fourni avec la disquette Master du DOS 3.3, et disponible en 3.2 (par DEMUFFIN).

PATCHER : inutile.

En matière de documentation de programme, Dakin est très lent, comparé à Apple Doc, surtout dans la version DOS 3.2. En effet, il faut dans ce cas pour l'utiliser booter, ou faire "RUN UT0001" ou "RUN UT0002" suivant le service

demandé. Afin de lister un programme et d'en obtenir la référence croisée de lignes et de variables, il faut, dans l'ordre :

- . charger le menu de la disquette 1 de Dakin
- . charger le programme de liste
- . charger le programme à documenter
- . recharger le menu pour arrêter le travail sur la disquette 1
- . charger le menu de la disquette 2 pour appeler le programme de référence croisée de variables
- . charger le programme de référence croisée de variables
- . recharger le programme à documenter
- . recharger le menu pour appeler le programme de référence croisée de numéros de lignes
- . charger le programme de référence croisée de numéros de ligne
- . recharger le programme à documenter
- . et, si on veut terminer sans utiliser le RESET, recharger le menu pour lui dire que l'on a fini.

Par comparaison, avec Apple Doc, le programme à documenter est chargé une fois pour toutes. On exécute alors consécutivement EXEC LISTER, EXEC VARDOC et EXEC LINEDOC. Un gain de temps gigantesque ...

Remarques sur le CRUNCHER : cet utilitaire élimine seulement les remarques si elles sont séparées et correspondent à un numéro de ligne non référencé par un GOTO ou GOSUB. Pour une remarque référencée, il restera (heureusement) le numéro de ligne et REM. Le REM demeure aussi quand la remarque était incorporée dans une séquence d'instructions séparées par des ":",

Limitations de Dakin :

- . très lent pour la documentation, surtout en DOS 3.2. En DOS 3.3, tout se trouve sur une disquette unique.
- . le LISTER n'est pas sur la même disquette que les autres programmes de documentation en DOS 3.2.
- . l'écriture des dates n'est pas libre; il faut absolument respecter le mode américain d'écriture des dates, sauf si l'on se trouve dans les douze premiers jours du mois !
- . il faut rentrer les paramètres de l'imprimante à chaque exécution.
- . certains programmes ne peuvent imprimer un nom de programme différent de celui sous lequel il est mémorisé sur disquette.

Limitation conjointe de Apple Doc et Dakin : pour la référence croisée de variables comme de numéros d'instruction, ces deux programmes ignorent tout ce qui est consécutif à un DATA, sous le même numéro d'instruction. On peut le constater sur notre exemple à l'aide de la ligne 150 : ni les variables, ni la référence à l'instruction 60 n'apparaissent dans les références croisées.

4. DOS Tool Kit.

DOS Tool Kit ne se limite pas à la documentation de programmes. Il comporte en fait trois parties fort différentes :

a) l'assistance à la programmation avec des fonctions de renumérotation (toutes les possibilités du RENUMBER, moins le bug classique), la numérotation automatique en option, une fonction de compression de remarques, divers gadgets et une référé-

rence croisée de variables.

b) un générateur de caractères haute résolution très flexible, permettant entre autres de superposer un texte quelconque à des images haute résolution. Passionnant et très riche.

c) un assembleur-éditeur que nous n'avons pas eu le temps de tester, mais dont on ne nous a dit que du bien à ce jour.

L'assistance à la programmation fournie par DOS Tool Kit suppose que celui-ci réside en mémoire en permanence durant la mise au point d'un programme Applesoft. Ne pouvant cumuler, je préfère franchement utiliser à cet effet le P.L.E. et le C.R.A.E. dont les services en matière d'assistance à la programmation me semblent sensiblement plus utiles.

Ceci n'enlève rien à l'intérêt des autres possibilités de ce progiciel, mais nous débordons avec le générateur de caractères haute résolution et l'assembleur éditeur le thème qui nous intéresse dans cet article.

5. Conclusion.

Pour ma part, j'utilise Apple Doc en documentation pendant la mise au point des programmes. Dans cette phase, il vaut mieux gagner du temps qu'économiser le papier ou fignoler la présentation. Dakin me sert à produire la référence croisée du programme, et à le compresser en utilisant le CRUNCHER.

APPLE DOC

```
*****
*                                     *
*           ENVELOPPE                 *
*                                     *
*           4 DECEMBRE 1981           *
*                                     *
*****
```

```
10 GOTO 150
20 Z$ = ""
30 GET R$: IF ASC (R$) = 13 THEN RETURN
40 IF ASC (R$) < 32 OR ASC (R$) > 90 THEN 30
50 PRINT R$;Z$ = Z$ + R$; GOTO 30
60 TEXT : HOME : FOR L = 1 TO 6: PRINT N$(L) TAB( 13)": "R$(L);: HTAB 15:
  GOSUB 20: IF Z$ < > "" THEN R$(L) = Z$
70 PRINT : PRINT : NEXT : INVERSE : PRINT "ENREGISTREMENT CONFIRME";: NORM
  : INPUT " : ";R$: IF LEFT$ (R$,1) < > "0" THEN 60
80 R$(4) = R$(4) + " - " + R$(5);R$(5) = CHR$ (13)
90 PRINT : PRINT "ALLUMER L'IMPRIMANTE": GET R$:D$ = CHR$ (13) + CHR$ (
  4): PRINT : INPUT "ENVELOPPE LONGUE ? ";R$: PRINT D$"PR#1"
100 IF LEFT$ (R$,1) = "0" THEN PRINT : PRINT : PRINT
110 FOR L = 1 TO 4:R$(L) = LEFT$ (BL$,25 + 15 * (R$ = "0")) + R$(L): NEXT
  :R$(6) = " " + R$(6)
120 FOR L = 1 TO 9: PRINT : NEXT : FOR L = 1 TO 6: PRINT R$(L): NEXT
130 PRINT D$"PR#0": PRINT : INPUT "UNE AUTRE ENVELOPPE ? ";R$: IF LEFT$
  (R$,1) < > "0" THEN END
140 FOR L = 1 TO 6:R$(L) = "": NEXT : GOTO 60
150 DATA NOM,ADRESSE 1,ADRESSE 2,CODE POSTAL,VILLE,PAYS:BL$ = " ": FOR L =
  1 TO 6: READ N$(L):BL$ = BL$ + BL$: NEXT : GOTO 60
```

```

*****
*
*           ENVELOPPE           *
*
*  -->TABLE OF VARIABLES<--  *
*
*****

```

BL\$ - BLANC
110

D\$ - CHR\$(13)+CHR\$(4)
90 90 130

L - INDICE COURANT
60 60 60 60 110 110 110 120 120 120 140 140

N\$(*) - NOMS DES RUBRIQUES
60

R\$ - REPONSE
30 30 40 40 50 50 70 70 90 90 100 110 130 130

R\$(*) - REPONSES AUX RUBRIQUES
60 60 80 80 80 80 110 110 110 110 120 140

Z\$ - CHAINE FOURNIE EN REPONSE
20 50 50 60 60

END OF VAR. LIST

```

*****
*
*           ENVELOPPE           *
*
*  -->TABLE OF LN# XREFS<--  *
*
*****

```

20 - ROUTINE DE LECTURE GENERALISEE
60

30
40 50

60 - TRAITEMENT PRINCIPAL
70 140

150 - INITIALISATION
10

END OF LN# LIST

XX THE VARIABLE CROSS REFERENCE XX

PROGRAM NAME...ENVELOPPE
DATE...04/12/81

EL\$
110

D\$
90 90 130

L
60 60 60 60 110 110 110 120 120 120 140 140

N\$<
60

R\$
30 30 40 40 50 50 70 70 90 90 100 110 130 130

R\$<
60 60 80 80 80 80 110 110 110 110 120 140



Z\$
20 50 50 60 60

DAKIN

XX THE LINE CROSS REFERENCE XX

PROGRAM NAME...ENVELOPPE
DATE...04/12/81

20: 60
30: 40 50
60: 70 140 150
150: 10

		A TOULOUSE	
APPLE 		LE PLUS MODULAIRE DU DISQUE 5P AU DISQUE DUR LE PLUS CONNU	MICRO ORDINATEUR SHOP SOUBIRON SA. 9 RUE KENNEDY TOULOUSE TEL. 21.64.39./21.04.57. TELEX. LPS.INF 521075 F
LOGICIELS	PET 2001 CBM3001 CBM8001	PARTICULIER SCIENTIFIQUE GESTION	100M² D'EXPOSITION
STOCKS PAYE FACTURATION TRAITEMENT DE TEXTE COMPTABILITE GENERALE GESTION DES COMMANDES BIJOUTIER, MEDICAL, DENTISTE GESTION DE FICHIERS ASSURANCES	HP 85 	DU SCIENTIFIQUE AU BUREAU D'ETUDE	S.A.V. DEPANNAGE SOUS 48 H REGION MIDI PYRENEES
FORMATION	PCC 2000 ISTC 5000	LIBRAIRIE SPECIALISE INFORMATIQUE	VIDEO GENIE EG 3003 DU GENIE POUR PETITS ET GRANDS
STAGES D'INITIATION BASIC, PASCAL, FORTRAN	SHARP PC 1211 MZ 80	DU MOINS CHER AU PARTICULIER	
CHOIX EN MICRO INFORMATIQUE			

Une incursion dans les mystères du DOS

JEAN-FRANCOIS DUVIVIER

Voici quelques modifications qui vous permettront de disposer d'un DOS personnalisé, avec certaines particularités bien utiles. Ces changements peuvent s'appliquer indifféremment à un DOS 3.2 ou 3.3 mais les adresses ne sont valables que pour un 48 K. Dans les autres cas, vous devez retrancher à toutes les adresses citées la valeur hexadécimale \$4000 pour un 32 K, \$8000 pour un 16 K.

1. Elimination de la pause lors d'un catalogue (lorsque celui-ci est trop long pour tenir dans un écran).

Normal: *ADA4: 16 (Poke 44452,22)
Modifié: *ADA4: FF (Poke 44452,255)

Lors d'un catalogue, le DOS tient à jour dans la case mémoire \$B39D le nombre de lignes à éditer avant d'effectuer une pause. Cette valeur est initialisée avec le contenu de l'adresse ADA4, normalement 22. Lorsqu'on l'initialise avec une valeur supérieure à 120, on supprime totalement la pause, le catalogue ne pouvant contenir que 120 lignes en DOS 3.3 et 96 en DOS 3.2.

2. Arrêt du catalogue par la touche "ESC" lors d'une pause.

Normal: *AE39: 20 0C FD
 *BCE0: zone inutilisée
Modifié: *AE39: 20 E0 BC
 *BCE0: 20 0C FD C9 9B D0 05 68
 68 4C 7F B3 60

Le saut à la pause du catalogue est remplacé par un saut à l'adresse \$BCE0. Le programme placé à cette adresse attend alors qu'une touche soit appuyée et la compare avec "ESC". Si cette touche est différente, le catalogue est continué. Dans le cas où l'on a appuyé sur "ESC", le programme rend la main à l'utilisateur en effectuant un saut à la routine appelante.

3. Affichage des programmes effacés lors d'un catalogue.

Normal: *ADD9: 30 4A
Modifié: *ADD9: EA EA

Le DOS teste pour chaque programme un octet du "directory" afin de savoir s'il est effacé. La modification annule le branchement conditionnel qui suit en le remplaçant par deux instructions NOP (No Operation). Dans le cas d'un programme effacé, une colonne à droite contiendra un caractère inversé, ce qui permet de les identifier rapidement. A noter que toute trace est perdue dans le directory lorsque le DOS a recopié un autre programme à la place de celui qui a été effacé.

4. Bootez sur un programme binaire.

Normal: *9E42: 06

Modifié: *9E42: 34 pour BRUN
*9E42: 14 pour EXEC

Sur un DOS normal, le programme HELLO ne peut être qu'en BASIC. La modification vous permet d'avoir un programme HELLO en binaire ou sous forme d'un fichier. La procédure est la suivante: initialiser la disquette avec un programme BASIC HELLO, après avoir fait la modification; sauver ou copier le fichier binaire (BINAIRE), détruire HELLO et changer le nom de BINAIRE en HELLO (RENAME BINAIRE, HELLO).

5. Suppression du NOMON lors de l'entrée dans le DOS.

Normal: *9DD7: 8D 5E AA
Modifié: *9DD7: EA EA EA

Lors de l'entrée à chaud dans le DOS (3DOG qui est équivalent à 9DBFG pour un 48 K) ou lors de l'appui sur un RESET (cas de l'autostart), le DOS est réinitialisé et l'option NOMON est mise. La modification ci-dessus permet de garder l'état du MON antérieur, empêchant la remise à jour de l'octet \$AA5E qui indique les commandes MON C,I,0 en action.

6. Annulation de la commande INIT.

Normal: *A884: 49
Modifié: *A884: 5B

Il peut être intéressant, surtout avec un DOS modifié, d'annuler la commande INIT afin d'éviter de recopier le DOS sur une nouvelle disquette. D'autre part, une fois la commande INIT enlevée, on peut se servir d'une partie de la page \$B6 afin d'y stocker d'autres routines. Le DOS maintient à jour un dictionnaire des commandes afin de pouvoir les reconnaître (voir l'article de C. Guérin). La modification change le premier "I" de INIT par un "J", caractère non accessible au clavier. La commande INIT ne sera alors plus reconnue.

7. Secteurs libres à chaque commande CATALOG.

Normal: *A884: 49
*ADC3: 20 2F AE
*B600: partie de INIT
Modifié: *A884: 5B
*ADC3: 20 00 B6
*B600: A2 0C 20 4A F9
A9 00 85 40 85 41 A0 C8
18 B9 F2 B3 F0 0E 0A 90
FB 48 E6 40 D0 02 E6 41
68 18 90 F0 88 D0 E9 A6
40 A5 41 AC 00 E0 C0 20
D0 07 20 1B E5 20 2F AE
60 20 24 ED 20 2F AE 60

La longueur de la routine impose de la placer en \$B600 et donc d'annuler la commande INIT. Cette routine est appelée après l'inscription du libellé DISK VOLUME et avant le saut de deux lignes précédant la liste des programmes. Elle compte à partir des informations disponibles dans le buffer VTOC (\$B3F3 à B47B) les secteurs disponibles puis les affiche sur écran. Le déroulement normal du

catalogue est ensuite repris.

8. Chargement du DOS modifié sur disquette.

Il peut être intéressant, une fois que vous avez modifié votre DOS, de pouvoir en disposer chaque fois que vous bootez. La première solution consiste à construire un programme HELLO avec toutes les modifications que vous souhaitez. Une fois ce programme exécuté, votre DOS en RAM est modifié.

La seconde solution consiste à initialiser une disquette avec votre DOS modifié. Celui-ci sera alors incorporé sur les pistes 0,1 et 2 de la disquette et disponible chaque fois que vous booterez. A noter que l'on ne peut charger ainsi la dernière modification, qui neutralise la commande INIT.

La troisième solution est de recopier le DOS en RAM sur les pistes 0,1,2 de la disquette, mais sans toucher aux autres pistes et en laissant par conséquent vos programmes intacts. Cette méthode vous permet de modifier le DOS chargé sur des disquettes déjà utilisées. Il suffit pour cela de taper les lignes suivantes:

```
*300: A9 9D 8D BC B5 20 C2
      B7 20 4A B7 60
*AA5D: 09
*AA5F: 0
*AA75: C8 C5 CC CC CF (il s'agit en fait du nom du programme sur lequel
                        on veut booter. S'il fait moins de 4 caractères,
                        compléter avec des espaces, dont le code ASCII
                        est A0).
```

Le lancement se fait par 300G ou par un CALL 768 à partir du BASIC.

Trucs et astuces

Programme mystère ...

Le premier lecteur qui nous fera parvenir l'explication complète de ce que fait le programme suivant gagnera un abonnement à quatre numéros de Pom's. Ce programme requiert le Basic Integer en ROM, c'est-à-dire la carte Integer ou la carte langage.

```
>LIST
20733 CALL -936: TAB 19: VTAB 13
20734 DEVINETTE= ASC("POMS")* PEEK (-1823)/ PEEK (-1595)*POMS ^ POMS
20735 POKE PEEK (-7782),12: POKE PEEK (-4710),184
20736 POKE 2069, PEEK ( RND (4096))
20737 PRINT "POMS"
20738 POKE POMS,POMS
20739 POP
```



apple en province

Afin de mieux tirer profit de toutes les possibilités d'**APPLE**, et de ses extensions. Loin des contacts anonymes, pour un rapport plus humain, des revendeurs régionaux spécialisés vous proposent des logiciels sur mesure, standards, ou d'apprentissage. Un service technique avant et après-vente. N'hésitez pas à contacter le revendeur le plus proche pour un conseil ou un renseignement

Anancy/Favergeres

74210

Bayonne

64100

Bordeaux

33000

Clermont-Ferrand

63000

Epernay

51200

Lyon

69003

Marseille 2

13000

Montpellier

34000

Nancy/Laxou

54250

Orléans

45000

Perpignan

66000

Rouen

76100

Strasbourg

67000

Toulon

83100

Valenciennes

59300

EUROPROCESS

Siège social : Doussard

(50) 44.31.12

LE CALCUL INTEGRAL

3, rue Aristide-Briand

(59) 55.43.47

BOUTISOFT 33

9, rue de la Lande

(56) 91.55.08

NEYRIAL

5, Bd Desaix

(73) 35.02.70

MAGENTA GESTION

7, av. A.-Thévenet Magenta

(26) 53.19.93

CIRCE

9, rue P.-Florence

(78) 54.31.95

ORDITEL

Siège social : BELCODEN

(42) 04.44.00

IFI-MICRO INFORMATIQUE

9-12, rue Castilhon

(67) 58.58.28

CENTER

SEMITEC

69, rue Mareville

(8) 340.43.38

AMC

13, rue des Minimes

(38) 62.62.58

MAB

2, place de Catalogne

(68) 34.04.46

CONSEIL COMPUTER

20, quai Cavalier-de-la-Salle

(35) 63.36.06

CILEC

18, quai Saint-Nicolas

(88) 37.31.61

S I A

Lepaillon, avenue de Brunet

(94) 23.74.30

MICROMEGA

38, rue de Famars

(27) 46.89.22

Si cette publicité vous intéresse, contactez le
CALCUL INTEGRAL

Applications de graphiques haute resolution

GUY MATHIEU

A. Quelques rappels

Le graphique dit de "haute résolution" présente sur Apple II les caractéristiques suivantes :

- . 280 x 192 sans texte, ou 280 x 160 avec 4 lignes de texte.
- . 8 couleurs, dont deux noirs et deux blancs. Il existe sur le marché des cartes permettant d'obtenir 16 couleurs en HGR mais cette option n'est pas compatible avec l'Applesoft.
- . 2 pages écran dites HGR et HGR2. Voir en ce qui concerne l'organisation de ces pages l'article de J.F. Duvivier "Les Adresses Du Graphique" dans le No 1 de POM'S.

Un point donné de l'écran ne peut avoir n'importe quelle couleur : les points des colonnes paires peuvent être bleus ou rouges et ceux des colonnes impaires peuvent être jaunes ou verts. Le blanc n'est qu'une illusion d'optique provoquée par des doublets (paires de points contigus).

Voici un programme surprise :

```
LIST
10 REM :COUCOU-COULEURS
20 HOME
30 INPUT "COULEUR CHOISIE ? (0 A 7) ";C
40 INPUT "X INITIAL (0 A 279) ";X0
50 HGR : HCOLOR= C
60 FOR X = X0 TO 279 STEP 2
70 HPLOT X,80 + 50 * SIN (X / 6) TO X,80
80 NEXT
```

Vous pourrez, selon les valeurs de C et de X0 obtenir du vert, du bleu, du rouge ... ou rien du tout. Mais jamais vous n'obtiendrez du blanc.

B. La réalisation des graphiques HGR

Il existe deux grandes voies pour obtenir des graphiques HGR.

L'une consiste à utiliser des périphériques spécifiques, avec leurs cartes interfaces et leurs logiciels: table à digitaliser, crayon lumineux. Nous n'aborderons pas cette possibilité ici.

La seconde consiste à écrire des programmes graphiques. Il existe sur Apple II deux séries différentes d'instructions graphiques.

La première série permet d'obtenir directement sur écran des traits ou des points;

on utilise dans ce cas les instructions :

```
HCOLOR
HPLOT X,Y
HPLOT X1,Y1 TO X2,Y2
```

La seconde série utilise une table de "shapes", formes digitalisées et préalablement stockées en mémoire. Outre les paramètres de recherche d'une forme F dans une table, on utilise les instructions :

```
ROT (rotation, de 0 à 64)
SCALE (échelle, entiers de 1 à 255)
DRAW F AT X,Y
XDRAW F AT X,Y
HCOLOR
```

N.B.: HCOLOR n'a pas d'effet direct sur l'instruction XDRAW qui dessine la forme dans la couleur complémentaire à celle du fond: noir sur blanc, bleu sur jaune, etc. On voit la puissance de cette instruction: redessiner une forme en utilisant la même instruction XDRAW une deuxième fois revient à effacer la forme sans toucher à l'image de fond. L'effacement par le noir ... efface tout.

C. Les programmes graphiques

Qu'il s'agisse de dessin direct sur l'écran ou de l'utilisation de shapes, les instructions utilisent un certain nombre de paramètres (couleurs, coordonnées, ...). Trois moyens viennent à l'esprit pour fournir au programme ces paramètres.

Le premier consiste à les introduire directement dans le programme :

```
HGR
HPLOT 10,10 TO 260,150
```

trace une ligne allant de X=10 et Y=10 (proche de l'angle haut gauche) à X=260 et Y=150 (proche de l'angle bas droit)

On tracera couramment par cette méthode un cadre, des lignes de rappel ...

La seconde consiste à calculer les paramètres par programme. Le programme:

```
10 HGR
20 X0=140 : Y0=96 : R=48
30 FOR I=0 TO 6.28 STEP 0.1
40 HPLOT X0 + R*COS(I), Y0 + R*SIN(I)
50 NEXT I
```

permet de tracer par points un cercle au centre de l'écran.

Il existe un troisième moyen, qui consiste à passer par un artifice pour entrer en machine, pendant le déroulement du programme, les divers paramètres. On peut pour cela utiliser le clavier: il suffit de décider arbitrairement qu'une certaine touche représente un certain mouvement, et de "lire" cette touche. On travaillera alors à partir d'un point de coordonnées X, Y et on calculera un nouveau point de coordonnées X+dX, Y+dY (dX et dY pouvant prendre par exemple, les valeurs -1, 0 ou 1, selon la touche pressée).

On peut aussi utiliser les poignées (paddles) destinées aux jeux. Et ceci peut se faire de deux façons différentes.

On peut transformer les valeurs lues sur les deux poignées en coordonnées. Par

exemple :

```
X = PDL (0) * 279/255
```

```
Y = PDL (1) * 191/255
```

Les coordonnées X et Y sont alors introduites dans une instruction telle que HPLOT, DRAW ou XDRAW.

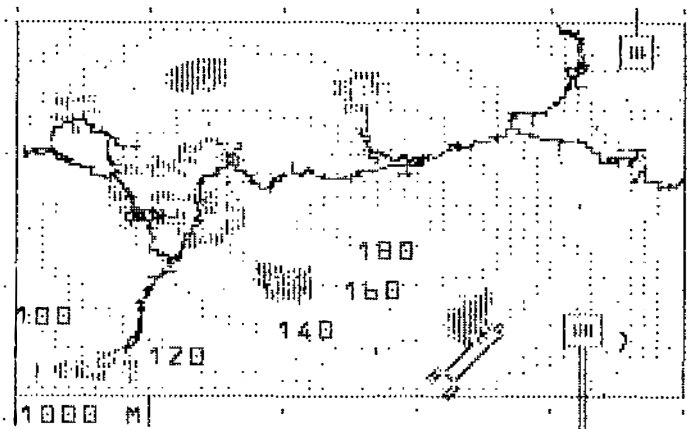
On peut aussi travailler comme avec le clavier en faisant mouvoir un point à partir de sa position actuelle, avec par exemple :

```
dX = INT ( PDL(0)/100 ) -1
```

```
dY = INT ( PDL(1)/100 ) -1
```

dX et dY prendront alors les valeurs -1, 0 ou 1.

Cette dernière méthode a été utilisée pour obtenir la carte ci-dessous.



D. L'utilisation des shapes

Le mode "shape" possède plusieurs vertus.

- * Une shape même complexe est tracée sur l'écran de façon apparemment instantanée.
- * Une "table de shapes" permet de disposer sur appel simple, d'un lot de formes classiques relativement à un problème donné (composants électroniques, lettres ...)
- * Les options ROT et SCALE permettent la mise en oeuvre d'effet de mouvement et de loupe.
- * L'instruction XDRAW, grâce à ses possibilités d'effacement, permet de mouvoir des shapes, par exemple pour désigner un point, simuler un mouvement mécanique, etc.

Le programme ci-après est destiné à faire tourner une forme sur elle-même en fonction de son déplacement. La disquette fournie contient la carte dessinée et une shape représentant un mobile que l'on peut déplacer sur la carte en utilisant les poignées de jeu.

Ce programme de démonstration ne permet que huit sens de déplacement et huit

orientations du mobile. Il serait facile de généraliser.

```

JLIST
10 HOME
20 REM *****
   * PROGRAMME COPIC *
   * DEPLACEMENT MOBILE *
   *****

30 HIMEM: 36000: REM PROTECTION
   TABLE DE SHAPES
40 D$ = CHR$(4)
50 PRINT D$"BLOADMOBILE": REM C
   HARGEMENT DE LA TABLE DE SHA
   PE
60 HGR
70 PRINT D$"BLOADCARTE": REM CH
   ARGEMENT DE LA CARTE
80 REM *****
   * INITIALISATION *
   *****

90 POKE 232,160: POKE 233,140
100 ZG = 20:ZD = 260:ZH = 10:ZB =
   150: REM LIMITATION DES DEP
   LACEMENTS
110 ZX = 200:ZY = 122: REM COORD
   ONNES POINT DE DEPART
120 ZR = 0:ZS = 3
130 ZN = 2: REM SHAPE NO 2 - ROT
   ATION 0 - ECHELLE 3
140 GOTO 360
150 REM ** LECTURE PADDLE **
160 ZP = PDL (ZP)
170 ZP = INT ( SQR (ZP) / 5.5) -
   1: REM DONNE A ZP LES VALEU
   RS -1,0,1 QUAND LA LECTURE D
   U PDL ZP VARIE DE 0 A 255

180 RETURN
190 REM ** TRACE MOBILE **
200 ROT= ZS
210 XDRAW ZN AT Z6,Z7
220 ROT= ZR
230 XDRAW ZN AT ZX,ZY

240 Z6 = ZX:Z7 = ZY
250 Z5 = ZR
260 RETURN
270 REM ** BORNES COORDONNEES **

280 IF ZY < ZH THEN ZY = ZH
290 IF ZY > ZB THEN ZY = ZB
300 IF ZX > ZD THEN ZX = ZD
310 IF ZX < ZG THEN ZX = ZG
320 RETURN
330 REM ** LECTURE POUSSOIR **
340 IF PEEK ( - 16287) < = 127
   THEN RETURN : REM SI ON A
   APPUYE SUR LE POUSSOIR DU P
   DL 0, ALORS Z=I (DRAPEAU) --
   > FIN
350 Z = 1: RETURN
360 REM *****
   * PROGRAMME *
   * PRINCIPAL *
   *****

370 Z = 0
380 ROT= ZR: SCALE= ZS
390 GOSUB 230
400 REM ** DEPLACEMENT **
410 ZP = 0: GOSUB 150:Z8 = ZP
420 ZP = 1: GOSUB 150:Z9 = ZP
430 ZX = ZX + 2 * Z8:ZY = ZY - 2 *
   Z9: REM DEPLACEMENT DE 2 EN
   2
440 GOSUB 270
450 IF Z8 = 1 AND Z9 = 0 THEN ZR
   = 0: GOTO 470
460 ZR = (2 * Z9 + Z8 * Z9) * 8 +
   32: REM 1090 ET 1100 DONNEN
   T LES VALEURS 0,8,16,...,56 A
   ZR SELON LE SENS DU DEPLA
   CEMENT
470 GOSUB 200
480 GOSUB 330
490 IF Z = 0 THEN 410
500 END

```

E. Quelques conseils pour travailler en HGR

* HGR, contrairement à HGR2, permet d'afficher 4 lignes en bas de l'écran: commentaires, questions à poser, etc. L'adressage de ces lignes se fait par des VTAB à partir de 21, et leur effacement par un HOME classique.

* Si le programme de traitement est volumineux (plus de 6K), la page écran HGR est

partiellement occupée par le programme. Il faut alors employer HGR2.

* Pour éviter que les variables et les tableaux ne détruisent la page haute résolution utilisée, il convient de placer un LOMEM juste au-dessus (soit 16385 ou 24577 selon que l'on utilise HGR ou HGR2).

* Pour éviter que les variables alphanumériques ne détruisent les tables de shapes, placer un HIMEM (36000 dans notre exemple).

* Pour éviter les messages d'erreur quand les coordonnées sont le résultat de calculs, limiter par programme les dites coordonnées (de 0 à 279 pour X et de 0 à 159 ou 191 pour Y).

* Attention : HGR et HGR2 sont des instructions qui effacent l'écran. Pour conserver une image en vue de la faire réapparaître, utiliser la technique des POKES (voir POM'S No 1, "Les adresses du graphique").

* Pour sauver une image sur disque

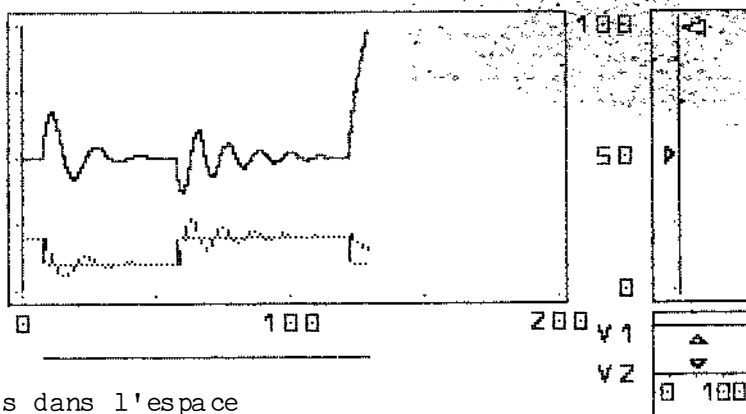
- . depuis HGR : BSAVE NOM, A\$2000, L\$2000
- . depuis HGR2 : BSAVE NOM, A\$4000, L\$2000

F. Quelques exemples de graphique haute-résolution

N.B. En noir et blanc, ces exemples normalement très colorés se trouvent dévalorisés.

a. Un panneau de contrôle avec enregistreur à trois voies (extrait d'un cours de régulation):

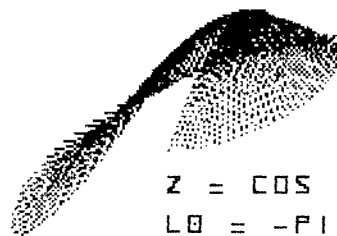
- cadres par l'instruction HPLOTT à paramètres dans le programme;
- courbes calculées dans un système d'équations différentielles;
- aiguilles des indicateurs en shapes.



b. Deux surfaces dans l'espace

Coordonnées cartésiennes

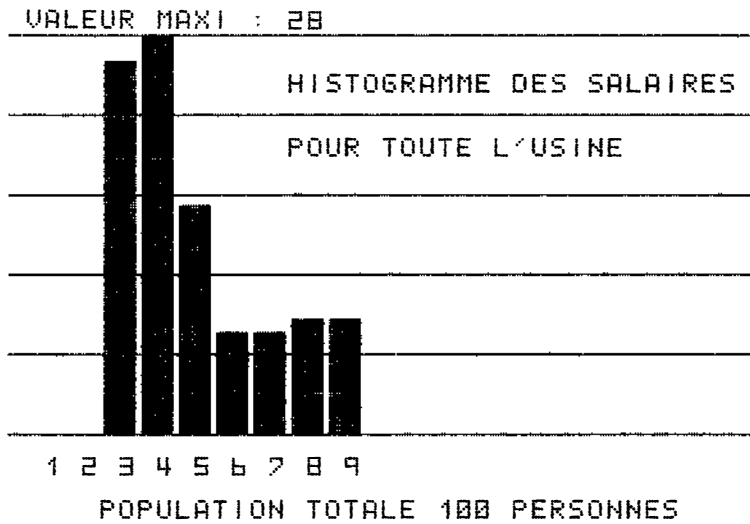
Coordonnées semi-polaires



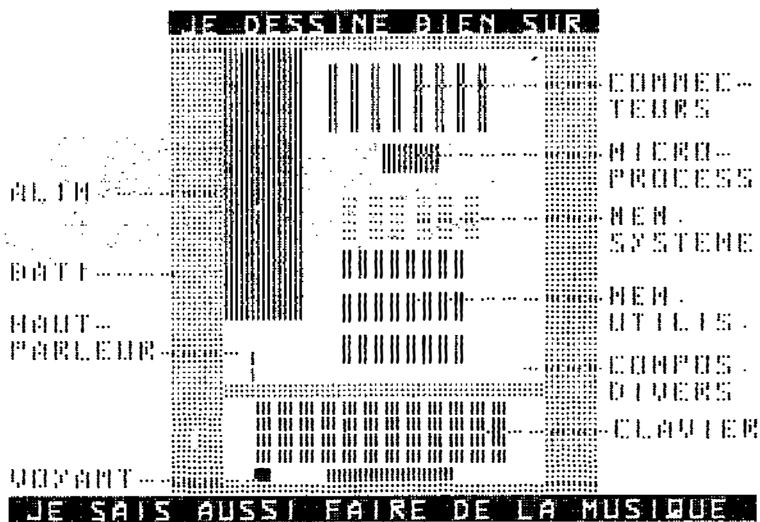
$$\begin{aligned}
 Z &= \sin X \times \sin Y \\
 X0 &= 0 & XN &= 4 \text{ PI} \\
 Y0 &= -2 \text{ PI} & YN &= 0
 \end{aligned}$$

$$\begin{aligned}
 Z &= \cos L \\
 L0 &= -\text{PI}/2 & LN &= \text{PI}/2
 \end{aligned}$$

c. Histogramme commenté. Cet histogramme a été obtenu à partir du programme STAGE: statistiques d'aide à la gestion. Les textes sont obtenus par des POKES directement dans la page graphique haute résolution.



d. Structure de l'ordinateur Apple II (extrait d'un audio-visuel MICRO). Les textes sont obtenus en employant des sous-programmes de dessins de caractères.



Les programmes HELLO des disquettes POM'S utilisent également des graphiques à haute résolution. A cet égard, n'hésitez pas à nous envoyer vos contributions pour les prochains numéros: tout est permis, même l'utilisation du jeu de la vie.

Les limites des editeurs de texte sur Apple

HERVE THIRIEZ

L'Apple n'a pas été conçu au départ comme matériel de traitement de texte. Cependant, de nombreux utilisateurs de l'informatique individuelle ne peuvent envisager d'acheter un matériel spécialisé pour la rédaction de leurs textes. Quelles sont les limitations auxquelles ils sont confrontés en utilisant un Apple, et dans quelle mesure peuvent-ils les tourner ?

1. Limitations de l'Apple.

Dans sa configuration normale, l'Apple II ne possède pas de clavier AZERTY et fonctionne sur un écran en 40 colonnes. De plus, les caractères minuscules et accentués n'apparaissent pas naturellement sur l'écran, et peu d'imprimantes savent imprimer la totalité des caractères accentués, quelle que soit la gymnastique à laquelle on se prête - mais il s'agit là d'une limitation relative aux imprimantes et non au processeur.

2. Principaux logiciels de traitement de texte.

Il existe une grande variété de logiciels de traitement de texte sur Apple; citons les plus renommés et les mieux diffusés dans la CEE, dans l'ordre alphabétique : Apple Pie, Apple Writer (I et II), Easywriter, Magic Window, Superscribe, Super-Text, Wordstar (avec carte Z80), Write-On. Nous nous limiterons dans notre analyse comparative à trois de ces systèmes.

3. Critères d'évaluation comparative.

Bien entendu, un certain nombre de fonctions essentielles sont assurées par chacun de ces logiciels, avec des commandes diverses et des temps de réponse variables. Ces fonctions de base sont :

- . mise à zéro de la mémoire
- . entrée d'un texte
- . sauvegarde sur disquette
- . chargement d'un texte en mémoire
- . modification du texte résident (ajouts et/ou suppressions)
- . recherche et remplacement de chaînes
- . impression sous formats variables

Nos critères généraux d'évaluation sont les suivants : la qualité de la documentation, la facilité d'apprentissage, la commodité d'utilisation et le temps de réponse. Le premier tableau évalue les systèmes retenus pour l'analyse selon leurs principales fonctions. Le second fournit une appréciation générale selon les quatre critères retenus ci-dessus.

4. Analyse commentée de trois systèmes.

a) Apple Writer.

Les avantages particuliers d'Apple Writer sont sa grande commodité d'utilisation, sa rapidité de fonctionnement et sa facilité de modification. En effet, de nombreux articles ont été consacrés dans diverses revues à des "patches" d'Apple Writer permettant de le rendre compatible avec telle ou telle carte 80 colonnes,

avec l'impression de caractères spéciaux, ... L'article de Jean-Louis Meillaud, dans le prochain numéro de Pom's, montrera d'ailleurs comment créer le 'ç' et le 'ù'. Remarquons enfin l'existence d'un programme d'apprentissage des commandes Apple Writer fourni avec la disquette d'origine.

Par contre, au titre des inconvénients, citons l'impossibilité de voir sur écran comment le texte que l'on rédige sera mis en page : on est forcé de tirer un texte "à blanc" pour constater la mise en page découlant des paramètres de formatage décidés au moment du lancement. Autre inconvénient : Apple Writer est lent à l'impression.

Bug : quand on demande à Apple Writer de justifier un texte en utilisant la totalité des 80 colonnes, il déplace le dernier caractère de la ligne en première colonne : il faut donc ne pas justifier de texte en 80 colonnes !

b) Easywriter.

Easywriter a été écrit en Forth, un langage très puissant dont nous parlerons un de ces jours. Easywriter permet l'incorporation de marqueurs ignorés à l'impression, afin de pouvoir aisément effectuer des transferts de blocs de texte. Le mode de stockage sur disquette d'Easywriter autorise la mémorisation sur un système en DOS 3.2 de 15% de caractères en plus par rapport aux autres logiciels. Une touche spéciale fait en outre apparaître sur l'écran la liste des instructions, ce qui facilite grandement l'apprentissage des commandes. Easywriter étant offert en standard sur le micro IBM, on peut s'attendre rapidement à des versions améliorées.

L'inconvénient majeur d'Easywriter réside dans l'impossibilité dans laquelle se trouve l'utilisateur, si compétent soit-il (Jean-François Duvivier a déclaré forfait), de pénétrer dans le code afin d'effectuer des modifications.

Bug : dans la version dont nous disposons, en cas de manipulation de plusieurs disquettes, Easywriter réécrit le catalogue au moment de la sauvegarde, sans le relire préalablement. Si l'on a changé de disquette depuis le dernier chargement, le catalogue de l'ancienne disquette écrase celui de la nouvelle, et le fichier s'inscrit dans une zone qui était libre sur l'ancienne disquette !

c) Magic Window.

A notre avis, de tous les systèmes disponibles sur le marché actuellement, Magic Window possède le meilleur rapport performance/prix. La grande originalité de ce progiciel tient à la particularité suivante : au lieu d'avoir un curseur mobile se déplaçant ligne par ligne sur l'écran, le système fournit sur l'écran une fenêtre se promenant dans les quatre directions à l'intérieur de la page formatée pour l'impression. L'utilisateur visualise ainsi par morceaux (choisis) la page telle qu'elle sortira sur l'imprimante. Les différents menus sont très faciles à utiliser.

L'inconvénient majeur de Magic Window tient à l'impossibilité de remplacer dans un texte une chaîne par une autre : il faut se positionner par une fonction de recherche à l'endroit visé, puis modifier à la main la chaîne. Nous attendons avec impatience la prochaine version, dans laquelle ce problème doit être résolu. S'il est très facile de réduire la largeur de la page en cours de création de texte, il est par contre mal commode de l'augmenter; il faut donc prévoir de créer et sauvegarder un texte en 80 colonnes, et le réduire au moment de l'impression, si l'on désire éventuellement l'utiliser dans divers formats.

d) Autres systèmes.

Nous avons commenté de façon détaillée ces trois logiciels que nous connaissons bien, mais nous pouvons aussi vous faire part de nos impressions sur d'autres systèmes.

. Wordstar - A tout seigneur, tout honneur ! Ce système est le plus répandu mondialement, du fait de son appartenance à la programmation CPM. Il est donc disponible sur tous les systèmes micro-informatiques équipés d'une carte Z80 (Softcard) et d'une carte 80 colonnes. Wordstar comporte une grande variété de commandes, qui ont inspiré les concepteurs de la plupart des autres systèmes.

. Superscribe présente son texte en haute résolution, ce qui permet d'afficher 70 caractères par ligne. Il est alors possible d'avoir une vue générale du texte. A noter que l'utilisation de la haute résolution ne porte pas atteinte à la rapidité de fonctionnement.

. Katexte est l'un des rares traitements de texte écrits en Pascal, et profite donc de la puissance de l'éditeur de ce langage. Notons aussi qu'il s'agit là de la réalisation d'une SSCI française, la société Ka.

5. Comparaison des commandes.

Mode d'évaluation :

++ : excellent
 + : bon
 0 : moyen
 D : disponible

Nous n'avons fait apparaître dans le tableau ci-dessous que les critères selon lesquels les trois systèmes différaient.

	Apple Writer	Easywriter	Magic Window
Disquette protégée	Non	Oui	Oui
Zone mémoire disponible	31 K	11 K	32 K
Suppression de mot	D		
Suppression de paragraphe	D		
Capacité de stockage	+	++	+
Menu des commandes		D	
Rapidité de la frappe	++	+	++
Rapidité des commandes	++	0	++
Tabulation			D
Justification au centre	D		D
Sauvegarde partielle	D		D
Sortie en fichier texte			D
Chargement partiel			D

6. Comparaison globale selon des critères généraux.

	Apple Writer	Easywriter	Magic Window
Qualité de la documentation	+	++	++
Facilité d'apprentissage	++	+	++
Commodité d'utilisation	+	+	++
Temps de réponse	++	+	++

7. Conclusion.

Initialement, seul Magic Window était conçu pour travailler avec une carte 80 colonnes. Les versions récentes d'Apple Writer et Easywriter fonctionnent aussi avec des cartes 80 colonnes spécifiques. Des systèmes à base de clavier AZERTY voient le jour actuellement. Nous avons utilisé pour la rédaction du numéro 2 le système IEF avec Apple et Olympia, cette dernière pouvant servir d'entrée (facultative) ou de sortie. Le système de traitement de texte de l'Apple ///, un Apple Writer très amélioré, fera l'objet d'une analyse dans le prochain numéro.

Un exemple de hard copy

CHRISTIAN GUERIN

Les imprimantes sont grosso modo de trois types principaux, selon leurs capacités graphiques. Certaines ne peuvent imprimer que des caractères normaux; c'était le cas le plus fréquent jusqu'à ces derniers mois. D'autres, telles que l'Epson Type I ou la série OKI, sont capables d'imprimer des caractères semi-graphiques, que nous décrivons dans le paragraphe suivant. Les dernières, enfin, impriment des graphiques à haute résolution; c'est le cas de la Centronics 739, des Epson Type II, 82 et 100, de la Paper Tiger, de la Seiksha et de la Silentye (entre autres).

Les caractères semi-graphiques sont divisés en six cases, 3 en hauteur et 2 en largeur, dont chacune peut être noire ou blanche. Il y a donc 64 caractères semi-graphiques possibles. L'association de ces caractères permet de reconstituer une image.

Le programme que nous présentons ici permet d'effectuer une "hard copy" d'une des deux pages haute résolution de l'Apple sur une imprimante semi-graphique du type OKI Microline 80.

Vous pouvez utiliser ce programme en mode direct, ou à partir d'un autre programme en procédant de la manière suivante :

POKE 771,32 - édition de la page 1
ou POKE 771,64 - édition de la page 2

POKE 772,0 - mode normal
ou POKE 772,255 - mode inverse

CALL 768

Ce programme est écrit pour fonctionner avec un type de carte interface spécifique implantée dans le slot 1.

Si la commande de votre carte est différente, ou si vous ne l'avez pas mise dans le slot 1, il suffit de modifier le sous-programme OUT en \$3C0.

Si le programme fonctionne, mais que votre imprimante ne saute pas de ligne, modifiez-le de la manière suivante :

CALL -151
3AA : A9 0A 20 C0 03
retour à Applesoft et sauvegarde

Imprimante Epson

Afin d'adapter ce programme au cas de l'Epson, il faut ajouter \$20 aux caractères graphiques et modifier les ordres de formatage de l'imprimante :

- . mode condensé
- . mode interligne serré
- . retour en fin de programme

Note de la rédaction

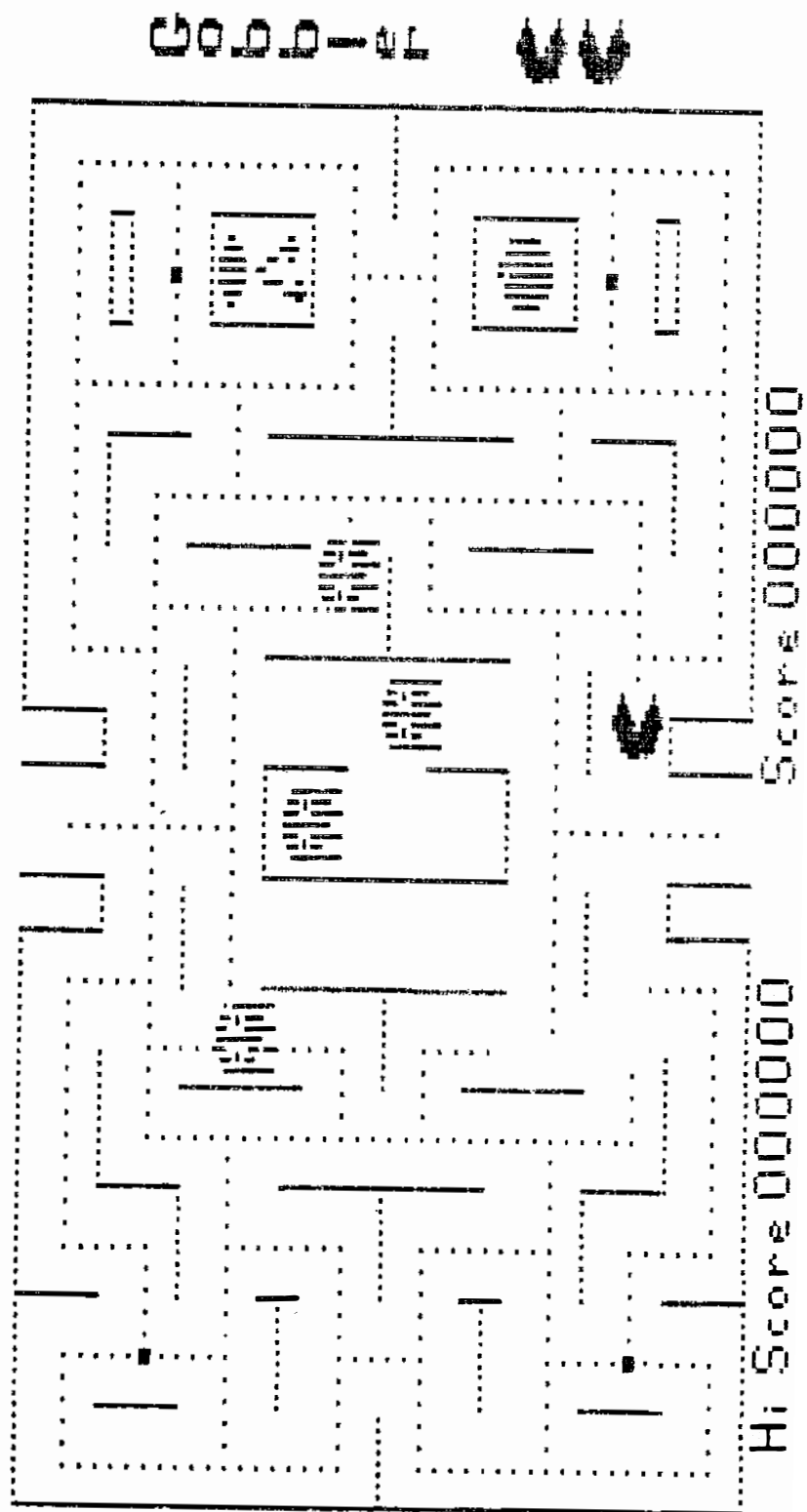
Ne disposant pas d'une OKI, nous n'avons pu tester ce programme. Par contre, nous avons pu admirer l'exemple de copie d'écran réalisé par Christian Guérin sur la page graphique du jeu Snoggle.

ICALL -151

*300.3CD

```

0300- 4C 05 03 20 00 20 B0 03
0308- A9 38 20 B9 03 A9 1D 20
0310- C0 03 A0 01 84 FD 88 A9
0318- BF 85 FE A5 FD 48 98 48
0320- A2 03 20 71 03 C6 FE 20
0328- 71 03 E6 FE 06 FD 10 05
0330- A9 01 85 FD C8 CA D0 EA
0338- A5 FF 4A 4A 4D 04 03 C0
0340- 28 D0 02 29 03 09 80 20
0348- C0 03 A5 FE C9 01 F0 0B
0350- 68 A8 68 85 FD C6 FE C6
0358- FE D0 C0 68 68 20 A0 03
0360- C0 28 D0 B3 A9 36 20 B9
0368- 03 A9 1E 20 C0 03 4C B0
0370- 03 A5 FE 48 29 C0 85 06
0378- 4A 4A 05 06 85 06 68 85
0380- 07 0A 0A 0A 26 07 0A 26
0388- 07 0A 66 06 A5 07 29 1F
0390- 0D 03 03 85 07 B1 06 25
0398- FD 18 F0 01 38 66 FF 60
03A0- A9 20 20 C0 03 A9 0D 20
03A8- C0 03 EA EA EA EA EA 60
03B0- A0 0A 20 A0 03 88 D0 FA
03B8- 60 48 A9 1B 20 C0 03 68
03C0- 48 AD 90 C0 29 02 F0 F9
03C8- 68 8D 91 C0 60 38
    
```



**END OF PASS 1
 **END OF PASS 2

```

0800      1  ;
0800      2  ;
0800      3  ;
0800      4  ;
0800      5  ;
0800      6  ;
0800      7  ;
0800      8  ;
0800      9  ;
0800     10  ;
0800     11  ;
0800     12  DL      EPZ $6      ;POINTEUR DE DEBUT DE LIGNE
0800     13  ;
0800     14  PPO     EPZ $FD     ;POSITION DU POINT DANS UN OCTET
0800     15  NL      EPZ $FE     ;NUMERO DE LA LIGNE
0800     16  CG      EPZ $FF     ;CARACTERE GRAPHIQUE
0800     17  ;-----
0300     18          ORG $300
0300     19  ;
0300     20  4C0503   JMP DEBUT
0303     21  ;
0303     22  20      PAGE     BYT $20      ;PAGE A EDITER (1=$20, 2=$40)
0304     23  00      MODE     BYT $0      ;0=NORMAL , $FF=INVERSE
0305     24  ;
0305     25  20B003  DEBUT   JSR SAUT      ;SAUT DE 10 LIGNES
0308     26  A938          LDA #$38
030A     27  20B903  JSR ESC      ;PASSAGE EN MODE INTERLIGNE SERRE
030D     28  A91D          LDA #$1D
030F     29  20C003  JSR OUT      ;PASSAGE EN MODE CARACTERE CONDENSE
0312     30  ;-----
0312     31  ;          DEBUT DU PROGRAMME DE HARD COPIE
0312     32  ;-----
0312     33  A001          LDY #1
0314     34  84FD          STY PPO      ;ON COMMENCE SUR LE PREMIER POINT
0316     35  88          DEY          ;DE LA PREMIERE LIGNE
0317     36  ;
0317     37  A9BF     OKI.1  LDA #!191   ;1 LIGNE = 191 POINTS
0319     38  85FE          STA NL
031B     39  ;
031B     40  A5FD     OKI.2  LDA PPO
031D     41  48          PHA          ;ON SAUVEGARDE LA POSITION ACTUELLE
031E     42  98          TYA
031F     43  48          PHA
0320     44  A203          LDX #3      ;3 FOIS 2 POINTS PAR CARACTERE
0322     45  ;
0322     46  207103  OKI.3  JSR ADL
0325     47  C6FE          DEC NL      ;ON PREND 2 POINTS L'UN
0327     48  207103  JSR ADL      ;EN DESSOUS DE L'AUTRE
032A     49  E6FE          INC NL
032C     50  06FD          ASL PPO
032E     51  1005          BPL OKI.4
0330     52  A901          LDA #1      ;PUIS ON PASSE AU
0332     53  85FD          STA PPO     ;POINT D'A COTE
0334     54  C8          INY
0335     55  ;
0335     56  CA     OKI.4  DEX
0336     57  D0EA          BNE OKI.3
  
```



```

0338          58 ;
0338 A5FF     59     LDA CG
033A 4A       60     LSR
033B 4A       61     LSR
033C 4D0403   62     EOR MODE      ;MODE INVERSE OU NORMAL
033F          63 ;
033F C028     64     CPY #!40
0341 D002     65     BNE OKI.5      ;UNE SEULE LIGNE DANS
0343 2903     66     AND #3        ;LE DERNIER CARACTERE
0345          67 ;
0345 0980     68     OKI.5  ORA #!80    ;LES CARACTERES GRAPHIQUES DOIVENT
0347 20C003   69     JSR OUT      ;AVOIR LE BIT 7 A 1
034A          70 ;
034A A5FE     71     LDA NL
034C C901     72     CMP #1
034E F00B     73     BEQ OKI.6
0350          74 ;
0350 68       75     PLA
0351 A8       76     TAY
0352 68       77     PLA          ;ON RESTORE LA POSITION POUR
0353 85FD     78     STA PFO      ;PASSER AU CARACTERE SUIVANT
0355 C6FE     79     DEC NL
0357 C6FE     80     DEC NL
0359 D0C0     81     BNE OKI.2
035B          82 ;
035B 68       83     OKI.6  PLA
035C 68       84     PLA          ;FIN D'UNE LIGNE
035D          85 ;
035D 20A003   86     JSR FIN.L
0360 C028     87     CPY #!40      ;FIN DU HARD COPIE
0362 D0B3     88     BNE OKI.1
0364          89 ;-----
0364          90 ;     FIN DU PROGRAMME
0364          91 ;-----
0364 A936     92     LDA #!36
0366 20B903   93     JSR ESC          ;PASSAGE EN MODE INTERLIGNE NORMAL
0369 A91E     94     LDA #!1E
036B 20C003   95     JSR OUT      ;PASSAGE EN MODE CARACTERE NORMAL
036E 4CB003   96     JMP SAUT      ;SAUT DE 10 LIGNES
0371          97 ;
0371          98 ;     *   CALCUL DE L'ADRESSE DE DEBUT   *
0371          99 ;     *           D'UNE LIGNE           *
0371         100 ;     *-----
0371         101 ;
0371 A5FE     102     ADL     LDA NL          ;EN FONCTION DU NUMERO
0373 48       103     FHA          ;DE CETTE LIGNE
0374 29C0     104     AND #!C0
0376 8506     105     STA DL
0378 4A       106     LSR
0379 4A       107     LSR
037A 0506     108     ORA DL
037C 8506     109     STA DL
037E 68       110     PLA
037F 8507     111     STA DL+1
0381 0A       112     ASL
0382 0A       113     ASL
0383 0A       114     ASL
0384 2607     115     ROL DL+1
0386 0A       116     ASL

```

```

0387 2607      117      ROL DL+1
0389 0A        118      ASL
038A 6606      119      ROR DL
038C A507      120      LDA DL+1
038E 291F      121      AND #$1F
0390 0D0303    122      ORA PAGE      ;ET DE LA PAGE
0393 8507      123      STA DL+1
0395 B106      124      LDA (DL),Y    ;ON RECUPERE L'OCTET DANS LA LIGNE
0397 25FD      125      AND PFO      ;PUIS LE POINT DANS L'OCTET
0399 18        126      CLC
039A F001      127      BEQ ADL.1     ;QUE L'ON UTILISE POUR CONSTITUER
039C 38        128      SEC
039D 66FF      129      ADL.1 ROR CG ;LE CARACTERE GRAPHIQUE
039F 60        130      RTS
03A0          131      ;
03A0          132      ;
03A0          133      ;
03A0 A920      134      FIN.L LDA #' '
03A2 20C003    135      JSR OUT
03A5 A90D      136      LDA #'M'-$40 ;RETURN
03A7 20C003    137      JSR OUT
03AA EA        138      NOP          ;SI L'OKI N'EST PAS DANS LE MODE
03AB EA        139      NOP          ; "AUTO LINE FEED"
03AC EA        140      NOP          ;METTRE ICI : LDA #'J'-$40
03AD EA        141      NOP          ;
03AE EA        142      NOP
03AF 60        143      RTS
03B0          144      ;
03B0          145      ;
03B0          146      ;
03B0 A00A      147      SAUT LDY #!10
03B2 20A003    148      SAUT.1 JSR FIN.L
03B5 88        149      DEY
03B6 D0FA      150      BNE SAUT.1
03B8 60        151      RTS
03B9          152      ;
03B9          153      ;
03B9          154      ;
03B9 48        155      ESC PHA
03BA A91B      156      LDA #!27
03BC 20C003    157      JSR OUT
03BF 68        158      PLA
03C0          159      ;
03C0          160      ;
03C0          161      ;
03C0 48        162      OUT PHA
03C1 AD90C0    163      READY LDA $C090
03C4 2902      164      AND #2
03C6 F0F9      165      BEQ READY
03C8 68        166      PLA
03C9 8D91C0    167      STA $C091
03CC 60        168      RTS
03CD          169      ;
03CD          170      ;
                                171      END

```

***** END OF ASSEMBLY

Le Pascal - Analyse du directory

MICHEL CRIMONT

Les manuels PASCAL fournis avec l'Apple donnent peu de renseignements sur le répertoire du disque, sa situation et sa composition. Or, le but du programme qui va suivre est d'assembler sur un seul disque les répertoires de tous les disques que l'on possède afin de réaliser un catalogue général permettant des recherches, des sélections en fonction du nom des disques, du type ou du genre des programmes, etc.

Pour que ce report se fasse automatiquement, il est tout d'abord nécessaire de savoir lire ce répertoire. Il commence au bloc 2 de chaque disquette et peut être défini comme un enregistrement.

```
REPertoire: record
    VOL      :DISQUE;
    CATALOG :array!1..77! of FICHER
end;
```

Il comporte en effet une première partie, que je nomme VOL, contenant les renseignements relatifs au disque (nom, date de création,...) et une seconde appelée CATALOG se présentant comme un TABLEAU de 77 enregistrements contenant chacun des renseignements relatifs à un fichier du disque. On retrouve la limitation à 77 du nombre maximum de fichiers par disque que l'on déclarera comme une constante MAXFILE=77.

Le type DISQUE comporte:
le numéro du bloc de début du répertoire, de type entier
le numéro du bloc de fin de répertoire, de type entier
le nom du disque de type String!7!
le nombre de blocs du disque, de type entier
la date de création du disque, de type date
et quelques valeurs appelées plus loin V1,V2,V3,V4 et que je n'ai pas identifiées, mais que l'on peut considérer de type entier.

Le type String!7! est une chaîne de caractères que l'on peut considérer comme PACKED ARRAY!0..7! of CHAR, l'octet 0 contenant la longueur de la chaîne (d'où la limitation à 255 caractères au maximum de ce type), et les octets suivant les caractères eux-mêmes.

Le type DATE est défini lui-même comme un enregistrement.

```
DATE =packed record
    MOIS :1..12;
    JOUR :1..31;
    AN   :1..99
end;
```

On obtient donc :

```

DISQUE =record
    DEBREP      :INTEGER;
    FINREP      :INTEGER;
    V1          :INTEGER;
    NOMDISQUE   :STRING!7!;
    NBRBLOC     :INTEGER;
    NBRFICHE    :INTEGER;
    V2          :INTEGER;
    DATEDISQUE :DATE;
    V3,V4       :INTEGER
end;

```

Chaque élément de CATALOG se définit par un enregistrement comportant:
le numéro du bloc de début de fichier, de type entier
le numéro du bloc suivant la fin de fichier, de type entier
le type de fichier, de type 0..7
le nom du fichier, de type STRING!15!
le nombre de bytes restant dans le dernier bloc, de type entier
la date du fichier de type date.

Le type de fichier est représenté par un chiffre de 0 à 7: 0=RIEN, 1=BAD, 2=CODE, 3=TEXT, 4=INFO, 5=DATA, 6=GRAPH, 7=PHOTO. Les types 4,6 et 7 ne sont pas utilisés.

On obtient alors:

```

FICHIER =record
    DEBFICHIER :INTEGER;
    FINFICHIER :INTEGER;
    TYPEFICHIER:0..7;
    NOMFICHIER :STRING!15!;
    DERNBLOC   :INTEGER;
    DATEFICHIER:DATE
end;

```

Il reste à définir le volume du disque qui doit être lu, que l'on appellera NUMLECTEUR. La procédure de lecture des blocs du répertoire sera donc simplement: UNITREAD(NUMLECTEUR,REPertoire,2048,2).

A noter que l'on peut remplacer 2048, qui représente le nombre d'octets transférés par l'appel de la fonction SIZEOF(REPertoire) qui renvoie le nombre d'octets du type répertoire. De même, un répertoire modifié (par exemple la date de mise à jour d'un fichier) doit être replacé sur le disque par UNITWRITE(NUMLECTEUR,REPertoire,2048,2).

La lecture et l'écriture se feront par précaution sous forme de fonctions renvoyant un booléen vrai seulement si la lecture ou l'écriture s'est effectuée correctement. Ce booléen servira de test au programme; ce sont les fonctions LIREREP et ECRIREREP.

Il peut être intéressant, si plusieurs programmes doivent accéder au répertoire, de constituer une UNIT placée en librairie à partir des déclarations de type DATE, DISQUE, FICHIER, de la variable NUMFICHIER et des deux fonctions LIREREP et ECRIREREP.

Le programme tel qu'il est prévu présente quelques procédures dont l'utilité n'est

pas évidente pour l'instant, mais qui sont utilisées par la suite du programme; il s'agit de :

PRINFO : évite d'avoir à se souvenir du code ASCII des caractères de contrôle
HOME : efface la fin de la ligne
EFB : efface le bas de l'écran à partir du curseur
BS : recule de un caractère
CR : retour de chariot
SON : déclenche un clic
INV : inverse les caractères à l'écran
(ne fonctionne qu'en PASCAL 1.1)
NORM : retour à l'affichage normal
PRENRETURN : ne prend que la touche "return" du clavier
PRENCAR: renvoie un caractère choisi dans un set donné
OUI : renvoie vrai si l'on tape "0".

La procédure LISTEFICHER est surtout destinée à contrôler que cette première partie fonctionne normalement. Quant au corps de programme actuel, il ne fait que réaliser la seule fonction "L" du FILER.

Bien entendu, il serait possible de lister toutes les autres informations du répertoire ou par exemple ne lister que les fichiers .CODE en menu et par un SETCHAINE chaîner directement le programme que l'on veut exécuter à partir du programme menu mis dans un système STARTUP, ou encore mettre à jour par programme des dates de fichiers, etc. Mais le but de cette première partie est simplement de montrer comment est constitué le répertoire de disque PASCAL. La deuxième partie montrera comment l'utiliser pour réaliser un catalogue général.

Il faut bien sûr utiliser les modifications du répertoire avec sagesse; on peut sans inconvénient modifier par exemple dans d'autres programmes le nom du disque, à condition de bien contrôler la longueur de la chaîne de caractères (STRING!7!) ou le nom d'un fichier (STRING!15!); les dates peuvent également être modifiées. Bien sûr, il ne peut être question de changer les références aux blocs début et fin de fichier, etc, sinon le disque deviendrait totalement illisible. On peut par contre créer des fichiers "fantômes" comme le fait la fonction "M" du FILER, ou remettre en service par ce moyen un fichier effacé par erreur si l'on connaît exactement le numéro des blocs de début et de fin.

BIBLIOGRAPHIE

LE PASCAL PAR LA PRATIQUE

Pierre LEBEUX et Henri TAVERNIER. éd. SYBEX

PROGRAMMER EN PASCAL

Daniel Jean DAVID et Jean-Luc DESCHAMPS. éd. P.S.I.

DECOUVREZ PASCAL SUR APPLE II

John COLIBRI. éd. MNEMODYNE

PROGRAMMING IN PASCAL (en anglais)

Peter GROGONO. éd. ADDISON-WESLEY

(* ===== PREMIERE PARTIE D'UN PROGRAMME DE CATALOGUE GENERAL *)

(*C (C) M. CRIMONT LE 28/10/81 *)

program LECREP;

const MAXFILE =77;

type CHOIDECA =set of CHAR;

DATE =packed record
MOIS :1..12;
JOUR :1..31;
AN :1..99
end;

DISQUE =record
DEBREP :INTEGER;
FINREP :INTEGER;
V1 :INTEGER;
NOMDISQUE :STRING!7!;
NBRBLOC :INTEGER;
NBRFICHE :INTEGER;
V2 :INTEGER;
DATEDISQUE :DATE;
V3, V4 :INTEGER
end;

FICHER =record
DEBFICHER :INTEGER;
FINFICHER :INTEGER;
TYPEFICHER:0..7;
NOMFICHER :STRING!15!;
DERNBLOC :INTEGER;
DATEFICHER:DATE
end;

var REPERTOIRE:record
VOL :DISQUE;
CATALOG :array!1..MAXFILE! of FICHER
end;

NUMLECTEUR:INTEGER;

HOME, BS, EFL, EFB, SON, INV, NORM, CR:CHAR;

function LIREREP(NUM:INTEGER):BOOLEAN; (* LECTURE DU REPERTOIRE *)
begin
(*\$I-*)
UNITREAD(NUM, REPERTOIRE, SIZEOF(REPERTOIRE), 2);
LIREREP:=IORESULT=0
(*\$I+*)
end;

```

function ECRIREP(NUM:INTEGER):BOOLEAN;      (* ECRITURE DANS LE REPERTOIRE *)
begin
  (*$I-*)
  UNITWRITE(NUM, REPERTOIRE, SIZEOF(REPERTOIRE), 2);
  ECRIREP:=IORESULT=0
  (*$I+*)
end;

procedure PRINFO;                          (* CARACTERES DE CONTROLE *)
begin
  HOME:=CHR(12);      EFL:=CHR(29);      BS:=CHR(8);      CR :=CHR(13);
  SON :=CHR(7);      INV:=CHR(18);      NORM:=CHR(20);      EFB:=CHR(11)
end;

procedure PRENRETURN;                      (* PREND UNIQUEMENT LE <RET> *)
var SORT :CHAR;
begin
  repeat
    READ(KEYBOARD, SORT)
  until EOLN(KEYBOARD)
end;

function PRENCAR(BONSET:CHOIDECA):CHAR;    (* SELECTION D'UN CARACTERE *)
var CH :CHAR;
    BON :BOOLEAN;
begin
  repeat
    READ(KEYBOARD, CH);
    if EOLN then CH:=CR;
    BON:=CH in BONSET;
    if not BON then WRITE(SON)
    else if CH in !' ?.,'^!' then WRITE(CH)
  until BON;
  PRENCAR:=CH
end;

function OUI:BOOLEAN;                      (* VRAIE SI L'ON TAPE 'O' *)
begin
  OUI:=PRENCAR(!'O', 'N') in !'O' !
end;

procedure LISTEFICHER;                     (* ESSAI DES PROCEDURES *)
var I,N :INTEGER;
begin
  N:=1;
  WRITE(HOME);
  with REPERTOIRE do
  begin
    WRITELN(' ':10, 'DISQUE:', VOL.NOMDISQUE);
    GOTOXY(0,3);
    for I:=1 to VOL.NBRFICHE do
    begin
      if CATALOG!I!.NOMFICHER<>' ' then
      with CATALOG!I! do
      begin
        WRITE(' ':5, NOMFICHER, ' ':20-LENGTH(NOMFICHER));
        with DATEFICHER do WRITELN(JOUR:2, '/', MOIS:2, '/', AN);
        N:=N+1
      end;
    end;
    if (N mod 15)=0 then

```

```

begin
  WRITELN;
  WRITE(INV,'TAPER (RETURN) POUR CONTINUER ',NORM);PRENRETURN;
  GOTOXY(0,3);WRITE(EFB);
  N:=1
end
end
end
end;

begin
  PRINFO;
  repeat
    WRITE(HOME);
    repeat
      GOTOXY(0,0);WRITE('NUMERO DU LECTEUR:');READLN(NUMLECTEUR)
    until NUMLECTEUR in !4,5,9..12!;
    if LIREREP(NUMLECTEUR) then LISTEFICHER
    else WRITELN('ERREUR EN LECTURE...');
    GOTOXY(0,22);WRITE(INV,'UN AUTRE DISQUE (O/N) ? ',NORM)
  until not OUI;
  WRITE(HOME)
end.

```

logma

Une informatique de gestion adaptée aux besoins des gestionnaires et réalisée par des gestionnaires,

ÉTUDIE

- opportunité d'utilisation de l'outil micro-informatique
- intégration entre informatique traditionnelle et personnelle
- politique de la communication dans l'entreprise

FORME

- formation à l'utilisation de la micro-informatique

RÉALISE

- réalisation de programmes à la demande

LIVRE

- livraison de systèmes clés en main, avec des progiciels de GESTION DE STOCK, PAYE, COMPTABILITE.

Nous sommes gestionnaires avant d'être informaticiens. L'informatique doit s'adapter à l'homme, et non l'inverse. L'outil micro-informatique répond particulièrement bien à ce souci de qualité et d'efficacité du travail, dans des conditions conviviales.

Nombreuses références en informatique traditionnelle - divers matériels - et en informatique individuelle - principalement Apple - auprès des PME et des groupes industriels.

logma s.a. Centre La Châtaigneraie - 29, avenue de Versailles - 78170 La-Celle-St-Cloud - Tél. : (3) 918.13.07

Trois secondes pour trier

NICOLE BREAUD

La fonction de tri est sans doute une des tâches les plus fondamentales de l'outil informatique. Mais les utilitaires de tri sont encore rares dans le domaine du micro.

Notre propos dans cet article est de :

- Présenter les différentes méthodes de tri et de les classer selon leurs performances.

- Proposer un tri par insertion programmé en assembleur qui ne vous demande que "trois secondes pour trier !"

A. Les méthodes de tri

Les théoriciens (Knuth, Wirth, Hoare et les autres) nous conseillent d'éviter de choisir des méthodes directes où le nombre d'échanges est rédhibitoire et nous proposent des méthodes basées soit sur une division en partitions du tableau, soit sur une structuration en arbre binaire des éléments.

Ces méthodes élaborées de tri sont en fait des dérivés des méthodes élémentaires dont on distingue cinq types: l'insertion, la sélection, l'échange, le comptage et la fusion.

De l'idée d'insertion, on retient qu'il s'agit à chaque pas d'insérer un élément nouveau dans une sous-liste déjà triée. C'est la méthode de tri élémentaire la plus efficace et ses performances rivalisent même dans certains cas avec celle des méthodes élaborées.

Selon l'idée de sélection, on définit un algorithme de recherche du plus petit élément qui sera placé en début de tableau, puis du second plus petit élément et ainsi de suite. Le temps d'exécution de cet algorithme est proportionnel au nombre d'éléments, ce qui est inacceptable pour des tableaux assez grands (plus de 50 éléments). Il faut préférer des méthodes de constitution d'un arbre binaire ayant des propriétés spécifiques qui amènent à un tri dont le temps d'exécution est proportionnel à $\log_2(N)$, N étant le nombre d'éléments à trier. Il s'agit du HEAPSORT.

De l'idée d'échange, on a imaginé l'algorithme de tri à bulles (BUBBLE) qui provoque une remontée "en surface" des plus petits éléments, mais ceci en un temps très élevé si le tableau est totalement en désordre. Il existe cependant un moyen plus rapide qui consiste à comparer des éléments situés à un écart donné, et à diminuer progressivement cet écart jusqu'à 1 (comme pour la méthode directe). Ces méthodes améliorées se dénomment SHELL et SHELL-METZNER. L'idée d'échange a aussi engendré la très belle méthode de tri par partitions et échanges dénommée QUICKSORT. Cet algorithme a un déroulement récursif; il s'appelle lui-même mais

avec des arguments de plus en plus simples à traiter. On le programme en BASIC grâce à une pile de réserve d'adresses de sous-tables à trier qui, avec une programmation adéquate, demande 20 cases au maximum.

De l'idée de comptage, on appréciera que cette méthode ne modifie pas l'ordre de groupes d'éléments identiques. Son efficacité, limitée, suppose donc une bonne distribution des clés sur un espace bien déterminé.

Enfin l'utilisation de la fusion se révèlera souvent bien utile. Il est en effet simple de reconstruire une liste à partir de sous-listes triées séparément. Cela permet par exemple de ne trier qu'avec une partie du fichier en mémoire vive, dans le cas d'une limitation de capacité.

CALL-A.P.P.L.E. a publié en Octobre 1980 un article sur les algorithmes de tri, ainsi que des tests comparatifs. Les résultats obtenus varient suivant le type et la taille des listes triées. Sur un fichier aléatoire de plus de cent éléments, le classement est le suivant: QUICKSORT, HEAPSORT, SHELL, SHELL-METZNER, INSERTION. Une comparaison des méthodes de tri vous est aussi offerte dans "la pratique de l'Apple", tome II (éditions du P.S.I.).

Il est à noter que la méthode d'insertion est la plus efficace de toutes pour les tris sur moins de vingt éléments et pour des fichiers d'éléments pré-triés. Ces performances alliées à sa simplicité en font une excellente ressource. La suite de cet article vous permettra de l'utiliser comme sous-programme assembleur de vos programmes BASIC.

B. Le programme TRINS. Tri par insertion

Si nous disposions du langage LISP (spécialement adapté au traitement de listes et à la récursivité), nous pourrions tout simplement dire:

"Trier, c'est insérer le 1er élément dans le tri du reste".

B1. Le programme en BASIC

On insère successivement dans la liste le second élément du tableau, puis le troisième, ... jusqu'au Nième du tableau. Les sous-listes déjà triées dans lesquelles ces éléments doivent s'insérer sont bien sûr:

```
A(1)
A(1),A(2)
etc
A(1),...,A(N)
```

Dès que l'élément à insérer K est plus petit qu'un élément de la sous-liste, ce dernier est décalé sur la droite pour permettre l'insertion dans des cases antérieures (éventuellement à la 1ère case si l'élément à insérer est plus petit que tous ceux de la sous-liste). S'il s'avère plus grand, il s'insère à la place suivante dans la sous-liste.

D'où le programme BASIC:

```
11LIST
10 REM *****
   * PROGRAMME DE TRI *
   * PAR INSERTION *
   * BASIC *
   *****
```

```
20 DIM A(255)
```

Pom's n° 2

```

30 INPUT "N? ";N
40 FOR I = 1 TO N:A(I) = INT (1
    0000 * RND (1)): PRINT A(I
    )" ";; NEXT I: PRINT
50 INPUT " DEPART DU TRI ";R$
60 FOR I = 2 TO N
70 K = A(I)
80 FOR J = I - 1 TO 1 STEP - 1
90 IF A(J) < K THEN 130
100 A(J + 1) = A(J)
110 NEXT J
120 A(1) = K: GOTO 140
130 A(J + 1) = K
140 NEXT I
150 INPUT "OPERATION TERMINEE ";
    R$
160 FOR I = 1 TO N: PRINT A(I)"
    ";; NEXT I

```

B2. Le programme en langage machine

Ce programme, plutôt académique, en langage machine permet le tri d'un tableau où chaque élément n'est qu'un seul octet et où l'on connaît l'adresse de début du tableau (contenue dans 6,7) et sa longueur (dans 8).

*300LL

```

0300- A0 01 LDY ##01
0302- C8 INY
0303- B1 06 LDA ($06),Y
0305- 85 0A STA $0A
0307- 98 TYA
0308- 48 PHA
0309- 38 SEC
030A- E9 01 SBC ##01
030C- A8 TAY
030D- B1 06 LDA ($06),Y
030F- C5 0A CMP $0A
0311- 90 09 BEC $031C
0313- E6 06 INC $06
0315- 91 06 STA ($06),Y
0317- C6 06 DEC $06
0319- 88 DEY
031A- 10 F1 BPL $030D
031C- A5 0A LDA $0A
031E- C8 INY
031F- 91 06 STA ($06),Y
0321- 68 PLA
0322- A8 TAY
0323- C4 08 CPY $08
0325- 90 DE BEC $0302
0327- 00 BRK

```

Le registre d'index Y évolue comme la variable I du programme précédent puis, avant la boucle J, Y est empilé avant d'être utilisé pour évoluer comme la variable J. En fin de boucle J, on récupère la variable I en dépilant la valeur de Y.

Pour tester le programme il faut au préalable charger 6 et 7 de l'adresse basse et

de l'adresse haute du premier octet du tableau à trier, et charger 8 du nombre d'octets à classer par ordre croissant de valeur comprises entre -128 et +127.

B3. Rappel préliminaire

Avant de rentrer au coeur de TRINS, revenons sur l'implantation des variables réelles en Applesoft et l'utilisation de sous-programmes pour nombres flottants.

Chaque valeur réelle occupe cinq octets: un pour l'exposant (avec un décalage de 128) et quatre pour la mantisse. Dans le cas d'un tableau de nombres réels à une dimension, on passe au nombre suivant en incrémentant le curseur de cinq unités.

Pour déplacer les éléments de la liste, on utilisera deux sous-programmes:

- MOVFM : transfert d'un nombre M vers l'accumulateur flottant FAC.
- MOVMF : transfert du FAC vers une zone de mémoire M.

Les arguments de ces sous-programmes sont les adresses basses et hautes dont les contenus constituent l'adresse de la première case de la zone de cinq octets. Pour MOVFM, il faut utiliser les registres A et Y respectivement pour l'adresse basse et haute du pointeur. Pour MOVMF, ce sont les registres X et Y.

Pour comparer deux nombres réels, nous en effectuons la soustraction dans le FAC et analysons le signe du résultat en appelant SIGN. Après l'appel de cette routine, l'accumulateur A contient 00, 01 ou FF suivant que le FAC est nul, positif ou négatif.

B4. le programme en assembleur

Les paramètres à donner au programme et les adresses où placer ces valeurs sont:

- A1: l'adresse de l'élément A(1) du tableau à trier (dans 6,7)
- N : le nombre d'éléments du tableau (dans 10)
- K : l'adresse d'une variable réelle permettant de conserver la première valeur de A(I) (dans 8,9).

Les variables utilisées sont:

- AI : adresse de l'élément A(I).
- AJ : celle de A(J).
- I et J : les indices des boucles.

L'adresse de l'élément suivant A(J+1) est calculée et rangée dans X et Y pour être immédiatement utilisée par MOVMF. Les sous-programmes SAVE et RESTOR permettent de reconstituer le contexte originel, spécifié par les valeurs des registres A,X,Y,P et S.

**END OF PASS 1			0800		9	AI	EPZ \$1E
**END OF PASS 2			0800		10	AJ	EPZ \$1B
			0800		11	K	EPZ \$08
0800	1	;	0800		12	I	EPZ \$19
0800	2	;TRI INSERTION	0800		13	J	EPZ \$1A
0800	3	; A(N)	0300		14		ORG \$300
0800	4	; N<256	0300		15		OBJ \$800
0800	5	;	0300		16	;	
0800	6	E EQU !5	0300	204AFF	17	TRI	JSR SAVE
0800	7	A1 EPZ 06	0303	A50A	18		LDA N
0800	8	N EPZ \$0A	0305	49FF	19		EOR #\$FF

0307	18	20		CLC	0374	A606	76		LDX	A1	
0308	6902	21		ADC	#2	0376	A407	77		LDY	A1+1
030A	8519	22		STA	I	0378	202BEB	78		JSR	MOVFM
030C	A507	23		LDA	A1+1	037B	4C9303	79		JMP	IS
030E	851F	24		STA	AI+1	037E	A508	80	AJINFK	LDA	K
0310	851C	25		STA	AJ+1	0380	A409	81		LDY	K+1
0312	A506	26		LDA	A1	0382	20F9EA	82		JSR	MOVFM
0314	18	27	ECLI	CLC		0385	A41C	83		LDY	AJ+1
0315	6905	28		ADC	#E	0387	A51B	84		LDA	AJ
0317	851E	29		STA	AI	0389	18	85		CLC	
0319	9002	30		BCC	TAIK	038A	6905	86		ADC	#E
031B	E61F	31		INC	AI+1	038C	AA	87		TAX	
031D	A41F	32	TAIK	LDY	AI+1	038D	9001	88		ECC	IL
031F	20F9EA	33		JSR	MOVFM	038F	C8	89		INY	
0322	A608	34		LDX	K	0390	202BEB	90	IL	JSR	MOVFM
0324	A409	35		LDY	K+1	0393	A51E	91	IS	LDA	AI
0326	202BEB	36		JSR	MOVFM	0395	E619	92		INC	I
0329		37	;			0397	F003	93		BEQ	FIN
0329	A519	38		LDA	I	0399	4C1403	94		JMP	BCLI
032B	18	39		CLC		039C	203FFF	95	FIN	JSR	RESTOR
032C	650A	40		ADC	N	039F	60	96		RTS	
032E	851A	41		STA	J	03A0		97	;		
0330	A51F	42		LDA	AI+1	03A0		98	MOVFM	EQU	\$EB2B
0332	851C	43		STA	AJ+1	03A0		99	MOVFM	EQU	\$EAF9
0334	A51E	44		LDA	AI	03A0		100	FSUE	EQU	\$E7A7
0336	38	45	BCLJ	SEC		03A0		101	SIGN	EQU	\$EB82
0337	E905	46		SBC	#E	03A0		102	SAVE	EQU	\$FF4A
0339	851B	47		STA	AJ	03A0		103	RESTOR	EQU	\$FF3F
033B	B002	48		BCS	COMP			104		END	
033D	C61C	49		DEC	AJ+1						
033F	A41C	50	COMP	LDY	AJ+1						
0341	20F9EA	51		JSR	MOVFM						
0344	A508	52		LDA	K						
0346	A409	53		LDY	K+1						
0348	20A7E7	54		JSR	FSUE						
034B	2082EB	55		JSR	SIGN						
034E	F002	56		BEQ	KINFAJ						
0350	102C	57		BPL	AJINFK						
0352	A51B	58	KINFAJ	LDA	AJ						
0354	A41C	59		LDY	AJ+1						
0356	20F9EA	60		JSR	MOVFM						
0359	A41C	61		LDY	AJ+1						
035B	A51B	62		LDA	AJ						
035D	18	63		CLC							
035E	6905	64		ADC	#5						
0360	AA	65		TAX							
0361	9001	66		BCC	OK						
0363	C8	67		INY							
0364	202BEB	68	OK	JSR	MOVFM						
0367		69	;								
0367	A51B	70		LDA	AJ						
0369	C61A	71		DEC	J						
036B	D0C9	72		BNE	BCLJ						
036D	A508	73		LDA	K						
036F	A409	74		LDY	K+1						
0371	20F9EA	75		JSR	MOVFM						

***** END OF ASSEMBLY

*300 3A0											
0300-	20	4A	FF	A5	0A	49	FF	18			
0308-	69	02	85	19	A5	07	85	1F			
0310-	85	1C	A5	06	18	69	05	85			
0318-	1E	90	02	E6	1F	A4	1F	20			
0320-	F9	EA	A6	08	A4	09	20	2B			
0328-	EB	A5	19	18	65	0A	85	1A			
0330-	A5	1F	85	1C	A5	1E	38	E9			
0338-	05	85	1B	B0	02	C6	1C	A4			
0340-	1C	20	F9	EA	A5	08	A4	09			
0348-	20	A7	E7	20	82	EB	F0	02			
0350-	10	2C	A5	1B	A4	1C	20	F9			
0358-	EA	A4	1C	A5	1B	18	69	05			
0360-	AA	90	01	C8	20	2B	EB	A5			
0368-	1B	C6	1A	D0	C9	A5	08	A4			
0370-	09	20	F9	EA	A6	06	A4	07			
0378-	20	2B	EB	4C	93	03	A5	08			
0380-	A4	09	20	F9	EA	A4	1C	A5			
0388-	1B	18	69	05	AA	90	01	C8			
0390-	20	2B	EB	A5	1E	E6	19	F0			
0398-	03	4C	14	03	20	3F	FF	60			
03A0-	A9										

B5. Utilisation de la routine

Le petit programme présenté ici donne un exemple de l'utilisation de la routine de tri écrite en assembleur par un programme BASIC.

```
10 REM *****
    * PROGRAMME DE TRI *
    * PAR INSERTION *
    * APPEL SUB-ROUTINE *
    *****

20 DIM A(255)
30 INPUT "N? ";N: POKE 10,N
40 K = 0:X = 0:I = 0
50 FOR I = 1 TO N:A(I) = INT (1
    0000 * RND (1)): PRINT A(I
    )" ";: NEXT I: PRINT
60 INPUT "DEPART DU TRI ";R$
70 X = A(1): POKE 6, PEEK (131): POKE
    7, PEEK (132)
80 X = K: POKE 8, PEEK (131): POKE
    9, PEEK (132)
90 CALL 768
100 INPUT "OPERATION TERMINEE ";
    R$
110 FOR I = 1 TO N: PRINT A(I)"
    "": NEXT I
```

Les instructions ** 30, 70 et 80 ** permettent de transférer les arguments à la routine, qu'on aura chargée préalablement (BLOAD TRINS-ASSEMBLEUR). Dans les adresses 131 et 132, on trouve le pointeur de la dernière variable utilisée par la dernière instruction BASIC, donc l'adresse de A(1) et celle de K.

Le programme TRINS-APPEL, qui fait appel à la routine assembleur est vingt fois plus rapide que son équivalent TRINS-BASIC (trois secondes au lieu de soixante) pour trier une liste d'une centaine d'éléments. Pour les plus exigeants, une routine QUICKSORT donnera de meilleurs résultats pour des tableaux de plus de vingt éléments. Mais le résultat est déjà fort appréciable pour un tri si facile à utiliser et peu encombrant.

BIBLIOGRAPHIE

- N.Wirth "Algorithms + Data Structures = Programs". Prentice Hall 1976
- Tremblay, Sorrenson "An Introduction To Data Structures With Applications". International Student Edition. Mc Graw Hill 1976
- Kernigham, Plauger "Software Tools". Addison Wesley 1976
- Meyer, Baudoin "Méthodes de Programmation". Eyrolles 1978
- Corge "Eléments d'Informatique". Larousse 1975
- Weston "Comparing ten sort algorithms". Call-A.P.P.L.E. Octobre 80
- Nicole Bréaud-Pouliquen "La Pratique de l'Apple II". Vol I, II et III. Editions du P.S.I. 1981

Sprechen Sie DOS ?

CHRISTIAN GUERIN

Ce programme va vous permettre d'adapter les commandes du DOS à vos préférences linguistiques.

Dans le premier numéro de POM'S, j'avais présenté un programme assembleur qui permettait de ne taper que la première lettre de certaines instructions données. Mais cette méthode pouvait gêner d'autres "inputs" à l'écran et devait être utilisée avec précaution. Il est à noter qu'elle peut être exploitée avec profit à d'autres fins, par exemple pour éviter des saisies redondantes telles que taper "Facture" ou "Compte" en toutes lettres.

Le programme présenté ici s'attaque au coeur du problème en modifiant la syntaxe du DOS à l'intérieur même de celui-ci. La table des noms des ordres que le DOS va lire en RAM pour interpréter ce que vous tapez se trouve à l'adresse 43139 (\$A884) pour un 48 K. Chacune des 28 commandes est séparée de la suivante grâce au bit de poids fort de son dernier caractère. Le programme vous permet de saisir votre table personnalisée puis la recopie dans le DOS en fin de traitement.

Pour constituer votre DOS, tout vous est permis. La seule contrainte extérieure est un problème de place: la table complète ne peut contenir plus de 133 caractères. Dans le programme, la longueur de chaque ordre est limitée à 9 et bornée par le nombre de caractères encore disponibles. Nous avons d'autre part interdit les blancs, qui rendraient inutilisable l'instruction: par exemple si vous aviez à taper "CAT LOG" à l'écran, c'est "CAT" que le DOS irait comparer aux valeurs contenues dans la table. A noter que la saisie propose par défaut le nom standard des ordres du DOS, qui vous sont rappelés à l'écran. Cette option permet de redonner facilement à votre DOS sa syntaxe virginale.

Si vous désirez créer une disquette avec un DOS personnalisé, il vous suffit d'initialiser celle-ci après avoir modifié votre syntaxe. Utilisez pour cela la commande INIT ou sa version modifiée (par exemple VIVE-POM'S).

N.B.: le programme est à adapter si vous ne possédez pas un 48 K. Il faut dans ce cas modifier l'adresse du début de la table (ôter 16*4096 pour un 32 K, 32*4096 pour un 16K).

```
10 REM *****
   *
   * PROGRAMME DE
   * MODIFICATION DES
   * ORDRES DU DOS
   *
   *****
70 NO = 1
80 HOME
90 REM *****
   * LECTURE DE LA
   * TABLE DANS LE DOS
   *****
20 DIM N$(28): REM NOM FINAL
30 DIM NO$(28): REM NOM INITIAL
40 AB = 43139: REM DEBUT TABLE
50 NC = 1: REM TAILLE TABLE
60 MC = 133: REM TAILLE MAXIMUM
100 FOR N = 1 TO 28
110 READ NO$(N)
120 C = PEEK (AB + NC)
130 IF C > 128 THEN 150
140 C$ = C$ + CHR$ (C):NC = NC +
   1: GOTO 120
```

```

150 N$(N) = C$ + CHR$(C - 128)
160 C$ = "" : NC = NC + 1
170 NEXT N
180 REM *****
    * AFFICHAGE DES *
    * NOMS A L'ECRAN *
    *****

190 TEXT : HOME
200 FOR I = 1 TO 14
210 VTAB 3 + I : PRINT NO$(I) : TAB(
10) : N$(I) : TAB( 19) : "! "
220 VTAB 3 + I : HTAB 22 : PRINT N
O$(I + 14) : TAB( 32) : N$(I +
14)
230 NEXT I
240 REM *****
    * MENU *
    *****

250 VTAB 19 : PRINT "TAPEZ -->
POUR UN DEPLACEMENT POSITIF"

260 PRINT " <-- POUR UN D
EPLACEMENT NEGATIF"
270 PRINT " ESC POUR MODI
FIER UN ORDRE"
280 PRINT " F POUR LA F
IN DU PROGRAMME"
290 INVERSE : FLASH : GOSUB 690 :
NORMAL
300 REM *****
    * ATTENTE D'UN *
    * CARACTERE *
    *****

310 C = PEEK ( - 16384) : IF C <
128 THEN 310
320 POKE - 16368,0
330 REM TEST SI ESCAPE
340 IF C = 155 THEN 520
350 REM TEST SI -->
360 IF C = 149 THEN GOSUB 690 : N
O = NO + 14 * (NO < > 28) -
27 * (NO > 14) : GOTO 290
370 REM TEST SI <--
380 IF C = 136 THEN GOSUB 690 : N
O = NO - 14 * (NO < > 1) +
27 * (NO < 15) : GOTO 290
390 REM TEST SI F
400 IF C - 128 < > ASC ("F") THEN
290
410 REM *****
    * FIN DE PROGRAMME *
    *****

420 HOME : VTAB 13 : PRINT " MO
DIFICATION DE LA TABLE DU DO
S"
430 NC = 1
440 FOR I = 1 TO 28

450 FOR J = 1 TO LEN (N$(I))
460 POKE AB + NC, ASC ( MID$(N$
(I),J,1))
470 NC = NC + 1
480 NEXT J
490 POKE AB + NC - 1, ASC ( RIGHT$(
N$(I),1)) + 128
500 NEXT I
510 HOME : END
520 REM *****
    * MODIFICATION *
    * D'UN ORDRE *
    *****

530 VTAB 18 : CALL - 958 : PRINT
: PRINT "MODIFICATION DE L'O
RDRE " : NO$(NO)
540 CD = MC - NC + LEN (N$(NO)) :
IF CD > 9 THEN CD = 9
550 PRINT CD : " CARACTERES DISPON
IBLES"
560 VTAB 21 : INPUT N$
570 IF N$ = "" THEN N$ = NO$(NO)

580 IF LEN (N$) > 9 THEN 520
590 IF NC - LEN (N$(NO)) + LEN
(N$) > MC THEN PRINT "" : GOTO
520
600 OK = 1 : FOR I = 1 TO LEN (N$
) : BL$ = MID$(N$,I,1) : OK =
OK * (BL$ < > " ") : NEXT I :
IF OK = 0 THEN 520
610 NC = NC - LEN (N$(NO)) + LEN
(N$)
620 N$(NO) = N$ : GOTO 180
630 REM *****
    * TABLE DES ORDRES *
    * ORIGINAUX DU DOS *
    *****

640 DATA "INIT","LOAD","SAVE","R
UN","CHAIN","DELETE"
650 DATA "LOCK","UNLOCK","CLOSE"
,"READ","EXEC","WRITE"
660 DATA "POSITION","OPEN","APPE
ND","RENAME","CATALOG"
670 DATA "MON","NOMON","PR#","IN
#","MAXFILES","FF"
680 DATA "INT","BSAVE","BLOAD","
BRUN","VERIFY"
690 REM *****
    * POSITIONNEMENT *
    * A L'ECRAN *
    *****

700 REM
710 VTAB 3 + NO - 14 * (NO > 14)

720 HTAB 10 + 22 * (NO > 14)
730 PRINT N$(NO) : RETURN

```


Des vers dans la pomme...

ALEXANDRE DUBACK

REPAREZ VOTRE RENUMBER, SI CE N'EST DEJA FAIT

Dans la rubrique "L'Apple Epluché" de l'Ordinateur Individuel, il y a longtemps, le problème du Renumber Apple a déjà été soulevé. Nous le rappelons et apportons ici une réponse plus complète au problème.

Essayez de faire renuméroter par le RENUMBER de votre disquette Master le programme suivant : 20 A=20*20

Si le résultat que vous obtenez est : 10 A=20*10

... il faut que vous fassiez le changement suivant :

```
LOAD RENUMBER
A=x:B=172:POKE x,B:POKE x+1,B-1
UNLOCK RENUMBER
SAVE RENUMBER
LOCK RENUMBER
```

avec les valeurs de x suivantes : x=14342 pour le DOS 3.2 avec Applesoft en RAM et x=4815 avec Applesoft en ROM. Pour le DOS 3.3 : x=14316 avec l'Applesoft en RAM et x=4789 avec Applesoft en ROM.

LES DRAMES DE L'APPEND - SUITE ET FIN (?)

Nous avons indiqué dans le numéro 1 de Pom's une méthode pour s'assurer que l'APPEND fasse apparaître une fin de fichier même dans le cas où la fin de fichier tombe sur une fin de secteur. Quand ce problème survenait, on s'en apercevait rapidement car la suite du fichier s'inscrivait au tout début du fichier.

Un autre "bug" de l'APPEND est nettement plus vicieux, car moins apparent. Il survient quand vous utilisez APPEND avec des fichiers de plus de 128 secteurs.

Lors d'un APPEND, le DOS commence normalement à écrire la suite des informations à partir du EOF (end of file) en écrasant celui-ci par la première nouvelle information. Quand l'EOF se trouve à plus de 128 secteurs au-delà du début de fichier, l'APPEND réalise l'adjonction des nouvelles informations à partir du byte suivant l'EOF, et donc sans écraser celui-ci.

A partir de ce moment, chaque APPEND ultérieur écrit les nouvelles informations en débutant au byte suivant ce premier EOF. Cela ressemble à ce que faisait l'APPEND avec le bug décrit dans le premier numéro de Pom's, mais c'est plus difficile à déceler, car il y a au moins 128 bons secteurs dans le fichier !

Nous essaierons de vous donner la solution dans le numéro 3 de POM's. En attendant, ne faites pas d'APPEND sur des fichiers de plus de 128 secteurs ...

Survol de l'Apple III

BRUNO RIVES

Que se cache-t-il derrière Apple /// ?

Le dernier-né de la firme californienne n'est sorti que tout récemment en Europe. Il s'agit d'un ordinateur personnel dont les caractéristiques se rapprochent plus d'un mini que d'un micro. De mémoire centrale 128 K octets extensible à 256 K, doté d'un micro-processeur deux fois plus rapide que celui de l'Apple II, possédant un système d'exploitation tourné vers l'utilisateur final, capable de supporter des disques durs de bureau d'une capacité de cinq millions d'octets, l'arrivée de l'Apple /// est vraiment un événement.

Les caractéristiques du matériel

L'Apple /// possède en standard une mémoire centrale de 128 K. Elle peut être portée à 256 K par l'addition de mémoires supplémentaires et d'un lecteur de disquettes intégré. Des connecteurs incorporés peuvent relier l'appareil à trois autres lecteurs de disquettes, sans interface supplémentaire.

Le clavier est plus élaboré que celui de l'Apple II, comportant notamment une partie numérique séparée ainsi que deux touches de fonction, quatre de direction, toutes étant à répétition automatique. Le clavier peut être équipé en version AZERTY accentuée. L'affichage permet plusieurs modes au choix. En mode texte, on obtient 80 colonnes sur 24 lignes en noir et blanc, 40 sur 24 en noir et blanc ou en 16 couleurs, l'écran pouvant être divisé en fenêtres. En mode graphique, on dispose au choix de 560 points par 192 en noir et blanc, 280 par 192 en noir et blanc ou en couleurs limitées, ou 140 par 192 en 16 couleurs.

En standard, nous trouvons également une interface RS232C pour connection sur tout type d'imprimante série ou pour une utilisation en communication, une interface pour imprimante SILENTYPE, une sortie analogique, une prise vidéo et une sortie couleur aux normes américaines mais facilement transformable pour la France.

La plaque de base de l'Apple /// possède en outre des slots d'entrée-sortie (comme sur l'Apple II) permettant de connecter quatre périphériques supplémentaires. Des cartes d'interface sont déjà développées sur Apple /// dont une carte parallèle universelle et la carte Thunderlock qui permet de remédier pour l'instant à l'absence de carte horloge intégrée.

La grande nouveauté est incontestablement aussi la sortie quasi simultanée de PROFILE, un disque dur de cinq millions d'octets à usage lui aussi individuel. Il s'agit d'un disque 5 pouces 1/4, technologie Winchester, dont les caractéristiques de fiabilité et d'utilisation (absence de bruit, totale compatibilité avec les systèmes et programmes) le mettent à la portée de quiconque. Il est possible de connecter jusqu'à quatre PROFILES sur un même APPLE ///, ce qui porte à 20 méga-

octets (plus les possibilités de disquette) la mémoire de masse en ligne.

Le système d'exploitation

SOS (Sophisticated Operations Systems) est le nouveau système d'exploitation développé par Apple pour l'Apple ///. Ses caractéristiques justifient son titre:

* Traitement de fichiers : le système d'exploitation de l'Apple /// considère tout périphérique comme un fichier que l'on peut ouvrir et fermer, dans lequel on peut lire et écrire, que ce soit un fichier sur disque, une imprimante ou un modem. Donc plus de problème pour spécifier un périphérique ou un autre; ils sont tous gérés de la même manière. On peut créer des catalogues et des sous-catalogues de fichiers et donc les trier selon les critères voulus.

* Gestion de la mémoire centrale: l'Apple /// possède 128 K ou 256 K de mémoire vive avec un micro-processeur 8 bits; comment cela est-il possible ? Grâce à deux aspects très sophistiqués du système. Le premier est le "Bank Switching" : l'espace mémoire est découpé en banques de 32 K et le système permet de passer d'une banque à l'autre de façon transparente comme sur les minis ou les gros systèmes. La deuxième ressource est le "zero page switching" qui permet à chaque banque d'avoir sa propre page zéro. Il ne s'agit plus d'une mémoire physique comme sur l'Apple II mais d'une mémoire virtuelle.

* La gestion des interruptions est intégrée dans le SOS et se situe à deux niveaux: au niveau système, les interruptions provenant des périphériques sont facilement récupérables grâce à des sous-programmes spécialement prévus (nul besoin d'écrire ses propres sous-programmes en langage machine). Au niveau utilisateur, il est très facile de profiter en BASIC ou Pascal de ces possibilités par des fonctions de langage comme ONKBD qui est activée dès que l'on touche au clavier.

Le logiciel d'exploitation

Il faut tout d'abord signaler les deux langages qu'Apple a choisi d'implanter sur Apple /// :

- Le Business BASIC est une version améliorée de l'Applesoft et contient des instructions telles que PRINT USING et IF THEN ELSE qui manquaient sur l'Apple II. Les possibilités de précision sont améliorées puisque les nombres peuvent atteindre 19 chiffres, la longueur des noms de variables peut comprendre 64 caractères et toutes les possibilités du SOS sont utilisables au niveau du BASIC.

- l'Apple /// Pascal est une version de l'UCSD Pascal dont nous ne décrirons pas les caractéristiques de langage. Il permet d'utiliser toutes les possibilités du système d'exploitation SOS et est le meilleur outil de développement sur Apple ///.

La compatibilité entre l'Apple II et l'Apple /// est totale au niveau du Pascal (il suffit de recompiler les programmes). Pour le BASIC, on peut soit utiliser l'Apple /// en mode simulation Apple II, soit faire passer les programmes BASIC sur l'Apple /// en utilisant un programme de transfert (il faudra bien sûr les modifier pour les rendre compatibles avec le Business BASIC).

De nombreux utilitaires sont disponibles, le plus important étant destiné à la gestion des périphériques (configuration, copie, opérations sur les fichiers...). Tous les sous-programmes de gestion des périphériques sont également prévus et peuvent être inclus dans le système d'exploitation en fonction du traitement:

lecteurs de disquettes, Profile, Silentye, RS232C, Qume, graphique ...

Les programmes d'application

Apple estime qu'en un an autant de programmes auront été développés sur Apple /// qu'en six ans sur Apple II. De nombreuses sociétés de logiciel produisent déjà des applications sur Apple /// et de nouveaux logiciels apparaissent toutes les semaines. Pour l'instant, Apple propose plusieurs produits:

* Un traitement de texte, Applewriter ///, qui n'a rien à envier aux plus sophistiqués des programmes de ce type, permettant notamment la connection avec d'autres fichiers SOS et la possibilité d'écrire des procédures chaînées de traitement de texte rendant aisés le courrier automatique, le calcul de devis, etc.

* VISICALC ///, qui est la nouvelle version de VISICALC adaptée à l'Apple ///. Plus rapide et travaillant en 80 colonnes, il a également l'avantage de pouvoir gérer de plus gros modèles.

* MAIL LIST MANAGER, programme de gestion de fichier individuel et d'impression d'étiquettes.

* ACCESS ///, programme de communication qui permet de relier l'Apple à un autre système informatique.

* Un programme de tracé graphique, BUSINESS GRAPHICS, qui permet de transcrire sous forme de graphiques sophistiqués l'information venant de programmes tels que VISICALC /// ou d'autres programmes d'application.

En conclusion, Apple /// est en fait destiné à un marché différent de celui de l'Apple II, même si certains utilisateurs de l'Apple II haut de gamme opteront pour ce nouveau produit. Tout le monde n'a pas besoin de 128 K de mémoire centrale, d'interfaces de communication ou de disques durs de 5 millions d'octets. Par contre, il y a un créneau réservé d'ordinaire aux mini-ordinateurs ou aux systèmes de time-sharing, et un besoin non encore satisfait par l'informatique classique auquel Apple /// répond parfaitement. Destiné aux Etats-Unis aux utilisations personnelles et professionnelles telles que le traitement de texte, le graphique, la gestion de tableaux ou de fichiers personnels, il devrait être utilisé en France non seulement pour ces fonctions, mais aussi pour des applications d'informatique classique telles que la comptabilité, la paie, la facturation, l'analyse des ventes ...

Note de la rédaction.

Il faut que nous tempérions l'enthousiasme bien compréhensif de notre ami Bruno Rives par quelques commentaires soulignant les problèmes éventuels relatifs à l'Apple ///.

Les premiers 10.000 Apple /// distribués aux Etats-Unis ont effectivement eu des problèmes de fiabilité de circuits et de surchauffe, conséquence secondaire de l'ambition de faire coexister tant de possibilités à l'intérieur d'un ordinateur individuel. Il semblerait que ces problèmes aient disparu, à la suite de la refonte d'un certain nombre d'éléments par Apple, sur un budget additionnel de 5 millions de dollars. Il n'y a pas suffisamment d'utilisateurs en France en ce

moment pour que nous puissions en dire plus à ce sujet. Nous attendons d'en avoir un sous la main ...

Faisons le point des remarques négatives que nous avons pu relever à l'égard de l'Apple /// :

a) c'est très bien de pouvoir disposer de 128K, mais il n'en reste en fait que 70 après chargement du SOS. Pour disposer de plus de place, il faudra libérer les zones mémoire occupées par des fonctions du SOS inutiles pour un traitement donné.

b) en matière de gestion de fichiers, notons un catalogue d'apparence beaucoup plus professionnelle. Il semble cependant y avoir un problème quand on réécrit un fichier: on court pour l'instant vers des ennuis si l'on réécrit un fichier sans l'avoir annulé auparavant, à cause de la présence d'EOFs parasites quand le fichier a été raccourci. Cela fait perdre un des avantages du catalogue dans lequel apparaissent la date de création et la date de dernière modification du fichier, car celles-ci sont alors forcément identiques.

c) l'Apple /// ne dispose pas encore de manuels en français aussi bien finis que ceux de l'Apple II. En outre, il manque des utilitaires indispensables tels que PLE et DAKIN, pour n'en citer que deux. Par contre ceux-ci fonctionnent toujours aussi bien en simulation de l'Apple II.

d) le clavier numérique séparé améliore significativement la rentrée de données. Ce n'aurait pas été une mauvaise idée d'y incorporer les touches +,-,*,/ et =.

e) le problème de l'horloge prévue initialement n'a pas été résolu. L'arrivée de la carte intégrée est annoncée pour début 1982 (?)

f) le passage à 560 points horizontaux dans le mode graphique relève d'un très bon sentiment. Pourquoi n'avoir pas aussi amélioré la définition verticale ?

g) les lecteurs de disquette supplémentaires ne sont pas compatibles avec ceux de l'Apple II. En cas de panne, pas de cannibalisation possible avec un Apple II !

Malgré ces critiques, les rares utilisateurs que nous connaissons, après avoir souffert des bugs de la version initiale du matériel, sont tout à fait satisfaits de l'Apple ///. Les prix publics annoncés sont de 23.700 F HT pour l'Apple /// (sans moniteur), 23.700 F pour le PROFILE, 2.500 F pour le moniteur et 4.050 F par lecteur supplémentaire. Ce n'est hélas pas donné : avec un second lecteur, le moniteur, une imprimante moyenne et une carte parallèle, on arrive à un budget de 36.000-37.000 F HT !



LIBRAIRIE LA NACELLE

INFORMATIQUE • ÉLECTRONIQUE • AUTOMATISME • MICROPROCESSEUR

**TOUS OUVRAGES ET ABONNEMENTS
FRANÇAIS ET ÉTRANGERS**

Tous les ouvrages français ou étrangers signalés dans cette revue peuvent être obtenus ou commandés à La Nacelle

2, rue Campagne-Première 75014 PARIS - Tél. 322 56 46

Métro Raspail - Parking à la hauteur du 120 bd du Montparnasse

ouvert tous les jours lundi compris, sans interruption de 9 h 30 à 18 h 50, samedi fermeture à 17 h 50.

Le jeu de la vie

JEAN-FRANCOIS DUVIVIER

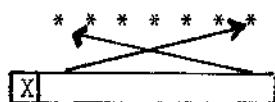
Cet article fait suite à ceux consacrés à ce jeu dans l'Ordinateur individuel dans les numéros 27,28 et 29 (mai à août 1981). On se contentera ici de rappeler que le jeu consiste à faire évoluer une population, génération après génération. Une population est un ensemble de points, situés aux intersections des lignes d'une grille carrée infinie. Les règles d'évolution sont impératives; il n'y a pas de marge de manoeuvre et l'avenir de la population initiale est en fait prédéterminé dès le départ. A chaque génération, des points meurent, d'autres naissent. Ainsi va la vie ...

Chaque cellule peut avoir jusqu'à huit voisines immédiates. Une cellule naît dans une position vide si elle a exactement trois voisines. Elle meurt de neurasthénie avec une voisine, et succombe à l'étouffement si elle en a quatre ou plus.

Cette série d'articles a attiré ma curiosité et m'a donné l'envie de réaliser mon propre programme. La folie des grandeurs aidant, me voici à rêver d'un jeu rapide (mois de 10 secondes à chaque génération) avec 53760 cellules. Pourquoi ce chiffre bizarre ? Parce qu'il correspond exactement au nombre de points affichés en haute résolution.

A ce stade, l'Applesoft ne pouvait satisfaire aucune de mes exigences. Seule solution: l'assembleur. Un rapide calcul me permit de pouvoir espérer atteindre mon but.

La mise au point a pris quelque temps. En effet je me suis laissé prendre au piège de belle manière. L'affichage haute résolution est fait dans l'ordre inverse des bits. Le point le plus à gauche correspond au bit 0, celui le plus à droite au bit 6.



Bits 7 6 5 4 3 2 1 0

Après les corrections nécessaires, voici le programme prêt pour les essais en grandeur nature. Les résultats sont à la hauteur de mes espérances: 5 secondes pour un écran pratiquement vide, 9 secondes dans les cas les plus défavorables.

Vous prendrez; je l'espère, plaisir à exécuter ce programme et à repérer les populations fixes (aucun changement), stables (figures se répétant à une fréquence plus ou moins longue), ou vouées à l'extinction !

A. Structure du Programme

1. Routine COMPT (comptage)

Entrée: registre A

Sortie: registre B

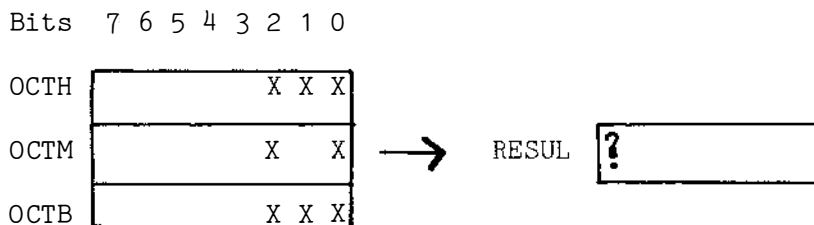
Fonction: à partir d'un octet transmis dans le registre A, compter le nombre de cellules vivantes (c'est à dire le nombre de bits à 1) sur les trois bits de plus faible poids. Ce résultat est disponible dans le registre Y.

2. Routine C.BIT (calcul d'un bit)

Entrée: OCTH (octet haut)
OCTM (octet milieu)
OCTB (octet bas)

Sortie: RESULT (résultat)

Fonction: calculer l'état de la cellule de la génération n+1 à partir des 8 cellules voisines de la génération n.



Chacun des trois octets est pris séparément et envoyé à la routine COMPT afin de calculer le nombre de voisins. A noter que le bit 1 de OCTM ne doit pas être pris en compte dans le calcul.

En fonction du résultat contenu dans le registre Y, 3 cas sont possibles :

* voisins <2 ou>3, la cellule de la génération suivante doit être vide. RESULT est marqué d'un bit à 0 (label MORT).

* voisins = 3, il y a une naissance. Quelle que soit l'état de la cellule à la génération n, la cellule de la génération n+1 est vivante. Un bit à 1 est marqué dans RESULT (label VIE).

* voisins = 2, la cellule reste inchangée. On recopie le bit 1 de OCTM dans RESULT.

3. Routine C.BYT (calcul d'un octet)

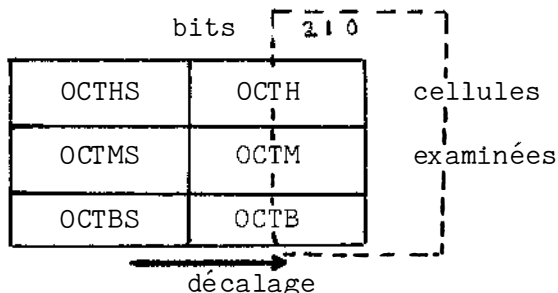
entrée: OCTH, OCTM, OCTB (octets haut, milieu et bas)
OCTHS, OCTMS, OCTBS (octets suivants)

Sortie: RESULT (résultat)

Fonction: générer un octet résultat, c'est à dire l'état de sept cellules. Le bit 7 de chaque octet n'est pas affiché par Apple (information couleur) et ne cons-

titue donc pas une cellule au niveau de l'écran. Il est forcé à zéro dans RESUL.

Pour utiliser C.BIT, la routine C.BYT doit présenter la cellule examinée en position 1 dans OCTM. D'autre part, le calcul du bit 6 doit prendre en compte pour voisins l'état des bits 0 des octets suivants (attention le bit 7 n'est pas pris en compte). Il est donc nécessaire de travailler sur deux octets qui seront décalés ensemble vers la gauche avant chaque appel de C.BIT.



Après 7 appels de C.BIT, on obtient dans REGUL, sur 7 bits, les 7 cellules de la génération suivante.

4. Routine C.LGN (calcul d'une ligne)

Entrée: LIGN1 (adresse de début de la ligne supérieure)
 LIGN2 (adresse de début de la ligne examinée)
 LIGN3 (adresse de début de la ligne inférieure)
 LIGNR (adresse de début de la ligne résultat)

Sortie: les 280 cellules de la génération n+1 correspondant à la ligne examinée sont stockées à partir de l'adresse LIGNR.

Fonction: Génération d'une ligne résultat. Lors des quarante appels de C.BYT nécessaires à l'élaboration d'une ligne, la routine C.LGN doit remettre à jour les informations OCTHS, OCTMS et OCTBS. Elle doit également stocker RESUL dans la ligne résultat.

5. Routine C.ECR (calcul d'un écran)

Entrée: HPAG (numéro de la page graphique affichée: \$20 pour la page 1, \$40 pour la page 2).

Sortie: stockage des 53760 cellules de la génération n+1.

Variables: LIGN1, LIGN2, LIGN3, LIGNR, NOLIGN (numéro de ligne courante).

Fonction: calculer pour chacune des 192 lignes, les adresses de début de la ligne courante, de la précédente et de la suivante, puis appeler la routine C.LGN.

Le calcul est facilité par l'utilisation de routines haute résolution de l'Applesoft:

* HPOSN (\$F411): donne à partir d'un numéro de page (HPAG) et d'un numéro de ligne (NOLIGN), l'adresse du début de cette ligne en mémoire (résultat dans LIGN).

* INCRY (\$F504): donne à partir de la ligne courante, l'adresse de début de la ligne suivante (résultat dans LIGN).

* DECRY (\$F4D5): donne à partir de la ligne courante, l'adresse de début de la ligne précédente (résultat dans LIGN).

La routine C.ECR utilise ces trois routines pour calculer les adresses LIGN1,

LIGN2 et LIGN3. L'adresse LIGNR est obtenue à partir de LIGN2 en effectuant une translation de page (décalage de \$2000). A noter que l'emploi des routines INCRY et DECRY permet de faire le tour de l'écran (la ligne suivant la ligne 192 étant la ligne 0).

6. Début de programme

Le programme commence par un affichage de la page résolution No 1. La boucle permet à chaque génération d'afficher la page calculée et de remettre à jour la variable HPAG.

B. Utilisation du programme

1. Placez-vous en Applesoft.
2. Chargez en page haute résolution 1 votre population initiale (ou tout autre dessin haute résolution).
3. Exécutez le programme VIE.OBJET.

1	;	*****	0800	32	PAGE2	EQU	\$C055
2	;	* * *	0800	33	DECRY	EQU	\$F4D5
3	;	* JEU DE LA VIE *	0800	34	INCRY	EQU	\$F504
4	;	* * *	0800	35	;		
5	;	* COPYLEFT *	0800	36	;		
6	;	* * *	0800	37		ORG	\$800
7	;	* POM'S *	0800	38	;		
8	;	* * *	0800	39	;		
9	;	*****	0800	40	;	DEBUT DU PROGRAMME	
10	;		0800	41	;	=====	
11		OCTH EPZ \$00	0800	42	;		
12		OCTM EPZ \$01	0800	43	;		
13		OCTB EPZ \$02	0800 AD50C0	44		LDA	HIRES1
14		OCTHS EPZ \$03	0803 AD52C0	45		LDA	HIRES2
15		OCTMS EPZ \$04	0806 AD57C0	46		LDA	HIRES3
16		OCTBS EPZ \$05	0809	47	BOUCLE:		
17		RESUL EPZ \$06	0809 AD54C0	48		LDA	PAGE1
18		SAVX EPZ \$07	080C A920	49		LDA	#\$20
19		SAVY EPZ \$08	080E 85E6	50		STA	HPAG
20		LIGN1 EPZ \$0A	0810 202008	51		JSR	C.ECR
21		LIGN2 EPZ \$0C	0813 AD55C0	52		LDA	PAGE2
22		LIGN3 EPZ \$0E	0816 A940	53		LDA	#\$40
23		LIGNR EPZ \$10	0818 85E6	54		STA	HPAG
24		LIGN EPZ \$26	081A 202008	55		JSR	C.ECR
25		NOLIGN EPZ \$E2	081D 4C0908	56		JMP	BOUCLE
26		HPAG EPZ \$E6	0820	57	;		
27		HPOSN EQU \$F411	0820	58	;		
28		HIRES1 EQU \$C050	0820	59	;	MISE A JOUR DE L'ECRAN	
29		HIRES2 EQU \$C052	0820	60	;	=====	
30		HIRES3 EQU \$C057	0820	61	;		
31		PAGE1 EQU \$C054	0820	62	;		

```

0820          63  C.ECR:
0820 A900      64          LDA #00
0822 85E2      65          STA NOLIGN
0824 2011F4    66          JSR HPOSN
0827          67  ECR1:
0827 20D5F4    68          JSR DECRY
082A A526      69          LDA LIGN
082C 850A      70          STA LIGN1
082E A527      71          LDA LIGN+1
0830 850B      72          STA LIGN1+1
0832 2004F5    73          JSR INCRY
0835 A526      74          LDA LIGN
0837 850C      75          STA LIGN2
0839 8510      76          STA LIGNR
083B A527      77          LDA LIGN+1
083D 850D      78          STA LIGN2+1
083F 8511      79          STA LIGNR+1
0841 38        80          SEC
0842 E920      81          SBC ##20
0844 A6E6      82          LDX HPAG
0846 E040      83          CFX ##40
0848 F003      84          BEQ ECR2
084A 18        85          CLC
084B 6940      86          ADC ##40
084D 8511      87  ECR2  STA LIGNR+1
084F 2004F5    88          JSR INCRY
0852 A526      89          LDA LIGN
0854 850E      90          STA LIGN3
0856 A527      91          LDA LIGN+1
0858 850F      92          STA LIGN3+1
085A 206608    93          JSR C.LGN
085D E6E2      94          INC NOLIGN
085F A5E2      95          LDA NOLIGN
0861 C9C1      96          CMP ##C1
0863 D0C2      97          BNE ECR1
0865 60        98          RTS
0866          99  ;
0866          100 ;
0866          101 ; CALCUL D'UNE LIGNE
0866          102 ; =====
0866          103 ;
0866          104 ;
0866          105 C.LGN:
0866 A000      106          LDY #00
0868 E10A      107          LDA (LIGN1),Y
086A 0A        108          ASL
086B 8500      109          STA OCTH
086D E10C      110          LDA (LIGN2),Y
086F 0A        111          ASL
0870 8501      112          STA OCTM
0872 E10E      113          LDA (LIGN3),Y
0874 0A        114          ASL
0875 8502      115          STA OCTB
0877          116 LGN1:
0877 C8        117          INY
0878 E10A      118          LDA (LIGN1),Y
087A 8503      119          STA OCTHS

```

*800.BFD									
0800-	AD	50	C0	AD	52	C0	AD	57	
0808-	C0	AD	54	C0	A9	20	85	E6	
0810-	20	20	08	AD	55	C0	A9	40	
0818-	85	E6	20	20	08	4C	09	08	
0820-	A9	00	85	E2	20	11	F4	20	
0828-	D5	F4	A5	26	85	0A	A5	27	
0830-	85	0B	20	04	F5	A5	26	85	
0838-	0C	85	10	A5	27	85	0D	85	
0840-	11	38	E9	20	A6	E6	E0	40	
0848-	F0	03	18	69	40	85	11	20	
0850-	04	F5	A5	26	85	0E	A5	27	
0858-	85	0F	20	66	08	E6	E2	A5	
0860-	E2	C9	C1	D0	C2	60	A0	00	
0868-	B1	0A	0A	85	00	B1	0C	0A	
0870-	85	01	B1	0E	0A	85	02	C8	
0878-	B1	0A	85	03	B1	0C	85	04	
0880-	B1	0E	85	05	88	C0	27	D0	
0888-	08	A9	00	85	03	85	04	85	
0890-	05	20	9F	08	A5	06	4A	91	
0898-	10	C8	C0	28	D0	D9	60	86	
08A0-	07	84	08	A2	07	20	BC	08	
08A8-	46	03	66	00	46	04	66	01	
08B0-	46	05	66	02	CA	D0	EE	A6	
08B8-	07	A4	08	60	A0	00	A5	00	
08C0-	F0	03	20	F0	08	A5	02	F0	
08C8-	03	20	F0	08	A5	01	F0	05	
08D0-	29	FD	20	F0	08	C0	02	90	
08D8-	13	C0	04	B0	0F	C0	03	F0	
08E0-	07	A5	01	4A	4A	66	06	60	
08E8-	38	66	06	60	18	66	06	60	
08F0-	4A	90	01	C8	4A	90	01	C8	
08F8-	4A	90	01	C8	60	00			

			08EC	169	C.BIT:		
			08EC	A000	170	LDY ##00	
			08BE	A500	171	LDA OCTH	
087C	B10C	120	LDA (LIGN2),Y	08C0	F003	172	BEQ BIT1
087E	8504	121	STA OCTMS	08C2	20F008	173	JSR COMPT
0880	B10E	122	LDA (LIGN3),Y	08C5	A502	174	BIT1 LDA OCTB
0882	8505	123	STA OCTBS	08C7	F003	175	BEQ BIT2
0884	88	124	DEY	08C9	20F008	176	JSR COMPT
0885	C027	125	CPY ##27	08CC	A501	177	BIT2 LDA OCTM
0887	D008	126	BNE LGN2	08CE	F005	178	BEQ BIT3
0889	A900	127	LDA ##00	08D0	29FD	179	AND #%11111101
088B	8503	128	STA OCTHS	08D2	20F008	180	JSR COMPT
088D	8504	129	STA OCTMS	08D5	C002	181	BIT3 CPY ##2
088F	8505	130	STA OCTBS	08D7	9013	182	BCC MORT
0891		131	LGN2:	08D9	C004	183	CPY ##4
0891	209F08	132	JSR C.BYT	08DB	E00F	184	BCS MORT
0894	A506	133	LDA RESULT	08DD	C003	185	CPY ##3
0896	4A	134	LSR	08DF	F007	186	BEQ VIE
0897	9110	135	STA (LIGNR),Y	08E1	A501	187	LDA OCTM
0899	C8	136	INY	08E3	4A	188	LSR
089A	C028	137	CPY ##28	08E4	4A	189	LSR
089C	D0D9	138	BNE LGN1	08E5	6606	190	ROR RESULT
089E	60	139	RTS	08E7	60	191	RTS
089F		140	;	08E8		192	VIE:
089F		141	;	08E8	38	193	SEC
089F		142	;	08E9	6606	194	ROR RESULT
089F		143	;	08EB	60	195	RTS
089F		144	;	08EC		196	MORT:
089F		145	;	08EC	18	197	CLC
089F		146	;	08ED	6606	198	ROR RESULT
089F		147	C.BYT:	08EF	60	199	RTS
089F	8607	147	STX SAVX	08F0		200	;
08A1	8408	148	STY SAVY	08F0		201	;
08A3	A207	149	LDX ##7	08F0		202	;
08A5		150	BYT1:	08F0		203	COMPTAGE
08A5	20EC08	151	JSR C.BIT	08F0		204	;
08A8	4603	152	LSR OCTHS	08F0		205	;
08AA	6600	153	ROR OCTH	08F0		206	COMPT:
08AC	4604	154	LSR OCTMS	08F0	4A	207	LSR
08AE	6601	155	ROR OCTM	08F0	4A	208	BCC COM1
08B0	4605	156	LSR OCTBS	08F1	9001	209	INY
08B2	6602	157	ROR OCTB	08F3	C8	210	COM1 LSR
08B4	CA	158	DEX	08F4	4A	211	BCC COM2
08B5	D0EE	159	BNE BYT1	08F5	9001	212	INY
08B7	A607	160	LDX SAVX	08F7	C8	213	COM2 LSR
08B9	A408	161	LDY SAVY	08F8	4A	214	BCC COM3
08BB	60	162	RTS	08F9	9001	215	INY
08EC		163	;	08FB	C8	216	COM3 RTS
08EC		164	;	08FC	60	217	;
08EC		165	;	08FD		218	;
08EC		166	;	08FD		219	END
08EC		167	;				
08EC		168	;				

***** END OF ASSEMBLY

Une routine bien pratique

Il est des routines qui vous attirent irrésistiblement en raison de leur originalité ou de leur ingéniosité. La routine moniteur de S.H. LAM en est une illustration parfaite.

La technique utilisée permet à l'utilisateur de placer dans une chaîne de caractères des commandes à destination du moniteur. La principale utilisation est de placer en mémoire un court programme en langage machine avec une grande facilité et sans un nombre invraisemblable de POKES ou de READ/DATAs.

La scène se passe en trois actes:

1. La chaîne est scindée en caractères qui sont placés dans le buffer clavier à la suite les uns des autres (à partir de l'adresse \$100 ou 512 en décimal).
2. Le CALL-144 met en oeuvre les routines qui examinent le buffer clavier et s'assurent de l'exécution des commandes spécifiées.
3. Le retour au BASIC. Il est possible, à partir du moniteur, d'incorporer plusieurs commandes sur une ligne si on a pris soin d'incorporer un séparateur (caractère neutre, le "N" par exemple). Aussi lors de l'acte 1 a-t-il été ajouté subrepticement la valeur N D9C6G (en Applesoft) à la chaîne de caractères prévue. Lors de l'acte 2, cette commande est exécutée en dernier, après toutes celles la devançant. Les sous-programmes \$D9C6 (Applesoft) ou \$E88A (Integer) permettent de revenir au mode RUN (exécution) au début de la prochaine ligne BASIC à exécuter. Bien que plus lente que des POKES et des READ/DATAs, cette routine a l'avantage d'être plus simple à utiliser. Succomberez-vous à son charme ?

<pre> 10 REM ***** * * ROUTINE S.H.LAM * * * POUR APPLESOFT * * ***** 20 REM 100 Y\$ = "300:FF FF FF FF FF FF" 110 GOSUB 63900: END 63900 REM ROUTINE MONITEUR 63905 REM APPLESOFT DE S.H.LAM 63910 Y\$ = Y\$ + " N D9C6G" 63920 FOR I = 1 TO LEN (Y\$) 63930 POKE 511 + I, ASC (MID\$ (Y\$,I,1)) + 128 63940 NEXT I: POKE 72,0: CALL - 144 63950 RETURN </pre>	<pre> 10 REM ***** * * ROUTINE DE S.H.LAM * * * POUR INTEGER * * ***** 30 REM 50 DIM Y\$(255) 100 Y\$="300:FF FF FF FF FF" 110 GOSUB 32000 120 END 32000 REM ROUTINE DE S.H.LAM POUR INTEGER 32010 Y\$(LEN(Y\$)+1)=" N E88AG" 32020 FOR I=1 TO LEN(Y\$) 32030 POKE 511+I, ASC(Y\$(I)) 32040 NEXT I 32050 POKE 72,0: CALL -144 32060 RETURN </pre>
--	--

Les clubs ont la parole

Apple Club - Avenue Paul Gilson 387 - 1620 Drogendos - Belgique

Nous sommes heureux de vous faire part de la création d'un club d'utilisateurs de l'Apple II.

Les objectifs de ce club sont d'apporter à ses membres une aide tant au niveau hardware que software et d'avoir accès à la programmàthèque du club.

Notre club tiendra ses réunions dans les locaux aimablement prêtés par P.N.B. Computer Center.

Contactez V. Brunette (02) 376 4244 ou Y. Fischer (02) 356 1817

Ma Pomme - 6 rue Paul Saunière - 75016 Paris - France

Message du président : "Utilisateur d'Apple, je me suis mis à la recherche d'un club et me suis fait piéger dans quelque chose qui n'avait de club que le nom! A partir de là, j'ai décidé de créer le 'chez soi' de tous les utilisateurs d'Apple. C'est à ce moment-là que j'ai eu la chance de rencontrer celui qui allait devenir notre secrétaire ... Nous désirons regrouper tous les utilisateurs d'Apple, de quelque horizon qu'ils viennent, pour une association de type 'défense du consommateur' ".

Le club compte actuellement cinquante membres, dans l'hexagone et outre-mer. Leur niveau de compétence est aussi varié que leur origine géographique, s'étendant du novice à l'expert. Les activités de la section parisienne sont essentiellement axées autour de réunions bimensuelles consacrées à des sujets particuliers (langage LISP, Microdial, ...) ou généraux (documentation, logiciels récents, utilitaires, cartes, ...). Notre objectif est simple : être utiles.

Le club tient à déclarer qu'il respecte la loi concernant la copie des logiciels copyrightés et espère avoir une action sur les prix auxquels ils seront proposés.

Téléphoner tous les jours de 20H à 24H au (1) 558 0578.

logma

Une informatique de gestion adaptée aux besoins des gestionnaires et réalisée par des gestionnaires.

ÉTUDIE

- opportunité d'utilisation de l'outil micro-informatique
- intégration entre informatique traditionnelle et personnelle
- politique de la communication dans l'entreprise

FORME

- formation à l'utilisation de la micro-informatique

RÉALISE

- réalisation de programmes à la demande

LIVRE

- livraison de systèmes clés en main, avec des progiciels de GESTION DE STOCK, PAYE, COMPTABILITE.

Nous sommes gestionnaires avant d'être informaticiens. L'informatique doit s'adapter à l'homme, et non l'inverse. L'outil micro-informatique répond particulièrement bien à ce souci de qualité et d'efficacité du travail, dans des conditions conviviales.

Nombreuses références en informatique traditionnelle - divers matériels - et en informatique individuelle - principalement Apple - auprès des PME et des groupes industriels.

Revue de presse -JFD

Depuis quelques mois, de nombreux livres et revues sur Apple ont paru. J'en ai sélectionné cinq parmi les plus récents. Deux d'entre eux, par leur exceptionnelle qualité, me semblent se hisser nettement au-dessus des publications actuelles (POM'S excepté) et devoir servir de bible à tous ceux qui souhaitent savoir ce qui se passe sous le capot de leur Apple.

* Beneath Apple DOS. Don Worth et Pieter Lechner. Quality Software (176p. en anglais).

Après tout ce qui a déjà été écrit sur ce diable de petite machine, il est bien difficile de nos jours de se passionner pour un livre. Et pourtant !... Je n'hésiterai pas à dire que cette publication a été l'un des faits majeurs de cette année. Il aurait très bien pu s'appeler "Tout ce que vous avez toujours voulu savoir sur le DOS sans jamais le trouver dans le manuel du DOS".

Avant d'aller plus loin, je soulignerai que ce livre n'est pas destiné aux débutants mais s'adresse à des lecteurs ayant déjà complètement assimilé le manuel Apple et désirant comprendre l'architecture interne du DOS et son utilisation à partir de l'assembleur.

Ce livre, très bien présenté sous un format identique à celui des manuels Apple, est divisé en huit chapitres.

1. Une introduction
2. L'évolution du DOS et les différences entre chaque version (3,3.1,3.2,3.3)
3. Le formatage d'une disquette: le pourquoi et le comment.
4. L'organisation de la disquette reprend un certain nombre d'informations provenant du manuel du DOS, mais en les approfondissant. Où est placé le DOS, le catalogue ? Que sont la VTOC et les tables de renvoi pistes/secteurs ? Comment sont stockés les fichiers, les programmes en binaire, Applesoft ou Integer ? Autant de réponses précises et claires.
5. La structure du DOS explique comment se passe le "boot" et décrit l'utilisation de la mémoire par le DOS.
6. L'utilisation du DOS à partir de l'assembleur: comment lire un secteur, travailler sur un fichier, faire un catalogue...
7. Comment singulariser et modifier le DOS (voir l'article de Christian Guérin).
8. Le schéma logique du DOS. Le chapitre examine pas à pas chaque routine du DOS et sa fonction. Seul regret, le DOS étant "copyrighté" par Apple, il n'a pu être présenté sous forme de listing commenté. Cependant, la clarté des explications et les différentes indications données permettent une compréhension aisée.

A la suite de ces huit chapitres figurent trois annexes. La première fournit le listing et les explications de petits programmes en assembleur très précieux: initialisation d'une seule piste, lecture complète d'une piste, etc. La deuxième donne quelques explications très sommaires sur les méthodes de protection utilisées pour empêcher la copie. La troisième constitue un glossaire de douze pages de tous les termes employés et des pages où ils sont commentés.

En conclusion, il s'agit d'un manuel excellent et pratiquement sans défaut que je recommande à tous les passionnés de l'assembleur.

* What's Where In The Apple. An Atlas To The Apple Computer. Pr. William F. Luebbert (110p. en anglais)

La publication dans un article de MICRO, the 6502 Journal, d'un atlas de 7 pages comprenant les principales adresses de l'Apple avait déjà valu à son auteur une certaine considération. Mais que dire après une si belle récidive ?

Scindé en deux parties, cet atlas rassemble plus de deux mille adresses en mémoire. Page zéro, pages graphiques, DOS, Integer, Applesoft, Moniteur: tout est passé au crible.

La première partie est organisée par adresses croissantes. Pour chacune sont donnés la valeur hexadécimale, la valeur décimale, le label éventuel et une explication. Dans le cas de routines, y sont également consignés les registres du 6502 qui sont modifiés. La deuxième partie reprend les mêmes informations mais par ordre alphabétique des labels.

Seul reproche à formuler, je n'ai pu y trouver l'explication de certains sigles figurant en face de chaque adresse. J'ai également pu trouver (et n'en suis pas peu fier) quelques adresses spéciales n'y figurant pas !

De présentation dense et concise (parfois trop), cet atlas représente le principal outil de travail sur Apple dès que l'on veut s'aventurer hors des octets battus.

* Apple II Monitors Peeled. Apple Computer Inc. (90p. en anglais).

La première version de ce manuel, écrit par William E. Dougherty, date de Mai 1979 et était distribuée à ses membres par le club américain A.P.P.L.E. (Apple Pugetsound Program Library Exchange). Séduit et convaincu de son utilité, Apple Computer Inc. a révisé et amélioré son contenu pour en éditer un manuel de même format que ceux que nous connaissons.

Exclusivement dédié aux moniteurs (l'ancien et l'Autostart), l'Apple II Monitors Peeled explicite l'usage des différentes variables en page zéro et de toutes les routines utilisées dans le moniteur. Quelques programmes en Applesoft permettent d'utiliser immédiatement les informations acquises.

Très intéressant pour les lecteurs programmant en assembleur, il peut également être recommandé pour les débutants qui désirent apprendre comment fonctionne le moniteur: grâce au style clair et précis et à de nombreuses explications, ils ne seront pas trop perdus. A quand la traduction française ?

* Call-A.P.P.L.E. In Depth NO 1. All About Applesoft (135p. en anglais).

Edité en hors-série par le magazine "Call-A.P.P.L.E.", ce numéro est entièrement consacré à l'Applesoft. Il regroupe de nombreux articles s'intéressant tant à la structure interne de l'Applesoft (stockage des programmes et de ses variables en mémoire, utilisation des adresses en page zéro...) qu'à son utilisation (overlay, rapidité...). A noter que les nombreux programmes présentés dans la revue peuvent vous être livrés sur disquette.

* Micro Apple 1. Micro, the 6502 Journal (en anglais).

Ce numéro 1 regroupe trente articles publiés dans la revue entre 1977 et 1981, et comprend une disquette d'accompagnement. D'un intérêt limité pour le programmeur confirmé, il peut cependant être intéressant pour des débutants qui y trouveront des programmes courts et simples dont ils pourront s'inspirer.

Courrier des lecteurs

J'ai trouvé une erreur dans votre revue ! Page 42, section 7, il faut lire CTRL-J à la place de CTRL-M. Ce n'est pas important, mais cela me fournit un bon prétexte pour vous écrire.

Evidemment, je tiens à vous féliciter pour la tenue de votre revue, c'est bien, continuez, n'hésitez pas à écrire des programmes en assembleur ... Un haut niveau de programmation est apprécié.

En ce qui concerne les erreurs dans les articles, j'ai apprécié le sérieux avec lequel elles ont été traquées. Malgré cela, des "vers" passent à travers le tamis ...

C'est pourquoi je vous supplie, si par mégarde une "bogue" se produit, ne la cachez pas à vos lecteurs et faites les corrections nécessaires dans le numéro suivant, en gros sans se cacher.

*

E
r
f

-

A
F
t
r
f
f
e

5
1
3

5
6
71

81

Le POKE se trouve dans la zone CPM I/O. A priori, cela fonctionne sans problème, mais ce qui m'inquiète, c'est que Microsoft ne l'ait pas prévu.

** P. Bekaert - Le Mesnil St Denis **

Quelqu'un sait-il pourquoi ?

J'ai pris un abonnement, ne connaissant pas votre revue ... J'espère en tirer partie, mais j'avoue bien sincèrement que le niveau est haut pour mes minimes connaissances. Le problème des débutants est bien simple, car ceux-ci sont livrés à eux-mêmes sans documentation, ou bien avec une documentation en anglais, et surtout sans conseils ni professeurs. Ils sont donc tenus d'apprendre seuls et ce n'est pas facile.

Que me donnera l'informatique, je l'ignore. mais c'est intéressant ...

Formatez vos programmes

LIST

```

100 REM *****
      * FORMATEZ VOS *
      * PROGRAMMES DE *
      * BELLE MANIERE *
      *****

```

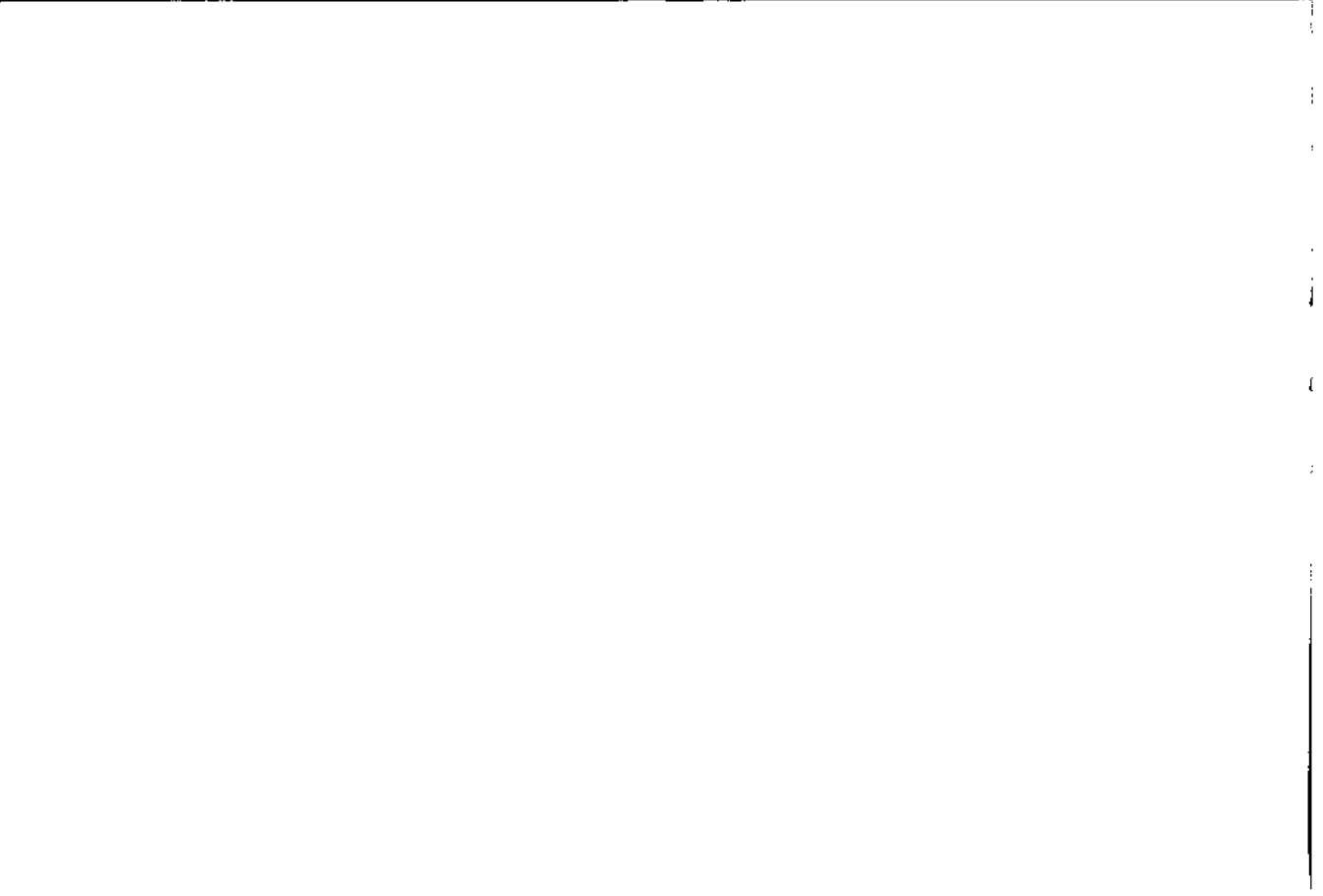
N'avez-vous jamais eu envie d'agrémenter vos programmes de petits tableaux comme dans l'exemple présenté à gauche ? Si vous avez essayé, vous vous êtes aperçu que cela demandait beaucoup de travail car la présentation du texte à l'écran est différente de sa physionomie dans un listing.

Le BASIC, Applesoft ou Integer, introduit en effet des espaces supplémentaires aux caractères que vous frappez. C'est la raison pour laquelle l'utilisation du "Poke 33,33" est si précieuse pour les corrections.

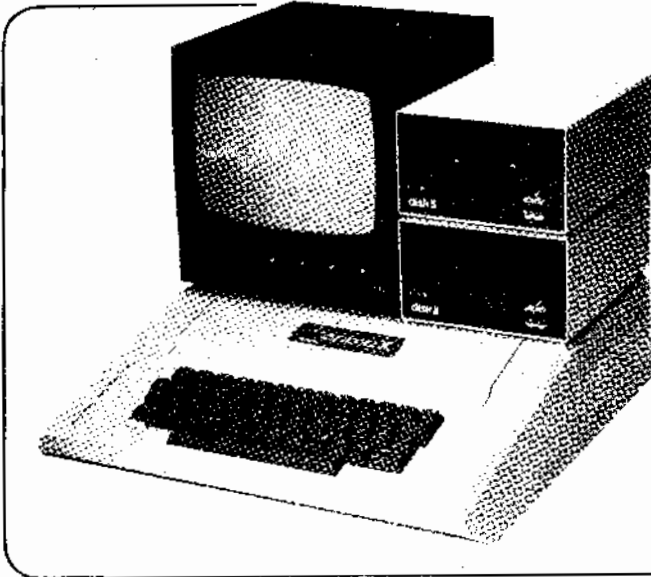
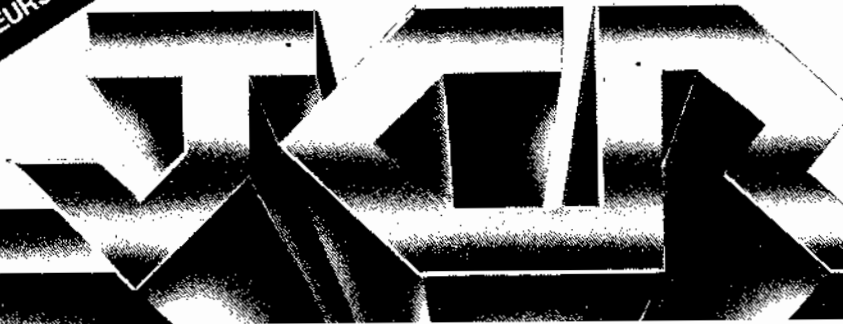
La solution à tous vos problèmes de présentation de programmes réside dans les instructions suivantes:

- Poke 33,28 (en Applesoft)
- Poke 33,32 (en Integer)

Après avoir procédé au POKE adéquat, vous pouvez entrer au clavier les REM suivant le format que vous souhaitez. Quand vous avez terminé, utilisez l'instruction "POKE 33,40" ou la commande TEXT (ou encore appuyez sur Reset si vous avez l'Autostart) pour rétablir la fenêtre de visualisation normale. Vous pouvez alors lister le programme, les REMarques resteront bien en place.



RECHERCHONS
VENDEURS-DEMONSTRATEURS



**PROMOTIONS
SPECIALES**
réservées
aux lecteurs de
"POM'S"



	PRIX F.T.T.C.		PRIX
APPLE II + 16 K	8 500	CARTE APPLESOFT	1 450
APPLE II + 32 K	8 750	CARTE PASCAL	3 240
APPLE II + 48 K	8 990	CARTE LANGAGE	1 580
APPLE III 128 K	26 000	CARTE MICROSOFT Z80	2 750
DISK II + CONTROLEUR	4 720	CARTE MICROGRAPHIQUE	5 200
DISK II	3 375	CARTE IEEE 488	3 100
DISK III SUPPL	4 400	CARTE PROTOTYPE	160
DISK 8" 2*256 K	19 150	CARTE PROTO APPLE III	390
DISK 8" 2*512 K	24 100	CARTE RECON. PAROLE	1 450
DISQUE DUR 10 MGA	41 500	CARTE MICROMUSIC	1 320
MODULATEUR N/B KIT	220	CARTE APPLECLOCK	2 160
CARTE COULEUR SECAM	1 050	CARTE TIMER CCS	1 150
CARTE COULEUR R.V.B.	920	CARTE VIDEOTHERM 80C	2 970
INTERFACE //	1 350	CARTE SMARTEM 80C	2 180
INTERFACE // APPLE III	1 600	SYLENTYPE II	2 800
INTERFACE V24. RS 232.	1 450	KIT SYLENTYPE II → III	270
CARTE INTEGER	1 450	EXTENSION 16 K RAM	250

PROMOTION DU MOIS

APPLE II 48 K + DISK II avec Contrôleur + Moniteur Vert 9" + 10 disquettes
14 400 F.T.T.C.

Liste détaillée des produits contre enveloppe timbrée.

Ces prix vous seront consentis sur présentation de cette publicité.

POM'S N°2

J.C.R. Electronique - 58 rue Notre-Dame-de-Lorette - 75009 Paris - Tél. : (1) 282.19.80

Expédition dans toute la France - Matériel garanti 1 an pièces et M.-O. - Ouvert du mardi au samedi de 10 h à 13 h et de 14 h à 19 h
En raison des fluctuations monétaires, ces prix sont susceptibles d'être modifiés sans préavis. Nous consulter pour confirmation.



DIFFUSÉ
PAR :



P.S.I. Diffusion
B.P. 88 - 77400 LAGNY
Tél. (6) 007 59 31

concurrente votre micro-informatique!

Vous êtes industriel, chercheur, enseignant, particulier, membre de profession libérale.

Nous sommes une équipe d'ingénieurs ayant une position de leader sur le marché de la micro-informatique et nous avons une expérience unique en matière d'installation et maintenance de systèmes.

Nous sommes faits pour nous entendre.



Micro Informatique Diffusion

51 bis, AVENUE DE LA RÉPUBLIQUE, 75011 PARIS. TÉL : 357.83.20.

● **Micro-ordinateurs.** Apple Commodore Pertec.

● **Périphériques.**

Disquettes, disques durs, imprimantes, terminaux clavier-écran, tables traçantes, tables à digitaliser.

● **Interfaces.**

Pour terminaux de tous types (V 24 RS 232C, 8 bits parallèles). Entrées analogiques. Sorties analogiques. IEEE-488. Entrées BCD. Cartes base de temps horloge. Calcul rapide.

Ouvert de 9h à 12h et de 14h à 19h. Sauf le dimanche.