

poom's

La revue francophone des utilisateurs de l'Apple

Disque virtuel 16 K

Le Calculateur entier

Des trucs pour Apple Writer

PEEKs et POKEs en Pascal

Un éditeur de shapes

L'Apple //c est né !

Initiation à l'assembleur (2)



NUMERO 12 • PRIX 40 F

M2366-12-40 F

ISSN : 0294-6068

SPiD
PRÉSENTE

LE N°2

LISTE DES POINTS DE VENTE

- 06000 - MAD'S - NICE - (93) 88.04.79
06210 - EVOLUTION 2000 - MANDELIU - (93) 49.81.61
08600 - MICRO-BOUTIQUE JCR - GIVET - (24) 55.01.23
10000 - MICROPOLIS - TROYES - (25) 79.03.79
11000 - I ÉLEC VIDEO CLUB - CARCASSONNE - (68) 47.08.94
11000 - R 2 I INFORMATIQUE - NARBONNE - (68) 65.15.83
12000 - BASE 2 SOCODETI - RODEZ - (65) 42.50.05
13004 - ALLIANCE - MARSEILLE - (91) 86.35.99
13005 - ELP INFO - MARSEILLE - (91) 94.91.13
13006 - MD SYSTÈME - JCR BOUTIQUE - MARSEILLE - (91) 37.62.33
13200 - LUDO - ARLES - (90) 96.79.03
14000 - OMB-VASSARD TILLETTE - CAEN - (31) 93.48.09
16000 - S.A. LHOMME - ANGOULÈME - (45) 92.27.37
18000 - AVENIR INFORMATIQUE - BOURGES (48) 65.16.57
19100 - MICROMATIC - BRIVE - (55) 87.15.17
19100 - INFORMATIC 19 - BRIVE - (55) 87.77.08
21000 - C.M.G. MICRO LEADER - DIJON - (80) 30.12.70 +
24100 - MICRO CYRANO INFORMATIQUE - BERGERAC -
(16) 56.06.12
25206 - I.T.A. MONTBELLARD - MONTBELLARD CEDEX - (81) 94.50.65
26000 - DOMICA - VALENCE - (75) 41.14.75
26500 - ECA ÉLECTRONIQUE - BOURG-LES-VALENCE - (75) 42.68.88
29000 - L'ORDINATEUR 29 - QUIMPER - (98) 95.92.70
30000 - DISCOUT INFORM. SERVICE - NIMES - (66) 23.74.21
31000 - MICRO DIFFUSION - TOULOUSE - (61) 22.81.17
33000 - MICRO DIFFUSION - BORDEAUX - (56) 81.11.99
33800 - ETS COCA - BORDEAUX - (61) 92.91.78
34006 - JCR BOUTIQUE - MONTPELLIER - (67) 58.84.37
34200 - BUREAU ORGANISATION - SETE - (67) 74.34.10
34500 - MARCELLEC - BEZIERS - (67) 37.11.13
37170 - LIM - CHAMBRAY-LES-TOURS - (47) 27.99.00
38500 - MICRO AVENIR - VOIRON - (76) 65.79.55
39000 - MICRO 39 - JEAN-PIERRE-ANDRÉ - LONS-LE-SAUNIER
(84) 24.45.39
41500 - T.I.M. - MER - (54) 81.62.47
42000 - DÉTROIT INFORMATIQUE - SAINT-ÉTIENNE - (77) 33.58.59
42100 - SAINT-ÉTIENNE COMPOSANTS - SAINT-ÉTIENNE
(77) 33.50.14
42300 - MICRO SYSTÈME RHONE-ALPES - ROANNE - (77) 68.67.99 +
44100 - SILICONE VALLÉE - NANTES - (40) 73.21.67
45000 - TÉLÉPHONIE BIS - ORLÉANS - (38) 54.34.34
47000 - JULIEN ÉLECTRONIQUE - AGEN - (58) 66.55.64
49000 - TEMPS X - ANGERS - (41) 88.95.07
49300 - CHOLET INFORMATIQUE - CHOLET - (41) 46.02.40
54000 - SEREC - NANCY - (8) 332.12.60
56000 - L'ORDINATEUR 56 - VANNES - (97) 42.52.20
56100 - L'ORDINATEUR 56 - LORIENT - (97) 64.52.54
57504 - ARGO INFORMATIQUE - SAINT-AVOLD - (87) 92.54.84 +
57800 - CM - FREYMING MERLEBACH - (87) 81.14.89
59000 - ETS BOULLANGER - LILLE - (20) 54.98.75
59000 - BÉCY INFORMATIQUE - LILLE - (20) 92.33.06
59400 - MICROSHOP - CAMBRAI - (27) 81.98.09 +
59500 - CID - DOUAI - (27) 88.47.20
59800 - M.B.D.C. - LILLE - (20) 57.91.87
60108 - QUENEUTTE - CREIL - (4) 425.04.26
60200 - LARDET S.A. - COMPIÈGNE - (4) 423.07.86
63000 - IMPACT - CLERMONT-FERRAND - (73) 92.17.55
64110 - ESPACE MICRO 64 - BAYONNE - (59) 59.41.55
64600 - INFORMATIQUE BASCO LANDAISE - ANGLET - (59) 31.96.05
66000 - SÉRIE INFORMATIQUE - PERPIGNAN - (68) 34.00.11
67150 - E.T.S. FRITSCHE - ERSTEIN - (88) 98.03.51
68000 - E.I.B. - COLMAR - (89) 93.68.35
69003 - B.I.M.P. - LYON (7) 860.81.7
69400 - MICRO INFORM.BEAUJOLAISE - VILLEFRANCHE-S/SAONE -
(74) 68.44.92
70000 - ÉLECTRO BOUTIQUE - VESOUL - (84) 76.49.52 +
71100 - AVENIR ÉLECTRONIQUE - CHALON/SAONE - (85) 48.73.35
71400 - C.H.B. ÉLECTRONIQUE - AUTUN - (85) 50.70.26
72000 - MICROTIQUE AESCULAPPE - LE MANS - (43) 24.97.80
73100 - L'ORDINATEUR - AIX-LES-BAINS - (79) 88.19.07
74102 - D.S.A. MICRO - ANNEMASSE - (50) 38.31.40
75001 - VIDEO SHOP - PARIS - (1) 296.93.95
75005 - HACHETTE - PARIS - 633.84.68
75006 - DURIEZ S.A. - PARIS - 329.05.60
75008 - ÉNERGY 8 - PARIS - 293.41.33
75009 - LE JEU ÉLECTRONIQUE - PARIS - 526.62.93 / 874.43.20
75009 - LPS BUREAU - PARIS - 878.26.45
75009 - J.C.R. ÉLECTRONIQUE - PARIS - 282.19.80
75010 - GÉNÉRAL VIDEO - PARIS - 206.50.50
75010 - LOGIC STORE - PARIS - 206.72.28
75011 - COCONUT INFORMATIQUE - PARIS - 355.63.00
75011 - P.I.T.B. - PARIS - 254.38.01
75012 - ELLIX - PARIS - 307.65.58
75014 - MIDEF - PARIS - 539.98.68
75015 - J.C.S. COMPOSANTS - PARIS - 355.96.22
75015 - ILLEL CENTRE - PARIS - 554.97.48
75016 - PENTASONIC - PARIS - 524.23.16
75016 - ANTIGONE - PARIS - 743.13.41
76600 - MICRO MAX - LE HAVRE - (35) 41.77.47
76600 - V.P.C. BUREAU - LE HAVRE - (35) 42.49.21
76600 - L'ORDINATEUR - LE HAVRE - (35) 21.54.55
77000 - ÉPSILON - MELUN - 437.51.95
80000 - LOGIC - AVIENS - (22) 95.54.84
80000 - P.S.I. ÉLECTRONIQUE - TOULON - (94) 93.11.20
86011 - LISTE INFORMATIQUE - POITIERS CEDEX - (49) 41.43.86
87000 - MICROLIM - LIMOGES - (55) 34.10.12 +
89100 - MINI LOISIRS - SENS - (86) 64.41.91
89100 - LASOBIKOR YONNE - SENS (86) 64.51.26 +
91210 - VIDEOTRONIC - DRAVEIL - 940.28.30
92100 - AXIOME - BOULOGNE - 604.02.21
92100 - OLIG - BOULOGNE BILL. - (1) 605.05.59
94100 - DIXMA - SAINT-MAUR - 885.98.92
98000 - MICROTEK 2 - MONACO - (93) 30.67.67 +
88009 - A.V.M. - ÉPINAL (29) 82.14.97

SUCCÈS OBLIGE

Le deuxième d'une
longue série de guide
des logiciels.
Plus d'un tiers de nou-
veautés.

AU SOMMAIRE :

— Une sélection de 416
programmes en Anglais
ou en Français pour :
APPLE - ATARI - COMMODORE
V20 et C64 - EPSON HX 20 -
ORIC 1 et ORIC ATMOS - IBM PC
- SINCLAIR ZX81 et SPECTRUM
TRS 80 - THOMSON TO 7 -
HECTOR.

— Les fiches techniques de
chaque programme compre-
nant :

La description précise du pro-
gramme.
Son prix moyen constaté.
Sa compatibilité avec tel ou tel
micro.

— En plus vous trouverez :
Des conseils pour choisir et acheter
le programme que vous cherchez.
Des index pour trouver facilement
ce que vous cherchez.

**EN VENTE 15 F CHEZ VOTRE
DISTRIBUTEUR OU 15 F + 5 F
DE PORT EN RENVOYANT LE
COUPON CI-DESSOUS.**



SPiD

LA HAUTE FIABILITÉ

BON DE COMMANDE A RENVoyer A SPiD - 39, RUE V.-MASSÉ - 75009 PARIS

Je désire recevoir le "GUIDE DES LOGICIELS" Printemps 1984
Je joins 20 F en chèque (15 F + 5 F de port) en règlement.

Nom

Adresse

Code et ville



pom's n°12

Sommaire	Page	Langage *	Niveau **
Disque virtuel 16K par Michel Haag	7	A	P-T
Charts Unlimited à l'essai par Guy Lapautre	11		T
Le Calculateur entier par Jacques Duma	15	A	P-T
Exp (Pi*Sqr (163)) est-il entier ? par Olivier Herz	21		T
Analyse de Magic Mailer par Guy Lapautre	22		T
Initiation à l'assembleur (2) par Gérard Michel	23	B-A	T
Patch du DOS 3.3 par Yvan Koenig	28	B	P-T
Des tableaux de dimension variable par Gérard Michel	29	A	P-T
& ONERR GOTO par Dimitri Auer	33	A	M-T
Programme assembleur à la fin d'un programme BASIC par Roland Jost	35	B	M-T
Des trucs pour Apple Writer II et //e par Hervé Thiriez	39		T
Adaptez Apple Writer 1.1 à l'Apple //e par Jacques Leroy	40		T
Un éditeur de shapes par Thierry Le Tallec	41	B-A	P-T
Nombres flottants en langage machine par Erick Ringot	45	B-A	P
Un mois avec le Macintosh par Hervé Thiriez	53		T
L'Apple //c est né ! par Guy Mathieu	54		T
PEEKs et POKEs en Pascal par Jacques Pino	55	P	T
Les disquettes de Pom's par Olivier Herz	56		T
Imprimez les codes ESC de votre PLE par Pascal Fontanier	59	B	T
Tracé rapide de cercles par Dominique Bernardi	61	A	M-T
La micro-informatique au Japon par Olivier Herz	62		T
Pom's a vu MEM/TERM par Guy Lapautre	63		T
La famille Apple // par Daniel Garric	67		T
Micro-informations par Jean-Michel Gourévitch	69		T
Courrier des lecteurs par Alexandre Duback	71		T
Courrier des clubs	73		T
Bibliographie	73		T

* Langage : B(asic) - A(ssembleur) - P(ascal). (B) signifie : relatif au BASIC.

** Niveau : D(ébutant) - M(oyen) - P(rofessionnel) - T(ous).

P-T signifie : programme utilisable par les débutants, mais dont la compréhension est de niveau "Professionnel".

Les annonceurs

ACER : p. 44 / ALPHA SYSTEMES : p. 4 / APPLE : p. 75 / B.F.I. : p. 6 / CEDIC-NATHAN : p. 51 / COMPUTER 3 : p. 38 / HELLO : p. 57 / M.B.D.C. : p. 72 - MICRO EXPO : p. 52 / MICRO-PERIPH : p. 66 / L'ORDINATEUR INDIVIDUEL : p. 76 / PENTASONIC : p. 14 / P.S.I. Diffusion : p. 64-65 / SATELCOM : p. 58 / SICOB : p. 10 / SPID : p. 2 / TELECOMPO : p. 38

Éditions MEV — 64-70 rue des Chantiers — 78000 Versailles

Directeur de la publication : Hervé Thiriez. Imprimerie Rosay, 94300 Vincennes. Imprimé en France. Dépôt légal : 2^e trimestre 1984.



LE FLASH !!! ON EST FAIT L'UN POUR L'AUTRE...

Nos cartes coprocesseurs permettent de transformer littéralement un Apple // en lui donnant une vitesse 2 à 20 fois supérieure en Applesoft ou en Pascal et/ou une compatibilité en assembleur ou en langage évolué avec les nouveaux micros du marché.

LA PERFORMANCE



carte AD 8088

16 bits, CP/M-86
APPLESOFT ultra rapide !

Sans modifier une ligne des programmes APPLESOFT, la vitesse est multipliée par un coefficient de 2 à 4. La carte permet d'utiliser CP/M-86 bientôt MS/DOS, le processeur arithmétique rapide 8087 a une vitesse époustouflante. Mémoire extensible à 192 Ko. Fonctionnement simultané du 6502 et de plusieurs cartes AD 8088 possible.



carte 6502 C

100% compatible
4 fois plus rapide.

Quel que soit le programme, quel que soit le langage, le microprocesseur 6502 C (3.6 MHz) divise par 3,6 le temps d'exécution. Aucune modification ni matériel, ni logiciel. La carte comporte 64 Ko de mémoire ; 2 versions : Apple II +, et Apple //e.



carte 68000

Vitesse, puissance. Avenir.

Le microprocesseur le plus puissant. 3 versions 8, 12.5 et 14 MHz. 128 Ko de mémoire sur la carte extensible à 812 Ko. Entièrement compatible avec APPLESOFT et DOS 3.3 qu'elle accélère dans un rapport de 3 à 20 selon les programmes. UCSD-p System complet disponible en version IV.I avec PASCAL, BASIC, FORTRAN. Développements possibles sous UCSD pour IBM PC et XT, VICTOR S1, RAINBOW 100, GOUPIL III, etc., sans recompilation. Le 68000 équipe LISA et MC INTOSH !

alpha
SYSTEMES

departement
diffusion

29, bd gambetta - 38000 grenoble
tél. (76) 43.19.97

DISTRIBUTEURS DANS TOUTE LA FRANCE

LA MICRO SANS FRONTIERE

D'aucuns annonçaient la mort lente d'Apple, étant donné la concurrence de l'IBM PC et des nouveaux 16 et 32 bits à haut rapport performance/prix, tels que le QL de Sinclair. Or, tout d'abord, on constate que les nouveaux produits concurrents n'ont pas toujours une naissance sans problème : ainsi, le QL connaît-il en ce moment des difficultés avec des bugs dans son QDOS et des erreurs de conception dans un de ses circuits spéciaux. En outre, la sortie du Macintosh a prouvé qu'il fallait encore compter avec Apple : la demande est telle aux Etats-Unis qu'il est question de lancer une campagne de publicité sur le thème « Nous savions qu'il s'en vendrait un million ; nous ne savions pas que cela se ferait en deux semaines... ».

Ce douzième numéro de Pom's vous apporte la seconde partie de la rubrique « Apprentissage de l'assembleur » assurée par **Gérard Michel**. Certains m'ont fait déjà fait savoir, à la suite de la première leçon, que cette initiation était la meilleure qu'ils aient jamais vu. Merci pour Gérard. Sur le plan pédagogique, **Erick Ringot** épluche aussi pour vous la façon dont les nombres flottants sont gérés par votre ordinateur favori.

Grâce au programme de **Michel Haag**, tout le monde pourra, sur Apple II Plus et //e, transformer la carte d'extension 16K en lecteur de disquettes virtuel. Il est alors possible d'y loger tous ces utilitaires d'usage fréquent, ou de s'en servir pour y placer les tables de référence des programmes Basic. En compensation, les « Pascaliens » apprendront avec **Jacques Pino** comment faire des PEEKs et des POKEs en Pascal.

Comme d'habitude, vous trouverez dans ce Pom's des tas de programmes tous plus utiles les uns que les autres. **Roland Jost** montre comment cacher des programmes en assembleur à la fin des programmes Basic. **Dominique Bernardi** illustre sa façon de tracer rapidement des cercles en écran haute résolution. **Dimitri Auer** vous permet enfin de créer des tableaux à dimension variable en Basic. **Yvan Koenig** propose un patch du DOS 3.3 permettant d'éliminer plusieurs de ses bugs et limitations. **Thierry Le Tallec** complète ses contributions antérieures par un éditeur de shapes de toute beauté. Les utilisateurs d'Apple Writer sont gâtés cette fois-ci : **Jacques Leroy** transforme Apple Writer I pour qu'il fonctionne sur un //e, et je fournis de nombreux trucs d'utilisation pour Apple Writer II et //e. Pour terminer, vous verrez avec **Pascal Fontanier** comment imprimer les codes ESCape de votre imprimante.

Pom's vous offre encore dans ce numéro une contribution importante avec le Calculateur Entier de **Jacques Duma** ; vous pourrez effectuer avec votre Apple II des calculs avec 500 chiffres significatifs. Comme il fallait 15 pages rien que pour le listing source en assembleur, nous n'avons pu le mettre que sur la disquette d'accompagnement ; par contre, les lecteurs courageux (ou obstinés) pourront reprendre le programme à partir du code objet publié avec l'article... **Olivier Herz** utilise le Calculateur Entier pour s'attaquer à la question métaphysique que tous se posent : $EXP(PI * SQR(163))$ est-il entier ?

Les bancs d'essai restent au goût du jour : **Guy Lapautre** a décortiqué Magic Mailer et Charts Unlimited ; il nous présente en outre le nouveau service MEM/TERM. L'intégrité de nos bancs d'essai est garantie par le fait que nous ne recevons pas de publicité en échange de nos analyses.

La sortie du Macintosh signifie-t-elle la disparition progressive de la ligne Apple II et //e ? L'article de **Daniel Garric** et la rubrique Micro-Informations, tenue à présent par **Jean-Michel Gourévitch**, vous prouvent le contraire : les nouveaux produits //c et //x (?) vont compléter la gamme en respectant autant que possible le sacro-saint principe de la compatibilité. **Olivier Herz** complète ces réflexions sur le marché de la micro-informatique en nous parlant du Japon.

Hervé Thiriez

Ont collaboré à ce numéro : Dimitri Auer – Dominique Bernardi – Jean-Michel Boyer – Alexandre Duback – Jacques Duma – Pascal Fontanier – Daniel Garric – Jean-Michel Gourévitch – Michel Haag – Olivier Herz – Roland Jost – Yvan Koenig – Guy Lapautre – Jacques Leroy – Thierry Le Tallec – Guy Mathieu – Gérard Michel – Jacques Pino – Erick Ringot – Hervé Thiriez. **Rédacteurs :** Alexandre Avrane – Gérard Michel. **Dessins :** Laurent Bidot. **Directeur de la publication - rédacteur en chef :** Hervé Thiriez - **Siège social et abonnements :** Editions MEV - 49, rue Lamartine - 78000 Versailles - **Rédaction et courrier des lecteurs :** 59, Bd Glatigny - 78000 Versailles - Tél. : (3) 951.24.43.

Régie publicitaire : Force 7 - Anne Jourdan - 5, place du Colonel Fabien - 75010 Paris - Tél. : 240.22.01.

Diffusion auprès des boutiques et librairies : PSI Diffusion - BP 86 - 77402 Lagny-sur-Marne Cedex.

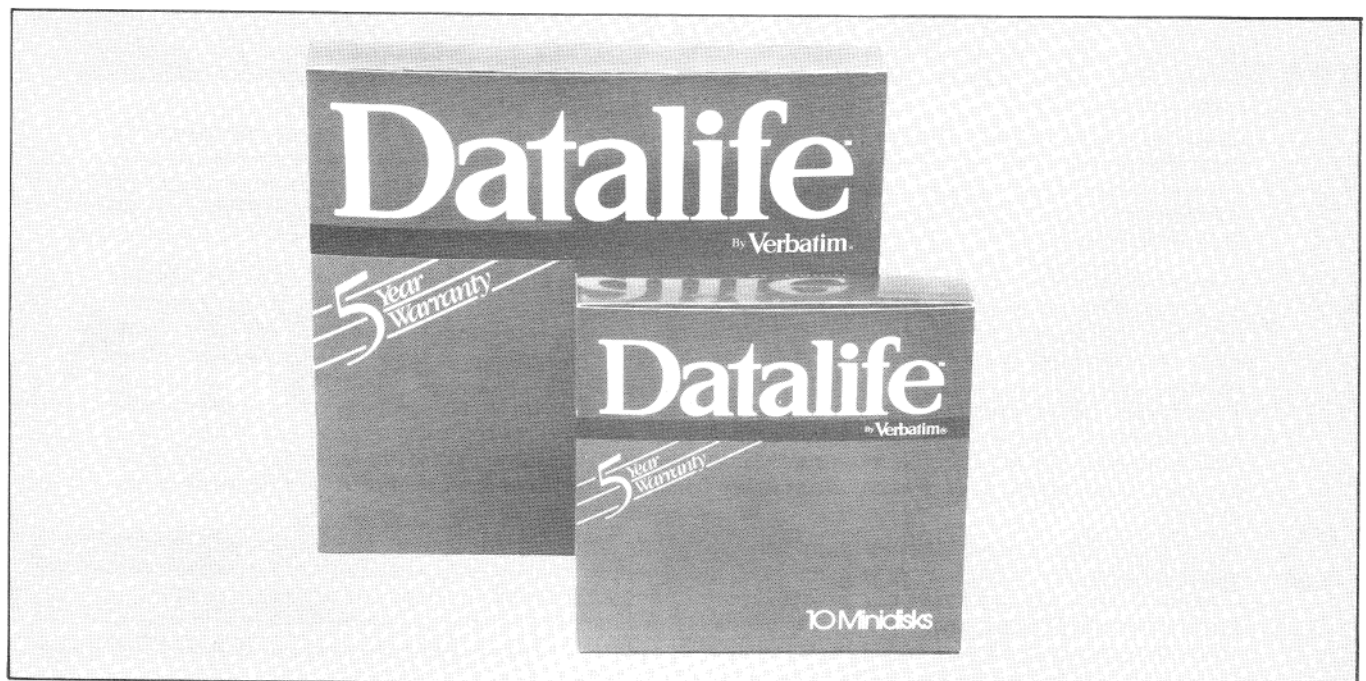
Composition : Télécompo - 13-15, avenue du Petit Parc - 94300 Vincennes - Tél. : 328.18.63.

Impression : Rosay - 47, avenue de Paris - 94300 Vincennes - Tél. 328.18.63.

Datalife

BY Verbatim®

DISQUETTES ET MINI DISQUETTES TOUTES CONFIGURATIONS



- Certification unitaire 100% sans erreur.
- Durée de vie : 30 millions de révolutions (standard de l'Industrie 3,5 millions de révolutions).
- Anneau de renforcement en standard sur le 5 1/4 ''.
- 5 1/4 '' en 48 et 96 TPI, simple et double face.

BFI ELECTRONIQUE - 9 RUE YVART -
75015 PARIS.
Tél. 533-01-37.

Disque virtuel 16 K

Michel Haag

Ce programme transforme les 16K de la carte langage en pseudo-disquette (disque virtuel) : c'est-à-dire que la carte langage se comportera comme un lecteur de disquettes à accès très rapide (et à capacité restreinte). Il est transparent au Basic et au File Manager du DOS. Le programme FID fonctionne sans difficulté avec la pseudo-disquette ainsi formée par les 16 K de la carte langage.

Exemples d'utilisation

On a souvent besoin, lors de la mise au point d'un programme, de plusieurs utilitaires. Ceux-ci peuvent être disponibles sur la pseudo-disquette.

Certains programmes doivent fréquemment accéder à un fichier texte pour le modifier ou le compléter. Si on le transfère sur la pseudo-disquette, il ne sera plus nécessaire d'attendre que la disquette se mette à tourner et qu'elle veuille bien présenter le secteur voulu sous la tête de lecture. Les lectures et écritures prendront moins de temps, surtout si le fichier est à accès direct.

Remarque aux débutants

Si les paragraphes expliquant le fonctionnement du programme vous dépassent, vous pouvez sauter directement à la partie "Utilisation" de cet article. On peut en effet parfaitement utiliser ce programme sans pour autant comprendre son fonctionnement (combien de conducteurs connaissent la mécanique ?).

Fonctionnement du programme

A chacune des 64 pages de MEV de la carte langage est associé un secteur. Les trois premières pages de la première tranche de 4K de la carte langage, comprises entre \$D000 et \$D2FF, correspondent aux trois premiers secteurs de la piste 17. Ils contiennent la VTOC (Volume Table Of Contents) en page \$D0 et un catalogue (directory) de deux secteurs. Les 13 autres pages de cette tranche correspondent aux secteurs 3 à 16 de la piste 3. Les pistes 4, 5 et 6 sont associées à la seconde tranche de 4K et à la zone \$E000-\$FFFF de la carte langage.

Le programme a été scindé en deux modules : un module d'initialisation, RWLCINIT (utilisé une seule fois) et un module d'accès aux "pistes" et "secteurs" de la carte langage. Ce second module, baptisé RWLC, reste en mémoire après l'initialisation.

Module RWLCINIT

Ce premier module construit la VTOC et un catalogue de 2 secteurs

```
1 ;
2 ;*****
3 ;*
4 ;*          RWLC          *
5 ;*
6 ;*          MICHEL  HAAG (NOV 83)  *
7 ;*
8 ;*****
9 ;
10          ORG $BEBO
11          OBJ $800
12 ;
13 ;EMULATEUR DE DISQUETTE
14 ;
15 ;  UTILISE LA CARTE LANGUAGE POUR SIMULER
16 ;  UNE DISQUETTE DE 4 PISTES
17 ;
18 IOBP      EPZ $48
19 PTR       EPZ $2A          ;PAGE = "SECTEUR"
20 BUFPTR    EPZ $3E
21 SECT      EPZ $2D
22 TRACK     EPZ $2E
23 ;
24 TRK       EQU $4
25 SEC       EQU $5
26 BUF       EQU $8
27 CMD       EQU $0C
28 DRV       EPZ $02
29 ;
30 RWLC      STY IOBP
31           STA IOBP+1
32           LDY #DRV          ;TOUT APPEL DE RWTS EST
33           LDA (IOBP),Y      ;AIGUILLE SUR RWLC. LE
34           CMP #2            ;CONTROLE EST REDONNE A RWTS
35           BEQ RWLC1         ;SI LE LECTEUR EST D1
36           JMP $BDO3
37 ;
38 ;TRANSFERT DE POINTEURS EN PAGE ZERO
39 ;
40 RWLC1     LDY #BUF          ;BUFFER DANS BUFPTR
41           LDA (IOBP),Y
42           STA BUFPTR
43           INY
44           LDA (IOBP),Y
45           STA BUFPTR+1
46           LDY #SEC          ;SECTEUR DANS SECT
47           LDA (IOBP),Y
48           STA SECT
49           LDY #TRK          ;PISTE DANS TRACK
50           LDA (IOBP),Y
51           STA TRACK
52 ;
53 ;DETERMINATION DU BLOCK DE 4K
54 ;
55           CMP #$11          ;VTOC OU DIR.
56           BEQ RAM1
57           CMP #3
58           BEQ RAM1          ;PISTE 3 = 1ERS 4K DE LA L.C..
59           BMI ERR
60           CMP #7
61           BPL ERR
62 RAM2     LDA $C083          ;PISTE 4,5 OU 6 :
63           LDA $C083          ;ON COMMUTE LA PREMIERE ZONE
64           LDA #00           ;DE 4K DE LA CARTE
65           STA PTR
66           LDA TRACK         ;PTR = DO+$10*(TRACK-4)+SECT
67           SEC
68           SBC #$04
69           ASL
```


sur la carte langage. Il procède ensuite à quelques modifications du DOS. Sur la table d'occupation du VTOC, seuls les secteurs existant effectivement sont signalés comme étant libres. RWLCINIT met ensuite un BRK (\$00) en \$BEAF et inhibe la commande DOS INIT. Il place enfin un JMP \$RWLC à l'entrée de RWTS en \$BD00.

Module RWLC

Ce second module débute en \$BEB0 et comporte 144 octets. Il recouvre donc en partie la commande INIT du DOS. Tout appel de RWTS est aigillé vers RWLC. Si le lecteur concerné est le lecteur 1, RWLC redonne le contrôle à RWTS. Si c'est le lecteur 2, RWLC détermine la page de la carte langage qui est invoquée et procède aux échanges demandés entre cette page et le buffer pointé par l'IOB. Si les appels de RWTS sont effectués par le File Manager du DOS, il sera automatiquement tenu compte de la VTOC créée par RWLCINIT. Le File Manager ne fera référence qu'à des secteurs existant effectivement sur la carte langage. Si par contre on ne souhaite pas tester soi-même les numéros de secteur et piste, il faut éviter d'accéder directement à la carte langage par RWTS, qui continuera cependant à donner des résultats parfaitement normaux sur le lecteur 1.

Utilisation du programme

Ce programme ne peut fonctionner que si une carte langage est installée, et non déjà utilisée par le programme (ou un langage, ou un DOS relogé). Il faut faire, dans l'ordre :

BLOAD RWLC.OBJ
BRUN RWLCINIT.OBJ

On peut à la place exécuter un fichier EXEC qui contiendrait ces deux commandes. La pseudo-disquette s'est maintenant substituée au second lecteur qui, par conséquent, ne peut plus être utilisé par le programme. Tout ordre du DOS suivi de ",D2" invoquera dorénavant la pseudo-disquette. De plus, comme dans le cas d'un lecteur réel, le disque utilisé par défaut restera le disque virtuel jusqu'au prochain appel du drive 1 (par exemple avec CATALOGD1). Les programmeurs possédant deux lecteurs pourront déconnecter RWLC et retrouver le second lecteur par POKE 48825,0. Pour reconnecter RWLC, il suffira de faire POKE 48825,2.

Pour échanger des fichiers entre la disquette 1 et la pseudo-disquette, il sera commode d'utiliser le FID. Enfin un dernier avertissement : les informations contenues sur la carte lan-

```

70 ASL
71 ASL
72 ASL
73 ADC #$D0
74 ADC SECT ;TOUJOURS BRANCHE
75 STA PTR+1 ;PISTE 3 OU $11 :
76 BCC EXQ
77 RAM1 LDA #$00 ;CALCUL DE LA PAGE
78 STA PTR
79 CLC
80 LDA SECT
81 ADC #$D0 ;COMMUTE LA 2IEME ZONE
82 STA PTR+1
83 LDA $C08B ;"LECTURE" OU "ECRITURE"
84 LDA $C08B
85 EXQ LDY #CMD
86 LDA (IOBP),Y
87 CMP #0
88 BEQ RET
89 CMP #1
90 BEQ READ
91 CMP #2 ;PROVOQUE UN I/O ERROR
92 BEQ WRITE ;COMMUTE LA MEM
93 ERR SEC
94 RET LDA $C082
95 LDA $C08A
96 RTS ;LIT UN "SECTEUR"
97 ;
98 READ LDY #$00
99 BR1 LDA (PTR),Y
100 STA (BUFPTR),Y
101 DEY
102 BNE BR1
103 CLC
104 BCC RET
105 WRITE LDY #$00 ;ECRIT UN "SECTEUR"
106 BW1 LDA (BUFPTR),Y
107 STA (PTR),Y
108 DEY
109 BNE BW1
110 CLC
111 BCC RET
112 LENGTH = *-RWLC
113 END

```

```

1 ;
2 ;*****
3 ;* *
4 ;* RWLCINIT *
5 ;* *
6 ;* MICHEL HAAG (NOV 83) *
7 ;* *
8 ;*****
9 ;
10 ORG $8000
11 OBJ $803
12 ;
13 ;
14 PTR EPZ $2A
15 ;
16 ;
17 ; INITIALISE UN VTOC ET UN DIRECTORY SUR LA PREMIERE
18 ;ZONE DE 4K DE LA CARTE, PAGE D0, D1 ET D2. MODIFIE
19 ;LE DOS.
20 ;
21 INIT:
22 ;
23 LDA $C08B ;VALIDE LA 1ERE TRANCHE DE 4K
24 LDA $C08B ;DE MEV DE LA CARTE LANGAGE
25 LDA #0 ;MET LE VTOC ET LE DIR A 0
26 STA PTR
27 LDA #$D0
28 STA PTR+1

```


gage sont volatiles et disparaissent avec l'extinction de l'appareil; il faudra donc penser (en fin de séance) à transférer sur une disquette réelle le contenu de la carte langage si l'on souhaite le conserver.

Pour les adresses du DOS et la construction de la VTOC, voir Beneath Apple DOS, de Worth et Lechner. Pour l'inhibition de la commande INIT, voir "Une incursion dans les mystères du DOS" (Pom's 2).



Récapitulation RWLC.OBJ

```

BEB0- 84 48 85 49 A0 02 B1 48
BEB8- C9 02 F0 03 4C 03 BD A0
BEC0- 08 B1 48 85 3E C8 B1 48
BEC8- 85 3F A0 05 B1 48 85 2D
BED0- A0 04 B1 48 85 2E C9 11
BED8- F0 25 C9 03 F0 21 30 40
BEE0- C9 07 10 3C AD 83 C0 AD
BEE8- 83 C0 A9 00 85 2A A5 2E
BEF0- 38 E9 04 0A 0A 0A 0A 69
BEF8- D0 65 2D 85 2B 90 11 A9
BF00- 00 85 2A 18 A5 2D 69 D0
BF08- 85 2B AD 8B C0 AD 8B C0
BF10- A0 0C B1 48 C9 00 F0 09
BF18- C9 01 F0 0C C9 02 F0 14
BF20- 38 AD 82 C0 AD 8A C0 60
BF28- A0 00 B1 2A 91 3E 88 D0
BF30- F9 18 90 ED A0 00 B1 3E
BF38- 91 2A 88 D0 F9 18 90 E1

```

Récapitulation RWLCINIT.OBJ

```

8000- AD 8B C0 AD 8B C0 A9 00
8008- 85 2A A9 D0 85 2B A9 00
8010- A0 00 91 2A 88 D0 FB E6
8018- 2B A5 2B C9 D3 D0 EF A9
8020- D0 85 2B A0 06 B9 8F 80
8028- 91 2A 88 10 F8 A9 7A 8D
8030- 27 D0 A9 03 8D 30 D0 A9
8038- 01 8D 31 D0 8D 37 D0 A9
8040- 23 8D 34 D0 A9 10 8D 35
8048- D0 A2 04 A9 FF A0 00 99
8050- 44 D0 C8 99 44 D0 C8 C8
8058- C8 CA D0 F3 A9 F8 8D 45
8060- D0 E6 2B A0 01 A9 11 91
8068- 2A C8 A9 02 91 2A 18 AD
8070- 82 C0 AD 8A C0 A0 03 B9
8078- 8B 80 99 00 BD 88 10 F7
8080- A9 00 8D AF BE A9 5B 8D
8088- 84 A8 60 4C B0 BE EA 04
8090- 11 01 03 00 00 FE

```

```

29 B3 LDA #0
30 LDY #00
31 B2 STA (PTR),Y
32 DEY
33 BNE B2
34 INC PTR+1
35 LDA PTR+1
36 CMP #D3
37 BNE B3
38 ;
39 ;VTOC
40 ;
41 LDA #D0 ;8 PREMIERS OCTETS DU VTOC
42 STA PTR+1
43 LDY #06
44 B1 LDA VTOCTBL,Y
45 STA (PTR),Y
46 DEY
47 BPL B1
48 LDA #S7A ;OCTETS 8 A S37
49 STA $D027
50 LDA #S03 ;DERNIERE PISTE ALLOUEE
51 STA $D030
52 LDA #S01
53 STA $D031
54 STA $D037 ;255 OCTETS/SECT
55 LDA #S23 ;S23 (35) PISTES
56 STA $D034
57 LDA #S10 ;S10 SECTEURS/PISTE
58 STA $D035
59 ;
60 ;TABLE D'OCCUPATION DU VTOC
61 ;
62 LDX #4 ;LES PISTES 3, 4 ET 5
63 LDA #FF ;SONT LES SEULES LIBRES
64 LDY #0
65 B4 STA $D044,Y
66 INY
67 STA $D044,Y
68 INY
69 INY
70 INY
71 DEX
72 BNE B4
73 LDA #SFB ;SAUF LES SECTEURS 0,1 ET 2
74 STA $D045 ;DE LA PISTE S11
75 ;
76 ;DIRECTORY
77 ;
78 INC PTR+1 ;PROCHAIN SECTEUR DU DIR.
79 LDY #1 ;PISTE S11, SECTEUR 2
80 LDA #S11
81 STA (PTR),Y
82 INY
83 LDA #2
84 STA (PTR),Y
85 CLC
86 LDA $C082 ;COMMUTE LA MEM
87 LDA $C08A
88 ;
89 ;MODIFIE LE DOS
90 ;
91 LDY #03 ;AIGUILLE TOUT APPEL
92 B5 LDA PATCHTBL,Y ;DE RWTS SUR RWLC
93 STA $BDOO,Y
94 DEY
95 BPL B5
96 LDA #S00 ;EMPECHE UNE INITIALISATION
97 STA $BEAF
98 LDA #S5B ;PROVOQUE UN SNTX ERR POUR
99 STA $A884 ;TOUTE UTILISATION DE INIT
100 RTS
101 PATCHTBL JMP $BEBO ;MODIFICATION DU DOS
102 NOP
103 VTOCTBL BYT $4,$11,$1,$3,0,0,$FE
104 LENGTH = *-INIT
105 END

```

SPECIAL SICOB UN SICOB TRES SPECIAL.

14-19 MAI
CNT-PARIS
LA DEFENSE

PROGICIELS,
MINI, MICRO-ORDINATEURS
(JOURNÉES GRAND PUBLIC: 18 ET 19 MAI)

A 319

PUBLICIS

Information: SICOB (1) 261.52.42 - 4, place de Valois - 75001 Paris



Charts Unlimited à l'essai

Guy Lapautre

Charts Unlimited est un éditeur graphique (ou plus précisément un aide à la réalisation de dessins) vendu \$195 par Graphware (5084 Mosiman Road - Middletown - OH 45042). Il se présente sous la forme désormais classique d'un classeur rigide, avec documentation et une disquette. Ce programme n'est pas encore, à notre connaissance, vendu en France.

Il est destiné à créer et à imprimer des schémas et dessins tels que : organigrammes, ordinogrammes de traitement, synoptique de processus industriels, etc. Il ne peut pas servir directement à représenter des séries numériques comme le font des logiciels tels que Visiplot, Décisionnel Graphique (essai dans le Pom's 10), Business Graphics ou PFS-Graph.

Les avantages et inconvénients de Charts Unlimited sont évalués en fonction de ce qu'un utilisateur est en droit d'attendre d'un tel logiciel, avec parfois référence implicite ou explicite à d'autres logiciels de même objectif. Ils sont présentés sous 7 têtes de chapitres :

- 1 - Apprentissage du logiciel
- 2 - La "page" écran
- 3 - Fonctions de dessin disponibles
- 4 - Objets et symboles
- 5 - L'édition sur imprimante
- 6 - Utilitaires : gestion des disques, liaisons externes
- 7 - Documentation

Apprentissage du logiciel

L'abord du logiciel est assez aisé : il se fait grâce à un "Tutorial" qui consiste à tracer un organigramme d'entreprise, et permet de couvrir la plupart des fonctions du logiciel.

Une fonction "AU SECOURS", invoquée par la touche "?", propose un simple synoptique des commandes disponibles.

On pourra s'étonner du choix fait pour les lettres plus ou moins mnémotechniques permettant l'appel des commandes (même en Anglais) : E (pour "ERASE") ou D (pour "DASH LINE"), d'accord, mais B pour désigner des "OBJETS STANDARDS" ou Y pour définir des objets...

Il faut s'habituer au mode de déplacement du curseur : 8 lettres pour les 8 directions dans toutes les commandes, sauf en mode texte où l'on retrouve les 4 flèches de direction (pour l'Apple //e).

Il est assez commode de configurer son système imprimante/interface ... à condition que ces deux matériels figurent dans les listes proposées. Sinon, quelques tâtonnements supplémentaires semblent conduire en général à un résultat satisfaisant.

Avantages : Simplicité d'emploi - Nombreuses fonctionnalités d'un abord aisé - Ecran clair.

Inconvénients : Codes de commande pas toujours évidents - Quelques problèmes dans le maniement du curseur.

Synthèse : Dans l'ensemble, pas de gros reproches, malgré quelques inconvénients mineurs. Un "Tutorial" bien fait. On commence à "se débrouiller" avec Charts Unlimited en 2 heures environ.

La "page" écran

L'écran est une fenêtre de 37 colonnes sur 19 lignes dans une page de 123 colonnes sur 90 lignes. Des échelles placées en haut et à gauche de l'écran montrent à quel endroit on se trouve dans la page.

En bas, une ligne de commande permet de savoir quelle est la fonctionnalité en cours. Chaque commande est appelée par une lettre code. On en ressort TOUJOURS par la touche ESC.

Nous avons déjà parlé du positionnement du curseur. Sauf en mode texte, il se fait par le groupe assez traditionnel de lettres "U I O J K N M <" pour obtenir les 8 directions de déplacement; il vaut donc mieux être en mode US sur un clavier //e. Les 4 flèches de direction (version //e) servent à un déplacement rapide, jusqu'aux bords de l'écran.

En mode texte, les déplacements se font avec les 4 flèches; RETURN renvoie à la ligne suivante, DEL efface le caractère sous le curseur (et non le précédent...) et CTRL-I crée un espace (pour insertion).

On a souvent besoin de désigner une zone rectangulaire de l'écran : sauvegarde, duplication, ... Ceci se fait commodément en se plaçant successivement dans les angles supérieur gauche et inférieur droit, et en "marquant" ces angles par RETURN.

Par ailleurs, une fonction "VIEW" permet d'obtenir, dans un petit rectangle en bas et à droite, un fac-similé réduit de la page entière.

Avantages : Ecran clair - Vision très commode de la page entière - Mode unique d'entrée et sortie de chaque commande - Puissance de la notion de "zone".

Inconvénients : Quelques petits problèmes de déplacement du curseur - La désignation de zones est parfois une commodité ... dangereuse : on peut, en l'omettant, perdre complètement le résultat d'une séance de travail.

Synthèse : Là encore, les avantages l'emportent sur les inconvénients. On regrettera toutefois qu'il n'y ait pas un verrouillage de la fonction de sauvegarde quand on a omis de désigner préalablement la zone à sauvegarder.

Les fonctions de dessin disponibles

Elles sont désignées ci-dessous par leur équivalent en français.

TEXTE : permet l'écriture de tous les caractères imprimables. L'écriture ne peut pas traverser un dessin déjà fait. Elle est limitée à l'intérieur d'une case, ou à une zone vide,...

LIGNES : droites pointillées (Dash) ou pleines (Solid) tracées grâce aux 8 touches de direction. Quelques petits problèmes de jonction selon les sens successifs de tracé.

OBJETS : on désigne par ce vocable des éléments de dessin pré-établis. Il existe un jeu dit "standard" de 36 éléments, qu'on peut manipuler, agrandir, déformer, ... (un mini "Lisa-Draw", par anticipation). L'utilisateur peut créer autant de jeux qu'il le souhaite (on trouve une certaine analogie avec le logiciel de traitement de texte Gutenberg - voir Pom's 11). Mais les fonctions d'agrandissement ne sont pas disponibles.

SYMBOLES : analogues aux objets, au nombre de 26, non transformables. L'utilisateur ne peut pas créer ses propres jeux.

DUPLICATION : reproduction, 1 ou n fois, d'une "zone", à intervalles équidistants sur une ligne (encore une idée reprise et développée dans Lisa-Draw).

EFFACEMENT : d'un point, ou d'une zone.

Avantages : Nombreuses fonctionnalités que l'on ne retrouve qu'avec des outils plus puissants mais sensiblement plus coûteux.

Inconvénients : Ils ne seraient sans doute pas cités si on n'avait pas vu plus fort depuis... mais pas sur le même matériel ni au même prix...

On pourra, à la rigueur, regretter 3 choses : l'impossibilité d'agrandir les objets utilisateur, le manque d'accès aux points élémentaires sur l'écran (voir Micro Painter, E-Z Draw ou Koala Pad), et enfin l'absence d'un outil de calcul associé.

Synthèse : Certains produits récents, sur des ordinateurs plus puissants, ou sur Apple II avec des compléments matériels et logiciels peu économiques, nous ont habitués à d'excellentes performances dans le dessin. Charts Unlimited n'atteint évidemment pas ces performances. Mais, par sa simplicité comme par son prix, c'est un outil remarquable aux nombreuses fonctionnalités.

Objets et symboles

La palette offerte comprend une planche de 36 objets dits "standard", dont beaucoup sont inspirés des dessins d'ordinogrammes, et une planche de 26 symboles variés. La fonction de création permet à l'utilisateur de créer autant de planches de 26 objets qu'il le souhaite, en partant d'objets ou symboles existants, de tracés nouveaux et de textes.

Les objets standards peuvent de plus être agrandis dans une ou deux directions (suivant les objets), et par conséquent déformés. Un cercle peut devenir un ovale, un petit carré un grand rectangle, etc. Symboles et objets utilisateur ne peuvent pas être modifiés dans leurs forme et dimensions.

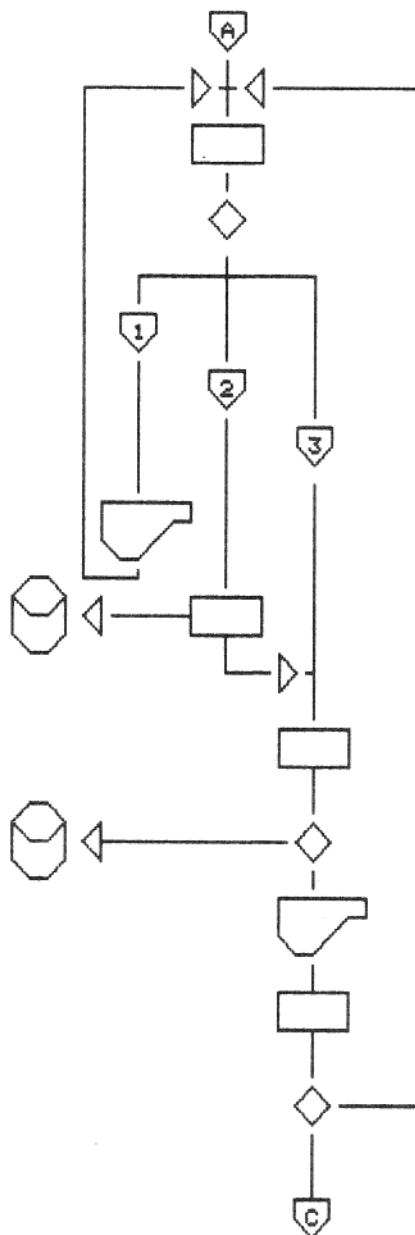
Avantages : Tracés rapides, et très simples dès lors qu'on a constitué sa (ses) propre(s) planche(s) d'objets - Appel rapide, mais seules les planches d'objets standards et de symboles peuvent être visualisées à l'écran.

Inconvénients : Il est regrettable que les objets créés par l'utilisateur ne disposent pas des mêmes fonctionnalités que les objets standards - Impossibilité de travailler au niveau du point élémentaire.

Synthèse : L'idée des "planches d'objets", qui a été reprise par des produits modernes plus dispendieux, est une des grandes forces de Charts Unlimited. Quel dommage que l'utilisateur ne puisse pas, à son gré, utiliser ses propres objets en place des objets standards.

L'édition sur imprimante

Un large choix d'imprimantes et d'interfaces est proposé à l'utilisateur, à partir d'un menu de configuration très simple à utiliser. Le problème risque de se compliquer un peu si l'imprimante et/ou l'interface qu'on possède ne figurent pas dans la liste. Mais un peu d'habitude permet en général de faire des assimilations.



Il est possible d'imprimer les graphiques, et ce en simple ou double dimension, verticalement ou horizontalement sur le papier. La simple dimension, dite "normale" est en fait très condensée (elle permet, sur papier 21x31, l'impression d'un graphe de 123 colonnes dans le sens de la largeur du papier. Mais les dessins sont en réalité raccourcis, ce qui introduit une déformation des objets et symboles.

L'impression en double dimension ne double en fait que la largeur, restituant aux objets leur forme normale. Mais il est nécessaire, pour un schéma dépassant une soixantaine de colonnes, de travailler "verticalement". A ce moment, un schéma de 123 colonnes tient juste dans une feuille de papier 21x31, mais seules les 2/3 des lignes supérieures sont imprimées.

Avantages : Un bon choix d'imprimantes et d'interfaces - Deux possibilités de dimensions d'impression et deux sens pour chacune d'elles.

Inconvénients : Une lenteur d'impression presque désespérante (les temps se comptent en longues minutes !).

Synthèse : Bonnes possibilités, pour qui est patient...

Utilitaires : gestion des disques, liaisons externes

Le temps de chargement du programme est assez long, mais on est ensuite tranquille. La configuration imprimante, étant sauvegardée sur le disque programme, est chargée avec celui-ci si on ne souhaite pas la modifier.

Les fonctions classiques d'échange avec le disque sont présentes. Bien que le travail avec deux lecteurs soit plus confortable, les transferts sont suffisamment peu nombreux pour qu'on puisse se contenter d'un seul lecteur.

A signaler un grave inconvénient de la fonction de sauvegarde : si on n'a pas pris la précaution de désigner la zone à sauvegarder, l'option par défaut n'est pas "TOUT" mais "RIEN" (pas d'avertissement : on a simplement l'impression d'un dysfonctionnement qui disparaît après deux RETURN) ... ce qui bien entendu risque de faire perdre un long temps de travail et, pour comble de malheur, efface sans pitié ce qui avait été préalablement sauvegardé sous le même nom.

Par ailleurs, notons que Charts Unlimited est un logiciel isolé, non prévu pour communiquer avec d'autres logiciels commerciaux.

Avantages : Transferts peu nombreux - Configuration imprimante chargée avec le programme.

Inconvénients : Le manque de protection, ou au minimum d'avertissement, contre une sauvegarde sans désignation préalable de zone - L'isolement du logiciel, qui ne peut pratiquer d'échange avec aucun autre.

Synthèse : Peu de remarques, si ce n'est qu'il est difficilement admissible de voir dans un logiciel plein de qualités une telle absence de sécurité.

Documentation

La documentation, en anglais seulement (il faut dire que le produit nous est arrivé directement des USA), comporte deux parties distinctes.

La première est un "Tutorial", qui enseigne l'essentiel du logiciel autour d'un exercice sur la construction de l'organigramme d'une entreprise. C'est simple et bien fait.

La seconde est classiquement présentée par fonctions. Les fonctions apparaissent dans l'ordre alphabétique de la lettre-code qui sert à les appeler, donc sans aucune logique. Si l'on se souvient par ailleurs que les mnémoniques sont parfois étranges, on conçoit qu'il eût été préférable de choisir un autre ordre !

Ne croyez pas que l'index final va apporter une aide supplémentaire : il est alphabétique lui aussi...

Enfin, pas de synoptique comme cela existe maintenant pour la plupart des logiciels.

Avantages : Bonne approche par le "Tutorial".

Inconvénients : Ordre de présentation peu logique dans la partie "Manuel de Référence" - Absence d'un tableau synoptique - Documentation

en anglais seulement, mais pour un produit encore non commercialisé en France.

Synthèse : C'est propre et bien présenté, comme c'est généralement le cas pour les documentations américaines - Mais, à l'exception de la partie apprentissage, c'est fort peu commode à exploiter.

Conclusion

En avance par rapport aux logiciels de dessin qu'on voit maintenant sur le marché, Charts Unlimited reste un produit très attrayant et très compétitif, puisqu'il n'exige aucun accessoire matériel.

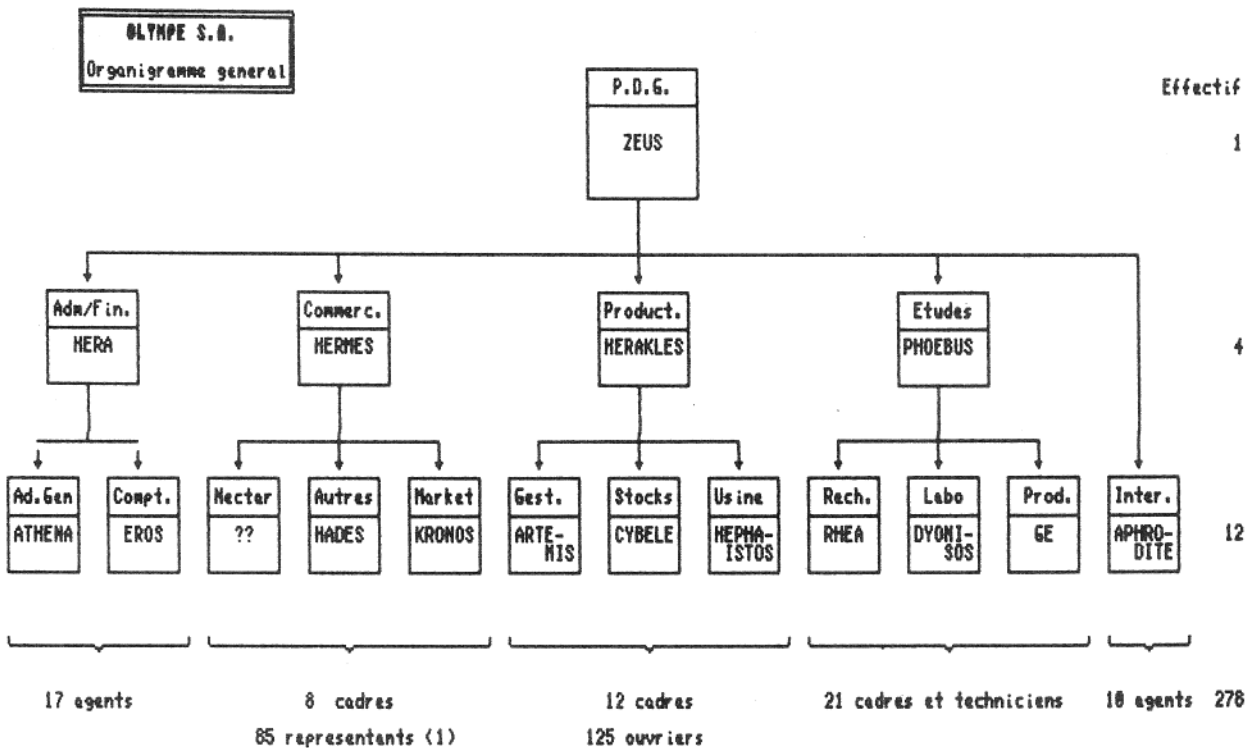
Parmi ses faiblesses : L'isolement du logiciel qui, par ailleurs, ne dispose d'aucun outil de calcul - L'impossibilité de descendre au niveau du point élémentaire - Un traitement moins complet des objets utilisateur que des objets standards - Une insuffisance de sécurité au niveau de la sauvegarde. Et, à un moindre titre : Une impres-

sion extrêmement lente - Un certain érotisme dans la désignation des commandes - Une documentation peu facile à exploiter.

Parmi ses forces : De très nombreuses fonctionnalités, capables de satisfaire un utilisateur exigeant - Un écran clair, avec rappel possible d'un schéma complet pour contrôle - D'intéressantes solutions d'impression - Une approche aisée et un apprentissage rapide - De très utiles notions de zones, d'objets et de symboles. Et, à un moindre titre : Un bon tutorial - Une agréable facilité de configuration du système d'impression.

Pour les possesseurs d'Apple //e

Attention à la touche "CAPS LOCK", qui doit toujours être enfoncée (sauf en mode texte) : dans le cas contraire, le logiciel "ne comprend rien", refuse toute entrée sur la ligne réservée et ne prévient pas (mais ne va pas jusqu'à se bloquer).

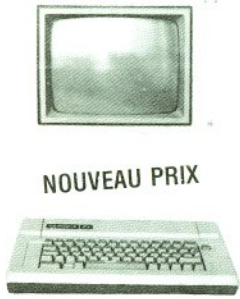


(1) dont environ:
1/2 en GRECE
1/2 à l'étranger

Effectif total : 295
Etabli le 31/1/1984
Le P.D.G. signe ZEUS

Impression CHARTS UNLIMITED

DUO DISK PROMOTION APPLE II



NOUVEAU PRIX

Esthétique, rapide, silencieux, le DUO DISK vous propose des drives plus modernes pour un prix identique à celui de 2 lecteurs séparés. Capacité 286 K

Comprendant : 1 UC APPLE IIE
1 MONITEUR APPLE
1 DUO DISK
1 APPLE FAN

16380 F

Comprendant :
* 1 UC MACINTOSH 128K
* 1 SOURIS



MACINTOSH

CARTE LANGAGE 16K APPLE



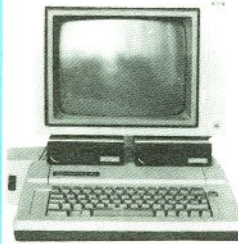
Disponible uniquement pour APPLE II+ elle est utilisée essentiellement pour des applications langages type PASCAL..... **1330 F**

Disponible également en compatible (sans ROM F8)..... **695 F**

- * 1 IMAGE WRITER
 - * 1 SET DE LOGICIEL
- CLEF EN MAIN caractéristiques :
— CPU 88000 16/32 bits
— 64K de ROM
— 128K de RAM
— Horloge 8MHz
— 4 gène monophoniques
— Lecteur 3 pouces
— Résolution graphique par point 512 * 342
— Interface série
— Clavier détachable

ENSEMBLE APPLE II

NOUVEAU PRIX



Comprendant :
* 1 UC APPLE IIE
* 1 MONITEUR APPLE
* 1 LECTEUR 5 POUCES
* 1 SYSTEME APPLE FAN

12800 F

APPLE III

NOUVEAU PRIX



Comprendant :
* 1 UC APPLE III 256K
* 1 MONITEUR APPLE
* 1 DISQUE DUR 5Mo
* 1 LOGICIEL SOS

524 23 16

JEUX APPLE



Trijeu	395 Frs
Naja 1	250 Frs
Naja 2	395 Frs
Galaxiel	350 Frs
Point bac maths1	295 Frs
Point bac maths2	295 Frs
Logo	1490 Frs
Sorcellerie	650 Frs
Téléjeux	550 Frs
Tic tac jeux	550 Frs
Echec et max	460 Frs
Zendar	396 Frs
Flaster blaster	330 Frs
Pinball conts set	440 Frs
Polywool	396 Frs
Chopflipper	361 Frs
Fathoms 40	397 Frs
Zaxxon	454 Frs
Genesis	567 Frs
Chess (niveau 7)	866 Frs
Flight simulator	595 Frs
RDF 1985	360 Frs

APPLE FAN

495 F

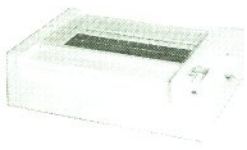
Quand votre APPLE est chargé en cartes d'extension, il a besoin d'un meilleur refroidissement. APPLE FAN est un ventilateur carrossé qui se fixe à l'extérieur et ne nécessite aucun perçage pour sa fixation.

PENTASONIC

**Penta 8
Penta 13
Penta 16**

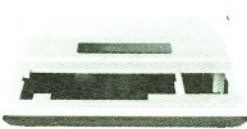
34, rue de Turin, 75008 Paris
Tél. 293 41 33
Métro : Liège, St-Lazare, Place Chichy.
10, bd Arago 75013 Paris - Gobelins
Tél. 336 26 05 Métro : Gobelins
(service correspondance et magasin)
5, rue Maurice Bourdon, 75016 Paris
Tél. 524 23 16 (Pont de Grenelle)
Métro Charles Michels, Téléx 614 789
Prix TTC promise à titre indicatif pour un volume en fonction des approvisionnements

IMAGE WRITER 5810 F



L'imprimante conçue par APPLE pour votre APPLE. Caractéristiques :
— 180 caractères par secondes
— bidirectionnelle
— matrice 7 * 9 prog. 16 * 8
— interface RS 232
— majuscules, minuscules
— graphique point par point

COFFRET APPLE 698 F



Dimensions identiques au coffret APPLE. Vendu avec ou sans découpe numérique

JOYSTICK 320 F



En plastique souple moulé ce joystick résistera à vos enfants sans difficultés. Autre avantage, il dispose de 4 switchs de façon à pouvoir être utilisé à la place des paddes.

FLOPPY 5 POUCES COMPATIBLE APPLE



2690 F

Ces floppys ont l'avantage de travailler à une vitesse nettement supérieure à celle des lecteurs standards, associée à une technologie plus moderne. (moteur entraînement direct et suspension cadre tendu).

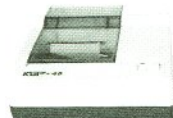
PARALLELE APPLE

1360 F



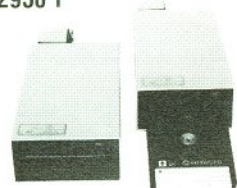
A ne pas confondre avec les cartes VIA ou PIA, cette interface est spécialisée dans le contrôle des imprimantes (APPLE ou autres).

UNE IMPRIMANTE CLEFS EN MAIN POUR VOTRE APPLE.



AK 40 1464 F

FLOPPY APPLE 3 POUCES 2950 F



Savez-vous que le temps de transfert d'un lecteur 3 pouces est sensiblement identique à celui d'un disque dur ? D'où l'intérêt du lecteur HI TACHI. Autre avantage la taille des supports et leur solidité. Ce lecteur peut théoriquement adresser 50 K de mémoire mais pour rester compatible avec le DOS 3.3 il est limité à 143 K.

LOGICIEL APPLE

- * VISICALC (français) 2700 F
- * MULTIPLAN 2420 F
- * VISIPILOT 1630 F
- * ORCA : assembleur 1490 F
- * APPLE WRITER : traitement de texte 1572 F
- * BUSINESS GRAPHIQUE 1375 F
- * QUICK FILE : traitement de fichier 790 F
- * VISIFILE : création et gestion de fichier 3320 F
- * APPLE LOGO 1600 F
- * APPLE PASCAL : langage 1820 F
- * CX BASE 200 3290 F
- * CX BASE 200 + CX TEXTE 3990 F

CARTE APPLE-TELL



6820 F

Distribuée par FEEDER cette carte est l'OUTIL DE COMMUNICATION.
— modem intégré
— gestion télex
— possibilité de mémorisation sur disquette des messages modem ou télex.

550 F



CARTE 6522 VIA APPLE

Commande de processus ? Vous avez besoin d'une carte 6522. Elle vous permet de disposer de 32 lignes que vous pouvez définir comme des entrées ou des sorties.

CARTE 8088 APPLE



Réservée à des applications professionnelles qui nécessitent des traitements temps réels très rapides. Cette carte est vendue complète avec son logiciel **5622 F**

CARTE TAXAN COULEUR



Indispensable pour interfacer votre APPLE avec un moniteur TAXAN.

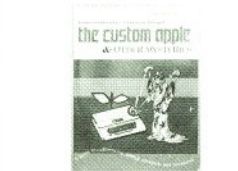
APPLE II+ **695 Frs**
APPLE IIE **1640 Frs**
*Option 80 colonnes incluse.

CLAVIER TYPE APPLE



Splendide clavier genre IBM pour votre APPLE II+.

OTHER MYSTERIES



Ces livres sont coûteux mais représentent le NEC PLUS ULTRA de la littérature technique. Tous les secrets de votre TRS80 ou de votre APPLE dévoilés. Version américaine non sous titrée.

- TRS DOS 2,3 decoded and other mysteries 296,00 F
- The custom apple 315,00 F
- Basic disk I/O faster and better 365,00 F
- How to do it on TRS 80 378,20 F
- TRS 80 disk and other mysteries 284,00 F
- Machine language disk I/O 378,20 F
- The custom TRS 80 378,20 F
- Basic faster and better 276,00 F

80 Colonnes APPLE

Ces 2 cartes distribuées par APPLE sont indispensables dans la majorité des applications professionnelles de votre APPLE IIE (visicalc, writer).
80 colonnes **998 Frs**
80 colonnes étendue **2379 Frs**

*étend la capacité RAM à 128K

INTERFACE SERIE APPLE

Cette carte permet 2 modes de travail
— liaison RS232 classique avec signaux de HAND-SHAKE
— liaison RS232 type MODEM bi-directionnelle.

1360 F

CARTE EVE



Extension 80 colonnes 64K RAM et sortie parallèle
2970 F

RGB EX 3520 F

Moniteur couleur entrée RVB. Bande passante vidéo 15 Mhz. Résolution horizontale 380. Résolution verticale 262.

RGB II 4732 F

Moniteur couleur entrée RVB. Bande passante vidéo > 15 Mhz. Résolution horizontale 510. Résolution verticale 262.

KP 810 5790 F

Imprimante 80 c 140 cps. Totalement compatible FX80. Qualité courrier avec introduction feuille à feuille type machine à écrire.

KP 910 7926 F

Imprimante 132 c 140 cps. Même caractéristiques que la 810.

CARTE 6809 APPLE 2800 F

Vendue avec sa disquette FLEX et son manuel cette carte donne à votre APPLE la puissance du CPU 6809 du MOTOROLA.

DISQUETTES



- 3" **72,00 F**
- 3,5" **79,00 F**
- 5" SF SD **22,50 F**
- 5" DF DD **33,00 F**
- 5" 96 TPI **39,80 F**
- 5" DF DD 10 sec **43,00 F**
- 5" DF DD 16 sec **44,00 F**
- 8" SF DD **44,00 F**
- 8" DF DD **54,00 F**



Z80 avec CP/M APPLE 3100 F

La carte Z80 les disquettes CP/M (les vraies), plus des manuels nombreux et complets, voici ce que vous propose le KIT CP/M de MICROSOFT. Fonctionne sur APPLE II+ ou IIE.

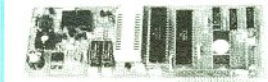
Carte Z80, également disponible en version compatible sans CP/M et sans manuel **995 Frs**

KOALA PAD 1470 F

Cette tablette de digitalisation permet de créer toutes les formes et tous les graphismes en direct, comme sur une table à dessin. Vendue avec manuels en français et logiciel pour APPLE **1470 Frs**

PROGRAMMATEUR DE MEMOIRES

Avec son logiciel intégré sur ROM le romble permet de programmer les 2708-2516-2532-2732-2758 en 21 ou 25V



1562 F

En liaison directe avec la RAM APPLE vous pouvez visualiser ou modifier directement le code hexa contenu dans votre original.

CARTE RAM 128K APPLE



Compatible APPLE II+ ou APPLE IIE cette carte dispose de son propre système de «REFRESH» et donc ne nécessite pas de connexions avec la RAM d'origine. Elle peut émuler les cartes légende et saturne.

Complète sans logiciel **1990 Frs**

Disponible également sous forme de CI seul **398 Frs**

Le Calculateur Entier

Jacques Duma

Cryptographes et arithméticiens, à vos claviers !

Ce Calculateur Entier permet de faire des opérations avec de très grands nombres. Il accepte jusqu'à 500 chiffres par nombre pour les calculs. Le "prompt" qui indique que le programme attend vos ordres est le signe pour-cent "%".

Les règles de syntaxe de ce Calculateur Entier sont formalisées en notation BNF. Les unités syntaxiques définies sont entre chevrons ("`<`" et "`>`"). Les termes de leur définition suivent les trois symboles " ::= ". Le signe "!" sépare les différentes formes que peut prendre une unité syntaxique. Les parties optionnelles sont notées entre accolades ("{" et "}")". "RETURN" signifie : taper la touche RETURN. Tous les autres symboles qui ne sont pas entre chevrons forment l'alphabet du langage du Calculateur.

Mise en route

Le code source Lisa est fourni (CALCULATEUR.L), ainsi que le code objet en binaire (CALCULATEUR.B). La version standard est assemblée à l'adresse 2112 (décimal) ou \$840 (hexa). Pour lancer le programme, il suffit de faire :

```
BRUN CALCULATEUR.B
```

L'écran est effacé et on peut lire, en haut et à gauche :

```
FAIRE : CALL 2112
```

Si l'on exécute cette instruction, on se retrouve sous le contrôle du Calculateur Entier. Le prompt est le signe pour-cent "%".

Si, par accident, on quitte le programme (par exemple en faisant RESET ou une erreur DOS), on peut y retourner sans perdre les variables en cours d'utilisation. Pour cela, il faut faire :

```
& (en Applesoft) ou CTRL-Y (en mode moniteur).
```

Si, à la suite d'une sortie, on désire réinitialiser le programme, il faut faire :

```
CALL 2112 (en Applesoft) ou 840G (en mode moniteur).
```

Si la position du programme en mémoire ne vous convient pas, vous pouvez l'assembler à une autre adresse (à condition qu'elle soit comprise entre \$805 et \$66D0). Tout ce qui précède reste valable,

seule l'adresse d'appel change. Au début, vous verrez apparaître sur votre écran :

```
FAIRE : CALL aaaa
```

où aaaa sera l'adresse décimale correspondant à l'adresse hexa d'origine de votre programme. Si par exemple vous mettez "ORG \$805" en ligne 2 du texte source, le programme affichera :

```
FAIRE : CALL 2053
```

Etudions à présent les possibilités de ce programme.

Commandes

Le Calculateur Entier est un interpréteur de lignes. Chaque ligne de commande contient au plus 250 caractères et doit se terminer par "RETURN".

```
<Ligne> ::=  
  <Commande>"RETURN" ||  
  <Commande DOS>"RETURN"  
<Commande> ::=  
  { <Dièse> } <Instruction> ||  
  <Commande> :  
  { <Dièse> } <Instruction>  
<Dièse> ::= <Constante> #
```

Il peut y avoir plusieurs instructions par ligne, séparées par deux-points ":". Les blancs ne sont pas pris en compte.

Le <Dièse> optionnel devant une <Instruction> représente un facteur répétiteur. La constante qui le précède, et qui doit être un entier positif inférieur à 256, indique combien de fois la ligne doit être exécutée (voir la section sur les facteurs répétiteurs).

Une <Commande DOS> est une commande du DOS standard. En fait, on utilisera seulement les commandes CATALOG, BSAVE (pour sauvegarder les résultats d'un travail), BLOAD (pour récupérer un travail en cours), PR#0 et PR#1 (pour diriger l'affichage vers l'écran ou l'imprimante). Une commande du DOS doit bien sûr être seule sur sa ligne.

Exemples de lignes :

```
%BLOAD nomfichier "RETURN"  
%?2*3 : A =2/3;R : $ : @ "RETURN"
```

Le signe pour-cent "%" est le prompt et la touche RETURN est tapée en fin de ligne. Nous verrons plus loin la signification du "\$" et de l'arobas "@".

Instructions

Il y a trois types d'instructions, pour l'initialisation du système, les affectations, et les affichages.

```
<Instruction> ::= & || , || v || <Affectation> || <Affichage> || <Rien>  
<Rien> ::=
```

La virgule (,) est une pause entre deux instructions. A l'exécution, trois bips sont émis et il faut appuyer sur une touche pour continuer. Si CTRL-C est appuyé à ce moment-là, il y a une interruption normale.

Le guillemet (") permet d'introduire des remarques dans le déroulement d'un calcul. Tout ce qui suit le (") jusqu'au (:) suivant ou jusqu'à la fin de la ligne est simplement recopié à l'écran (les espaces sont supprimés).

Par exemple :

```
%"ZERO : A=0 : ?A :  
A=A+1 : "UN : ?A : "FINI  
affichera :  
ZERO  
= 00  
UN  
= 01  
FINI
```

<Rien> signifie qu'une instruction peut être vide (dans une ligne, on peut avoir ":" sans rien entre les deux-points séparateurs).

Initialisation

& est l'instruction qui permet de désaffecter toutes les variables en cours d'utilisation. Cette instruction, avant de réinitialiser tout le système, demande confirmation par :

```
RAZ (O/N) ?
```

Si la réponse est O pour OUI, le système est réinitialisé; dans le cas contraire, on retourne au niveau de commande sans rien perdre.

Attention : le chargement d'un fichier de variables à partir d'une disquette fait perdre tout le contenu actuel de la mémoire.

Instructions d'affectation

Ce programme n'est qu'un calculateur et non un analyseur d'expressions arithmétiques. On ne peut donc faire qu'une opération par instruction, mais on peut avoir plusieurs instructions par ligne.

Remarque : si les nombres sont très grands, les calculs peuvent durer un temps appréciable; il ne faut surtout pas s'impatienter.

Les additions et soustractions sont rapides.

Une multiplication de deux nombres de 250 chiffres peut durer jusqu'à 15 secondes.

L'élevation à une puissance peut atteindre 25 secondes (2¹⁶⁸⁵ par exemple).

La division est la plus longue : la division d'un nombre de 500 chiffres par un nombre de 1 chiffre peut prendre 60 secondes.

```
<Affectation> ::= <Variable> =
  <Expression> { ; <Variable> }
<Expression> ::=
  { - } <Primaire> || { ( ) } <Primaire>
  <Opérateur> <Primaire>
<Primaire> ::=
  <Variable> || <Constante>
<Opérateur> ::=
  + || - || * || / || ^ || ' ,
```

Les opérations permises sont donc l'addition, la soustraction, la multiplication, la division entière (avec "/"), qui peut fournir son reste (à condition que les deux opérandes soient positifs), la puissance entière à exposant positif (avec "^") et enfin la multiplication par une puissance de 10 (avec l'apostrophe).

Voici quelques exemples :

```
A = -3 + 123 ; B = A ^ 5 ; C = A / B ;
D = -A * C ; C = 12 - A ; A = A / A ; A :
C = 33 / 5 ; B
```

Pour la division entière, on a avec le reste la possibilité d'une affectation double. Exemple :

```
Q = 13 / 5 : la valeur 2 est affectée à la variable Q.
```

```
Q = 13 / 5 ; R : la valeur 2 est affectée à Q et le reste 3 à R.
```

```
R = 13 / 5 ; R : la valeur 2 est affectée à R, puis 3 est aussi affecté à R. Le 2 est donc perdu, mais on a ainsi obtenu le reste de la division en utilisant une seule variable ...
```

L'option { ; <Variable> } n'est acceptée par le Calculateur, à la suite d'une affectation, que dans le cas de la division. Dans les autres cas, une erreur est détectée.

L'apostrophe (') permet de faire très rapidement les multiplications et divisions par les puissances de 10 (des décalages). X^N est égal à X*(10^N). X et N peuvent être positifs ou négatifs. Par exemple, si N=5 et M=-5, alors :

```
? 123456789'N donne =12 3456 7890 0000
```

```
? 123456789'M donne =01 2345
```

Si l'exposant de 10 est négatif, cette opération ne rend pas de reste, contrairement à la division.

Limites - Les règles suivantes doivent être respectées :

- les deux opérandes d'une division

doivent être positifs;

- il n'est pas possible de diviser par zéro;
- l'exposant d'une puissance doit être positif;
- tout résultat après calcul doit avoir moins de 509 chiffres.
- l'exposant de 10 dans (') ne doit pas dépasser 511 en valeur absolue.

Si l'on désire affecter à une variable un nombre de plus de 250 chiffres, il faut procéder par étapes car la ligne est limitée à 250 caractères. Ainsi, un nombre de 400 chiffres sera affecté à la variable X en deux lignes :

```
% X=(la suite des 200 premiers chiffres)'200 "RETURN"
% X=X+(la suite des 200 derniers chiffres) "RETURN"
```

Variables

On dispose de 26 variables distinctes. Chaque variable a pour identificateur une des lettres de l'alphabet (A..Z) et pour valeur un nombre entier relatif, c'est-à-dire un entier positif, nul ou négatif.

```
<Variable> ::=
A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q
|R|S|T|U|V|W|X|Y|Z
```

A l'initialisation du système, les variables ne sont pas affectées (elles n'ont pas de valeurs).

L'utilisation d'une <Variable> non initialisée dans une <Expression> provoque le message d'erreur :

```
? VARIABLE 'x' NON AFFECTEE
```

Pour utiliser une <Variable>, il faut donc l'initialiser en lui affectant pour valeur une <Constante> ou une <Expression>.

Constantes

Une constante est un nombre entier positif.

```
<Constante> ::= <Chiffre> ||
  <Constante><Chiffre>
<Chiffre> ::= 0|1|2|3|4|5|6|7|8|9
```

Il y a cependant une limitation à la taille des constantes. On ne peut pas entrer plus de 250 caractères au clavier; de plus, après calcul, le nombre le plus grand ne peut avoir plus de 508 chiffres.

Les blancs étant non significatifs dans une ligne, on peut entrer des espaces entre les groupes de chiffres d'une constante, ce qui facilite l'écriture des très grands nombres.

Si une <Constante> est utilisée de manière répétitive dans un calcul, il vaut mieux l'affecter à une <Variable> et utiliser la <Variable>. Les calculs seront plus rapides. Le codage d'une constante est en effet assez long.

Instructions d'affichage

Ces instructions permettent de visualiser les résultats sur l'écran (ou sur l'imprimante, si l'on a fait préalablement PR#i avec l'imprimante sur le Slot i).

```
<Affichage> ::=
  $ || @ || ? <Expression>
```

?<Expression> affiche la valeur de l'expression. Le Calculateur passe à la ligne et écrit "=" suivi de la valeur. Si l'expression est une division, deux résultats sont donnés : le quotient puis le reste.

Les nombres sont écrits avec un espace tous les quatre chiffres pour faciliter la lecture. Un caractère CR (Carriage Return) est envoyé tous les 79 caractères pour que les sorties sur imprimante soient formatées correctement.

On peut afficher les valeurs de toutes les variables en service à un instant donné par @. Ceci écrit :

AFFICHAGE DES VARIABLES AFFECTEES : variables et valeurs, par ordre alphabétique

L'affichage écran est rapide. On peut l'interrompre entre chaque variable en tapant sur la barre d'espacement; on le relance de la même façon. CTRL-C permet d'arrêter l'affichage et de récupérer le prompt.

On peut aussi afficher en hexadécimal l'adresse et la longueur de la zone mémoire à sauvegarder pour conserver les valeurs calculées et affectées aux variables en service.

```
"$" donne l'adresse et la longueur de la zone mémoire à sauvegarder par un BSAVE. Le format est le suivant : ZONE A SAUVEGARDER A$xxxx,L$yyyy où xxxx est l'adresse du début de la zone à sauvegarder et yyyy sa longueur, le tout en hexadécimal.
```

Par exemple, pour sauvegarder les variables calculées sur la disquette sous le nom de fichier "MES-CALCULS", il faut faire :
%BSAVE MES-CALCULS,A\$xxxx,L\$yyyy "RETURN"

La valeur de A\$xxxx dépend de l'adresse à laquelle le programme a été assemblé (on a normalement A\$1700). La valeur de L\$yyyy est L\$0070 à l'initialisation du système. Chaque variable affectée occupe 256 octets supplémentaires (\$100). On aura ainsi "L\$0370" si 3 variables ont été affectées.

Pour récupérer ultérieurement les valeurs sur lesquelles on travaille, il suffira de faire :

```
%BLOAD MES-CALCULS "RETURN"
```


Attention : le contenu de la mémoire avant le BLOAD est perdu.

Les facteurs répéteurs

Une ligne normale est exécutée une seule fois. Si le <Dièse> est utilisé, on peut exécuter une <Ligne> ou une fin de ligne plusieurs fois.

La constante qui précède le dièse (#) doit être un entier positif inférieur ou égal à 255. Ce nombre indique combien de fois il faut exécuter tout ce qui suit le dièse jusqu'à la fin de la ligne (jusqu'au "RETURN").

Il peut y avoir plusieurs fois le <Dièse> sur une même ligne. Chaque facteur répéteur porte sur ce qui le suit jusqu'à la fin de la ligne. Le Calculateur n'est pas programmable, mais le <Dièse> permet de faire assez facilement des calculs répétitifs (comptages, factorielles).

S'il y a trop de facteurs répéteurs sur une même ligne, il est possible que le programme soit mis en défaut par saturation de la pile. Il vaut donc mieux éviter d'utiliser plus de 15 fois le dièse dans la même ligne (la contrainte est raisonnable ...).

La syntaxe impose que chaque dièse soit séparé du suivant par au moins une instruction (éventuellement vide). Il y a donc au moins un deux-points entre chaque paire de dièses. Si l'on veut afficher 500 fois la valeur 3, il faut taper la ligne suivante :

```
% 10# : 50 # ?3 "RETURN"
```

Voici quelques exemples :

Le signe pour-cent "%" au début de la ligne est le prompt. La suite est tapée par l'utilisateur jusqu'au "RETURN". Les blancs sont non significatifs (on peut ne pas en mettre). Le reste est affiché par le Calculateur :

```
?? 2+3 "RETURN"
```

```
=05
```

```
% 3# ?2+3 "RETURN"
```

```
=05
```

```
=05
```

```
% A=0 : 100# A=A+1
```

```
% ?A
```

```
=0100
```

```
% A=0 : F=1 : 10# @ : 10#A=A
```

```
+1 : F=A*F
```

```
AFFICHAGE DES VARIABLES
```

```
AFFECTEES :
```

```
A=00
```

```
F=01
```

```
AFFICHAGE DES VARIABLES
```

```
AFFECTEES :
```

```
A=10
```

```
F=0362 880
```

```
AFFICHAGE DES VARIABLES
```

```
AFFECTEES :
```

```
A=20
```

```
F=0243 2902 0081 7664 0000
```

```
AFFICHAGE DES VARIABLES
```

AFFECTEES :

```
A=30
```

```
F=0265 2528 5981 2191 0586 3630
```

```
8480 0000 00....
```

Cet exemple calcule dans F la factorielle de A. Il y a deux boucles imbriquées. La première affiche les variables affectées ("@" est exécuté 10 fois), et exécute la seconde boucle. La seconde incrémente A et multiplie A par F. Les résultats ne sont donc affichés qu'une fois sur dix. Le dernier affichage est A=90 et F la valeur de factorielle 90 (90!). On termine alors les deux boucles par le calcul de factorielle 100 (100!), mais la valeur n'est pas affichée. Si l'on fait "@", on affichera A=100 et F=100! A titre d'information, le dépassement de capacité est obtenu pour la factorielle de 256 (256!).

Un autre exemple utile :

```
% N= 1225313 : K=3 : 10# : 60#
```

```
Q=N/K;R : X=1/R : K=K+2 "RE-
```

```
TURN"
```

```
? DIVISION PAR ZERO
```

```
% @ "RETURN"
```

```
AFFICHAGE DES VARIABLES AF-
```

```
FECTEES :
```

```
K=0997
```

```
N=0122 5313
```

```
Q=1229
```

```
R=00
```

```
X=00
```

```
%N=1229 : K=3 : 20# Q=N/K;R :
```

```
X=1/R : K=K+2 "RETURN"
```

```
% @ "RETURN"
```

```
AFFICHAGE DES VARIABLES AF-
```

```
FECTEES :
```

```
K=43
```

```
N=1229
```

```
Q=28
```

```
R=40
```

```
X=00
```

La première ligne permet de chercher si le nombre N=1225313 est premier. Pour cela, on le divise par tous les nombres impairs à partir de K=3 et ceci 600 fois (on arrive ainsi à la racine de N). Il faut donc ajouter 2 à K 600 fois (10# :60 #).

Le quotient est Q, le reste est R. Si N n'est pas premier, le reste R prendra la valeur zéro à un certain moment. L'instruction (X=1/R) permet d'interrompre le calcul à cet instant (car il y a division par zéro).

Dans notre exemple, ceci se produit après 4 minutes de calcul. On affiche alors les variables, ce qui permet de voir que N=K*Q, c'est-à-dire que 1225313 = 997 * 1229 et n'est donc pas premier.

On essaye de même de décomposer 1229 en facteurs premiers. Mais, cette fois, on reprend la main sans erreur. Le nombre 1229 est donc premier.

Pour des grands nombres, le temps de calcul reste hélas prohibitif.

Interruptions

On sait déjà qu'il est possible de programmer des arrêts (,), d'interrompre ou de reprendre l'affichage avec la barre d'espace. On peut aussi arrêter les calculs par CTRL-C. Cette interruption est non destructive, car elle n'est pas prise en compte n'importe quand. Si l'on fait CTRL-C durant l'affichage, le message d'interruption apparaît à l'écran et on retourne au niveau de commande. La valeur des variables (à cet instant) n'est pas modifiée.

Si l'on fait CTRL-C au cours d'un calcul, on peut attendre l'interruption entre deux et une dizaine de secondes. En effet, le CTRL-C n'est pris en compte que dans les divisions et les puissances, mais pas dans les multiplications. Dans le cas d'une affectation, si le calcul est interrompu, l'affectation n'a pas lieu et la valeur de la variable demeure inchangée.

Si le CTRL-C intervient au cours d'un calcul avec facteur répéteur, il est difficile de savoir si l'interruption a lieu au cours d'un calcul (s'il y a "/" ou "*" dans le calcul), ou si l'interruption a lieu entre deux répétitions. Il faut alors regarder la valeur des compteurs que l'on a utilisés dans la boucle pour évaluer la position du point d'interruption.

Remarque : il n'est pas possible de reprendre un calcul interrompu. Soyez donc patients, car les temps de calcul sont parfois importants.

Il peut aussi se produire des interruptions accidentelles. Si l'on tape la touche RESET, on se retrouve soit sous contrôle de l'Applesoft (ROM Autostart), soit sous le contrôle du moniteur (ancienne ROM).

Pour retourner dans le Calculateur sans perdre le travail en cours, il faut faire "&" (à partir de l'Applesoft) ou CTRL-Y (à partir du moniteur). Si une erreur est faite dans une commande du DOS, on quitte aussi le Calculateur pour se retrouver sous contrôle du Basic. La procédure de retour est la même.

Attention : si on fait RESET pendant l'affectation d'une variable, celle-ci sera incorrectement affectée; des erreurs pourront alors se produire avec cette variable. Méfiez-vous par conséquent du RESET.

Pour utiliser l'Applesoft et le Calculateur ensemble, il faut :

- assembler le programme à l'adresse \$66D0 (ORG \$66D0 et OBJ &800)

- après sauvegarde correcte du code binaire, faire BRUN;

```
FAIRE CALL 26320 apparaît à
```

```
l'écran. Il faut alors faire :
```

```
HIMEM:26320 : CALL 26320
```

Le Calculateur est alors protégé contre le Basic.

Entrées-Sorties

Si une imprimante est branchée sur le slot i, faire PR#i pour sortir les résultats sur papier. La présentation à l'imprimante n'est pas identique à celle de l'écran.

Certains CR ne sont transmis qu'à l'écran pour plus de lisibilité, mais les lignes ne sont pas sautées sur l'imprimante. De plus, les très grands nombres sont présentés par lignes de 80 caractères sur l'imprimante.

Si les sorties se font sur papier, on ne voit rien à l'écran. Mais si votre interface d'imprimante permet de conserver l'image écran grâce à un caractère de contrôle, vous pouvez entrer ce caractère (sans tenir compte de l'erreur de syntaxe qu'il provoquera). Tout doit fonctionner normalement.

Exemples

Voici le résultat d'une séquence de travail normale. Ce qui suit le signe pour-cent "%" est tapé au clavier par l'utilisateur, suivi de "RETURN"; le reste est affiché par le Calculateur Entier.

CALCULATEUR ENTIER

VERSION 1.2

PAR J. DUMA

<06.01.1984>

??12345+123

=0124 68

%%\$

ZONE A SAUVEGARDER A\$182C,
L\$0070

%A=-123456789: \$: @: B =
A + A : @

ZONE A SAUVEGARDER A\$1700,
L\$0170

AFFICHAGE DES VARIABLES
AFFECTEES :

A= -0123 4567 89

AFFICHAGE DES VARIABLES
AFFECTEES :

A= -0123 4567 89

B= -0246 9135 78

%?63/5 : ?-12*12 : ?2 ^ 100 : ?C

= 12

= 03

= -0144

=0126 7650 6002 2822 9401 4967

0320 5376

? VARIABLE 'C' NON AFFECTEE

%A= A+ -D

? PRIMAIRE INCORRECT

%A=3/0

? DIVISION PAR ZERO

%A=5/3:A : ?A : ?-3/A

=02

? OPERANDE NEGATIF

% A=0 : 20# : 20# A=A+1

% ?A

= 0400

%... etc... etc...

Résumé des règles syntaxiques

<Ligne> ::=

<Commande>"RETURN" ||

<Commande DOS>"RETURN"

<Commande> ::=

{ <Dièse> } <Instruction> ||

<Commande> : { <Dièse> }

<Instruction>

<Dièse> ::=

<Constante>#

<Instruction> ::=

& || , || " || <Affectation> ||

<Affichage> || <Rien>

<Affichage> ::=

\$ || @ || ? <Expression>

<Affectation> ::=

<Variable> = <Expression>

{ ; <Variable> }

<Expression> ::=

{ - } <Primaire> || { - } <Primaire>

<Opérateur> || <Primaire>

<Primaire> ::=

<Variable> || <Constante>

<Variable> ::= <Lettre>

<Constante> ::= <Chiffre> ||

<Constante> <Chiffre>

<Opérateur> ::= + || - || * || / || ^ ||

<Lettre> ::=

A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|
P|Q|R|S|T|U|V|W|X|Y|Z

<Chiffre> ::= 0|1|2|3|4|5|6|7|8|9

<Rien> ::=

Les messages d'erreur

? ERREUR DE SYNTAXE

? INTERRUPTION DE L'UTILISATEUR

L'utilisateur a fait CTRL-C.

? FACTEUR REPETITEUR INCORRECT

Facteur > 255, ou il manque #.

? DIVISION PAR ZERO

? OPERANDE NEGATIF

Les opérandes d'une division doivent être positifs.

? RESULTAT TROP GRAND

Le résultat du calcul a plus de 508 chiffres.

? EXPOSANT NEGATIF

Pour une puissance, l'exposant doit être positif.

? AFFECTATION INCORRECTE

Il peut manquer le signe (=).

? PRIMAIRE INCORRECT

? VARIABLE 'x' NON AFFECTEE

La variable dont le nom est (x) n'est pas en service.

? EXPRESSION INCORRECTE

? MAUVAIS OPERATEUR

NDLR : le Calculateur est très utile pour lire le livre passionnant de François Le Lonnais "Les nombres remarquables", chez Herman. Encore une fois, Pom's vous offre un programme qui, à lui seul, se vendrait plus cher que l'abonnement d'un an avec disquettes. Nous n'avons pas publié le programme source, car il y en aurait pour 15 pages en deux colonnes... Le source, ainsi que la récapitulation, se trouve comme d'habitude sur la disquette d'accompagnement du numéro. Les droits de l'auteur seront bien entendu établis sur la base de la longueur totale de l'article, incluant ces 15 pages non publiées. ■

Calculateur Entier Code objet

*840.1840

0840- A9 4C 8D F5 03 8D F8 03

0848- A9 5E 8D F6 03 8D F9 03

0850- A9 09 8D F7 03 8D FA 03

0858- 20 58 FC 20 8C 17 C6 C1

0860- C9 D2 C5 BA A0 C3 C1 CC

0868- CC A0 87 00 A2 40 A9 08

0870- 20 24 ED 20 8E FD 20 8E

0878- FD A9 4C 8D 40 08 A9 8B

0880- 8D 41 08 A9 08 8D 42 08

0888- 4C D3 03 20 58 FC 20 9B

0890- 17 A0 AD AD AD AD AD AD

0898- AD AD AD AD AD AD AD AD

08A0- AD AD AD AD AD AD AD AD

08A8- CC C3 D5 CC C1 D4 C5 D5

08B0- D2 A0 C5 CE D4 C9 C5 D2

08B8- 8D A0 AD AD AD AD AD AD

08C0- AD AD AD AD AD AD AD AD

08C8- AD AD AD AD AD AD AD AD

08D0- A0 A0 A0 A0 D6 C5 D2 D3

08D8- C9 CF CE A0 A0 B1 AE B2

08E0- 8D A0 A0 A0 A0 A0 A0 A0

08E8- AD AD AD AD AD AD AD AD

08F0- AD AD AD AD AD AD AD AD

08F8- A0 A0 A0 A0 DO C1 D2 A0

0900- A0 CA AE A0 C4 D5 CD C1

0908- 8D A0 A0 A0 A0 A0 A0 A0

0910- AD AD AD AD AD AD AD AD

0918- AD AD AD AD AD AD AD AD

0920- A0 A0 A0 A0 BC B0 B6 AE

0928- B0 B1 AE B1 B9 B8 B4 BE

0930- 8D A0 A0 A0 A0 A0 A0 A0

0938- AD AD AD AD AD AD AD AD

0940- AD AD AD AD AD AD AD AD

0948-	7F	A9	00	9D	1C	18	CA	10	OBC8-	CO	OB	A9	DB	CD	CO	OB	DO	OE48-	DO	FA	88	DO	F4	EE	13	18	
0950-	FA	A9	00	8D	99	18	A9	70	OBDO-	EE	20	F9	17	4C	D8	09	48	OE50-	4C	33	0E	EE	13	18	4E	13	
0958-	8D	98	18	4C	C9	09	20	58	OBD8-	29	1F	0A	0A	AA	BD	2D	18	OE58-	18	A9	00	38	ED	13	18	8D	
0960-	FC	20	9B	17	AD	AD	AD	AD	OBEO-	DO	02	68	60	A9	AO	20	ED	OE60-	2D	18	AD	2E	18	ED	8D	72	0E
0968-	AD	AD	AD	AD	AD	AD	AD	AD	OBEB-	FD	68	C9	CO	DO	O2	A9	AO	OE68-	AD	2F	18	8D	73	0E	AE	2D	
0970-	AD	AD	AD	AD	AD	AD	AD	8D	OBFO-	20	ED	FD	A9	BD	20	ED	FD	OE70-	18	BD	FF	FF	C9	00	DO	09	
0978-	AO	C3	C1	CC	C3	D5	CC	C1	OBF8-	A9	AO	20	ED	FD	BD	2C	18	OE78-	EE	2D	18	E8	DO	F3	CE	2D	
0980-	D4	C5	D5	D2	A0	C5	CE	D4	OC00-	10	04	A9	AD	DO	O2	A9	AO	OE80-	18	60	AE	14	18	CA	DO	03	
0988-	C9	C5	D2	8D	A0	AD	AD	AD	OC08-	20	ED	FD	BD	2E	18	8D	24	OE88-	4C	A6	0E	CA	DO	03	4C	CA	
0990-	AD	AD	AD	AD	AD	AD	AD	AD	OC10-	OC	8D	2F	OC	BD	2F	18	8D	OE90-	OE	CA	DO	03	4C	D8	OE	CA	
0998-	AD	AD	AD	AD	AD	AD	AD	8D	OC18-	25	OC	8D	30	OC	BD	2D	18	OE98-	DO	03	4C	4F	OF	CA	DO	03	
09A0-	AO	D2	C5	DO	D2	C9	D3	C5	OC20-	AA	AO	OF	BD	FF	FF	2D	DA	OEA0-	4C	2A	10	4C	51	10	AD	1C	
09A8-	AO	D3	D5	D2	A0	D3	CF	D2	OC28-	FD	EO	FF	F0	20	E8	BD	FF	OEA8-	18	CD	20	18	DO	06	8D	2C	
09B0-	D4	C9	C5	8D	A0	AD	AD	AD	OC30-	FF	20	DA	FD	EO	FF	F0	15	OEB0-	18	4C	50	12	20	1B	12	DO	
09B8-	AD	AD	AD	AD	AD	AD	AD	AD	OC38-	88	F0	09	A9	AO	20	ED	FD	OEB8-	03	4C	B5	13	90	09	20	49	
09C0-	AD	AD	AD	AD	AD	AD	AD	8D	OC40-	E8	4C	23	OC	20	8E	FD	E8	OECO-	14	20	D4	13	20	70	14	4C	
09C8-	00	D8	20	3A	FF	A2	F8	9A	OC48-	AO	10	4C	23	OC	20	8E	FD	OEC8-	98	12	AD	20	18	49	FF	8D	
09D0-	A9	00	48	48	48	4C	DF	09	OC50-	20	F9	17	60	20	1F	OD	20	OEDO-	2C	18	8D	20	18	4C	A6	OE	
09D8-	20	D1	15	C9	BA	FO	3B	68	OC58-	D1	15	F0	03	4C	FF	15	20	OED8-	20	1B	12	B0	06	20	49	14	
09E0-	FO	16	AA	CA	68	8D	D3	15	OC60-	F9	17	A9	CO	20	D7	OB	20	OEE0-	20	D4	13	20	B5	13	AD	1C	
09E8-	A8	68	8D	D2	15	48	98	48	OC68-	9A	15	AD	14	18	C9	04	DO	OEE8-	18	4D	20	18	8D	2C	18	AD	
09F0-	8A	48	20	B7	15	4C	1A	0A	OC70-	OB	20	22	14	A9	CO	20	D7	OEF0-	1E	18	8D	76	12	8D	54	13	
09F8-	68	DO	1B	68	DO	19	48	48	OC78-	OB	20	9A	15	4C	D8	09	20	OEF8-	AD	1F	18	8D	77	12	8D	55	
0A00-	48	A9	FF	48	A9	01	48	48	OC80-	EF	15	B0	03	4C	FF	15	8D	OF00-	13	AD	22	18	8D	19	13	AD	
0A08-	A9	A5	85	33	20	6A	FD	EO	OC88-	11	18	A9	00	8D	12	18	20	OFO8-	23	18	8D	1A	13	AD	2E	18	
0A10-	00	F0	F5	4C	DF	09	68	4C	OC90-	C9	15	C9	BD	F0	03	4C	E1	OF10-	8D	79	12	8D	73	12	AD	2F	
0A18-	DF	09	20	C9	15	F0	B9	20	OC98-	16	20	1F	OD	AE	14	18	EO	OF18-	18	8D	7A	12	8D	74	12	20	
0A20-	DF	15	90	79	29	OF	8D	10	OCA0-	04	DO	18	20	D1	15	C9	BB	OF20-	12	13	F0	08	20	6E	12	CE	
0A28-	18	20	C9	15	20	DF	15	90	OCA8-	DO	11	20	C9	15	20	EF	15	OF28-	15	18	DO	F8	20	4E	13	CE	
0A30-	53	0E	10	18	AD	10	18	0A	OCB0-	BO	03	4C	E1	16	8D	12	18	OF30-	1D	18	DO	03	4C	A6	16	20	
0A38-	0A	6D	10	18	8D	10	18	20	OCB8-	20	C9	15	20	D1	15	F0	03	OF38-	12	13	F0	08	20	6E	12	CE	
0A40-	D1	15	29	OF	18	6D	10	18	OC00-	4C	E1	16	AD	11	18	29	1F	OF40-	15	18	DO	F8	20	4E	13	EE	
0A48-	8D	10	18	20	C9	15	20	DF	OC08-	0A	0A	AA	BD	2D	18	F0	09	OF48-	21	18	DO	D3	4C	62	0E	AD	
0A50-	15	90	31	0E	10	18	AD	10	OCDO-	20	5F	15	20	22	14	4C	OC	OF50-	20	18	F0	03	4C	6D	16	AD	
0A58-	18	0A	90	03	4C	45	16	0A	OC08-	0D	EE	99	18	AD	2C	18	9D	OF58-	1C	18	F0	03	4C	6D	16	AD	
0A60-	90	03	4C	45	16	6D	10	18	OCE0-	2C	18	AD	2D	18	9D	2D	18	OF60-	2D	18	C9	FF	DO	16	AD	2E	
0A68-	90	03	4C	45	16	8D	10	18	OCE8-	AD	2E	18	9D	2E	18	AD	2F	OF68-	18	8D	75	OF	AD	2F	18	8D	
0A70-	20	D1	15	29	OF	18	6D	10	OCF0-	18	9D	2F	18	AD	1C	18	8D	OF70-	76	OF	A2	FF	BD	FF	FF	DO	
0A78-	18	90	03	4C	45	16	8D	10	OCF8-	2C	18	AD	1D	18	8D	2D	18	OF78-	03	4C	89	16	20	49	14	20	
0A80-	18	20	C9	15	C9	A3	F0	03	OD00-	AD	1E	18	8D	2E	18	AD	1F	OF80-	D4	13	20	70	14	A9	01	8D	
0A88-	4C	45	16	AD	D2	15	48	AD	OD08-	18	8D	2F	18	AD	12	18	DO	OF88-	1A	18	A9	00	8D	1B	18	AD	
0A90-	D3	15	48	CE	10	18	AD	10	OD10-	03	4C	8D	09	8D	11	18	A9	OF90-	1E	18	8D	63	13	8D	31	13	
0A98-	18	48	20	C9	15	20	D1	15	OD18-	00	8D	12	18	4C	C3	OC	20	OF98-	8D	3E	13	AD	1F	18	8D	64	
0AA0-	DO	03	4C	D8	09	A2	05	DD	OD20-	C9	15	DO	03	4C	4E	17	18	OFA0-	13	8D	32	13	8D	3F	13	AD	
0AA8-	FE	17	F0	06	CA	10	F8	4C	OD28-	A9	2C	6D	98	18	8D	2E	18	OFA8-	26	18	8D	7A	13	8D	89	13	
0AB0-	7F	OC	8A	0A	AA	BD	04	18	OD30-	8D	1E	18	8D	22	18	8D	26	OFB0-	8D	A7	13	8D	AC	13	8D	9E	
0AB8-	8D	C2	0A	BD	05	18	8D	C3	OD38-	18	8D	2A	18	A9	18	6D	99	OFB8-	13	8D	A3	13	AD	27	18	8D	
0AC0-	0A	4C	FF	FF	20	C9	15	F0	OD40-	18	8D	2F	18	8D	1F	18	8D	OF00-	7B	13	8D	8A	13	8D	A8	13	
0AC8-	06	20	ED	FD	4C	C4	0A	20	OD48-	23	18	8D	27	18	8D	2B	18	OFC8-	8D	AD	13	8D	9F	13	8D	A4	
0AD0-	8E	FD	20	F9	17	4C	D8	09	OD50-	A9	00	8D	14	18	EE	1F	18	OFD0-	13	20	1B	12	F0	02	B0	11	
0AD8-	20	C9	15	F0	03	4C	FF	15	OD58-	EE	23	18	EE	23	18	EE	27	OFD8-	20	5D	13	EE	1A	18	DO	06	
0AE0-	20	3A	FF	20	8C	17	52	41	OD60-	18	EE	27	18	EE	27	18	EE	OFE0-	EE	1B	18	20	B7	15	4C	D1	
0AE8-	5A	60	60	68	4F	6F	4E	69	OD68-	2B	18	EE	2B	18	EE	2B	18	OFE8-	OF	AD	1A	18	DO	03	CE	1B	
0AF0-	60	7F	A0	00	20	OC	FD	C9	OD70-	EE	2B	18	20	B5	13	20	D4	OFF0-	18	CE	1A	18	AD	1A	18	OD	
0AF8-	CF	DO	03	4C	8B	08	20	8C	OD78-	13	20	FB	13	20	E5	14	20	OFF8-	1B	18	F0	1C	20	2A	13	20	
0B00-	17	88	88	88	88	88	88	88	OD80-	D1	15	C9	AD	DO	14	20	C9	1000-	74	13	CE	25	18	20	1B	12	
0B08-	88	88	88	88	88	88	88	AO	OD88-	15	20	EC	OD	AD	2C	18	49	1008-	FO	02	B0	DD	20	CA	12	20	
0B10-	AO	AO	AO	AO	AO	AO	AO	AO	OD90-	FF	8D	2C	18	20	D1	15	DO	1010-	99	13	20	B7	15	4C	05	10	
0B18-	AO	AO	AO	8D	00	4C	D8	09	OD98-	0A	60	20	EC	OD	20	D1	15	1018-	20	62	0E	20	97	14	20	49	
0B20-	20	C9	15	F0	03	4C	FF	15	ODA0-	DO	01	60	C9	AB	DO	04	A9	1020-	14	20	D4	13	20	70	14	4C	
0B28-	20	3A	FF	20	3A	FF	20	3A	ODA8-	01	DO	2B	C9	AD	DO	04	A9	1028-	62	0E	20	22	14	20	E5	14	
0B30-	FF	8D	10	CO	AD	00	CO	10	ODB0-	02	DO	23	C9	AA	DO	04	A9	1030-	20	B5	13	AD	C6	13	8D	44	
0B38-	FB	20	B7	15	8D	10	CO	4C	ODB8-	03	DO	1B	C9	AF	DO	04	A9	1038-	10	AD	C7	13	8D	45	10	A2	
0B40-	D8	09	20	C9	15	F0	03	4C	ODC0-	04	DO	13	C9	DE	DO	04	A9	1040-	FF	A9	01	9D	FF	FF	20	OC	
0B48-	FF	15	20	8C	17	DA	CF	CE	ODC8-	05	DO	0B	C9	A7	DO	04	A9	1048-	15	AD	20	18	F0	03	4C	C5	
0B50-	C5	A0	C1	A0	D3	C1	D5	D6	ODDO-	06	DO																

10C8-	8D	16	18	4A	4A	6D	16	18	1348-	DO	F3	CE	1D	18	60	AO	04	15C8-	60	EE	D2	15	DO	03	EE	D3
10D0-	8D	16	18	68	29	OF	6D	16	1350-	18	A2	FF	3E	FF	FF	CA	DO	15D0-	15	AD	FF	FF	C9	AO	FO	F1
10D8-	18	8D	16	18	20	07	12	AD	1358-	FA	88	DO	F4	60	AO	04	18	15D8-	C9	8D	FO	02	C9	BA	60	20
10E0-	14	18	C9	05	FO	03	4C	27	1360-	A2	FF	3E	FF	FF	CA	DO	FA	15E0-	D1	15	C9	BO	BO	01	60	C9
10E8-	11	AD	19	18	OD	18	18	FO	1368-	88	DO	F4	CE	1D	18	AE	1D	15E8-	BA	90	02	18	60	38	60	20
10F0-	2F	18	6E	19	18	6E	18	18	1370-	18	4C	3D	13	AO	04	18	A2	15F0-	D1	15	C9	C1	BO	01	60	C9
10F8-	90	14	20	BE	14	20	97	14	1378-	FF	3E	FF	FF	CA	DO	FA	88	15F8-	DB	90	02	18	60	38	60	20
1100-	20	D8	0E	20	OC	15	AD	19	1380-	DO	F4	CE	25	18	AE	25	18	1600-	8C	17	8D	BF	AO	C5	D2	D2
1108-	18	OD	18	18	FO	12	20	97	1388-	BD	FF	FF	C9	00	DO	09	EE	1608-	C5	D5	D2	AO	C4	C5	AO	D3
1110-	14	20	49	14	20	D8	0E	20	1390-	25	18	E8	DO	F3	CE	25	18	1610-	D9	CE	D4	C1	D8	C5	8D	8D
1118-	E5	14	20	B7	15	4C	E9	10	1398-	60	F8	18	A2	FF	BD	FF	FF	1618-	00	4C	C9	09	20	8C	17	8D
1120-	20	BE	14	20	70	14	60	AD	13A0-	69	01	9D	FF	FF	CA	BD	FF	1620-	BF	AO	C9	CE	D4	C5	D2	D2
1128-	2C	18	8D	1A	18	20	49	14	13A8-	FF	69	00	9D	FF	FF	CA	DO	1628-	D5	DO	D4	C9	CF	CE	AO	C4
1130-	20	B5	13	20	D4	13	20	70	13B0-	F5	D8	4C	82	13	AD	2E	18	1630-	C5	AO	CC	A7	D5	D4	C9	CC
1138-	14	20	49	14	A9	00	18	6E	13B8-	8D	C6	13	AD	2F	18	8D	C7	1638-	C9	D3	C1	D4	C5	D5	D2	8D
1140-	19	18	6E	18	18	90	02	A9	13C0-	13	A2	00	A9	00	9D	FF	FF	1640-	8D	00	4C	C9	09	20	8C	17
1148-	FF	48	AD	19	18	FO	03	4C	13C8-	E8	DO	FA	8D	2C	18	A9	FF	1648-	8D	BF	AO	C6	C1	C3	D4	C5
1150-	A6	16	AD	1A	18	FO	4E	18	13D0-	8D	2D	18	60	AD	2E	18	8D	1650-	D5	D2	AO	D2	C5	DO	C5	D4
1158-	AD	2D	18	6D	18	18	90	04	13D8-	91	15	AD	2F	18	8D	92	15	1658-	C9	D4	C5	D5	D2	AO	C9	CE
1160-	68	4C	B5	13	A8	8D	21	18	13E0-	AD	1E	18	8D	94	15	AD	1F	1660-	C3	CF	D2	D2	C5	C3	D4	8D
1168-	AE	2D	18	AD	2E	18	8D	8A	13E8-	18	8D	95	15	AD	2C	18	8D	1668-	8D	00	4C	C9	09	20	8C	17
1170-	11	AD	2F	18	8D	8B	11	AD	13F0-	1C	18	AD	2D	18	8D	1D	18	1670-	8D	BF	AO	CF	DO	C5	D2	C1
1178-	22	18	8D	8D	11	8D	19	13	13F8-	4C	8E	15	AD	2E	18	8D	91	1678-	CE	C4	C5	AO	CE	C5	C7	C1
1180-	AD	23	18	8D	8E	11	8D	1A	1400-	15	AD	2F	18	8D	92	15	AD	1680-	D4	C9	C6	8D	8D	00	4C	C9
1188-	13	BD	FF	FF	99	FF	FF	E8	1408-	22	18	8D	94	15	AD	23	18	1688-	09	20	8C	17	8D	BF	AO	C4
1190-	C8	DO	F6	AD	2C	18	8D	20	1410-	8D	95	15	AD	2C	18	8D	20	1690-	C9	D6	C9	D3	C9	CF	CE	AO
1198-	18	68	FO	03	20	12	13	20	1418-	18	AD	2D	18	8D	21	18	4C	1698-	DO	C1	D2	AO	DA	C5	D2	CF
11A0-	70	14	4C	62	OE	38	AD	2D	1420-	8E	15	AD	1E	18	8D	91	15	1700-	8D	8D	00	4C	C9	09	20	8C
11A8-	18	ED	18	18	BO	03	4C	A6	1428-	AD	1F	18	8D	92	15	AD	2E	1708-	AO	DO	D2	C9	CD	C1	C9	D2
11B0-	16	A8	8D	1D	18	AE	2D	18	1430-	18	8D	94	15	AD	2F	18	8D	1710-	C5	AO	C9	CE	C3	CF	D2	D2
11B8-	AD	2E	18	8D	D7	11	AD	2F	1438-	95	15	AD	1C	18	8D	2C	18	1718-	C5	C3	D4	8D	8D	00	4C	C9
11C0-	18	8D	D8	11	AD	1E	18	8D	1440-	AD	1D	18	8D	2D	18	4C	8E	1720-	09	48	20	8C	17	8D	BF	AO
11C8-	DA	11	8D	54	13	AD	1F	18	1448-	15	AD	1E	18	8D	91	15	AD	1728-	D6	C1	D2	C9	C1	C2	CC	C5
11D0-	8D	DB	11	8D	55	13	BD	FF	1450-	1F	18	8D	92	15	AD	22	18	1730-	AO	A7	00	68	20	ED	FD	20
11D8-	FF	99	FF	FF	C8	E8	DO	F6	1458-	8D	94	15	AD	23	18	8D	95	1738-	8C	17	A7	AO	CE	CF	CE	AO
11E0-	AD	2C	18	8D	1C	18	68	FO	1460-	15	AD	1C	18	8D	20	18	AD	1740-	C1	C6	C6	C5	C3	D4	C5	C5
11E8-	10	AD	1D	18	C9	02	BO	03	1468-	1D	18	8D	21	18	4C	8E	15	1748-	8D	8D	00	4C	C9	09	20	8C
11F0-	4C	A6	16	CE	1D	18	20	4E	1470-	AD	22	18	8D	91	15	AD	23	1750-	17	8D	BF	AO	C5	D8	DO	D2
11F8-	13	20	22	14	4C	62	OE	18	1478-	18	8D	92	15	AD	2E	18	8D	1758-	C5	D3	D3	C9	CF	CE	AO	C9
1200-	2E	16	18	2E	17	18	60	18	1480-	94	15	AD	2F	18	8D	95	15	1760-	CE	C3	CF	D2	D2	C5	C3	D4
1208-	AD	16	18	6D	18	18	8D	18	1488-	AD	20	18	8D	2C	18	AD	21	1768-	C5	8D	8D	00	4C	C9	09	20
1210-	18	AD	17	18	6D	19	18	8D	1490-	18	8D	2D	18	4C	8E	15	AD	1770-	8C	17	8D	BF	AO	CD	C1	D5
1218-	19	18	60	AD	1D	18	CD	21	1498-	26	18	8D	91	15	AD	27	18	1778-	D6	C1	C9	D3	AO	CF	DO	C5
1220-	18	FO	06	BO	02	38	60	18	1500-	8D	24	18	AD	2D	18	8D	25	1780-	D2	C1	D4	C5	D5	D2	8D	8D
1228-	60	AD	1E	18	8D	45	12	AD	1508-	18	4C	8E	15	AD	2E	18	8D	1788-	00	4C	C9	09	20	AA	17	20
1230-	1F	18	8D	46	12	AD	22	18	1510-	91	15	AD	2F	18	8D	92	15	1790-	B1	17	20	BF	17	20	DF	17
1238-	8D	48	12	AD	23	18	8D	49	1518-	AD	2A	18	8D	94	15	AD	2B	1798-	4C	EB	17	20	AA	17	20	B1
1240-	12	AE	1D	18	BD	FF	FF	DD	1520-	18	8D	95	15	AD	2C	18	8D	17A0-	17	20	CF	17	20	DF	17	4C
1248-	FF	FF	DO	03	E8	DO	F5	60	1528-	28	18	AD	2D	18	8D	29	18	17A8-	EB	17	85	1B	86	1C	84	1D
1250-	AD	1E	18	8D	76	12	AD	1F	1530-	4C	8E	15	29	1F	OA	AA	AA	17B0-	60	BA	BD	03	01	85	19	BD
1258-	18	8D	77	12	AD	2E	18	8D	1538-	BD	2E	18	8D	91	15	BD	2F	17B8-	04	01	85	1A	4C	F2	17	AO
1260-	79	12	8D	73	12	AD	2F	18	1540-	18	8D	92	15	AD	2E	18	8D	17C0-	00	B1	19	FO	09	20	ED	FD
1268-	8D	7A	12	8D	74	12	A2	FF	1548-	94	15	AD	2F	18	8D	95	15	17C8-	20	F2	17	4C	C1	17	60	AO
1270-	F8	18	BD	FF	FF	7D	FF	FF	1550-	BD	2C	18	8D	2C	18	BD	2D	17D0-	00	B1	19	FO	09	20	FO	FD
1278-	9D	FF	FF	CA	DO	F4	D8	AD	1558-	18	8D	2D	18	4C	8E	15	AD	17D8-	20	F2	17	4C	D1	17	60	BA
1280-	1D	18	CD	2D	18	90	03	AD	1560-	11	18	29	1F	OA	AA	AA	AD	17E0-	A5	19	9D	03	01	A5	1A	9D
1288-	2D	18	8D	2D	18	CE	2D	18	1568-	2E	18	8D	91	15	AD	2F	18	17E8-	04	01	60	A5	1B	A6	1C	A4
1290-	FO	03	4C	62	OE	4C	A6	16	1570-	8D	92	15	BD	2E	18	8D	94	17F0-	1D	60	E6	19	DO	02	E6	1A
1298-	AD	1E	18	8D	BE	12	AD	1F	1578-	15	BD	2F	18	8D	95	15	AD	17F8-	60	A9	8D	4C	FO	FD	A6	A4
12A0-	18	8D	BF	12	AD	2E	18	8D	1580-	2C	18	9D	2C	18	AD	2D	18	1800-	CO	BF	AC	A2	D8	OA	42	OB
12A8-	C1	12	8D	BB	12	AD	2F	18	1590-	BD	FF	FF	9D	FF	FF	E8	DO	1808-	8A	OB	54	OC	20	OB	C4	OA
12B0-	8D	C2	12	8D	BC	12	A2	FF	1598-	F7	60	20	B7	15	48	AD	00	1810-	17	4C	C7	17	60	AO	00	B1
12B8-	F8	38	BD	FF	FF	FD	FF	FF	15A0-	CO	10	12	C9	AO	DO	OE	8D	1818-	19	FO	09	20	FO	FD	20	F8
12C0-	9D	FF	FF	CA	DO	F4	D8	4C	15A8-	10	CO	AD	00	CO	10	FB	20	1820-	17	4C	D7	17	60	BA	A5	19
12C8-	62	OE	AD	1E	18	8D	FC	12	15B0-	B7	15	8D	10	CO	68	60	48	1828-	9D	03	01					

EXP(PI*SQR(163)) est-il entier ?

Olivier Herz

Le problème

Il y a quelques années, la revue Sciences & Vie annonçait dans ses récréations mathématiques que EXP(PI*SQR(163)) était entier et valait 262.537.412.640.768.744, en avançant que des calculs poussés sur ordinateur n'avaient donné que des chiffres 9 dans les 2 millions de premières décimales. La revue parlait même d'une démonstration basée sur le nombre d'Euler (nombre irrationnel, comme PI et e).

Calcul sur TI 58

Je voulais en avoir le cœur net. Aussi, en 1979, j'attaquai le problème muni de ma calculatrice TI 58. J'utilisai pour cela un programme de multiplication de nombres de 50 chiffres. Comme toutes les autres opérations étaient manuelles à cause de la faible capacité de la calculatrice, j'utilisai de nombreuses astuces pour calculer la racine carrée et l'exponentielle. Après un après-midi de calcul (et un autre pour vérification), j'obtins un résultat à 50 chiffres (donc 31 décimales, dont au moins 25 devaient être bonnes en tenant compte des erreurs d'arrondi). Qu'obtins-je ?

Calcul en Basic

En 1981, muni d'un Apple, je décidai de confirmer ce résultat avec un programme Applesoft (joint sur la disquette d'accompagnement de ce numéro, afin que les lecteurs intéressés puissent "l'éplucher"). Ce programme utilise des tableaux pour stocker, par tranches de 4 chiffres, des grands nombres. De plus, il possède des routines pour faire des multiplications et additions de grands nombres ou des divisions de grands nombres par une constante d'au plus 4 chiffres. Là, les résultats sont obtenus bien plus rapidement (14 mn pour 80 chiffres, 5 mn avec un programme compilé).

Le Calculateur Entier

Enfin, tel Zorro, arrive le Calculateur Entier de Jacques Duma, muni de la division de grands nombres; je vais montrer dans les lignes qui suivent à quel point il devient alors facile de programmer ce calcul.

NDLR : Olivier Herz a rédigé son article à partir de la version 1.0 du calculateur. La lecture de cet article a amené Jacques Duma à réaliser et à nous envoyer la version 1.2, publiée dans ce numéro. Cette version a

ajouté la pause entre deux instructions (,), l'introduction des remarques (") et surtout la multiplication par une puissance de 10 (^). Ainsi, le programme adapté à cette version calcule-t-il le résultat en cinq minutes de moins que EXP.EXEC !

Nous avons décidé de ne pas modifier le texte original d'Olivier pour lui garder sa virginité initiale (au texte, bien sûr). Le lecteur relativisera donc certaines remarques et pourra d'autant mieux apprécier la qualité des analyses d'Olivier et l'utilité de ses observations.

1 - Comme on travaille uniquement sur des nombres entiers, il va falloir utiliser une constante multiplicative pour transformer des nombres décimaux en nombres entiers. Soit par exemple, pour des calculs sur 100 décimales (on pourrait aller presque jusqu'à 254=508/2 décimales) :

$K=10^{100}$

Bien entendu, K n'est pas nécessairement une puissance de 10. Voir à ce sujet la remarque (2).

2 - Pour calculer la racine carrée de 163, on utilise la méthode de Newton, par approximations successives avec la formule $U_{n+1} = (U_n + 163/U_n) / 2$, en prenant 13 comme approximation de départ. Soit :

$R=K*K$: $R=163*R$: $U=13*K$:

$7\#$: $X=R/U$: $U=X+U$: $U=U/2$

7 itérations suffisent pour avoir 100 bonnes décimales.

3 - PI est supposé connu (le Palais de la Découverte publie 2000 décimales sur une carte postale). Et l'on calcule PI*SQR(163) :

$R=3$ 14159 26535 89793 23846

26433 83279 50288 41971 69399

37510 58209 74944 59230 78164

06286 20899 86280 34825 34211

70679*U : R=R/K

4 - On obtient 40.109... et l'on va calculer EXP(40.109...) = EXP(40) * EXP(0.109...). On ne garde donc que la partie décimale du résultat :

$X = 40*K$: $R = R - X$

5 - On calcule son exponentielle par développement limité avec la formule $EXP(X) = 1 + X + X^2/2 + X^n/n!$ = somme des U_n , avec $U_0=1$ et $U_n = U_{n-1} * X/n$:

$S=K$: $I=0$: $U=K$: $45\#$: $I=I+1$:

$U=U*R$: $U=U/K$: $U=U/I$: $S=S+U$

On calcule suffisamment de termes (ici 45) pour que U_n soit négligeable.

6 - On calcule de la même façon, avec ici $X=1$:

$E=K$: $I=0$: $U=K$: $70\#$: $I=I+1$:
 $U=U/I$: $E=E+U$

7 - Il ne reste plus qu'à calculer et fournir le résultat (avec 100 décimales, dont peut-être 80 à 90 de significatives) :

$S=S*E$: $S=S/K$

?S

Que pensez-vous du résultat ? Sciences & Vie n'avait-il pas publié son article dans le numéro d'avril ?

Remarques

(1) Tous ces résultats peuvent être obtenus automatiquement grâce au fichier EXP.EXEC. Il suffit de se mettre dans le calculateur et d'EXECuter ce fichier pour ce faire; on obtient le résultat (ici 100 chiffres) en un peu plus de 20 minutes. Faire MON C,I,O au préalable si l'on veut voir l'avancement des calculs.

On peut en effet utiliser de véritables programmes pour ce calculateur grâce aux fichiers EXEC. Utilisez Applewriter II ou reportez-vous au numéro 3 de Pom's (p. 55) ou au recueil (p. 142) pour fabriquer de tels fichiers.

(2) Au vu de ces calculs, il semble qu'on puisse émettre un regret pour le calculateur. On aurait gagné beaucoup de temps dans les calculs pour nombres non entiers si l'on avait eu une opération donnant, à partir d'un grand nombre N et d'un entier (positif ou négatif) K de 3 chiffres au plus, le résultat $N*(10^K)$. Les nombres étant codés en BCD, ce calcul aurait été ultra-rapide.

(3) Bien entendu, en utilisant des astuces, on peut aller bien plus vite. Mais la méthode utilisée présente l'avantage d'être la plus simple.

Liste de EXP.EXEC

$K=10^{100}$

$R=K*K$: $R=163*R$: $U=13*K$:

$7\#$: $X=R/U$: $U=X+U$: $U=U/2$

$R=3$ 14159 26535 89793 23846

26433 83279 50288 41971 69399

37510 58209 74944 59230 78164

06286 20899 86280 34825 34211

70679*U : R=R/K

$X = 40*K$: $R = R - X$

$S=K$: $I=0$: $U=K$: $45\#$: $I=I+1$:

$U=U*R$: $U=U/K$: $U=U/I$: $S=S+U$

$E=K$: $I=0$: $U=K$: $70\#$: $I=I+1$:

$U=U/I$: $E=E+U$

$40\#$: $S=S*E$: $S=S/K$

?S

Analyse de Magic Mailer

Guy Lapautre

Présentation du produit

Magic Mailer (Autocourrier pour la version française) est un logiciel complémentaire à Magic Window (Autotexte), distribué par ARTSCI Inc. et diffusé en France par Ordinateur Express. Nous emploierons dans la suite les noms français.

La version que nous avons testée est théoriquement adaptée à l'Apple IIe, mais pas plus que la version 2 d'Autotexte; elle n'utilise pas les possibilités nouvelles du clavier. C'est d'ailleurs moins ennuyeux que pour Autotexte, car les mouvements de curseur sont beaucoup moins fréquents. C'est pourtant dommage...

Il convient de remarquer qu'Autocourrier est exclusivement destiné à l'impression et ne permet de créer aucun texte ni fichier. L'utilisateur est censé utiliser Autotexte pour créer les textes proprement dits et les fichiers (bien qu'il soit possible, dans certaines conditions, d'utiliser des fichiers exogènes, en créant un "moniteur").

La présentation sous reliure est correcte, sans avoir le raffinement de nombreux logiciels pour Apple. La documentation est rédigée en anglais. Une traduction en français est également fournie (avec quelques adaptations). Les auteurs préviennent qu'il s'agit d'une version provisoire. Nous y avons effectivement trouvé un certain nombre d'erreurs ou imprononciations. Telle qu'elle est, elle a le mérite d'exister.

Fonctionnalités de base

Autocourrier est avant tout un "adresseur automatique". Il permet d'adresser commodément des lettres-type à toutes les personnes enregistrées dans un fichier, ou à une sélection d'entre elles. La sélection, s'il y a lieu, se fait à partir du contenu de certains champs, avec une très large combinaison de conditions logiques.

Allant un peu plus loin, il est possible de faire figurer conditionnellement des mots, membres de phrases ou paragraphes permettant de personnaliser les lettres.

Les données éditables sont de 4 types :

- des textes figurant dans la "lettre" créée sous Autotexte;
- des données figurant dans le fichier associé;
- des "constantes", en particulier une date;

- des données entrées au clavier au fur et à mesure de l'impression.

En revanche, Autocourrier ne comporte strictement aucune possibilité de calcul, ce qui interdit son emploi pour des opérations, même très simples, du type facturation ou édition de devis...

Le document à éditer

Créé sous Autotexte, il se présente comme un document normal dans lequel on aurait opéré un certain nombre d'aménagements.

Des zones sont réservées aux données issues du fichier: elles sont repérées par des identificateurs que l'on retrouve dans le "moniteur" du fichier. Ces identificateurs permettent en outre de sélectionner l'édition de portions de texte selon le contenu des champs du fichier associé, de jouer avec les majuscules et minuscules...

Des symboles de justification peuvent être introduits dans le texte, si des zones de longueurs variables font que la justification systématique faite sous Autotexte ne puisse pas toujours convenir.

Des symboles spéciaux peuvent être utilisés, par exemple pour les ensembles titre/nom/prénoms, ou la date.

Ces options sont certes intéressantes, mais on aboutit assez vite à un texte ésotérique si on veut aller loin dans les possibilités du logiciel.

Le fichier associé

Qu'il soit créé sous Autotexte ou adapté à partir d'un fichier exogène, ce fichier comporte un premier enregistrement vide, qui indique sa structure (en rappelant que les identificateurs utilisés sont ceux qu'on retrouve dans le texte).

Bien entendu, rien n'interdit d'avoir un fichier comportant des rubriques non utilisées dans un texte, mais qui pourront l'être dans un autre (au même titre que le texte peut contenir des identificateurs non présents dans le fichier, et qui correspondront à des données à entrer au clavier).

Le fichier, considéré comme un véritable texte par Autocourrier, est par définition logé en mémoire centrale avec le texte lui-même. Il ne saurait donc être volumineux! Cette limitation en rend l'emploi difficile pour de grosses applications professionnelles. Il existe toutefois un palliatif, avec l'utilisation d'une forme de chaînage.



Signalons par ailleurs qu'on peut fort bien travailler sans fichier, en rentrant au clavier les données pour lesquelles des identificateurs ont été placés dans le texte, ce qui est une solution séduisante pour des applications de petit volume, en permettant une personnalisation de lettres à peu de frais, au niveau d'applications "personnelles/professionnelles".

Utilisations

Deux options de base amènent à quatre modes d'exploitation :

- en continu ou cas par cas;
- tout le fichier ou une sélection.

Dans la pratique, deux d'entre elles sont particulièrement utiles :

- en continu avec sélection (si on peut exprimer la sélection sous forme de critères logiques);
- cas par cas et tout le fichier (permettant une sélection à l'examen de chaque enregistrement).

Voilà un des points forts d'Autocourrier, d'autant plus que les possibilités de sélection logique sont très étendues et, comme le disent d'ailleurs les auteurs de la documentation, "plus simples à mettre en oeuvre qu'à expliquer" (un effort pourrait cependant être fait pour clarifier les explications...).

En résumé

Autocourrier est un honnête système d'adressage automatique, dont deux points forts sont les larges possibilités de sélection de rubriques et/ou de paragraphes, et un système très complet de sélection dans le fichier associé.

On regrettera principalement un certain ésotérisme des textes si on utilise beaucoup d'options, ainsi que l'absence complète de fonctions de calcul. Enfin, comme dans le cas d'Autocourrier, il est dommage que les auteurs n'aient pas pris en compte les possibilités accrues offertes par le clavier de l'Apple IIe. ■

Initiation à l'assembleur (2)

Gérard Michel

Puisque nous n'avons pas été submergés par un courrier nous intimant l'ordre de mettre fin sur le champ à cette série, nous la poursuivrons sans plus attendre par l'examen de nouvelles instructions choisies dans le répertoire du langage assembleur.

Les instructions de comparaison

Elles jouent en assembleur le même rôle fondamental qu'en Basic, en permettant d'orienter le programme vers différents traitements selon le résultat de tests effectués sur des variables caractéristiques.

On ne peut toutefois comparer directement deux variables quelconques, par exemple le contenu de l'adresse \$32 avec celui de l'adresse \$303, ni une variable quelconque à une valeur donnée, par exemple le contenu de l'adresse \$18 et la valeur #\$A0.

En fait, la comparaison "stricto sensu" n'est possible qu'entre l'un des registres du microprocesseur (accumulateur, registre X ou registre Y) et "autre chose" (variable ou constante).

Il faut cependant noter que certaines instructions agissant sur une case-mémoire, sans lien direct avec une véritable comparaison, peuvent donner lieu à des tests significatifs par le biais de ce que nous avons appelé "le dernier résultat", dans l'article précédent, lors de l'examen des instructions BEQ et BNE. Nous en trouverons d'ailleurs un exemple plus loin.

L'assembleur se distingue en outre du Basic en ce que l'opération de comparaison est réalisée séparément du test effectué sur son résultat. Ainsi, alors que l'on fait "IF A=B THEN..." en Basic, on écrit en assembleur une séquence du type : "Comparer A et B" puis "Si le résultat de la comparaison est l'égalité alors...". Le microprocesseur soustrait en fait B de A (mais sans changer la valeur de A), et positionne le "dernier résultat" en conséquence; le test porte donc en définitive sur la valeur de ce "dernier résultat".

Pour ce qui est des tests autorisés, nous ne retiendrons pour l'instant que ceux qui ont déjà été évoqués dans l'article précédent, à savoir l'égalité (testée par BEQ, soit "dernier résultat" = 0) et la différence (testée par BNE, soit "dernier résultat" <> 0). Nous traiterons des autres possibilités dans un prochain article, dans lequel sera précisée la

nature réelle de notre fameux "dernier résultat".

Comparaison avec l'accumulateur

Le code "mnémorique" de l'instruction est **CMP**, ce qui suffit pour indiquer que l'un des termes de la comparaison est l'accumulateur. Le second terme dépend de ce qui suit **CMP**.

On peut ainsi comparer à l'accumulateur soit une valeur donnée, soit le contenu d'une adresse. Dans cette seconde hypothèse, tous les modes d'adressage sont autorisés pour définir l'adresse concernée. Vous trouverez ci-après la liste des syntaxes possibles pour l'instruction **CMP** et les codes hexadécimaux correspondants (ADS désigne une étiquette symbolique affectée à l'adresse dans le programme en assembleur).

- **CMP** valeur = **CMP** # $\$$ NN = C9 NN (exemple : C9 A0)
- **CMP** adresse en page zéro = **CMP** ADS = **CMP** \$aa = C5 aa
- **CMP** adresse quelconque = **CMP** ADS = **CMP** \$aaaa = CD octet bas/octet haut (exemple : **CMP** \$90FF = CD FF 90)
- **CMP** "adresse page zéro", X = **CMP** ADS,X = **CMP** \$aa,X = D5 adresse
- **CMP** ADS,X = **CMP** \$aaaa,X = DD octet bas/octet haut
- **CMP** ADS,Y = **CMP** \$aaaa,Y = D9 octet bas/octet haut
- **CMP** (ADS,X) = **CMP** (\$aa,X) = C1 adresse
- **CMP** (ADS,Y) = **CMP** (\$aa,Y) = D1 adresse

Notez bien que le contenu de l'accumulateur n'est pas affecté par l'opération de comparaison, ce qui autorise les comparaisons en chaîne, telles que :

```
CMP #80
BEQ ETIQUETTE1
CMP #8D
BEQ ETIQUETTE2
```

Comparaison avec le registre X

Le code mnémorique est **CPX**. Là encore, il est possible de comparer X avec une valeur ou le contenu d'une adresse, mais seul l'adressage direct non indexé est autorisé. Le contenu de X demeure inchangé, ce qui permet également les comparaisons en chaîne.

- **CPX** valeur = **CPX** # $\$$ NN = E0 NN (par exemple, E0 04)
- **CPX** adresse page zéro = **CPX** ADS = **CPX** \$aa = E4 adresse
- **CPX** adresse quelconque = **CPX** ADS = **CPX** \$aaaa = EC octet bas/octet haut

Comparaison avec le registre Y

Le code mnémorique est **CPY**. Mêmes remarques générales que pour **CPX**.

- **CPY** valeur = **CPY** # $\$$ NN = C0 NN
- **CPY** adresse page zéro = **CPY** ADS = **CPY** \$aa = C4 adresse
- **CPY** adresse quelconque = **CPY** ADS = **CPY** \$aaaa = CC octet bas/octet haut

Augmenter ou diminuer le contenu d'une adresse INC

Cette instruction permet d'augmenter de 1 le contenu de l'adresse donnée après **INC** et correspondrait à "A=A+1" en Basic. Elle agit directement sur l'adresse concernée, sans qu'il soit nécessaire de passer par l'intermédiaire d'un registre.

Une adresse correspondant à un octet, son contenu ne peut prendre qu'une valeur comprise entre \$0 et \$FF (0 à 255 en décimal). Ainsi, si l'adresse **ADR** contient \$FF et que vous faites "INC **ADR**", vous trouverez 0 dans **ADR** (de même que 9+1=0 si on ne compte que sur un chiffre !).

De plus, notre fameux "dernier résultat" tient compte de la valeur de l'adresse après son traitement par **INC**; il devient nul si l'adresse contient 0 (comme dans le cas ci-dessus) et sera différent de 0 dans tous les autres cas. On peut donc effectuer des tests significatifs par **BNE** et **BEQ** en sortie d'une instruction **INC**.

Ceci est particulièrement utile lorsque l'on veut manipuler l'adresse-base d'un adressage indirect indexé par Y, ou, de manière générale, une valeur supérieure à 255 et qui occupe donc deux octets. Supposons ainsi que l'on ait à incrémenter de 1 en 1 une valeur stockée aux adresses \$18 (octet bas) et \$19 (octet haut). La séquence d'instructions à utiliser serait :

```
Ins.1 INC $18
Ins.2 BNE Ins.4
Ins.3 INC $19
```

Ins.4 ...

On augmente l'octet bas de 1; si le résultat n'est pas nul, on saute à la suite du programme, sinon on augmente l'octet haut de 1. Si l'on arrive à Ins.1 avec "0418" dans \$18-\$19, on aura "0518" en arrivant à Ins.4, mais si l'on arrive avec "FF18" on aura "0019" après être passé par les deux INC.

L'adressage indirect ne peut être utilisé avec INC.

- INC adresse page zéro = INC ADS = INC \$aa = E6 adresse
- INC adresse quelconque = INC ADS = INC \$aaaa = EE octet bas/octet haut
- INC "adresse page zéro", X = INC ADS, X = INC \$aa, X = F6 adresse
- INC "adresse quelconque", X = INC ADS, X = INC \$aaaa, X = FE octet bas/octet haut

DEC

Comme on pouvait légitimement s'y attendre, DEC permet de diminuer de 1 le contenu d'une adresse. Là encore, le "dernier résultat" évolue en fonction du contenu de l'adresse après DEC, mais on ne peut l'utiliser aussi facilement qu'avec INC pour la manipulation de valeurs sur deux octets, puisque l'octet haut doit diminuer lorsque l'octet bas vaut \$FF et non lorsqu'il vaut 0 (de même que l'on ne change de dizaine que lorsque le chiffre des unités tombe à 9 et non à 0).

En reprenant notre exemple précédent, la séquence d'instructions à utiliser avec DEC serait :

Ins.1 DEC \$18
Ins.2 LDA \$18
Ins.3 CMP # \$FF
Ins.4 BNE Ins.6
Ins.5 DEC \$19
Ins.6 ...

(on aurait pu également utiliser les registres X ou Y en lieu et place de l'accumulateur)

- DEC adresse page zéro = DEC ADS = DEC \$aa = C6 adresse
- DEC adresse quelconque = DEC ADS = DEC \$aaaa = CE octet bas/octet haut
- DEC "adresse page zéro", X = DEC ADS, X = DEC \$aa, X = D6 adresse
- DEC "adresse quelconque", X = DEC ADS, X = DEC \$aaaa, X = DE octet bas/octet haut

Saut incondicional : JMP

Les instructions BEQ et BNE, maintenant bien connues de tous, provoquent un branchement à un endroit

donné du programme si le résultat du test effectué sur le "dernier résultat" répond à leur attente (respectivement égal à 0 et différent de 0). De plus, nous avons vu que le branchement se fait en fonction d'un déplacement et non directement sur une adresse que l'on indiquerait derrière BEQ ou BNE. On se "branche" 10 octets plus bas, ou encore 48 octets plus haut, si la condition est satisfaite.

L'instruction JMP, en revanche, provoque un saut "inconditionnel" à l'adresse indiquée à sa suite; elle se rapproche en cela de l'instruction GOTO du Basic.

Ainsi, "JMP \$FDE0" (ou JMP ADR si ADR a été définie comme étiquette pour \$FDE0) provoquera un saut à l'adresse \$FDE0 dès que sera rencontré le code de JMP.

JMP diffère de JSR comme GOTO de GOSUB : un RTS ne ramènerait pas à l'instruction qui suit JMP mais à celle qui suit le dernier JSR effectué dont on ne serait pas encore revenu.

Il est possible d'utiliser JMP avec l'adressage indirect : le saut se fait alors à l'adresse contenue dans l'adresse indiquée après JMP.

- JMP adresse absolu = JMP ADS = JMP \$aaaa = 4C octet bas/octet haut
- JMP adresse indirect = JMP (ADS) = JMP (\$aaaa) = 6C octet bas/octet haut

La pile du microprocesseur

Lorsque le microprocesseur rencontre une instruction JSR, il doit stocker quelque part l'adresse du point de retour à laquelle ramènera le RTS de fin de la sous-routine. Il dispose pour cela d'un "silo" bien à lui, baptisé "pile". En simplifiant, on peut dire que l'adresse de retour est posée sur le sommet de la pile et que le processeur vient l'y reprendre lorsqu'il rencontre un RTS.

Considérons par exemple les instructions suivantes :

Adresse 0 JSR Adresse 2
Adresse 1 Suite du programme
Adresse 2 JSR Adresse 4
Adresse 3 RTS
Adresse 4 JSR Adresse 6
Adresse 5 RTS
Adresse 6 LDA \$F0
----- RTS

Après le "JSR Adresse 6", la pile contiendra dans l'ordre :

Adresse 5
Adresse 3
Adresse 1

Le RTS qui suit "Adresse 6" ramènera donc à "Adresse 5", dont le RTS ramènera alors à "Adresse 3"

qui provoquera finalement le retour à "Adresse 1". La pile est de type "dernier entré - premier sorti" : une nouvelle donnée à stocker est toujours posée sur le dessus et, chaque fois qu'il doit récupérer une information stockée dans la pile, le processeur prend toujours celle qui se trouve au sommet.

Le programmeur peut également utiliser cette pile pour des stockages temporaires. En particulier, deux instructions permettent d'empiler et de dépiler l'accumulateur :

- **PHA** (code 48) : dépose le contenu de l'accumulateur au sommet de la pile
- **PLA** (code 68) : transfère l'octet qui se trouve au sommet de la pile dans l'accumulateur.

Cela limite le nombre d'adresses à utiliser pour une sauvegarde momentanée d'information, dans le cas par exemple où l'on veut sauvegarder le contenu de certaines adresses avant l'appel d'une sous-routine qui utilise et modifie ce contenu, puis récupérer les valeurs initiales au retour de la sous-routine. De même si l'on doit calculer la nouvelle adresse-base d'un adressage indirect post-indexé par Y en ajoutant à l'ancienne adresse une valeur obtenue justement par adressage post-indexé sur la base de cette ancienne adresse (le principe consiste alors à mettre dans l'accumulateur l'octet bas de l'adresse, lui ajouter la donnée obtenue avec la valeur adéquate de Y, empiler l'accumulateur, charger ensuite l'accumulateur avec l'octet haut de l'adresse, lui ajouter la donnée obtenue avec la seconde valeur adéquate de Y, stocker le résultat dans l'octet haut de l'adresse, puis dépiler dans l'accumulateur et stocker ce dernier dans l'octet bas de l'adresse - ce que l'on ne pouvait faire dès le début puisque cela aurait modifié l'adresse-base avant que l'on ait récupéré la valeur correcte à ajouter à l'octet haut, d'où incohérence des résultats).

Lorsque l'on utilise la pile pour stocker des données, il convient de se rappeler que, pour le bon fonctionnement du système, tout ce qui a pu être déposé sur une information doit être retiré avant la récupération à bon escient de celle-ci. Le petit programme suivant, par exemple, conduirait à des résultats peu conformes aux espérances du programmeur :

LDA \$18
PHA
JSR SR

SR PLA
RTS

En effet, l'accumulateur est empilé, puis le "JSR SR" provoque le dépôt au sommet de la pile de l'adresse de retour. Dans la sous-routine SR, PLA transfère dans l'accumulateur l'octet qui se trouve au sommet de la pile, soit l'un des deux octets de l'adresse de retour, puis le RTS entraînera un branchement sur une "pseudo" adresse de retour composée de l'un des octets de l'adresse véritable et de la valeur empilée par PHA avant l'appel de la sous-routine. Il y a peu de chances pour que cette "pseudo" adresse soit la même que celle de l'instruction suivant "JSR SR" et, à moins que cela ne soit délibérément voulu, vous risquez quelques surprises...

Exemples d'application

Nous vous proposons trois petits programmes d'illustration, sans grand intérêt pratique il est vrai, mais qui offrent tous l'avantage de rendre visibles les résultats de leur exécution.

Exemple 1

Le programme Basic remplit l'écran avec des caractères quelconques avant d'appeler la routine en langage machine. Cette dernière attend une saisie au clavier de la forme "/C1/C2/RETURN" (par exemple: /A/T/) et remplace ensuite tous les caractères C1 affichés par des caractères C2 (par exemple tous les A par des T).

Analysons donc en détail le programme en assembleur, source de la routine machine, en rappelant tout d'abord que ORG est une "directive" de l'assembleur (ici LISA 1.5) donnant l'adresse à partir de laquelle doit se faire l'assemblage, tandis que EQU et EPZ sont deux directives permettant d'affecter une étiquette symbolique à une adresse (EPZ si l'adresse est en page zéro et EQU dans le cas contraire).

Ligne 2: \$FC22 est l'adresse de la

routine du moniteur qui exécute l'équivalent du VTAB de l'Applesoft, en fonction de la valeur trouvée à l'adresse \$25 (V), qui doit donc contenir la position verticale du curseur souhaitée.

Ligne 3: \$FDED est l'adresse de la routine qui fait exécuter le sous-programme de sortie de caractères.

Ligne 4: \$FDOC est l'adresse de la routine qui fait exécuter le sous-programme de saisie de caractères.

Ligne 5: \$FBC1 est l'adresse de la routine qui calcule l'adresse-base d'une ligne d'écran dont le numéro (0 à #\$17) lui est donné dans l'accumulateur (voir article précédent).

Ligne 6: \$FC9C est l'adresse de la routine qui efface une ligne à partir de la position du curseur jusqu'à la fin de la ligne (c'est la routine appelée par un CALL -868 en Basic).

Ligne 7: \$28-\$29 contient l'adresse-base de la ligne calculée par \$FBC1.

Ligne 8: \$24 est l'adresse dans laquelle doit être stockée la position horizontale du curseur.

Ligne 10: nous mettrons à l'adresse \$6 le code-écran du caractère à remplacer.

Ligne 11: \$7 contiendra pour sa part le code-écran du caractère de remplacement.

Lignes 12 à 14: on déplace le curseur à la ligne numéro \$17 (équivalent à VTAB 24).

Lignes 15 à 17: on se place au début de la ligne #\$17 et on l'efface.

Lignes 18 et 19: on se repositionne au début de la ligne.

Ligne 20: demande la frappe d'un caractère au clavier. Le code-écran du caractère sera dans l'accumulateur au retour de GET.

Ligne 21: #\$AF est le code-écran de "/". On veut donc savoir si le caractère tapé est "/".

Ligne 22: si ce n'est pas le cas, la demande est répétée.

Ligne 23: le caractère est bien "/", on l'affiche à l'écran.

Ligne 24: on en demande un autre.

Ligne 25: #\$8D est le code-écran de RETURN: notre caractère est-il un RETURN?

Lignes 26 et 27: si oui, on saute à S2, sinon on l'affiche.

Lignes 28 à 31: on charge le registre X avec la position horizontale du curseur (H) et on le compare à la valeur 2. Si X vaut 2, en effet, notre caractère doit être celui qu'il faut remplacer (H vaut 0 avant le premier GET, puis 1 après le COUT de "/", puis 2 après le COUT du premier caractère). Dans ce cas, on saute à S10 où le caractère est stocké dans C1 avant le retour à la demande de frappe au clavier.

Lignes 31 et 32: X est différent de 2, il s'agit donc du caractère de remplacement que l'on stocke dans C2 avant retour à la saisie.

Lignes 35 à 37: pour que la syntaxe soit respectée, H doit être à 5 lors de la frappe du RETURN (on a affiché 5 caractères aux positions 0 à 4). Si tel n'est pas le cas, le RETURN est refusé.

Lignes 38 à 39: on va examiner le contenu de toutes les lignes d'écran en commençant par la première (0).

Ligne 40: calcul de l'adresse-base d'une ligne.

Ligne 41: on commence à la première colonne de chaque ligne.

Ligne 42: on charge l'accumulateur avec le caractère situé dans la colonne Y de la ligne dont l'adresse-base est contenue dans ADB.

Lignes 43 et 44: on le compare à C1 et, s'il lui est identique, on arrive à la ligne 45. On saute à S3 dans le cas contraire.

Lignes 45 et 46: on remplace C1 par C2.

Lignes 47 à 49: Y=Y+1. La dernière colonne porte le numéro #\$27: si Y est différent de #\$28 on examine le caractère suivant dans la même ligne. Si Y vaut #\$28, la dernière colonne a été traitée et on passe à la ligne suivante.

Ligne 50: V=V+1.

Lignes 51 à 54: si V est différent de #\$18, il reste des lignes à examiner et on retourne en S5. Sinon, on revient au début de la routine pour demander d'autres caractères (on ne peut sortir que par RESET).

Exemple 1 : source assembleur

0300	1	ORG	\$300	0304	2022FC	14	JSR	VTAB		
0300	2	VTAB	EQU	\$FC22	0307	A900	15	LDA	#0	
0300	3	COUT	EQU	\$FDED	0309	8524	16	STA	H	
0300	4	GET	EQU	\$FDOC	030B	209CFC	17	JSR	CLR	
0300	5	BASCAL	EQU	\$FBC1	030E	A900	18	LDA	#0	
0300	6	CLR	EQU	\$FC9C	0310	8524	19	STA	H	
0300	7	ADB	EPZ	\$28	0312	200CFD	20	S1	JSR	GET
0300	8	H	EPZ	\$24	0315	C9AF	21	CMP	#\$AF	
0300	9	V	EPZ	\$25	0317	D0F9	22	BNE	S1	
0300	10	C1	EPZ	\$6	0319	20EDFD	23	JSR	COUT	
0300	11	C2	EPZ	\$7	031C	200CFD	24	S11	JSR	GET
0300	12	SO	LDA	#\$17	031F	C98D	25	CMP	#\$8D	
0300	A917				0321	F013	26	BEQ	S2	
0302	8525				0323	20EDFD	27	JSR	COUT	
					0326	A624	28	LDX	H	

0328	E002	29		CPX #2
032A	F005	30		BEQ S10
032C	8507	31		STA C2
032E	4C1203	32		JMP S1
0331	8506	33	S10	STA C1
0333	4C1203	34		JMP S1
0336	A624	35	S2	LDX H
0338	E005	36		CPX #5
033A	DOE0	37		BNE S11
033C	A900	38		LDA #0
033E	8525	39		STA V
0340	20C1FB	40	S5	JSR BASCAL
0343	A000	41		LDY #0
0345	B128	42	S4	LDA (ADB),Y
0347	C506	43		CMP C1
0349	D004	44		BNE S3
034B	A507	45		LDA C2
034D	9128	46		STA (ADB),Y
034F	C8	47	S3	INY
0350	C028	48		CPY #S28
0352	DOF1	49		BNE S4
0354	E625	50		INC V
0356	A525	51		LDA V
0358	C918	52		CMP #S18
035A	DOE4	53		BNE S5
035C	4C0003	54		JMP S0

Exemple 1 : récapitulation

```

0300- A9 17 85 25 20 22 FC A9
0308- 00 85 24 20 9C FC A9 00
0310- 85 24 20 0C FD C9 AF D0
0318- F9 20 ED FD 20 0C FD C9
0320- 8D F0 13 20 ED FD A6 24
0328- E0 02 F0 05 85 07 4C 12
0330- 03 85 06 4C 12 03 A6 24
0338- E0 05 D0 E0 A9 00 85 25
0340- 20 C1 FB A0 00 B1 28 C5
0348- 06 D0 04 A5 07 91 28 C8
0350- C0 28 D0 F1 E6 25 A5 25
0358- C9 18 D0 E4 4C 00 03

```

Exemple 1 : programme Basic

```

5 TEXT
10 D$ = CHR$(4): PRINT D$"BLOAD
INI2.OBJ": HOME : PRINT "LE T
EXTE QUI SUIT N'EST DESTINE Q
U'A REMPLIR L'ECRAN POUR AVOI
R DES CARACTERES";
20 PRINT "A MODIFIER POUR LA ROUT
INE EN LANGAGE-MACHINE...": PRINT
30 FOR I = 1 TO 350:C$ = CHR$(
2 + INT ( RND (1) * 59)): PRINT
C$" ";: NEXT : CALL 768

```

Exemple 2

Le programme est très court, mais plus "ambitieux" que le précédent (mais oui !). Il s'agit de modifier le sous-programme de sortie de caractères afin de rendre plus souple le mélange de textes affichés en mode normal et en mode inverse (pour éviter justement l'usage répété des instructions NORMAL et INVERSE).

Les adresses \$36 et \$37 (54-55 en décimal) contiennent l'adresse du sous-programme de sortie de caractères utilisé par le système. Au début du programme Applesoft d'illustration, nous y plaçons donc l'adresse de notre routine (\$300 soit octet bas = 0 et octet haut = \$3 = 3). Le CALL 1002 est utilisé pour rendre ce changement effectif en le faisant prendre en compte par le DOS (voir à ce sujet les explications détaillées données par Olivier Herz dans le courrier des lecteurs du numéro 11 de Pom's).

Le but à atteindre est de remplacer INVERSE par "Control-I" (que nous noterons CTRL-I) et NORMAL par CTRL-N, afin de pouvoir taper une instruction telle que :
PRINT "CTRL-ICECI EST UNCTRL-N EXEMPLE"

Exemple 2 : source assembleur

0300		1	ORG \$300
0300	C989	2	CMP #S89
0302	DO05	3	BNE S1
0304	A93F	4	LDA #S3F
0306	8532	5	STA \$32

0308	60	6	RTS
0309	C98E	7	S1
030B	D005	8	CMP #S8E
030D	A9FF	9	BNE S2
030F	8532	10	LDA #SFF
0311	60	11	STA \$32
0312	20FOFD	12	RTS
0315	60	13	S2
			JSR \$FDF0
			RTS

et de voir s'afficher CECI EST UN en inverse et EXEMPLE en normal (CTRL-I et CTRL-N seront bien sûr "invisibles" dans l'instruction affichée à l'écran).

Voyons maintenant comment la routine y parvient, sachant que le caractère à "sortir" lui arrive contenu dans l'accumulateur.

Lignes 2 et 3 : #S89 est le code-écran de CTRL-I. Si le caractère n'est pas CTRL-I, on saute à S1.

Lignes 4 à 6 : l'adresse \$32 contient un "drapeau" qui indique au système quel mode d'affichage vidéo utiliser. L'affichage est normal si \$32 contient #SFF, inverse pour #S3F et clignotant pour #S7F (en fait, la valeur de \$32 est utilisée pour modifier les codes-écran en fonction de l'affichage souhaité). Après avoir rencontré un CTRL-I, on met donc l'affichage en mode inverse, et l'on sort de la routine sans rien afficher pour l'instant.

Lignes 7 et 8 : #S8E est le code-écran de CTRL-N. Si l'accumulateur ne contient pas CTRL-N, on saute à S2.

Lignes 9 à 11 : après avoir rencontré CTRL-N, on remet l'affichage en mode normal et l'on ressort de la routine.

Lignes 12 et 13 : \$FDF0 est l'adresse du sous-programme "standard" de sortie de caractères. Vous aurez remarqué que l'on n'utilise pas cette fois \$FDED comme dans le programme précédent. En effet, \$FDED fait exécuter le sous-programme de sortie, par un JMP à l'adresse contenue dans \$36-\$37. En situation normale, cette adresse ramènerait à \$FDF0 (\$FDED étant alors pratiquement équivalent à \$FDF0), mais comme nous avons modifié le contenu de \$36-\$37, utiliser \$FDED dans notre routine reviendrait finalement à "JMP \$300", d'où retour au début de la routine et boucle infinie. Si l'accumulateur ne contient ni CTRL-I ni CTRL-N, on affiche donc le caractère correspondant, sachant que cet affichage tiendra compte de la valeur contenue dans \$32, puis on sort de la routine.

Le retour à la routine standard de sortie de caractères peut se faire par RESET. Par ailleurs, vous pourrez constater que si notre routine est active, le LIST d'un programme Applesoft contenant des parties de texte incluses entre CTRL-I et CTRL-N provoque également l'affichage de celles-ci en inverse dans les lignes de programme.

Exemple 2 : récapitulation

```
0300- C9 89 D0 05 A9 3F 85 32
0308- 60 C9 8E D0 05 A9 FF 85
0310- 32 60 20 F0 FD 60
```

Exemple 2 : programme Basic

```
10 PRINT CHR$(4)"BLOAD INI3.OBJ
   ": POKE 54,0: POKE 55,3: CALL
   1002
20 HOME : PRINT " UE
```

```
STION 1 : ";; INPUT Z$: PRINT
Z$: PRINT
30 INPUT "QUESTION 2 : ";Z$: PRINT
Z$
40 PRINT : INPUT "ENCORE UNE PETI
TE QUESTION ? ";Z$
50 REM LES MESSAGES "QUESTION 1"
, "QUESTION 2" ET "ENCORE UNE
PETITE QUESTION ? " SONT ENC
ADRES PAR DES CTRL-I ET CTRL-
N
```

Exemple 3

Après avoir modifié le sous-programme de sortie de caractères, pourquoi ne pas essayer de faire de même avec le sous-programme d'entrée ?

Cette fois, c'est aux adresses \$38 et \$39 (56 et 57 en décimal) qu'il faut stocker l'adresse de notre routine d'entrée avant d'exécuter un CALL 1002.

La routine affichera en inverse sur la dernière ligne de l'écran les caractères tapés, qui s'afficheront ensuite à la position du curseur dans les conditions habituelles. Si on tape RETURN ou si 40 caractères ont été affichés sur la dernière ligne, celle-ci s'efface et la sortie en inverse reprend à la première colonne.

Ce mode d'affichage double ne sera toutefois actif que lorsqu'on aura tapé CTRL-A, et il le restera jusqu'à la frappe de CTRL-B ou RESET (retour à la routine standard dans ce dernier cas).

Nous vous laissons le soin d'analyser le programme, au moyen des renseignements déjà donnés dans le cours de cet article et des ultimes précisions ci-après. Bon courage !

- \$FD1B est l'adresse du sous-programme standard de saisie de caractères. Mêmes remarques pour \$FD1B par rapport à \$FD0C que pour \$FDF0 par rapport à \$FDED.
- Les adresses \$6, \$7 et \$8 nous serviront de variables de travail.
- #\$81 est le code-écran de CTRL-A.
- \$FC58 est l'adresse de la routine HOME.
- #\$DD est le code-écran du crochet fermant (]) et \$400 est l'adresse de la première colonne de la première ligne de l'écran.
- #\$A0 est le code-écran de "espace", #\$82 celui de CTRL-B et #\$8D celui de RETURN.
- On utilise la pile pour sauver la valeur contenue dans \$24 (position horizontale du curseur) avant

l'appel à \$FC9C (lignes 32-33) et avant l'appel à \$FDF0 (lignes 57-58) qui réclament ou entraînent une modification de cette valeur.

- \$21 (33 en décimal) contient la valeur de la largeur de l'écran. On l'augmente avant "JSR \$FDF0" (ligne 60) et on la réduit ensuite afin que l'affichage de la dernière colonne de la dernière ligne ne provoque pas de "scroll" des autres lignes.
- La sous-routine S01, interne au programme, réalise l'équivalent de VTAB 24 avec en sus effacement de la dernière ligne dans les cas suivants : affichage de 40 caractères, frappe de RETURN, frappe de CTRL-B. Elle rétablit les valeurs des positions horizontale et verticale du curseur dans les lignes "normales" avant de rendre la main au programme principal.
- Avant de sortir de la routine (ligne 51 et ligne 68), on remet dans l'accumulateur le code-écran du caractère saisi afin d'en permettre l'affichage par les voies normales. ■

Exemple 3 : source assembleur

```
0300          1          ORG $300
0300 201BFD    2  SO     JSR $FD1B
0303 C981     3          CMP #$81
0305 D014     4          BNE S1
0307 A001     5          LDY #1
0309 8406     6          STY $6
030B 2058FC   7          JSR $FC58
030E E624     8          INC $24
0310 A200     9          LDX #0
0312 A9DD    10          LDA #$DD
0314 8D0004   11          STA $400
0317 A9A0    12          LDA #$A0
0319 D0E5    13          BNE S0
031B C982    14  S1     CMP #$82
031D D033    15          BNE S2
031F A000    16          LDY #0
0321 8406    17          STY $6
0323 202D03  18          JSR S01
0326 2022FC  19          JSR $FC22
0329 A9A0    20          LDA #$A0
032B D0D3    21          BNE S0
032D A525    22  S01    LDA $25
032F 8508    23          STA $8
0331 A917    24          LDA #$17
0333 8525    25          STA $25
0335 2022FC  26          JSR $FC22
```

```
0338 A406    27          LDY $6
033A F004    28          BEQ S02
033C A524    29          LDA $24
033E D00D    30          BNE S03
0340 A524    31  S02    LDA $24
0342 48      32          PHA
0343 A900    33          LDA #0
0345 8524    34          STA $24
0347 209CFC  35          JSR $FC9C
034A 68      36          PLA
034B 8524    37          STA $24
034D A508    38  S03    LDA $8
034F 8525    39          STA $25
0351 60      40          RTS
0352 A406    41  S2     LDY $6
0354 D001    42          BNE S4
0356 60      43          RTS
0357 C98D    44  S4     CMP #$8D
0359 D00D    45          BNE S3
035B C606    46          DEC $6
035D 202D03  47          JSR S01
0360 E606    48          INC $6
0362 2022FC  49          JSR $FC22
0365 A98D    50          LDA #$8D
0367 60      51          RTS
0368 8507    52  S3     STA $7
036A 202D03  53          JSR S01
036D A03F    54          LDY #$3F
```

```

036F 8432      55      STY $32
0371 A524      56      LDA $24
0373 48         57      PHA
0374 A507      58      LDA $7
0376 E621      59      INC $21
0378 20FOFD    60      JSR $FDF0
037B C621      61      DEC $21
037D 68         62      PLA
037E 8524      63      STA $24
0380 2022FC    64      JSR $FC22
0383 A0FF      65      LDY #$FF
0385 8432      66      STY $32
0387 A507      67      LDA $7
0389 60         68      RTS

```

Exemple 3 : récapitulation

```

0300- 20 1B FD C9 81 D0 14 A0
0308- 01 84 06 20 58 FC E6 24
0310- A2 00 A9 DD 8D 00 04 A9
0318- A0 D0 E5 C9 82 D0 33 A0

```

Bien qu'il ait conservé le même nom depuis le 25 août 1980, le DOS 3.3 a connu un certain nombre de modifications. En particulier, l'apparition de l'Apple //e en janvier 1983 a provoqué la modification du DOS (pour qu'il supporte la carte 80 colonnes) et une tentative (initialement infructueuse) de remédier au bug de l'APPEND (voir "Réparez votre APPEND" et "Les drames de l'APPEND" dans les Pom's 1 et 2); depuis novembre 1983, les disquettes MASTER fournies par Apple contiennent une version enfin corrigée de ce bug.

Je vous propose de modifier votre DOS de manière à le rendre plus agréable d'utilisation à l'aide d'un programme qui incorpore 6 patches :

- la commande TDUMP, déjà parue dans le Pom's 10 page 71, remplace CHAIN (syntaxe : TDUMP fichier [,S slot] [,D drive] [,V volume]); POKE 48361,3 permet de

visualiser les fichiers à accès direct; POKE 48361,18 rétablit l'état normal;

- correction du bug du catalogue (la longueur d'un fichier est indiquée modulo 255);
- mise à jour du DOS pour la version //e, y compris la rectification de l'APPEND;
- un patch permettant de sortir avec CTRL-Q du catalogue pendant la pause;
- un patch pour afficher le nombre de secteurs libres après l'indication du volume;
- un dernier patch qui place le curseur sur la commande en cas d'instruction erronée au DOS.

Assurez-vous que votre DOS est standard avant d'exécuter ce pro-

```

0320- 00 84 06 20 2D 03 20 22
0328- FC A9 A0 D0 D3 A5 25 85
0330- 08 A9 17 85 25 20 22 FC
0338- A4 06 F0 04 A5 24 D0 0D
0340- A5 24 48 A9 00 85 24 20
0348- 9C FC 68 85 24 A5 08 85
0350- 25 60 A4 06 D0 01 60 C9
0358- 8D D0 0D C6 06 20 2D 03
0360- E6 06 20 22 FC A9 8D 60
0368- 85 07 20 2D 03 A0 3F 84
0370- 32 A5 24 48 A5 07 E6 21
0378- 20 F0 FD C6 21 68 85 24
0380- 20 22 FC A0 FF 84 32 A5
0388- 07 60

```

Exemple 3 : programme Basic

```

10 PRINT CHR$(4)"BLOAD INI4.OBJ
   ": POKE 56,0: POKE 57,3: CALL
   1002: POKE 6,0: TEXT : HOME :
   POKE 35,22

```

Patch du DOS 3.3

Yvan Koenig

programme, car il utilise certaines zones libres de la mémoire, très connues et éventuellement déjà employées. En cas de doute, bootez d'abord sur votre disquette MASTER.

Attention ! Ces modifications sont incompatibles avec la plupart des programmes qui déplacent le DOS sur la carte langage.

Ces modifications n'annulent pas la commande INIT; il est donc possible d'initialiser des disquettes qui les contiendront en permanence. Dans ce cas, il est préférable de marquer sur l'étiquette qu'il s'agit d'un DOS 3.3 "patché", sinon vous pourriez ne pas comprendre pourquoi certains programmes (qui utiliseraient les mêmes adresses) ne marchent plus. ■

```

1 REM KOENIG 22/2/84
10 FLASH : PRINT "PATCH EN COURS": NORMAL
99 REM TDUMP FILE,S6,D1,VO
100 HEX$ = "A893:54 44 55 4D D0 N9D26:DE BC N BC
   DF:20 8E FD 20 A3 A2 20 8C A6 F0 12 20 ED
   FD AD 0 C0 10 F3 8D 10 C0 20 1B FD C9 91
   D0 E9 20 FC A2 60": GOSUB 300: CALL - 1
   44
119 REM CATALOG > 255 SECTORS
120 HEX$ = "AE42:A0 2 A9 B0 48 A5 44 D9 A4 B3 A5
   45 E9 0 90 10 85 45 A5 44 F9 A4 B3 85 44
   68 18 69 1 48 D0 E5 68 NADAF:C BD AE B3
   20 ED FD CA D0": GOSUB 300: CALL - 144
129 REM MISE A JOUR IIE
130 HEX$ = "B33E:18 AD BF B5 8D EC B5 6D E6 B5 8
   D E6 B5 AD C0 B5 8D ED B5 6D E4 B5 8D E4
   B5 90 3 EE E5 B5 60 0 0": GOSUB 300: CALL
   - 144
139 REM BUG BIG APPEND
140 HEX$ = "BA69:AE 5F AA E0 1C F0 5 A2 0 8E 5D
   B6 60 A9 FF 8D FB 4 8D C C0 8D E C0 4C 2F
   FB AD BD B5 8D E6 B5 8D EA B5 BA 8E 9B B

```

```

   3 4C 7F B3 0 0 0": GOSUB 300: CALL - 144
150 HEX$ = "A6BB:20 69 BA NB684:84 BA NB6AA:BB N
   BFD6:20 76 BA 4C 44 B7 NB9FD:0 0 0 NB652:
   5": GOSUB 300: CALL - 144
179 REM CTRL-Q: SORTIE CATALOGUE
180 HEX$ = "AE2F:20 8E FD CE 9D B3 D0 A 20 8D B7
   F0 F0 NB78D:20 C FD C9 91 60": GOSUB 300
   : CALL - 144
189 REM SECTEURS LIBRES
190 HEX$ = "B6B3:20 48 F9 86 44 86 45 A0 C8 18 B
   9 F2 B3 F0 E A 90 FB 48 E6 44 D0 2 E6 45
   68 18 90 F0 88 D0 E9 20 42 AE 4C 2F AE NA
   DC4:B3 B6": GOSUB 300: CALL - 144
209 REM CURSEUR SUR ERREUR
210 HEX$ = "A6FF:4C D9 B6 NB6D9:C6 25 C6 25 C6 2
   5 C6 25 20 22 FC 6C 5E 9D": GOSUB 300: CA
   LL - 144
299 END
300 HEX$ = HEX$ + " N D7D2G": FOR I = 1 TO LEN
   (HEX$): POKE 511 + I, ASC ( MID$(HEX$,I,
   1)) + 128: NEXT : POKE 72,0: RETURN

```


Des tableaux de dimension variable

Gérard Michel

Vous avez sans doute déjà constaté que l'Applesoft (comme la plupart des BASICs d'ailleurs) est particulièrement strict avec le dimensionnement des tableaux. Une instruction "DIM A\$(20)", par exemple, ne peut être suivie d'une instruction "DIM A\$(30)" sans que cela provoque l'émission d'un message d'erreur.

Cette excessive rigueur ne va pas sans quelques inconvénients pour le programmeur :

- Si un tableau doit comporter X variables "volatiles", ne servant que pour une petite partie du programme, et que l'on a besoin ensuite d'un tableau de même type mais de dimension Y, il faut soit définir deux tableaux, soit en dimensionner un seul à la plus grande des deux valeurs X et Y, mais toujours sans espoir de récupérer la place occupée par les tableaux, même s'ils deviennent inutilisés.
- Si l'on doit sauvegarder provisoirement un tableau A dans un tableau B, puis recharger ensuite les valeurs initiales dans A, la place occupée par B reste perdue.
- Un tableau dont le nombre d'éléments évolue par nature (table de référence d'enregistrements sur disquette par exemple) doit être défini avec la plus grande dimension prévue, même si elle n'est effectivement jamais atteinte, ou seulement à l'issue de plusieurs mois de fonctionnement du programme. Là encore, la place perdue peut être considérable, et l'on pourrait évoquer d'autres problèmes...

La routine présentée ici vise à réduire le nombre de ces inconvénients. Elle s'appelle au moyen de l'ampersand (&) et autorise deux types de manipulation :

- Suppression pure et simple d'un tableau, pour en récupérer la place-mémoire et permettre un redimensionnement éventuel, sans affecter les autres tableaux ou variables.
- Pour les tableaux à une seule dimension, possibilité d'incrémenter la DIM de 1 lorsque l'on veut rajouter un élément (et ce autant de fois que nécessaire), et possibilité de la décrémenter d'autant lorsqu'on en supprime un.

```
1 ;*****
2 ;*
3 ;*   MODIFICATION DE LA DIM DES   *
4 ;*   TABLEAUX                     *
5 ;*
6 ;*   CODE = DIM.VAR.OBJ            *
7 ;*
8 ;*   ASSEMBLEUR : LISA 1.5         *
9 ;*
10 ;*****
11   ORG $927C
12   OBJ $800
13 L1 EPZ $6
14 L2 EPZ $7
15 N1 EPZ $8
16 N2 EPZ $9
17 TAB2 EPZ $18
18 VP EPZ $1A
19 VP2 EPZ $1E
20   LDY $6B           ;INITIALISE TAB2 A L'ADRESSE DE
21   STY TAB2          ;DEBUT DE LA ZONE DES TABLEAUX
22   LDY $6C
23   STY TAB2+1
24   LDY #5           ;NOMBRE MAXIMUM D'OCTETS POUR
25   STY N1           ;UN ELEMENT D'UN TABLEAU (REEL)
26   LDX #0
27   STX L2
28   STX $1B
29   STX $1C
30   STX $1D
31   STX N2
32   BEQ $00          ;ON A DEJA LE CARAC. QUI SUIT &
33 DEB JSR $B1        ;ACQUISITION CARAC. SUIVANT DU PROGRAMME
34   BNE $0
35   INC $1B          ;DRAPEAU "FIN D'INSTRUCTION"
36   BNE $1
37 SO  CMP #'$'      ;TABLEAU ALPHANUMERIQUE ?
38   BEQ $2
39   CMP #'%'        ;TABLEAU ENTIER ?
40   BEQ $3
41   CMP #$C8        ; "+" ?
42   BEQ $2
43   CMP #$C9        ; "-" ?
44   BEQ $1
45 $00 STA L1,X      ;1ERE LETTRE DU NOM DU TABLEAU EN L1
46   INX             ;ET 2EME EVENTUELLE EN L2
47   BNE DEB
48 $2  INC $1D        ;DRAPEAU "NOMBRE D'OCTETS PAR ELEMENT"
49   LDA L2
50   ORA #$80        ;L2=L2+$80
51   STA L2
52 $0  LDY #1        ;A-T-ON UN "+" OU UN "-" DERRIERE
53   LDA ($B8),Y     ;LE CARACTERE QUE L'ON VIENT
54   JSR $BA         ;DE LIRE ?
55   BEQ $1          ;NON (FIN D'INSTRUCTION)
56   JSR $B1        ;OUI -> M.A.J. POINTEUR DE PROGRAMME
57   CMP #$C8       ; "+" ?
58   BNE $1
59 $2  JMP $3
60 $1  INC $1C        ;DRAPEAU "+" OU "-"
61   JMP $3
62 $1  JSR $10        ;SP RECHERCHE DANS ZONE DES TABLEAUX
63   JMP $50
64 $3  INC $1D
65   LDA L1
66   ORA #$80        ;L1=L1+$80
67   STA L1
68   JMP $2
69 $10 LDY #0
70   LDA (TAB2),Y
```

Structure d'un tableau en mémoire

Le pointeur de début de la zone des tableaux est donné par les adresses \$6B-\$6C, le pointeur de fin par \$6D-\$6E.

Quel que soit le type du tableau (chaîne de caractères, réel ou entier), sa représentation en mémoire débute toujours par les informations suivantes :

- Deux octets correspondant aux deux lettres du nom du tableau (même si ce dernier n'en compte qu'une, deux octets sont utilisés).
- Le nombre total d'octets occupés par le tableau, dans l'ordre poids faible / poids fort.
- Le nombre de dimensions.
- Pour chacune des dimensions, le nombre de ses éléments, dans l'ordre poids fort / poids faible.

Chaque élément occupe ensuite 2 octets s'il s'agit d'un tableau d'entiers, 3 s'il s'agit d'alphanumériques et 5 si ce sont des réels.

Les lettres définissant le nom du tableau permettent au système d'en reconnaître le type, et de distinguer par ailleurs A\$ de A ou de A%. Par rapport au code ASCII, mémorisé dans le programme, les noms mémorisés dans la zone des tableaux présentent les caractéristiques suivantes :

- Tableau "entier" : le code de chacune des deux lettres est augmenté de \$80 (s'il n'y a qu'une lettre, le deuxième octet contient donc \$80). "A%" serait codé "C1 80" et "BB%" donnerait "C2 C2"
- Tableau "alphanumérique" : seul le code de la seconde lettre est augmenté de \$80. "A\$" serait codé "41 80" et "BB\$" donnerait "42 C2".
- Tableau "réel" : les codes ne sont pas modifiés. "A" serait codé "41 00" et "BB" serait codé "42 42".

La routine cherchera donc dans la zone des tableaux celui dont le nom lui est indiqué après "&".

S'il s'agit seulement de le supprimer, elle déplacera vers le bas tout ce qui suit la fin du tableau, c'est-à-dire les autres tableaux s'il y en a, du nombre d'octets occupés par le tableau concerné, qui lui-même sera ainsi "écrasé". Ceci revient à effectuer un "MOVE" avec pour point d'arrivée l'adresse du tableau et pour point de départ l'adresse du tableau suivant. Les données du tableau sont évidemment perdues.

Pour incrémenter ou décrémenter la dimension, la routine effectuera un "MOVE" vers le haut ou vers le bas, d'un nombre d'octets correspondant au type du tableau (2, 3 ou 5) et mettra à jour le nombre d'octets oc-

```

71     CMP L1
72     BNE S4
73     INY           ;1ERE LETTRE TROUVEE
74     LDA (TAB2),Y
75     CMP L2
76     BEQ S5           ;2EME LETTRE TROUVEE
77 S4   JSR S40         ;SP -> ADRESSE DU TABLEAU SUIVANT
78     STA TAB2+1
79     LDA VP
80     STA TAB2
81     JMP S10         ;ON CONTINUE LA RECHERCHE
82 S40  LDY #2
83     LDA (TAB2),Y ;ON AJOUTE A TAB2 LE NOMBRE
84     CLC           ;D'OCTETS OCCUPES PAR LE TABLEAU, CE QUI
85     ADC TAB2     ;DONNE L'ADRESSE DU TABLEAU SUIVANT
86     STA VP
87     INY
88     LDA (TAB2),Y
89     ADC TAB2+1
90     STA VP2
91 S5   RTS
92 S50  LDA TAB2     ;DESTINATION DU "MOVE"
93     STA $42
94     LDA TAB2+1
95     STA $43
96     INY
97     LDA (TAB2),Y
98     STA N1       ;N1/N2 -> NBRE OCTETS SUPPRIMES
99     INY
100    LDA (TAB2),Y
101    STA N2
102    JSR S40
103    LDA VP       ;DEBUT DE LA ZONE A DEPLACER
104    STA $3C
105    LDA VP2
106    STA $3D
107 FIN1 LDA $6D     ;FIN DE LA ZONE A DEPLACER
108    STA $3E
109    LDA $6E
110    STA $3F
111    LDY #0
112    JSR $FE2C    ;MOVE
113    LDA $6D     ;M.A.J. POINTEUR "FIN ZONE TABLEAUX"
114    SEC
115    SBC N1
116    STA $6D
117    LDA $6E
118    SBC N2
119    STA $6E
120    LDX $1B
121    BNE FIN     ;ON EST DEJA EN FIN D'INSTRUCTION
122 FIN0 JSR $B1   ;MISE EN FIN D'INSTRUCTION
123 FIN  RTS
124 R3   JSR S10
125     LDY #4
126     LDA (TAB2),Y ;EST-CE BIEN UN TABLEAU A 1 DIM.
127     CMP #1
128     BNE FIN0
129     JSR S40
130     LDA $1D
131     BEQ R4     ;C'EST UN TABLEAU DE REELS
132     DEC N1     ;CALCUL NOMBRE OCTETS PAR ELEMENT
133     LDA N1
134     SEC
135     SBC $1D
136     STA N1
137 R4   LDA $1C
138     BEQ R5     ;C'EST UN "+"
139 R40  LDY #6     ;M.A.J. NOMBRE D'ELEMENTS
140     LDA (TAB2),Y ;QUI DIMINUE DE 1
141     SEC
142     SBC #1
143     STA (TAB2),Y
144     DEY
145     LDA (TAB2),Y
146     SBC #0
147     STA (TAB2),Y
148     LDY #2     ;M.A.J. NOMBRE OCTETS DU TABLEAU
149     LDA (TAB2),Y ;QUI DIMINUE DE N1
150     SEC
151     SBC N1
152     STA (TAB2),Y

```

cupés ainsi que le nombre d'éléments dans la dimension.

Mode d'utilisation

Suppression

La syntaxe générale est "& Nom du tableau", par exemple: &Q%, &RR\$, &TG... Si le nom comporte plus de deux lettres, il ne faut mettre derrière "&" que les deux premières et le type (&TA% pour TABLEAU% par exemple).

On peut utiliser des instructions du type :

```
10 DIM A$(10,2), BB(25) : CY%(0) = 2
20 &A$ : &BB : &CY%
30 DIM A$(55), CY%(10,20) ...
```

Il est indispensable que le tableau ait été effectivement créé avant sa suppression, soit par une instruction DIM, soit par l'affectation d'une valeur à un élément d'indice inférieur à 10 (dimension par défaut). Sinon, le système s'embarque pour un voyage sans retour !

Modification de la dimension

La syntaxe est "& Nom du tableau +" pour ajouter un élément (&AA\$+, &CT%+ ...) et "& Nom du tableau -" pour en enlever un (&S\$-, &VV- ...).

Comme précédemment, le tableau doit effectivement avoir été créé, d'une façon ou d'une autre, mais il doit en sus ne comporter qu'une seule dimension pour que la routine produise les résultats espérés.

Il est possible de réduire la dimension jusqu'à ce qu'il n'y ait plus aucun élément dans le tableau, mais les 7 octets de définition (lettres...) seront toujours là. Ainsi, la ligne suivante provoquerait une erreur :

```
10 DIM A(5) : FOR I=5 TO 0 STEP -1 : &A- : NEXT : DIM A(20)
```

Il suffit d'insérer "&A" entre le NEXT et la nouvelle instruction DIM pour que tout se passe bien.

Il est en revanche parfaitement possible d'écrire :

```
10 DIM A(5) : FOR I=5 TO 0 STEP -1 : &A- : NEXT : FOR I=0 TO 5 : &A+ : A(I) = I : NEXT
```

Le programme de démonstration listé ci-après présente de nombreux exemples d'application.

```
153 INY
154 LDA (TAB2),Y
155 SBC #0
156 STA (TAB2),Y
157 LDA VP
158 PHA
159 LDA VP2
160 PHA
161 JSR S40 ;NOUVELLE ADRESSE DU TABLEAU SUIVANT
162 STA S43 ;DESTINATION DU MOVE
163 LDA VP
164 STA S42
165 PLA
166 STA S3D ;DEBUT DE ZONE A DEPLACER (ANCIENNE
167 PLA ;ADRESSE DU TABLEAU SUIVANT)
168 STA S3C
169 JMP FIN1
170 R5 LDY #6
171 LDA (TAB2),Y ;NBRE ELEMENTS AUGMENTE DE 1
172 CLC
173 ADC #1
174 STA (TAB2),Y
175 DEY
176 LDA (TAB2),Y
177 ADC #0
178 STA (TAB2),Y
179 LDY #2
180 LDA (TAB2),Y ;NBRE OCTETS AUGMENTE DE N1
181 CLC
182 ADC N1
183 STA (TAB2),Y
184 INY
185 LDA (TAB2),Y
186 ADC #0
187 STA (TAB2),Y
188 LDA S6E
189 STA L2 ;ANCIENNE FIN ZONE DES TABLEAUX
190 STA N2 ;NOUVELLE FIN ZONE DES TABLEAUX
191 LDA S6D
192 STA L1 ;ANCIENNE FIN
193 CLC
194 ADC N1
195 STA S6D
196 STA N1 ;NOUVELLE FIN
197 BCC R6
198 INC S6E
199 INC N2
200 R6 LDY #0
201 R60 LDA (L1),Y ;"MOVE" VERS LE HAUT
202 STA (N1),Y
203 LDA L2
204 CMP VP2 ;VP/VP2 -> ANCIENNE ADRESSE DEBUT
205 BNE R8 ;DU TABLEAU SUIVANT QUI MARQUE LA
206 LDA L1 ;FIN DE LA ZONE A DEPLACER
207 CMP VP
208 BNE R8
209 JMP FIN0
210 R8 DEC L1
211 LDA L1
212 CMP #SFF
213 BNE R7
214 DEC L2
215 R7 DEC N1
216 LDA N1
217 CMP #SFF
218 BNE R60
219 DEC N2
220 JMP R60
221 DCM "INT"
222 END
```

Programme DIM.VAR.OBJ

```
927C- A4 6B 84 18
9280- A4 6C 84 19 A0 05 84 08
9288- A2 00 86 07 86 1B 86 1C
9290- 86 1D 86 09 F0 19 20 B1
```

```
9298- 00 D0 04 E6 1B D0 35 C9
92A0- 24 F0 11 C9 25 F0 33 C9
92A8- C8 F0 21 C9 C9 F0 20 95
92B0- 06 E8 D0 E2 E6 1D A5 07
92B8- 09 80 85 07 A0 01 B1 B8
92C0- 20 BA 00 F0 0F 20 B1 00
```



```

92C8- C9 C8 D0 03 4C 50 93 E6
92D0- 1C 4C 50 93 20 E5 92 4C
92D8- 11 93 E6 1D A5 06 09 80
92E0- 85 06 4C B4 92 A0 00 B1
92E8- 18 C5 06 D0 07 C8 B1 18
92F0- C5 07 F0 1C 20 00 93 85
92F8- 19 A5 1A 85 18 4C E5 92
9300- A0 02 B1 18 18 65 18 85
9308- 1A C8 B1 18 65 19 85 1E
9310- 60 A5 18 85 42 A5 19 85
9318- 43 C8 B1 18 85 08 C8 B1
9320- 18 85 09 20 00 93 A5 1A
9328- 85 3C A5 1E 85 3D A5 6D
9330- 85 3E A5 6E 85 3F A0 00
9338- 20 2C FE A5 6D 38 E5 08
9340- 85 6D A5 6E E5 09 85 6E
9348- A6 1B D0 03 20 B1 00 60
9350- 20 E5 92 A0 04 B1 18 C9
9358- 01 D0 F1 20 00 93 A5 1D
9360- F0 09 C6 08 A5 08 38 E5
9368- 1D 85 08 A5 1C F0 38 A0
9370- 06 B1 18 38 E9 01 91 18
9378- 88 B1 18 E9 00 91 18 A0
9380- 02 B1 18 38 E5 08 91 18
9388- C8 B1 18 E9 00 91 18 A5
9390- 1A 48 A5 1E 48 20 00 93
9398- 85 43 A5 1A 85 42 68 85
93A0- 3D 68 85 3C 4C 2E 93 A0
93A8- 06 B1 18 18 69 01 91 18

```



```

93B0- 88 B1 18 69 00 91 18 A0
93B8- 02 B1 18 18 65 08 91 18
93C0- C8 B1 18 69 00 91 18 A5
93C8- 6E 85 07 85 09 A5 6D 85
93D0- 06 18 65 08 85 6D 85 08
93D8- 90 04 E6 6E E6 09 A0 00
93E0- B1 06 91 08 A5 07 C5 1E
93E8- D0 09 A5 06 C5 1A D0 03
93F0- 4C 4C 93 C6 06 A5 06 C9
93F8- FF D0 02 C6 07 C6 08 A5
9400- 08 C9 FF D0 DB C6 09 4C
9408- E0 93

```

```

1 REM DEMO DIM VARIABLE
2 REM =====
5 HIMEM: 37499
10 PRINT CHR$(4)"BLOAD DIM.VAR.
   OBJ": POKE 1013,76: POKE 1014
   ,7 * 16 + 12: POKE 1015,9 * 1
   6 + 2
20 DIM B(5),T$(1),CC%(2,3),TT$(10)
30 FOR I = 0 TO 5:B(I) = 1000 * I
   : NEXT I: FOR I = 0 TO 2: FOR
   J = 0 TO 3:CC%(I,J) = I * J: NEXT
   : NEXT I: FOR I = 0 TO 10:TT$(
   I) = "TABLEAU TT$ " + STR$(
   I): NEXT
40 T$(0) = "PRE-CREE 0":T$(1) = "P
   RE-CREE 1": FOR I = 2 TO 20: &
   T$ + :T$(I) = "CREE NO " + STR$(
   I): PRINT FRE (0): NEXT I: FOR
   I = 0 TO 20: PRINT T$(I): NEXT
50 GET Z$: PRINT : FOR I = 20 TO
   6 STEP - 1: & T$ - : PRINT FRE
   (0): NEXT I: FOR I = 0 TO 5: PRINT
   T$(I): NEXT
60 GET Z$: PRINT : PRINT : & B: DIM
   AA$(0),B(0)
70 INPUT "1ER ELEMENT ";AA$(0)
80 INPUT "UN AUTRE : ";Z$: IF Z$ =
   "N" THEN 200
90 Z = Z + 1: & AA$ + : PRINT "ELE

```

```

MENT NO "Z" "": INPUT AA$(Z):
GOTO 80
200 PRINT : PRINT : FOR I = 0 TO
Z: PRINT AA$(I): NEXT I: GET Z
$: PRINT : PRINT
210 & CC%: DIM CC%(0)
220 FOR I = 1 TO 20: & B + :B(I) =
2 * I: & CC% + :CC%(I) = 10 *
I: NEXT I: FOR I = 0 TO 20: PRINT
B(I),CC%(I): NEXT I: GET Z$: PRINT
: PRINT
230 & TT$: FOR I = 0 TO 4:TT$(I) =
"NOUVEAU TT$ " + STR$(I): NEXT
240 FOR I = 20 TO 10 STEP - 1: &
B - : & CC% - : NEXT I: FOR I =
0 TO 9: PRINT B(I),CC%(I): NEXT
250 & AA$: DIM AA$(0)
260 FOR I = 1 TO 450: & AA$ + :AA
$(I) = "AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA$ " + STR$(I):
PRINT AA$(I)" "": PRINT FRE
(0): PRINT : NEXT
270 FOR I = 450 TO 1 STEP - 1: &
AA$ - : PRINT FRE (0): NEXT
: PRINT : FOR I = 0 TO 9: PRINT
B(I),CC%(I): NEXT
280 FOR I = 0 TO 4: PRINT TT$(I):
NEXT

```

& ONERR GOTO

Dimitri Auer

Faisant suite à l'article de Guy Mathieu, relatif à l'ergonomie des programmes, paru dans Pom's 6, voici une routine en langage machine qui viendra en aide aux programmeurs soucieux de créer des logiciels amicaux. J'attaque ici un problème qui n'a guère été soulevé dans l'article précité : le traitement des erreurs.

Le premier type d'erreur, le fameux "bug" qui ne surviendra qu'après quelques mois d'utilisation du programme, sera facilement éliminé grâce au programme d'analyse syntaxique d'Olivier Herz (cf. Pom's 6). Par contre, il est difficile de prendre en compte, et surtout de prévoir, les erreurs de manipulation éventuelles de la part de l'utilisateur. Je ne connais rien de plus désagréable que de voir un programme stopper son exécution en plein travail par un "I/O ERROR", un "DISK FULL" ou que sais-je encore...

C'est pour éviter ce genre de désagréments que l'Applesoft met à votre disposition l'instruction "ONERR GOTO", qui permet de "piéger" toutes les erreurs du DOS, celles du BASIC, et même la frappe au clavier du fameux CTRL-C. Mais le "ONERR GOTO" souffre de deux défauts majeurs :

- il n'empêche pas l'arrêt du programme par un RESET volontaire ou accidentel;
- si un trop grand nombre d'erreurs successives est traité, le désordre qui apparaît au niveau du pointeur de pile provoque un blocage du système (voir page 136 du manuel Applesoft).

Pour ces raisons, il m'a semblé utile de créer une routine de "ONERR GOTO" étendu, traitant ces deux problèmes.

Principes généraux

L'Apple dispose d'une routine de RESET appelée par l'intermédiaire d'un vecteur, baptisé SOFTEV, situé en page 3 aux adresses \$3F2 et \$3F3 (1010-1011). En modifiant le contenu de ces deux adresses, on peut envoyer le RESET à n'importe quelle adresse au sein de la mémoire de l'ordinateur. Pour signifier à l'Apple de ne pas effectuer un "départ à froid", il faut en outre stocker le résultat du "ou exclusif" entre le contenu de \$3F3 et la valeur \$A5 en \$3F4, ce que la routine du moniteur SETPWRC (\$FB6F) fera pour nous.

```
1 ; APPLE TOOL KIT EDITOR ASSEMBLER
2 *****
3 *
4 * & ONERR GOTO *
5 * *
6 * TRAITEMENT D'ERREUR APPLESOFT *
7 * AVEC EXTENSION AU 'RESET' *
8 * *
9 * COPYRIGHT 1983 *
10 * DIMITRI AUER *
11 * *
12 * *
13 *****
14 ;
15 ; EFFECTUE LES MEMES OPERATIONS QUE
16 ; LE CLASSIQUE ONERR GOTO DE L'APPLESOFT
17 ; MAIS S'APPLIQUE AUSSI A L'ENFORCEMENT
18 ; DE LA TOUCHE 'RESET'.
19 ;
20 ; SYNTAXE: & ONERR GOTO (NUM. LIGNE)
21 ;
22 ;
23 ; ORG $300
24 ;
25 ; CONSTANTES
26 ; -----
27 ;
28 ONERR EQU $A5 ; TOKEN DE ONERR
29 GOTO EQU $AB ; TOKEN DE GOTO
30 RESTORE EQU $AE
31 ;
32 ; VARIABLES
33 ; -----
34 ;
35 EXTLOC EQU $3F5 ; ADRESSE VERS LAQUELLE POINTE L'APPEL
36 TXTPTR EQU $B8 ; POINTEUR DANS LA LIGNE COURANTE APPL
37 CURLIN EQU $75 ; NUMERO DE LA LIGNE EN EXECUTION
38 ERRPTR EQU $F4 ; POINTEURS ONERR
39 ERRFLAG EQU $D8 ; FLAG D'ACTIVITE DE ONERR
40 SOFTEV EQU $3F2 ; VECTEUR DU RESET
41 ERRSTK EQU $DF ; VALEUR DU POINTEUR DE PILE
42 ; AVANT L'APPARITION DE L'ERREUR.
43 ;
44 ;
45 ;
46 ; ROUTINES EXTERNES
47 ; -----
48 ;
49 CHRGET EQU $00B1 ; CAPTE LE CARACTERE SUIVANT DANS LA L
50 SNERR EQU $DEC9 ; AFFICHE 'SYNTAX ERROR'
51 SETPWRC EQU $FB6F ; REAJUSTAGE DES VECTEURS DE RESET
52 DATA EQU $D995 ; DEPLACE TXTPTR EN FIN D'INSTRUCTION
53 ;
54 ;
55 ;
56 ; MISE EN PLACE DES VECTEURS &
57 ;
58 LDA #$4C ; JUMP
59 STA EXTLOC
60 LDA #>START
61 STA EXTLOC+1
62 LDA #<START
63 STA EXTLOC+2
64 RTS
65 ;
66 ;
67 ; PROGRAMME PRINCIPAL APPELE
68 ; PAR CHAQUE EMPLOI DE L'AMPERSAND &
69 ;
70 START CMP #ONERR ; TOKEN ONERR ?
71 BEQ TEST2
72 CMP #RESTORE ; DEUXIEME POSSIBILITE ?
73 BNE ERR ; SI NON ERREUR
74 BEQ STACK ; TRAITE & RESTORE
```

L'astuce utilisée ici consiste à revectoriser le RESET vers la routine de traitement du "SYNTAX ERROR". Chaque RESET sera alors interprété comme une erreur de syntaxe et pourra être capté aisément par ONERR GOTO.

Le programme

Il est évident que cela pourrait être effectué directement à partir du programme Applesoft par une série de POKEs et un CALL. Mais comme il sera bon, de toutes façons, d'utiliser la routine d'erreur préconisée par le manuel Applesoft, autant inclure le tout dans un petit utilitaire.

Pour conserver la souplesse du BASIC, nous utiliserons l'ampersand (&) pour appeler la routine en langage machine.

La première partie de cette routine met en place les vecteurs nécessaires à & en \$3F5-\$3F6 (lignes 58 à 64). A chaque appel de &, on passera ensuite directement à la seconde partie (ligne 67).

Descriptif du programme principal

- Lignes 67-77 : teste la syntaxe de l'appel soit & ONERR GOTO, soit & RESTORE. Si ONERR GOTO, branchement à OK. Si RESTORE, saut à la deuxième partie (traitement de pile). En cas d'échec des deux comparaisons, saut à la routine qui affichera un "SYNTAX ERROR" et stoppera le programme.
- Lignes 81-99 : activation de ONERR GOTO et mise en place des nouveaux vecteurs de RESET. Sortie via la routine DATA qui place le pointeur de ligne Applesoft en fin d'instruction (recherche un ";" ou un "0" de fin de ligne).
- Lignes 111-119 : réajuste le pointeur de pile à la valeur précédant l'apparition de l'erreur (voir manuel Applesoft). Cette section est appelée par & RESTORE.

```

75 TEST2 JSR CHRGET ; INSTRUCTION SUIVANTE
76 CMP #GOTO ; TOKEN GOTO ?
77 BEQ OK
78 ;
79 ERR JMP SNERR ; ENVOIE LE MESSAGE D'ERREUR
80 ;
81 OK JSR CHRGET ; AVANCE TXTPTR
82 LDA TXTPTR
83 STA ERRPTR
84 LDA TXTPTR+1
85 STA ERRPTR+1
86 LDA CURLIN
87 STA ERRPTR+2
88 LDA CURLIN+1
89 STA ERRPTR+3
90 ;
91 LDA ##80
92 STA ERRFLAG ; ACTIVATION DE ONERRGOTO
93 ;
94 LDA #)SNERR ; REVECTORISATION
95 STA SOFTEV ; DU
96 LDA #<SNERR ; RESET
97 STA SOFTEV+1
98 JSR SETPWRC
99 JMP DATA ; FIN
100 ;
101 ;
102 ; DEUXIEME PARTIE: & RESTORE
103 ; REAJUSTE LE POINTEUR DE PILE
104 ; AFIN D'EVITER LE 'PLANTAGE'
105 ; DU SYSTEME PAR SUITE D'ERREURS
106 ; REPETEES.
107 ;
108 ; A UTILISER DANS LA SECTION DE TRAITEMENT
109 ; DE L'ERREUR PAR UN & RESTORE
110 ;
111 STACK PLA ; SAUVEGARDE
112 TAY ; DES REGISTRES
113 PLA
114 LDX ERRSTK ; REAJUSTAGE DU POINTEUR
115 TXS ; DE PILE
116 PHA ; REGENERATION
117 TYA ; DES
118 PHA ; REGISTRES
119 JMP DATA ; FIN ET RETOUR AU BASIC

```

Remarque : les utilisateurs de disquettes auront à inclure un CALL 1002 au sein de leur routine de traitement d'erreur afin de reconnecter le DOS après un éventuel RESET.

Le petit programme de démonstration listé ci-après met en évidence le mode d'utilisation de cette routine.

Récapitulation

*300.35D

```

0300- A9 4C 8D F5 03 A9 10 8D
0308- F6 03 A9 03 8D F7 03 60
0310- C9 A5 F0 06 C9 AE D0 09
0318- F0 31 20 B1 00 C9 AB F0
0320- 03 4C C9 DE 20 B1 00 A5
0328- B8 85 F4 A5 B9 85 F5 A5
0330- 75 85 F6 A5 76 85 F7 A9
0338- 80 85 D8 A9 C9 8D F2 03
0340- A9 DE 8D F3 03 20 6F FB
0348- 4C 95 D9 68 A8 68 A6 DF
0350- 9A 48 98 48 4C 95 D9 D0
0358- DA E6 01 4C 33 03

```

10 REM

```

*****
*
*      & ONERR DEMO
* (C) 1983 DIMITRI AUER
*
*****

```

20 REM

```

30 REM PLOT HARGEMENT DE LA ROU
TINE
40 PRINT CHR$(4)"BRUNAMPER.ONER
R.OBJO"

```

```

50 REM ACTIVATION DE ONERR
100 & ONERR GOTO 1000

```

200 REM

```

*****
*
* PROGRAMME PRINCIPAL
*
*****

```



```

*
*****
210 TEXT : HOME
215 INVERSE : FOR I = 1 TO 40 * 3
    : PRINT " ";: NEXT : NORMAL
220 VTAB 2: HTAB 13: PRINT " & ON
    ERR DEMO "
240 PRINT : PRINT : PRINT "LIST20
    00,2010": PRINT : LIST 2000,2
    010
250 VTAB 21: INVERSE : PRINT "ESS
    AYEZ DE STOPPER LE PROGRAMME
    PAR CTRL-C OU RESET. ENFO
    NCEZ LA TOUCHE 'S' POUR EXECU
    TER LA LIGNE 2010,
    ";
255 NORMAL
260 POKE - 16368,0
270 VTAB 18: HTAB 15: PRINT "PAUS
    E..."
280 FOR PA = 1 TO 1000: NEXT PA
290 VTAB 18: HTAB 15: PRINT "
    "
295 FOR I = 1 TO 10
300 X = PEEK ( - 16384) - 128
310 IF X = ASC ("S") THEN 2000: REM
    PRODUIT L'ERREUR
315 NEXT

```

```

320 GOTO 260: REM BOUCLAGE
330 REM
999 REM
*****
*
* ROUTINE DE CAPTAGE DE
* L'ERREUR
*
*****
1000 & RESTORE : REM REMET A JO
    UR L POINTEUR DE PILE
1005 CALL 1002: REM ET EN PLACE L
    ES VECTEURS
1006 REM DU DOS EN CAS DE RESET
1007 HOME : VTAB 12: NORMAL
1008 FOR I = 1 TO 20: X = PEEK (4
    9200): NEXT
1010 PRINT "ERREUR!!!"
1020 PRINT : PRINT "CODE:" PEEK (
    222)
1030 PRINT "LIGNE NO:" PEEK (218)
    + 256 * PEEK (219)
1040 POKE - 16368,0
1050 PRINT : GET R$: GOTO 200
2000 REM EXEMPLE D'ERREUR
2010 PRINT LOAD : REM
    MOT RESERVE

```

Programme assembleur à la fin d'un programme BASIC

Roland Jost

Lorsqu'un programme BASIC utilise un ou plusieurs programmes en langage machine, la méthode la plus courante consiste à charger ceux-ci dans la page 3, ou en fin de mémoire vive, par des BLOAD.

Cette méthode présente deux inconvénients :

- 1) perte de temps au moment du chargement en mémoire ;
- 2) perte de place sur la disquette.

Incorporer le programme machine à la fin du programme BASIC est une autre méthode répandue. Il y a deux avantages. D'une part, le chargement se fait en une seule fois, donc gain de temps. D'autre part, la copie sur disquette ou cassette est plus simple. Inconvénient : si l'on veut modifier le programme BASIC, il faut recommencer toute la procédure.

Rajouter un programme machine à la fin d'un programme BASIC suppose l'exécution des opérations suivantes :

- 1) charger le programme BASIC
- 2) prévoir les modifications des adresses d'appel des sous-programmes

3) chercher la fin du programme BASIC par un PRINT PEEK(175) + 256*PEEK(176)

4) charger le programme binaire à cette adresse

5) modifier les adresses d'appel dans le programme BASIC. Attention, ceci ne doit pas modifier la longueur du programme BASIC.

6) Positionner le pointeur de fin de programme à la fin du programme binaire.

7) Sauver le programme (sous un autre nom par précaution).

Bien que ces opérations soient parfaitement réalisables à la main, elles sont fastidieuses et une erreur est toujours possible. Il est donc plus sûr d'automatiser tout le processus comme nous vous le proposons ici.

Le fichier INCLUSION

Ce fichier EXECutable vous demande le nom du programme BASIC, sauvegarde ses pointeurs, modifie les adresses de chargement du BASIC et charge le programme APPEND-PROG à l'adresse \$6000.

APPEND-PROG

Ce programme Applesoft, chargé par INCLUSION, vous demande le(s) nom(s) du ou des programmes binaires à utiliser. Taper RETURN s'il n'y en a plus.

Après exécution de APPEND-PROG, il ne vous reste plus qu'à sauvegarder votre programme BASIC, qui englobera les routines en langage machine.

Pour que l'opération se déroule normalement, il est nécessaire d'insérer dans les premières lignes de votre programme Applesoft une instruction du type :

P1 = \$\$\$\$\$: P2 = \$\$\$\$\$: Pn = \$\$\$\$\$

P1.....Pn étant les noms de vos sous-programmes binaires. Attention : les mots réservés du BASIC ne doivent pas être utilisés.

Les sous-programmes devront être appelés par des CALL P1.... CALL Pn.

Dans le cas d'une table de formes, utiliser une instruction telle que :

SHAPE = \$\$\$\$: POKE 233,
SHAPE / 256; POKE 232, SHAPE -
PEEK(233) * 256

Pour une bibliothèque de programmes utilisant & (Ampersand), insérer :

LIB = \$\$\$\$: POKE 1013, 76 :
POKE 1015, LIB/256 : POKE 1014,
LIB - PEEK(1015)* 256

Si vos routines ne sont pas relogeables, reportez-vous à l'article de Philippe François dans Pom's 7, ou faites les modifications nécessaires avant la sauvegarde du programme complet.

Quelques détails

EXEC INCLUSION

HOME : INVERSE : PRINT "INCLUSION DE PROGRAMMES ASSEMBLEUR A LA FIN D'UN PROGRAMME APPLESOFT " : NORMAL : PRINT

POKE 118,0 : POKE 43699,0 :
INPUT "NOM DU PROGRAMME
APPLESOFT " : N\$: POKE 43699,1

Cette ligne mérite quelques explications. L'adresse 118 (\$76, baptisée CURLIN) contient \$FF en mode immédiat. En mettant zéro, on trompe l'interpréteur. Ceci permet d'exécuter des commandes INPUT, GET en mode immédiat sans le fatidique "ILLEGAL DIRECT ERROR". POKE 43699,0 permet de déconnecter la commande EXEC. A la fin de l'instruction, un POKE 43699,1 rend la main au fichier EXEC.

PRINT CHR\$(4) "LOAD " N\$

POKE 26, PEEK(175) : POKE 27,
PEEK(176)

on sauve l'adresse de fin de programme.

POKE 103, 1 : POKE 104, 96 :
POKE 24576,0

modifie l'adresse de chargement d'un programme APPLESOFT.

RUN APPEND-PROG

APPEND-PROG

Le sous-programme en langage machine implanté en page 3 permet, au terme de l'exécution, de redonner aux pointeurs Applesoft des valeurs standards.

Pour tester la procédure, nous vous proposons d'utiliser le programme ESSAI-INCLUSION comme programme Applesoft. Les sous-programmes nécessaires sont INV (inversion de chaînes de caractères) et STRING (affiche x fois une chaîne). Ils vous sont offerts "en prime" !

INV

Certains BASICs, ainsi que PLE, possèdent une fonction permettant l'inversion d'une chaîne de caractères. Le programme assembleur INV réalise cette opération sur une chaîne ayant au plus 127 caractères.

Appel par : CALL 768,A\$ (CALL 768,nom de la chaîne)

STRING

Cette fonction existe dans certains BASICs. Elle permet d'afficher n fois la même chaîne de caractères.

Programme INV

```

0300- 20 BE DE JSR $DEBE : teste la virgule
0303- 20 E3 DF JSR $DFE3 : recherche variable
0306- 85 FA STA $FA : sauve les pointeurs
0308- 84 FB STY $FB :
030A- A0 00 LDY #00 :
030C- B1 FA LDA ($FA),Y :
030E- 85 FC STA $FC : sauve la longueur
0310- C8 INY :
0311- B1 FA LDA ($FA),Y :
0313- 48 PHA :
0314- C8 INY :
0315- B1 FA LDA ($FA),Y :
0317- 85 FE STA $FE : sauve l'adresse
0319- 68 PLA :
031A- 85 FD STA $FD :
031C- A4 FC LDY $FC : on balaye la chaîne
031E- 88 DEY :
031F- B1 FD LDA ($FD),Y :
0321- 48 PHA :
0322- 88 DEY :
0323- 10 FA BPL $031F : fin de chaîne ?
0325- A4 FC LDY $FC : on reconstitue
0327- 88 DEY : la chaîne en
0328- 68 PLA : r(cuperant la pile.
0329- 91 FD STA ($FD),Y :
032B- 88 DEY :
032C- 10 FA BPL $0328 :
032E- 60 RTS :
```

Programme STRING

```

0300- 20 BE DE JSR $DEBE : teste la virgule
0303- 20 F8 E6 JSR $E6F8 : {value la formule
0306- 86 06 STX $06 : sauvegarde de la valeur
0308- 20 BE DE JSR $DEBE : teste la virgule
030B- C9 22 CMP #22 : est-ce un "
030D- D0 05 BNE $0314 :
030F- 20 81 DE JSR $DE81 : saisie de la chaîne
0312- D0 0A BNE $031E :
0314- 20 E3 DF JSR $DFE3 : recherche la variable
0317- 20 6C DD JSR $DD6C : ce doit ~tre une chaîne
031A- 85 A0 STA $A0 :
031C- 84 A1 STY $A1 :
031E- 20 3D DB JSR $DB3D : affiche la chaîne
0321- C6 06 DEC $06 :
0323- D0 F9 BNE $031E : on recommence ?
0325- 20 B7 00 JSR $00B7 :
0328- C9 3B CMP #3B : teste le point-virgule
032A- D0 04 BNE $0330 :
032C- 20 B1 00 JSR $00B1 :
032F- 60 RTS :
0330- 4C 8E FD JMP $FD8E : retour chariot
```

Appel :

CALL 768,n,A\$
CALL 768,n,"chaîne"
CALL 768,n,A\$;
CALL 768,n,"chaîne";

Le point virgule évite un retour chariot après l'affichage.

REMARQUES IMPORTANTES

1 - les noms de vos sous-programmes ne doivent pas se terminer par un \$.

2 - Attention aux pages graphiques. Condensez votre programme BASIC

au maximum en combinant les lignes et en supprimant les remarques. Si nécessaire, n'utilisez que la page graphique 2.

```
100 REM PROGRAMME APPEND
101 REM =====
110 PRINT
120 DEF FN PE(X) = PEEK (X) + 2
    56 * PEEK (X + 1)
130 D$ = CHR$(4):G$ = CHR$(7)
140 AF = FN PE(26)
145 FF = AF * (AF < = 2200) + 220
    0 * (AF > 2200)
150 ONERR GOTO 430
160 GOSUB 250
170 HOME : FOR I = 0 TO K: PRINT
    B$(I): NEXT I: PRINT : IF K =
    0 THEN PRINT " EST";: GOTO 1
    90
180 PRINT "SONT";
190 PRINT " MAINTENANT INCLUS A V
    OTRE PROGRAMME BASIC. LES APP
    ELS (CALL) DE VOTRE PRO
    GRAMME BASIC ONT ETE MIS A JO
    UR."
200 PRINT : PRINT : PRINT " VOUS
    POUVEZ LISTER VOTRE PROGRAMME
    ET LE SAUVER SOUS LE NOM
    QUI VOUS CONVIENT."
210 DATA 169,1,133,103,169,8,13
    3,104,165,26,133,175,133,105,
    165,27,133,176,133,106,96,0
220 FOR I = 0 TO 21: READ DA: POKE
    768 + I,DA: NEXT I
230 CALL 768
240 END
250 REM RECHERCHE DES SOUS-PROGR
    AMMES
260 FOR I = 2048 TO FF: IF PEEK
    (I) < > 208 THEN 400
270 IF PEEK (I + 1) < > 36 OR PEEK
    (I + 2) < > 36 OR PEEK (I +
    3) < > 36 OR PEEK (I + 4) <
    > 36 OR PEEK (I + 5) < > 3
    6 THEN 400
280 FOR J = I - 1 TO I - 11 STEP
    - 1: IF PEEK (J) < 65 THEN
    K1 = J + 1: J = I - 11
290 NEXT J
300 B$ = "":
310 FOR N = K1 TO I - 1: B$ = B$ +
    CHR$( PEEK (N)): NEXT N
340 REM CHARGEMENT DES SOUS-PROG
    RAMMES
350 PRINT : PRINT B$" CHARGE A L
    'ADRESSE "AF
```

```
360 PRINT D$"BLOAD "B$",A"AF
370 B$(K) = B$
380 AF$ = RIGHT$ (" " + STR$
    (AF),5): FOR L = 1 TO 5: POKE
    I + L, ASC ( MID$( AF$,L,1)):
    NEXT L:I = I + 1
390 LG = FN PE(43616):AF = AF + L
    G: POKE 27,AF / 256: POKE 26,
    AF - PEEK (27) * 256:K = K +
    1
400 NEXT I
410 RETURN
420 REM TRAITEMENT D'ERREURS
430 IF PEEK (222) = 6 THEN FLASH
    : PRINT G$:G$"CE FICHER N'EX
    ISTE PAS": NORMAL : GOTO 210
440 FLASH : PRINT G$:G$"ERREUR D'
    EXECUTION ": GOTO 210
```

```
100 REM PROGRAMME DE TEST APPEND
101 REM
110 HOME
120 INV = $$$$$:STRING = $$$$$
130 V$ = CHR$(91) + "I": CALL ST
    RING,20,V$:
140 FOR J = 1 TO 19: HTAB 1: CALL
    STRING,1,V$;: HTAB 39: CALL S
    TRING,1,V$;: NEXT J
150 CALL STRING,20,V$
160 POKE 32,4: POKE 33,33
170 VTAB 3: INPUT "VOTRE PRENOM ,
    SVP ";A$
180 CALL INV,A$
190 VTAB 8: PRINT "MERCI ";: INVERSE
    : PRINT A$: NORMAL
200 FOR T = 1 TO 3000: NEXT
210 PRINT : PRINT "OH! EXCUSEZ MO
    I ";: INVERSE : PRINT A$: NORMAL
    : PRINT : PRINT "J'AI INVERSE
    LES LETTRES"
220 FOR T = 1 TO 3000: NEXT
230 CALL INV,A$
240 : PRINT : PRINT "VOILA,C'EST R
    ENTRE DANS L'ORDRE "
250 PRINT : PRINT : HTAB 10: PRINT
    "AU REVOIR ";: INVERSE : PRINT
    A$: NORMAL
260 PRINT : HTAB 10: CALL STRING,
    10 + LEN (A$),"#"
270 TEXT : VTAB 22
```


BONJOUR LES PRIX !!

NOS PRIX SONT F TTC

Carte langage	550	Speech card	390
Carte 128 k ram	1800	Carte horloge	600
Carte 80 colonnes	750	Joystick	200
Interface série	650	Ventilateur	320
Super série	1100	Contrôleur de drive auto switch 13/16	450
Interface parallèle	420	Microdrive 3"	2500
Grappler + buffer 16 k	1500	Moniteur vert 12"	950
Interface modem	2200	Disquettes 5" 1/4 par 1 boîte	180/boîte
Carte Z 80	750	" " " 5 boîtes	175/bôte
Wildcard	650	" " " 10 boîtes	170/boîte

AU-DESSUS, NOUS CONSULTER.

Carte bleue et visa acceptées
Crédit sous réserve d'acceptation du dossier

Computer 3

3, rue Papillon 75009 Paris - Tél. 523.51.15
(metro Poissonnière)



LA PHOTOCOMPOSITION EN PROLONGEMENT DE LA MICRO-INFORMATIQUE



TRANSMETTEZ-NOUS VOS TEXTES
PAR TÉLÉPHONE

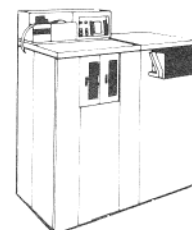
ou

DONNEZ-NOUS VOTRE DISQUETTE



Les textes de vos articles, catalogues, annuaires ou brochures saisis sur votre APPLE sont envoyés directement sur notre photocomposeuse.

Nous vous évitons ainsi, le coût et le temps de la saisie supplémentaire que nécessite le traitement traditionnel de la photocomposition avant l'impression des documents, si vous le désirez nous pouvons également nous charger de l'impression et du brochage.



NOTRE RÉFÉRENCE... LA REVUE POM'S

TELECOMPO 328.18.63

PHOTOCOMPOSITION
BUREAUTIQUE
TRANSMISSION DE DONNÉES

GESTION DE FICHIERS
MATÉRIEL DE
TRAITEMENT DE TEXTES

13 et 15 avenue du Petit Parc
94300 VINCENNES

Des trucs pour Apple Writer II et //e

Hervé Thiriez

Mon propos dans cet article n'est pas de vous fournir un mode d'emploi d'Apple Writer II ou //e (il existe en effet une fonction Help avec "PO-?", pomme ouverte et "?" simultanés), mais de vous donner des trucs d'utilisation indépendants les uns des autres.

Azerty ou Qwerty ?

Avec l'Apple //e, le choix entre le mode Azerty et le mode Qwerty n'est pas forcément évident. Quand on doit souvent taper des chiffres, il peut être préférable de taper en Qwerty (pour avoir les chiffres en minuscules) et d'utiliser les touches US permettant d'obtenir les caractères spéciaux français : les accolades ouvrantes (é) et fermantes (è), l'arrobas (à), la barre transversale inversée (ç), la barre verticale (ü) et le tilde (tréma).

Le soulignement

Le paramètre de souligné est par défaut le symbole "&", ce qui est assez fâcheux, car on a facilement recours à ce symbole, d'où quelques soulignements inattendus. Avant d'imprimer un texte, vérifiez donc qu'il ne comporte pas de "&" et modifiez le paramètre de souligné dans le cas contraire, en utilisant un caractère non employé dans votre fichier.

Si votre imprimante ne possède pas de mode souligné, vous pouvez annihiler totalement l'option soulignement en entrant, une fois parvenu dans le mode de changement des paramètres d'impression (avec "CTRL-P ?"), la séquence "PS" immédiatement suivie de RETURN.

Si votre imprimante possède un mode soulignement obtenu par des codes différents de ceux de l'imprimante matricielle Apple, vous devrez entrer vous-même les séquences de début et fin de soulignement. Bien entendu, vous perdrez dans ce cas la justification pour les lignes où apparaîtront ces symboles.

La coupure des mots

La touche CTRL-Z permet de basculer entre la présentation des mots à l'écran, coupés ou non. L'intérêt de l'option "coupure" réside dans le fait que l'on voit alors plus clairement s'il n'y a pas de RETURNS intempestifs à l'intérieur des paragraphes. Inversement, la "non-coupure" rend le texte plus lisible, quoiqu'à cet égard les opinions des utilisateurs puissent dif-

férer. En faisant "CTRL-Q G", on bascule entre l'affichage ou non des retours chariot.

Il y a une autre raison pour choisir l'un de ces deux modes : le déplacement vertical du curseur ne s'effectue pas de la même manière pour chaque mode. Ainsi, si l'on veut effectuer des corrections en fin de paragraphe, et en remontant dans le texte, il vaut mieux être en mode "coupure". Par contre, pour remonter le curseur verticalement sans décalage latéral, le mode "non-coupure" est nécessaire.

Le glossaire

Vous n'utiliserez jamais trop souvent le glossaire. Si votre Apple Writer est un outil d'usage fréquent, vous pouvez même avoir recours à plusieurs glossaires, chacun spécialisé pour une tâche précise (évitée cependant les confusions). On peut bien entendu mettre dans un glossaire un mot long tel que "micro-ordinateur", ou une formule de politesse comme "Veuillez agréer, Monsieur, ...". C'est ainsi que certaines lectrices reçoivent parfois de ma part des lettres aux formules douteuses.

Pour utiliser le glossaire avec l'Apple //e, ne vous fatiguez pas à entrer "CTRL-G" suivi du caractère : il suffit de faire "PO" (pomme ouverte) et le caractère simultanément.

La meilleure façon de concevoir des glossaires spécifiques consiste probablement à créer un noyau de glossaire, contenant par exemple tous les caractères de contrôle de l'imprimante (qualité courrier, compressé, caractères étendus, ...), qui servira de point de départ aux glossaires spécifiques.

On peut - ne nous en privons pas - utiliser des commandes de glossaire incorporant des caractères de contrôle ou des RETURNS.

Caractères de contrôle

Avec l'imprimante matricielle Apple, un "è" s'obtient avec la séquence "CTRL-V CTRL-H CTRL-V e", soit huit touches ... L'idéal serait de créer cette séquence dans le glossaire, pour pouvoir l'appeler par exemple avec "PO-e". Malheureusement, le glossaire refuse, dans son mode de création normal, la séquence "CTRL-V CTRL-H CTRL-V". Il faut alors sauvegarder le glossaire sur la disquette, puis le charger comme un fichier normal. On peut ensuite rentrer

la séquence de touches de la façon suivante : "CTRL-L CONTROLV (CONTROLV est un fichier qui se trouve sur la disquette Apple Writer //e) CTRL-V CTRL-H CTRL-V CTRL-L CONTROLV e". Le chargement de CONTROLV place simplement un caractère CTRL-V, matérialisé par un V en inverse, à la place du curseur, tandis que la séquence "CTRL-V CTRL-H CTRL-V" permet d'obtenir un CTRL-H (soit un H en inverse à l'écran). Il ne reste qu'à sauvegarder ce fichier glossaire modifié.

D'ailleurs, lors de la création ou de la mise à jour d'un glossaire, il est à la fois plus rapide et plus sûr de procéder ainsi que de passer par "CTRL-G ?" à chaque item.

Insertion de RETURNS

Un retour chariot est indiqué dans le glossaire par le symbole "\$" en Azerty, et "J" en Qwerty. Ainsi, pour marquer le passage en mode compressé et le changement correspondant des marges avec mon Oki92, j'ai créé la commande de glossaire "PO-c", directement à partir de "CTRL-G ?", en entrant "c CTRL-V CTRL-X CTRL-V \$ CTRL-V CTRL-\$ CTRL-V \$.mg10\$.md 121\$.jt". CTRL-X annule le mode qualité courrier et CTRL-\$ met en mode compressé. J'ai ajouté ici, pour faciliter la lecture, un blanc pour séparer les touches de contrôle, mais tout était contigu lors de la création de la commande.

Fichiers tampons

Plutôt que d'encombrer le glossaire avec des textes longs tels que formules de politesse, indication de changement d'adresse, exposition des conditions de vente, ..., il est préférable de créer autant de petits fichiers tampons que l'on chargera, le moment venu, avec un simple CTRL-L. En effet, le glossaire est limité à un total de 2048 caractères, même si l'on peut (voir la documentation du programme) chaîner des définitions du glossaire.

Recherche de chaînes

On se trompe parfois, lors de la recherche (et du remplacement) de chaînes en effectuant la recherche vers le haut alors qu'on souhaitait l'effectuer vers le bas, ou inversement. Pour éviter ce type d'erreur, chaque fois que je désire corriger un texte (en général du haut vers le bas), je précède les corrections d'un

CTRL-B qui, tout en me plaçant en haut du fichier, garantit que les recherches iront dans le bon sens.

Visualisation du texte

Quand on commence à utiliser la pagination, ou les titres en haut et bas de page, on souhaite parfois visualiser le résultat possible avant de lancer effectivement l'imprimante. C'est intéressant aussi pour tester les conséquences de la justification (faut-il ajouter des césures ?).

La solution, toute simple, consiste à indiquer avec "CTRL-P PD0" que l'imprimante (Printer Destination) est en fait l'écran. Si l'on a peur que tout cela défile trop vite, il suffit de prendre par exemple les paramètres d'impression "LI20" et "IP20" et d'indiquer avec "FE1" que l'on souhaite "imprimer" feuille à feuille.

L'impression

Pour imprimer sur du papier en bande un texte continu sans saut de page, il suffit d'entrer (après "CTRL-P ?") les paramètres d'impression suivants : MH0, MB0, LI1, IP1 (ou tout autre nombre à la place du "1", tant qu'il est égal pour LI et IP).

La lecture de fichier

Si vous essayez de lire un fichier dont le nom comporte un trait d'union "-", ce nom n'est pas reconnu en lecture. Heureusement, on peut alors, avec "CTRL-O B" renommer le fichier (là, le nom est reconnu) et lui donner un nom acceptable en lecture.

Mise à jour des paramètres

Les paramètres d'impression, accessibles par "CTRL-P ?", se mettent à jour automatiquement après chaque impression. Cela peut provoquer des surprises : si vous avez défini un titre haut (ou bas) directement dans le texte pour qu'il n'apparaisse pas sur la première page (la page de garde), il se trouvera dans les paramètres généraux après l'impression. Si donc vous lancez ensuite une seconde impression sans avoir pensé à effacer ce titre de la liste des paramètres par défaut, il apparaîtra à la seconde impression dès la page de garde (et cela décalera aussi toute numérotation des pages que vous auriez pu demander).

Le WPL

Le Word Processing Language (WPL) est un excellent langage

complémentaire (livré avec Apple Writer //e) permettant de créer des états personnalisés, d'effectuer des calculs, de préprogrammer en fait de multiples commandes d'Apple Writer.

Nous l'utilisons, chez Pom's, pour transformer automatiquement certaines séquences de caractères qui nous sont propres en celles attendues par la machine de photocomposition de notre imprimeur. Ainsi, Pom's est réalisé grâce à des textes préparés en Apple Writer //e, transformés par WPL, puis transmis par modem à l'ordinateur de l'imprimeur.

La documentation du WPL occupe à elle seule plus de 100 pages; comme on peut en outre travailler quotidiennement avec Apple Writer sans jamais avoir recours au WPL, nous n'en parlerons pas dans cet article.

Bibliographie

Bien entendu, nous vous conseillons (n'en déplaise aux copieurs frénétiques) la documentation du programme, dont la traduction française est bien faite. A part cela, vous pourrez lire "Donnez du caractère à votre imprimante" dans le Pom's 9, et "Un bug Apple Writer //e" dans le Pom's 10 (page 70). ■

Adaptez Applewriter 1.1 à l'Apple //e

Jacques Leroy

Le programme de traitement de textes Applewriter 1.1 des Apple II et II Plus utilise des modules qui permettent d'introduire des minuscules, certains caractères spéciaux et de pallier l'absence de touches de déplacement du curseur.

Son utilisation sur un Apple //e est par contre source de déboires (les espaces sont remplacés par une demi-barre transversale gauche, les chiffres deviennent des lettres minuscules) et cause d'insatisfactions (les possibilités nouvelles du clavier demeurent inutilisables). Il convient par conséquent de modifier ces modules afin d'obtenir le système adéquat.

Dans le Pom's 3, Jean-Louis Meilaud rappelait qu'Applewriter 1.1 est un logiciel "ouvert", à condition de se plonger dans les programmes en langage machine. Après un travail de bénédictin, sans avoir pratiqué antérieurement l'hexadécimal, j'ai réussi à détecter les modifications nécessaires pour adapter Applewriter 1.1 à l'Apple //e. Ces modifications sont à la portée de tous et n'utilisent que le moniteur de l'Apple.

Modification de TEDITOR

UNLOCK TEDITOR
BLOAD TEDITOR
RENAME TEDITOR, OLD.TEDITOR
LOCK OLD.TEDITOR
CALL -151 : passage en mode moniteur

*1505: EA EA EA EA

Minuscules et caractères spéciaux.

*09E0: EA

Un seul espace pour passer en mode "déplacement du curseur".

*09E9: 8B

*09F0: 8A

*09F7: 95

*09FE: 88

Utilisation des quatre flèches pour déplacer le curseur après la touche ESC, pour les corrections à faire dans le texte.

*080C: 83

*0810: C3

Neutralise le CTRL-C dont les effets sont maintenant devenus nocifs.

*3D0G : retour au Basic
BSAVE TEDITOR, A\$803, L\$1040
LOCK TEDITOR

Modification de PRINTER

UNLOCK PRINTER
BLOAD PRINTER
RENAME PRINTER,OLD.PRINTER
LOCK OLD.PRINTER
CALL -151

*12E1: EA EA Caractères spéciaux

*12E8: 00 Majuscules et

*12F6: 00 minuscules

*109D: EA EA EA EA

En-tête de page

*0F56: A1

Commandes d'impression désormais de la forme "IFJ" avec des majuscules.

*3D0G

BSAVE PRINTER, A\$803, L\$1040
LOCK PRINTER

NDLR - Ces transformations vous permettent d'utiliser Applewriter 1.1 sur un Apple //e sans devoir acquérir Applewriter //e. Par contre, les caractères étant recodés, vous ne pourrez avec cette nouvelle version d'Applewriter 1.1 ni lire vos anciens fichiers Applewriter 1.1, ni transformer les nouveaux fichiers Applewriter 1.1 en format Applewriter //e. ■

Un éditeur de shapes

Thierry Le Tallec

Ce programme vous permettra de constituer simplement des tables de shapes et de les modifier selon vos désirs (suppressions ou ajouts de shapes), de les visualiser, les sauver sur disquettes et les relire...

Mode d'emploi

Il ne devrait pas vous poser de problèmes, dans la mesure où le programme indique toujours la nature des traitements possibles et vous y donne accès au moyen de menus.

Pour la création d'une shape, vous disposez d'une "fenêtre" quadrillée représentant un agrandissement de la partie de "forme" en cours de création. La shape complète est dessinée en grandeur réelle et en continu à droite de cette fenêtre. Si vous touchez l'un des bords ce celle-ci au cours des déplacements du spot, elle se déplace également sur la shape, c'est-à-dire que la partie agrandie se décale à l'intérieur de la fenêtre.

Les touches I, J, K et M permettent les déplacements habituels. La touche P place un point à la position du spot dans la shape, et la touche F marque la fin de l'édition pour la création définitive et l'incorporation de la shape dans la table.

La touche E est utilisée pour l'effacement; elle annule l'effet de la dernière touche employée, qu'il s'agisse d'un déplacement ou de la mise en place d'un point. Une séquence de "E à répétition" permet donc de refaire en sens inverse tout le chemin accompli depuis le début de la création.

Une seule précaution à prendre lors de la création d'une shape : ne pas dépasser le nombre maximum de vecteurs autorisé pour chacune d'elles (le programme vous prévient lorsque vous approchez de la zone dangereuse).

Le programme

Il se compose d'un programme en Applesoft et d'une routine en langage machine "cachée" à la fin de celui-ci. Vous trouverez ci-après la récapitulation hexadécimale de cette routine : elle occupe les adresses \$1753 à \$1820 (5971 à 6176 en décimal). Les adresses \$1821 à \$19A1 (6176 à 6561) sont en outre utilisées comme zone de stockage.

Si vous rentrez vous-même ce programme en machine, vous devez donc mettre aux adresses \$AF-\$B0 (175-176), qui contiennent l'adresse de fin du programme Applesoft en mémoire, l'adresse \$19A1 (dans l'ordre A1 19), et ce après avoir saisi l'Applesoft et le code-machine, mais avant de sauver le tout sur disquette par une seule instruction SAVE. Vous pouvez faire, par exemple : POKE 175, 161 : POKE 176, 25 après retour au Basic, avant de taper SAVE "programme".

NDLR : lors de la lecture du programme Applesoft d'édition de shapes, vous trouverez certaines instructions quelque peu bizarres, à base de variables a priori inutiles dont le nom comporte un nombre variable de Z (voyez notamment la dernière ligne !). Si vous rentrez ce programme en machine, il ne vous sera pas nécessaire de reproduire ces instructions, l'essentiel étant que vous respectiez l'adresse de début de la routine en langage machine (\$1753) et que vous ne tapiez pas finalement un programme Applesoft plus long que celui publié ci-après (afin qu'il n'y ait justement pas conflit entre les deux parties du programme). Pour notre part, nous avons laissé ces instructions dans la liste afin qu'elle reste comparable au programme fourni sur la disquette d'accompagnement de ce numéro de Pom's.

Nous allons maintenant vous expliquer brièvement leur raison d'être. La version initiale du programme proposée par l'auteur comportait un grand nombre de caractères de contrôle, destinés notamment à améliorer la présentation du texte à l'écran (sauts de ligne). Mais ces caractères présentent un grave inconvénient : ils n'apparaissent pas lors d'un LIST du programme et peuvent même donner l'impression qu'une série d'espaces sépare les deux parties de texte qui les encadrent. Conséquence : le lecteur ne peut pas reconstituer le programme original à partir de la liste publiée. De plus, abusé par cette liste, il risque de taper un programme trop long, débordant sur les adresses réservées à la routine en langage machine, pour aboutir à un résultat déplaisant : un "plantage". Nous avons donc modifié ce programme, afin d'obtenir exactement le même nombre d'octets pour l'Applesoft, mais uniquement avec des caractères visibles.

Ceci nous fournit l'occasion de lancer un appel à l'intention de tous les auteurs qui envisagent de contribuer à Pom's : n'oubliez pas vos futurs lecteurs ! En d'autres termes, n'incorporez dans vos programmes que des caractères "listables et publiables", surtout si leur fonctionnement suppose l'emploi d'une routine "cachée". De même, il est bon de se rappeler que tous les possesseurs d'Apple ne disposent pas d'un système permettant l'affichage des minuscules à l'écran. Une instruction telle que : PRINT "Appuyez sur C pour continuer" risque fort de laisser perplexe l'utilisateur d'un Apple II+ standard, et il est alors douteux, justement, qu'il puisse continuer. ■

Programme SUPER-SHAPES Partie visible

```
1 ONERR GOTO 24
2 A = 5971 : B# = CHR# (4) : A% = 200
  : DEF FN A(B) = PEEK (B) +
  : 256 * PEEK (B + 1) : HCOLOR=
  : 3 : SCALE= 1 : ROT= 0 : DIM B%(A
  : %): GOTO 21
3 FOR C = - 1 TO 1 : FOR D = 0 TO
  : 1 : HPLOT B + D, E + C : NEXT : NEXT
  : HCOLOR= 3 : RETURN
```

```
4 FOR C = E - 3 TO E + 3 STEP 6 : HPLOT
  : B, C TO B + 1, C : NEXT : FOR C =
  : E - 2 TO E + 2 STEP 4 : HPLOT
  : B - 1, C TO B + 2, C : NEXT : FOR
  : C = E - 1 TO E + 1 : HPLOT B -
  : 2, C TO B + 3, C : NEXT : HCOLOR=
  : 3 : RETURN
5 POKE 34, 0 : INVERSE : VTAB 21 : HTAB
  : 1 : PRINT "DEBUT="F; SPC( 13 -
  : PEEK (36)):"FIN=";G; SPC( 25
  : - PEEK (36)):"SHAPE ";C%/"
  : H; SPC( 40 - PEEK (36)): NORMAL
  : POKE 34, 21 : RETURN
```

```

6 POKE 34,1: HOME : VTAB 4: PRINT
  "1. CATALOGUE DRIVE 1": PRINT
  : PRINT "2. CATALOGUE DRIVE 2
  ": PRINT : PRINT "3. "C#" TAB
  LE DE SHAPES": PRINT : PRINT
  "4. RETOUR AU MENU": PRINT : PRINT
  : PRINT "VOTRE CHOIX :";
7 GET A$:C = VAL (A$): IF C < 1 OR
  C > 4 THEN 7
8 PRINT C: IF C = 1 THEN PRINT B
  $"CATALOG,D1": GET A$: GOTO 6
9 IF C = 2 THEN PRINT B$"CATALOG
  ,D2": GET A$: GOTO 6
10 IF C = 4 THEN POP : GOTO 24
11 RETURN
12 IF I > 279 THEN I = 155
13 IF I < 155 THEN I = 279
14 IF J > 159 THEN J = 0
15 IF J < 0 THEN J = 159
16 IF E > 148 THEN E = 148: CALL
  A
17 IF B < 3 THEN B = 3: CALL A +
  3
18 IF E < 4 THEN E = 4: CALL A +
  6
19 IF B > 150 THEN B = 150: CALL
  A + 9
20 RETURN
21 K = 232:L = 233:M = 49152:N = 4
  9168
22 IF F = 0 THEN HOME : VTAB 10:
  PRINT "$300=768": PRINT "$40
  00=16384": PRINT "$6000=24576
  ": PRINT "$8000=32768": PRINT
  "$9000=36864": PRINT : PRINT
  "LOMEM=" FN A(105): PRINT "HI
  MEM=" FN A(115): INPUT "ADRES
  SE DEBUT (DECIMAL):";F: IF F <
  768 THEN F = 0: GOTO 22
23 POKE F,0:G = F: POKE K,F - INT
  (F / 256) * 256: POKE L, INT
  (F / 256):Z = 22
24 TEXT : HOME : INVERSE : PRINT
  "
  EDITEUR DE SHAPES
  ": NORMAL : VTAB 5
  : PRINT "1. CREER UNE TABLE D
  E SHAPES": PRINT "2. AJOUTER
  DES SHAPES": PRINT "3. SUPPRI
  MER DES SHAPES"
25 PRINT "4. VISIONNER VOS SHAPES
  ": PRINT "5. LIRE UNE TABLE D
  U DISQUE": PRINT "6. SAUVER L
  A TABLE SUR DISQUE": PRINT "7
  . TERMINER": POKE 34,20: HOME
  :H = PEEK (F): GOSUB 5:Z = Z
  Z
26 VTAB 23: HTAB 1: PRINT "VOTRE
  CHOIX :"; GET A$:C = VAL (A
  $): IF C < 1 OR C > 7 THEN 26
27 PRINT C: ON C GOTO 64,71,73,80
  ,86,89,91
28 VTAB 23: HTAB 2:0 = 0
29 PRINT CHR$ (7);:B = 80:E = 76
  : GOSUB 3:B% = 0:I = 216:J =
  79
30 B% = B% + 1: IF B% > A% - 10 THEN
  INVERSE : VTAB 21: HTAB 1: PRINT
  CHR$ (7);" VECTEUR #";B%; SPC(
  5);"(MAX = ";A%;")"; SPC( 40 -
  PEEK (36)): NORMAL : VTAB 23

```

```

: HTAB 2
31 GET A$: IF A$ = "P" THEN HPLLOT
  I,J: GOSUB 4:0 = 4: GOTO 31
32 IF A$ < > "I" THEN 35
33 IF 0 = 0 THEN HCOLOR= 0: GOSUB
  3
34 J = J - 1:E = E - 8: GOSUB 12: GOSUB
  3:B%(B%) = 0 + 0:0 = 0: GOTO
  30
35 IF A$ < > "J" THEN 38
36 IF 0 = 0 THEN HCOLOR= 0: GOSUB
  3
37 I = I - 1:B = B - 7: GOSUB 12: GOSUB
  3:B%(B%) = 3 + 0:0 = 0: GOTO
  30
38 IF A$ < > "K" THEN 41
39 IF 0 = 0 THEN HCOLOR= 0: GOSUB
  3
40 I = I + 1:B = B + 7: GOSUB 12: GOSUB
  3:B%(B%) = 1 + 0:0 = 0: GOTO
  30
41 IF A$ < > "M" THEN 44
42 IF 0 = 0 THEN HCOLOR= 0: GOSUB
  3
43 J = J + 1:E = E + 8: GOSUB 12: GOSUB
  3:B%(B%) = 2 + 0:0 = 0: GOTO
  30
44 IF A$ < > "E" THEN 53
45 B% = B% - 1:P = B%(B%): IF 0 =
  4 THEN HCOLOR= 0: HPLLOT I,J:
  GOSUB 4
46 0 = 0: HCOLOR= 0: GOSUB 3: ON B
  % = 0 GOTO 29: IF P > 3 THEN
  P = P - 4
47 IF P = 0 THEN E = E + 8:J = J +
  1
48 IF P = 1 THEN B = B - 7:I = I -
  1
49 IF P = 2 THEN E = E - 8:J = J -
  1
50 IF P = 3 THEN B = B + 7:I = I +
  1
51 GOSUB 12: IF B%(B%) > 3 THEN
  HCOLOR= 0: HPLLOT I,J: GOSUB 4
52 GOSUB 3
53 IF A$ < > "F" THEN 31
54 HOME : VTAB 23: PRINT "CREATIO
  N EN COURS":Q = 0:R = 0:S = 1
  : IF 0 = 4 THEN B%(B%) = 4:B%
  = B% + 1
55 C = PEEK (49200):P = B%(S):R =
  R + 1: IF R > 3 THEN R = 1
56 IF R = 1 THEN Q = P
57 IF R = 2 THEN Q = Q + 8 * P
58 IF R < > 3 GOTO 61
59 IF P < 4 THEN Q = Q + 64 * P
60 POKE T,Q:T = T + 1: IF P = 0 OR
  P > 3 THEN S = S - 1: IF Q <
  8 THEN POKE T - 1,Q + 64:Q =
  3:R = 1
61 S = S + 1: IF S < B% THEN 55
62 POKE T,Q: IF Q > 0 THEN T = T +
  1: POKE T,0
63 T = T + 1: HGR : DRAW C% AT 139
  ,79: RETURN
64 HOME : VTAB 23: INPUT "NOMBRE
  DE SHAPES (1-255) ?";H: IF H <
  1 OR H > 255 THEN 64
65 C% = 0: POKE F,H: FOR C = 1 TO
  3: POKE F + C,0: NEXT :G = F +
  2 * H + 2
66 HOME : PRINT " I": PRINT "J K

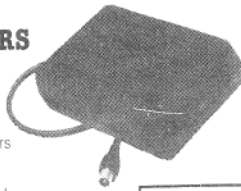
```


RENDEZ VOTRE APPLE * ENCORE "PLUS"

Cartes et accessoires additionnels compatibles APPLE II

POUR JEUX VIDEO ET MICRO-ORDINATEURS

INTERFACE
PHS 80
UNIVERSELLE
Compatible
tous micro-ordinateurs
et jeux vidéo.
Entrée PERITEL.
Sortie UHF - SECAM L
Régulateur de tension incorporé.



549 F

FLOPPY DRIVE pour APPLE 5 POUCES

2599 F



PROMOTION DISQUETTE POUR FLOPPY

5" SF-DD 48 TPI, l'unité **21 F**
par 10 pièces l'unité **19 F**, par 50 pièces l'unité **18 F**
3 POUCES MD3 «HITACHI»
- Capacité DD : 500 K octets.
disquette rigide protégée l'unité **65 F**

2950 F

«MONITOR BASE» SOCLE ORIENTABLE POUR MONITEURS NB OU COULEUR

S'oriente en toutes directions •
Angle de 12,5° en position avant et arrière (soit 25°)

- Mobile ou fixe avec blocage
- Patins antidérapants
- Supporte plus de 80 kg.



199 F

CARTE LANGAGE 16 K RAM



Pour extension du 48 K RAM en 64 K. Compatible FORTRAN PASCAL. LISP. BASIC
Entièrement équipée

549 F

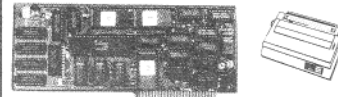
CARTE D'EXTENSION 128 K RAM



Emulation disk-drive sous DOS, PASCAL ou CP/M
Entièrement équipée

2190 F

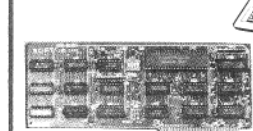
CARTE 80 COLONNES



80 car. x 24 lignes. Résolution 7 x 9. Compatible avec la plupart des traitements de texte BASIC, PASCAL, CP/M, MODEM
Entièrement équipée

749 F

CARTE Z 80



Fonctionne sous CP/M
Utilisation de tout logiciel sous CP/M.
Entièrement équipée

799 F

CARTE INTERFACE POUR 2 FLOPPY-DRIVE



Entièrement équipée

449 F

CARTE DE PROGRAMMATION 2716-2732-2764



Programmation lecture/copie chargement de programme directement sur 2716.
Entièrement équipée.

799 F

KIT GOLDEN

KIT GOLDEN

(modules montés, câblés, équipés)

Carte d'unité centrale avec 6502 et Z80 **3350 F**
Clavier ASC II **950 F**
Alimentation 220 V. 5 A **799 F**
Coffret **698 F**

5797 F

L'ENSEMBLE **5199 F**

Chaque élément peut-être acheté séparément.

CARTE D'UNITE CENTRALE double processeur 6502 et Z 80. 64 K RAM

Entièrement équipée (sans ROM)
7 slots d'extensions. Fonctionne sous CP/M

CLAVIER ASC II



68 touches. Alphanumérique. Majuscules; minuscules, décimales

ALIMENTATION 220 V. 5 A

COFFRET pour carte de base, clavier et pavé numérique.

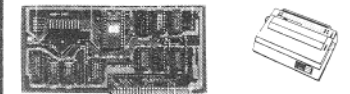
CARTE RVB

pour moniteur couleur



695 F

CARTE INTERFACE BUFFERISÉE IMPRIMANTE



Pour toutes marques sortie CENTRONIC'S - Buffer 64 K RAM.
Livrée équipée en 16 K (extension jusqu'à 64 K)

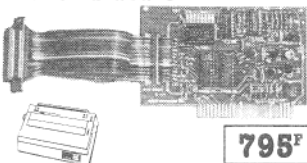
1690 F

CARTE INTERFACE POUR 4 IMPRIMANTES EN BATTERIE

Permet de brancher 4 imprimantes

799 F

CARTE DE CONNEXION série RS 232 C



795 F

CARTE «SPEECH»



Carte langage en Anglais et phonèmes

695 F

IMPRIMANTE SEIKOSHA GRAPHIQUE COMPACTE GP 100 A



SUPER PROMO **2250 F**

Interface parallèle en standard. 80 car./ligne. 50 car./sec. Impression en simple ou double largeur Papier normal. Entraînement par tracteurs ajustables.

Interface pour APPLE II ou IIE avec câble **890 F**

Papier pour GP 100 **180 F**

Les 1000 feuilles **99 F**

Ruban encreur GP 100 **99 F**

Interface pour APPLE II ou IIE avec câble **890 F**

Papier pour GP 100 **180 F**

Les 1000 feuilles **99 F**

Ruban encreur GP 100 **99 F**

Interface pour APPLE II ou IIE avec câble **890 F**

Papier pour GP 100 **180 F**

Les 1000 feuilles **99 F**

Ruban encreur GP 100 **99 F**

Interface pour APPLE II ou IIE avec câble **890 F**

Papier pour GP 100 **180 F**

Les 1000 feuilles **99 F**

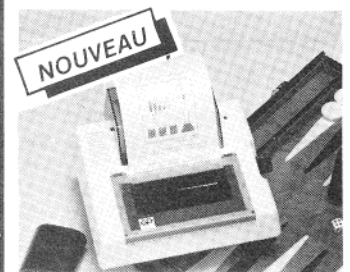
Ruban encreur GP 100 **99 F**

Interface pour APPLE II ou IIE avec câble **890 F**

Papier pour GP 100 **180 F**

Les 1000 feuilles **99 F**

Ruban encreur GP 100 **99 F**



IMPRIMANTE GP 50A SEIKOSHA

- Entraînement à friction • Graphique
- 2 épaisseurs de caractères
- Interface parallèle compatible CENTRONICS

1250 F

JOY-STICK



équipé de 2 trimes pour recherche du point zéro

169 F

MONITEURS



OCEANIC 14" couleur **3500 F**

ZENITH 12" écran vert **999 F**

Moniteur couleur RTC en module simple à monter. Avec Péritel, électronique et mécanique complet

2890 F

TABLE GRAPHIQUE **1890 F**

VENTILATEUR «FAN» pour Apple

495 F

EFFACEUR D'EPROM EN KIT

Complet avec notice

180 F

ALIMENTATION A DECOUPE

+ 5 V. 5 A. + 12 V. 1,5 A.
- 12 V. 0,5 A. - 5 V. 0,5 A

779 F

* APPLE est une marque déposée et appartient à APPLE COMPUTER S.A.

CONDITIONS GENERALES DE VENTES PAR CORRESPONDANCE
Pour éviter les frais de contre-remboursement, nous vous conseillons de régler vos commandes intégralement (y compris frais de port). FORFAIT DE PORT : 25 F.

ACER MICRO

42, rue de Chabrol, 75010 Paris.
Tél. 770.28.31.

Nombres flottants en langage machine

Erick Ringot

Introduction

Le but de cet article est double :

- approfondir les connaissances du lecteur en matière de structure des données en Applesoft et notamment sur celle des nombres flottants;
- répondre aux besoins des scientifiques en leur fournissant un programme de formatage de nombres.

Les thèmes suivants seront abordés :

- l'accumulateur flottant;
- représentation de flottants en mémoire;
- comment trouver une constante en ROM ?
- formatage de nombres scientifiques.

Il sera pratique d'utiliser le Moniteur Etendu de Thierry Le Tallec dans Pom's 8 et de prendre connaissance de l'article de Philippe François sur les programmes relogeables dans Pom's 7.

Accumulateur flottant : FAC

De même que le microprocesseur 6502 dispose d'un accumulateur 8 bits sur lequel il effectue les opérations essentielles d'addition et de soustraction, l'Applesoft comporte une zone mémoire analogue pour les nombres flottants qu'il manipule : l'accumulateur flottant baptisé FAC.

FAC est une zone de 6 octets s'étendant des adresses \$9D à \$A2, dans laquelle l'Applesoft stocke les nombres flottants (par opposition aux nombres entiers), sur lesquels il réalise des opérations évoluées telles que :

- les quatre opérations : + - x /
- des fonctions comme : SIN COS TAN EXP LOG ...
- des transformations comme : INT ABS ...

Pour comprendre sous quelle forme les flottants y sont enregistrés, il suffit de se souvenir comment - en notation décimale - nous décomposons les nombres. Tout nombre N s'écrit : $N = M \cdot 10^C$ où M désigne la MANTISSE, avec $0 <= |M| < 1$, et C désigne la CARACTERISTIQUE (ou exposant).

En binaire, on peut tout naturellement adopter les mêmes dispositions :

Exemple 1 : $N = 7.5$, soit $0.75 \cdot 10^1$

10

En décimal, $N = 0.75 \cdot 10^1$
Remarquons que $N = 15 \cdot 2^{-2} - 1$.
Nous multiplions N par 2 jusqu'à obtenir un nombre entier. Remarquons que cette opération n'est pas toujours possible, comme on peut le constater dans l'exemple 2.

En binaire, $N = 1111 \cdot 10^{-1}$
(15 en base 10 = 1111 en base 2)
(2 en base 10 = 10 en base 2)
Décalons : $N = 0.1111 \cdot 10^{11}$
(3 en base 10 = 11 en base 2)
soit en hexa : $N = 0.F \cdot 2^3$

L'accumulateur FAC est une zone de 6 octets dans laquelle les nombres sont codés de la façon suivante :

/C/M1/M/M/M/S/

C : caractéristique majorée de \$80

M1/M/M/M : mantisse sur 4 octets

S, l'octet d'extension, caractérise le signe du nombre :

- si $N > 0$ alors S est M1 avec le bit 7 à zéro

- si $N < 0$ alors S est égal à M1

notons que si $N = 0$ alors $C=0$

Pour notre exemple, 7.5 sera donc codé :

/83/F0/00/00/00/70/

7.5 a pour mantisse 0.F0 00 00 00 et pour exposant \$3.

En effet, il est positif, donc $S = \$F0$ avec le bit 7 (le plus fort) à zéro, or $\$F0 = 1111\ 0000$, donc $S = 0111\ 0000$ soit \$70.

Exemple 2 (plus complexe) : soit à coder $N = 7.4$

On multiplie par 2, soit $N = 14.8 \cdot 2^{-1}$

Comme ce n'est pas un entier, recommençons avec : $N = 29.6 \cdot 2^{-2}$

Cela ne convient toujours pas. Jusqu'où faut-il donc répéter le doublement ? Eh bien, nous savons que la mantisse (exprimée en binaire) est codée sur les quatre octets M1 M M M du FAC, soit $4 \cdot 8 = 32$ bits. La mantisse (ou du moins sa partie décimale) ne peut donc excéder $2^{32} = 4.294.967.296$. Il ne faut par conséquent pas poursuivre la multiplication par 2 au-delà de cette valeur.

Ainsi, $7.4 = 3972844749 \cdot 2^{-29}$.

Cette valeur de la mantisse est une valeur approchée car, derrière le dernier 9, il y a encore une décimale. Ceci dit, l'erreur est inférieure à 2^{-30} , soit $9 \cdot 10^{-10}$ (ce qui justifie les 10 chiffres significatifs de l'Applesoft).

Il reste à coder la mantisse en binaire, ou plus simplement en hexadécimal, grâce à des divisions succes-

sives par 16 :

$3\ 972\ 844\ 749 = 14 \cdot 16^7 + 12 \cdot 16^6 + 12 \cdot 16^5 + 12 \cdot 16^4 + 12 \cdot 16^3 + 12 \cdot 16^2 + 12 \cdot 16 + 13 = \$EC\ CC\ CC\ CD$

Il faut décaler le point décimal devant la mantisse, soit une translation de 32 bits. La nouvelle caractéristique vaut donc : $-29 + 32 = 3$.

En conclusion :

Caractéristique $C = \$80 + \$03 = \$83$

Mantisse M1 M M M = $\$EC\ CC\ CC\ CD$

Octet d'extension : on prend $M1 = \$EC = 1110\ 1100$; comme N est positif, on supprime le bit numéro 7 (le plus fort). Donc, $S = 0110\ 1100$ soit $S = \$6C$ en hexadécimal. Ainsi, 7.4 est codé $\$83\ EC\ CC\ CC\ CD\ 6C$ dans le FAC (C M1 M M M S).

On peut vérifier cet état de fait directement en mémoire; il faut pour cela charger le FAC avec la valeur de son choix, puis le consulter.

Vérification en mémoire

L'instruction Basic USR constitue une voie de passage entre l'Applesoft et le langage machine. Pour pouvoir l'utiliser, il faut préciser à l'Apple l'adresse du programme en langage machine que l'on désire appeler. Cette adresse (qui occupe 2 octets) doit être écrite dans les octets 11=\$B (octet faible de l'adresse) et 12=\$C (octet fort).

Ainsi, pour une routine implantée en 768=\$300, on tapera POKE 11,0 (adresse faible), puis POKE 12,3 (adresse forte) et l'Apple "saura" qu'à la prochaine utilisation de USR, il devra exécuter la routine à l'adresse \$300.

Si l'on écrit $Y = \text{USR}(X)$ où X et Y sont des variables numériques, X sera versé dans le FAC par l'Applesoft de façon automatique avant l'exécution de la routine en langage machine. Si cette routine prend en charge le FAC, alors Y contiendra le résultat de ses calculs.

Quant à nous, utilisons USR simplement pour voir comment l'Apple code les nombres dans le FAC. En \$300, nous mettons un octet nul \$0, qui est le code d'une interruption BRK stoppant toute procédure et donnant la main au moniteur : tapons POKE 768,0 à cette fin.

Nous profitons alors de cette interruption pour consulter directement le FAC en mémoire vive, dans les 6 octets de \$9D à \$A2.

Exemple 1 : codage de 7.5 en FAC.

Taper B=7.5:A=B:PRINT USR(A)
L'Apple s'interrompt comme prévu et l'astérisque indique le passage au mode moniteur.

Taper 9D.A2 pour consulter FAC. La réponse :

```
009D- 83 FO 00
00A0- 00 00 70
```

confirme nos prévisions.

Pour retourner à l'Applesoft, taper CTRL-C.

Exemple 2 : codage de 7.4 en FAC.

On reprend la procédure avec la valeur 7.4 (après avoir refait les POKes), puis 9D.A2, et la réponse devient effectivement :

```
009D- 83 EC CC
00A0- CC CD 6C
```

Les flottants "packés"

On peut reprocher au codage de flottant en FAC d'être redondant; l'octet d'extension S contient en effet deux informations: le signe du nombre et la valeur de M1, ce dernier est donc surabondant.

Par mesure d'économie, les flottants en mémoire ne seront donc plus codés que sur 5 octets selon la configuration suivante, dans laquelle M1 est remplacée par S :

/C/S/M/M/M/

C'est la forme compactée, "packée".

Ainsi, la valeur 7.4 est codée 83 EC CC CC CD 6C dans le FAC, mais on gagne un octet en mémoire "standard" en la codant : 83 6C CC CC CD.

Echange entre FAC et la mémoire

Maintenant que les deux types de codage nous sont connus, il serait facile de réaliser un petit programme en langage machine effectuant la transition de l'un vers l'autre et réciproquement. On imagine bien que l'Applesoft dispose déjà de telles routines :

```
MOVFM EQU $EB2B
MOVE Memory ← FAC
```

qui convertit et transfère le FAC en zone mémoire pointée par les registres Y (octet fort) et X (octet faible). En retour, l'accumulateur et le bit Z du registre d'état reflètent FACEXP (octet S de FAC).

(traduit de "What's Where in the Apple?" de F.Luebbert - Ed. Micro Inc.).

On trouve évidemment la routine réciproque :

```
MOVFM EQU $EAF9 :
MOVE FAC ← Memory
```

qui convertit et transfère vers FAC la mémoire pointée par le registre Y

(octet fort) et l'accumulateur A (octet faible).

Ces routines vont nous permettre de faire quelques exercices.

Passons en mode moniteur : CALL -151

Implantons le programme utilisé par USR en \$300 (00 03 aux adresses \$B-\$C) et en \$300 :

```
A0 03          LDY    #$03
```

l'octet fort de l'adresse du nombre est \$03

```
A2 10          LDX    #$10
```

l'octet faible de cette adresse est \$10

```
20 2B EB      JSR    $EB2B
```

transfère le FAC (chargé par USR) vers \$310

```
00           BRK
```

interruption

qui traduit FAC vers la zone de 5 octets commençant en \$310.

Repassons en Basic par CTRL-C (et RETURN) et entrons :

```
NEW
10 INPUT A : PRINT USR (A)
RUN
```

Répondons : 7.5 (RETURN) au "?"

Après le BREAK *, on peut consulter la zone \$310 par 310.314

On obtient alors la réponse :

```
83 70 00 00 00
```

On peut effectuer cette opération pour les nombres 1. et 10. dont nous aurons besoin plus tard :

1. est codé 81 00 00 00 00

10. est codé 84 20 00 00 00

0. est codé 00 00 00 00 00

PI/2 est codé 81 49 0F DA A2

(1.57079633)

Comment trouver une constante flottante en ROM ?

L'Applesoft a besoin de certaines constantes pour ses calculs : PI/2, Log(2) ... permettant de calculer toute fonction comme série entière.

Le programme MONITEUR.OBJ du Moniteur Etendu (Pom's 8) nous permet la recherche de constantes, à condition d'en connaître le codage, ce qui ne nous pose plus de problème.

Chargeons le moniteur : BRUN MONITEUR.OBJ.

La ROM s'étend de \$CFFF à \$FFFF. Pour la recherche d'une constante en ROM, il suffit de taper :

```
CFFF.FFFF/C S M M M (RETURN)
```

Le Moniteur Etendu donne l'adresse du premier octet s'il le trouve. On obtient ainsi les résultats suivants :

NOM	VALEUR	ADRESSE	CODE
ONE	1.	E913	81 00 00 00 00

```
TEN 10. E950 84 20 00 00 00
PI/2 PI/2 F066 81 49 0F DA A2
```

Formatage de nombres en notation scientifique

A titre d'exemple, nous allons écrire une routine en assembleur permettant de formater les nombres en notation scientifique.

Format choisi : il est fixe.

Les nombres formatés occuperont exactement 10 caractères selon le format :

sm.mmmErcc

s = signe

m.mmm = mantisse M telle que $1 < M < 10$ (4 chiffres significatifs dont 3 décimales)

E = signe d'exponentiation

r = signe de caractéristique

cc = caractéristique

Ainsi, -124.57 s'écrira -1.246E+02.

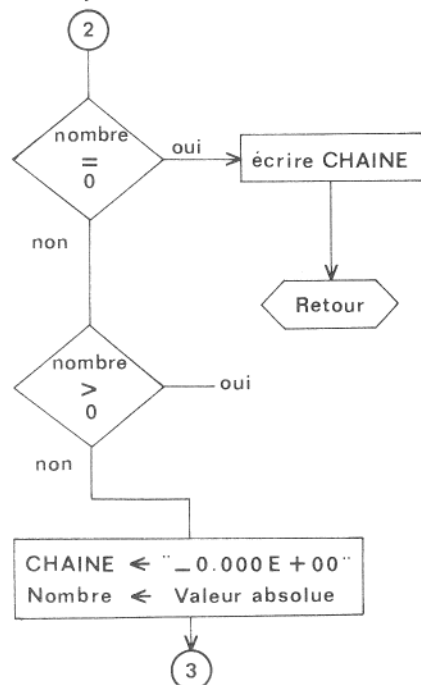
Algorithme

Le principe d'une telle routine est de transformer le nombre en chaîne de caractères.

1) Il faut d'abord initialiser la chaîne à zéro :

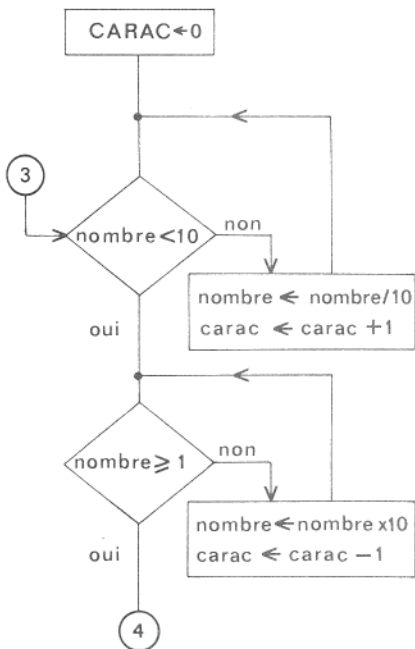
CHAINE = "+0.000E+00"

2) Il faut ensuite déterminer le signe du nombre et modifier si nécessaire le premier caractère de CHAINE en conséquence :



3) Le calcul de la mantisse et de la caractéristique se conduisent simultanément. Si le nombre est supérieur ou égal à 10, il faut le diviser par 10 et incrémenter la caractéristique. Si le

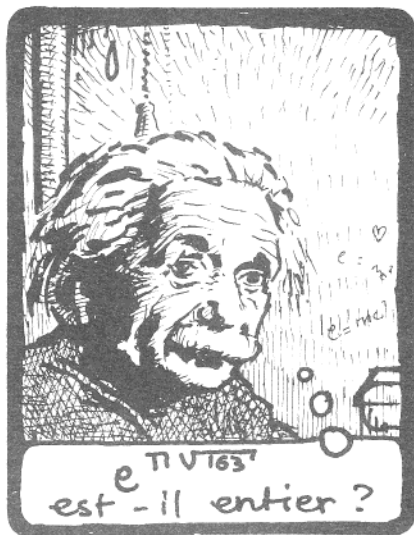
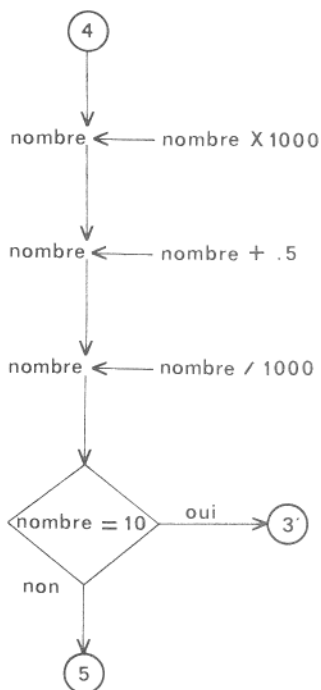
nombre est inférieur à 1, il faut le multiplier par 10 et décrémenter la caractéristique. La caractéristique est initialisée à zéro.



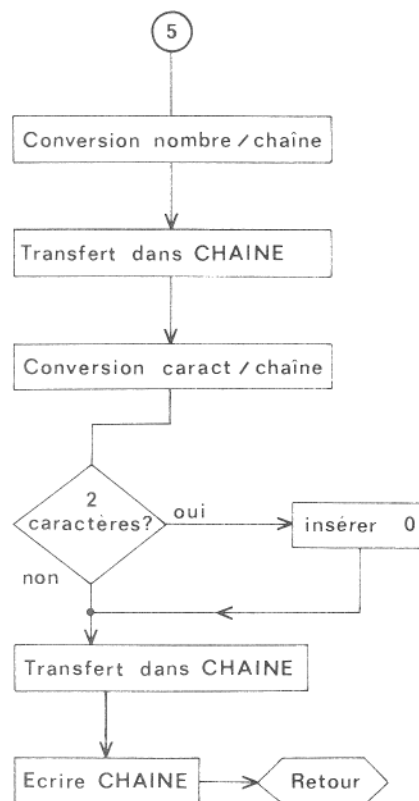
4) Ne gardant que trois décimales, il faut arrondir : pour cela, on multiplie par 1000, on ajoute .5 (afin d'arrondir et non tronquer), puis on divise par 1000.

Exemple : $n = 1.834627$
 Arrondir : $1000 * n = 1834.627$
 $1000 * n + .5 = 1835.127$
 Partie entière = 1835
 Divisée par 1000 : 1.835

Remarque : un problème peut survenir dans le cas particulier où $9.9995 < n < 9.9999$. En effet, l'arrondi donne alors 10. Il faut donc reformater :



5) On transforme NOMBRE en une chaîne de caractères, on la verse dans CHAINE ; on transforme également CARACT en une chaîne en prenant garde qu'elle comporte 3 caractères, y compris le signe, et on transfère dans CHAINE. Il ne reste plus qu'à écrire CHAINE.



Sous-programme Applesoft

Variables :
 CH\$: chaîne recevant la chaîne
 CA% : caractéristique
 NB : nombre à formater, puis mantisse
 (voir le listing de SCIENT.APP)

Routine en assembleur
 (DOS Tool Kit)

```

1 *****
2 * SCIENT.LIB *
3 * RELOGEABLE *
4 * * *
5 * FORMATAGE DE NOMBRES *
6 * EN NOTATION SCIENTIF. *
7 * * *
8 *****
9 ;
10 ;E.RINGOT, LE 16/01/83
11 ;
12 ; ORG $1000 ;POUR ASSEMBLAGE SEULEMENT
13 ;
14 ;-----
15 ;<1>-VARIABLES
16 ;-----
17 ;
18 CHAINE EQU $250 ;BUFFER
19 SIGMAN EQU CHAINE ;SIGNE DE MANTISSE
20 MANTIS EQU CHAINE+1 ;MANTISSE
21 POINT EQU CHAINE+2
22 SYMBEX EQU CHAINE+6 ;SYMBOLE 'E'
23 SIGEXP EQU CHAINE+7 ;SIGNE D'EXPOSANT
24 EXPOSA EQU CHAINE+8 ;EXPOSANT
25 FINCHA EQU CHAINE+10
26 EXPO EQU $260 ;CARACTERISTIQUE
27 FRESPC EQU $71 ;POINTEUR DE CHAINE
28 ONE EQU $E913 ;CONSTANTE FLOTTANTE 1.
29 TEN EQU $EA50 ;CONTIENT LE FLOTTANT 10.
30 ;
31 ;-----
32 ;<2>-ROUTINES ROM
33 ;-----
34 ;
35 CHRGET EQU $B7 ;N'AVANCE PAS LE POINTEUR
36 FRMNUM EQU $B1 ;CARACTERE DANS LE BASIC
37 FRMNUM EQU $DD67 ;FAC <= EXPRESSION
38 MUL10 EQU $EA39 ;FAC <= FAC * 10
39 DIV10 EQU $EA55 ;FAC <= FAC / 10
40 SIGN EQU $EB82 ;A <= SIGNE (FAC)
41 ABS EQU $EBAF ;FAC <= ABS (FAC)

```

Syntaxe

Nous vous proposons d'appeler une routine formatant et éditant les nombres. La syntaxe de la routine Applesoft sera :

CALL adr, nombre [:]

adr = adresse décimale de la routine qui sera relogeable

, = séparateur

nombre = argument à passer : constante, expression, variable numérique
[:] = spécification optionnelle empêchant le retour chariot après édition

On ne peut écrire qu'un seul nombre avec un tel CALL.

Commentaires sur le programme

Lignes 78-79 : CHRGET saisit la virgule dans le texte Applesoft et incrémente le pointeur TXTPTR. FRMNUM transforme l'expression numérique suivant la virgule et la transfère dans FAC.

Lignes 98-100 : STROUT écrit une chaîne de caractères pointée par les registres Y(MSB) et A(LSB); la chaîne doit se terminer par un octet nul - ne pas confondre avec un octet contenant "0" soit \$30.

Lignes 102-108 : CHRGET prend le caractère dans le texte Applesoft, mais sans incrémenter le pointeur TXTPTR. Si le caractère n'est pas ";", on envoie un retour chariot. Dans le cas contraire, on le passe grâce à CHRGET et on revient à l'Applesoft.

Lignes 116-134 : calcul de la mantisse et de la caractéristique EXPO. FCOMP compare FAC au nombre pointé par Y et A à la sortie.

A contient 0 si FAC = MEM

A contient 1 si FAC > MEM

A contient FF si FAC < MEM

Lignes 138-150 : procédure d'arrondi - la routine FADDH ajoute 0.5 à FAC.

Lignes 154-163 : transforme le FAC en chaîne grâce à FOUT. La chaîne est pointée par Y et A à la sortie (normalement, c'est \$100). On récupère cette adresse en FRESPEC (page 0), ce qui nous permet de la transférer vers MANTIS. On contrôle deux choses :

- arrêt du transfert si la chaîne se termine par 0;

- arrêt du transfert si 5 caractères au maximum sont transférés (ce qui nous met à l'abri des erreurs de calcul).

Lignes 167-174, 175-182, 183-189 : dans chacun de ces trois blocs, on écrit le signe de la caractéristique, on transforme la caractéristique en chaîne et on la transfère vers EX-POSA; si l'exposant ne comporte qu'un chiffre, on insère un "0" par devant.

```
42 FLOAT EQU $EB93 ;FAC <= A
43 INT EQU $EC23
44 FADDH EQU $E7A0 ;FAC <= FAC + 1/2
45 FCOMP EQU $EBB2 ;COMPARE FAC AVEC (Y,A)
46 FOUT EQU $ED34 ;CHAINE CARACTERES <= FAC
47 STROUT EQU $DB3A ;ECRIT UNE CHAINE
48 CROUT EQU $FD8E ;RETOUR CHARIOT
49 RETURN EQU $FF58 ;VOIR POM'S 7 PAGES 27-28
50 ;
51 ;-----
52 ;<3>-PROGRAMME DE FORMATAGE
53 ;-----
54 ;
55 ;INITIALISATION EXPOSANT
56 ;
57 LDA #0
58 STA EXPO
59 STA FINCHA
60 ;
61 ;INITIALISATION DE LA CHAINE A ZERO
62 ;
63 LDX #9
64 LDA #'0'
65 ZERO STA CHAINE,X
66 DEX
67 BNE ZERO
68 LDA #'+'
69 STA SIGMAN
70 STA SIGEXP
71 LDA #'E'
72 STA SYMBEX
73 LDA #'.'
74 STA POINT
75 ;
76 ;SAISIE NOMBRE A FORMATER
77 ;
78 JSR CHRGET ;PASSE LA VIRGULE
79 JSR FRMNUM ;L'EXPRESSION PASSE EN FAC
80 ;
81 ;SIGNE ET VALEUR ABSOLUE
82 ;
83 JSR SIGN ;A <= SIGNE(FAC)
84 BEQ ECRIT ;SI FAC = 0, TERMINE
85 ;
86 CMP #1 ;FAC POSITIF ?
87 BEQ ABSO ;OUI !
88 LDA #'-' ;NON !,ON CHANGE LE SIGNE
89 STA SIGMAN ;CONTENU EN SIGMAN
90 ;
91 JSR ABS ;FAC <= ABS (FAC)
92 ABSO CLV ;PROGRAMME RELOGEABLE
93 JSR RETURN ;VOIR POM' 7
94 BVC FORMAT ;PSEUDO JSR FORMAT
95 ;
96 ;ECRITURE DU NOMBRE FORMATE
97 ;
98 ECRIT LDY #<CHAINE ;(Y,A) POINTENT
99 LDA #>CHAINE ;VERS CHAINE
100 JSR STROUT ;ECRIT CHAINE
101 ;
102 JSR CHRGET ;PREND LE CARACTERE SUIVANT
103 CMP #3B ;EST-CE LE CARACTERE " ; " ?
104 BEQ RETOUR ;OUI!
105 JSR CROUT ;NON! RETOUR CHARIOT
106 RTS
107 RETOUR JSR CHRGET ;SAUTE CE";"
108 RTS
109 ;
110 ;FORMATAGE DE LA VALEUR ABSOLUE
111 ;
112 FORMAT TSX ;CF. POM'S 7
113 DEX
114 DEX
115 TXS
116 TENFAC LDY #<TEN ;COMPARE FAC A 10.
```


Etude comparative des routines

Place mémoire

Routine Applesoft compactée (sans REM) numérotée à partir de 100 de 10 en 10.

- programme = 350 octets
 - descripteurs 3 variables = 21 octets
 - chaîne = 10 octets
 - TOTAL
 = 381 octets

Routine assembleur \$FA = 244 octets, soit un gain de 36 % en place mémoire.

Mesures de rapidité

Exemple 1 :

FOR I = 1 TO 1000 : NB = I : NEXT

Temps d'exécution : 3 secondes.

Exemple 2 :

FOR I = 1 TO 1000 : NB = I : GOSUB 100 : NEXT

Temps d'exécution : 144 secondes.

Exemple 3 :

FOR I = 1 TO 1000 : CALL 4096,I : NEXT

Temps d'exécution : 60 secondes.

Temps unitaires :

routine Applesoft : 144-3:1000 = 141 ms

routine assembleur : 60-3:1000 = 57 ms, soit un gain de temps de 60%.

Conclusion

La pauvreté de l'Applesoft amène régulièrement le programmeur à écrire de petites routines palliant ses déficiences naturelles. Deux voies s'offrent à lui : le Basic et le langage machine. Le Basic est simple à écrire mais lent, gourmand en place mémoire et incapable de gérer des variables locales. Le langage machine est beaucoup plus souple d'utilisation mais demande un temps de programmation plus important et un éditeur-assembleur. Néanmoins, le présent article montre que l'utilisation intensive des routines présentes en ROM facilite grandement la tâche du programmeur; une bonne documentation s'impose alors. Enfin, la constitution d'une bibliothèque d'utilitaires, que l'on charge selon les besoins, est bien plus simple à réaliser en langage machine (pourvu que les routines soient relogeables) grâce à BLOAD, qu'en Basic où il faut jongler avec les HOLD, MERGE et compagnie. Un premier élément de cette bibliothèque (LIBRARY en anglais, d'où le nom du code objet) est proposé aujourd'hui. Chacun peut y apporter sa pierre ...

```

117 LDA #>TEN
118 JSR FCOMP
119 CMP #\$FF ;A < 10. ?
120 BEQ ONEFAC ;OUI!
121 JSR DIV10 ;FAC <- FAC / 10.
122 INC EXPO ;NON! INCREMENTE L'EXPOSANT
123 CLC
124 BCC TENFAC ;SAUT INCONDITIONNEL
125 ;
126 ONEFAC LDY #<ONE ;COMPARE FAC A 1.
127 LDA #>ONE
128 JSR FCOMP
129 CMP #\$FF ;A < 1. ?
130 BNE ARROND ;NON!
131 JSR MUL10 ;FAC <- FAC * 10.
132 DEC EXPO ;OUI! DECREMENTE L'EXPOSANT
133 CLC
134 BCC ONEFAC ;SAUT INCONDITIONNEL
135 ;
136 ;ARRONDI A TROIS DECIMALES
137 ;
138 ARROND JSR MUL10 ;3 MULTIPLICATIONS DE
139 JSR MUL10 ;FAC PAR 10.
140 JSR MUL10
141 JSR FADDH ;AJOUTE 1/2 A FAC
142 JSR INT ;PREND LA PARTIE ENTIERE
143 JSR DIV10 ;3 DIVISIONS DE
144 JSR DIV10 ;FAC PAR 10.
145 JSR DIV10
146 ;
147 LDY #<TEN ;POUR LE CAS
148 LDA #>TEN ;OU L'ON PASSERAIT
149 JSR FCOMP ;DE 9.9995 A 10
150 BEQ TENFAC ;CORRECTION
151 ;
152 ;TRANSFERT MANTISSE DANS CHAINE
153 ;
154 JSR FOUT ;TRADUIT LA MANTISSE EN CHAINE
155 STY FRESPC+1 ;ADRESSE TRANSFEREE
156 STA FRESPC ;EN FRESPC LIBRE
157 LDY #0
158 MOVMAN LDA (FRESPC),Y ;TRANSFERT VERS CHAINE
159 BEQ CALEXP ;CARACTERE NUL = FIN DE CHAINE
160 STA MANTIS,Y
161 INY
162 CPY #5 ;5 CARACTERES MAXI
163 BNE MOVMAN
164 ;
165 ;TRANSFERT DE L'EXPOSANT
166 ;
167 CALEXP LDA EXPO
168 JSR FLOAT ;FAC <= EXPO
169 JSR SIGN ;A <= SIGNE DE L'EXPOSANT
170 CMP #\$FF ;NEGATIF ?
171 BNE ABSEXP ;NON!
172 LDA #'-' ;OUI!, CHANGER LE SIGNE
173 STA SIGEXP
174 JSR ABS
175 ABSEXP JSR FOUT ;EXPOSANT EN CARACTERES
176 STY FRESPC+1
177 STA FRESPC ;POINTEUR
178 LDY #0
179 MOVEXP LDA (FRESPC),Y
180 STA EXPOSA,Y
181 INY
182 CPY #2
183 BNE MOVEXP
184 CMP #0 ;CARACTERE NUL ?
185 BNE FINEXP ;NON!
186 LDA EXPOSA
187 STA EXPOSA+1
188 LDA #'0'
189 STA EXPOSA ;SINON, DECALAGE
190 FINEXP BIT RETURN
191 RTS
    
```

Programme SCIENT.OBJ

```

1000- A9 00 8D 60 02 8D 5A 02
1008- A2 09 A9 B0 9D 50 02 CA
1010- D0 FA A9 AB 8D 50 02 8D
1018- 57 02 A9 C5 8D 56 02 A9
1020- AE 8D 52 02 20 B1 00 20
1028- 67 DD 20 82 EB F0 12 C9
1030- 01 F0 08 A9 AD 8D 50 02
1038- 20 AF EB B8 20 58 FF 50
1040- 16 A0 02 A9 50 20 3A DB
1048- 20 B7 00 C9 3B F0 04 20
1050- 8E FD 60 20 B1 00 60 BA
1058- CA CA 9A A0 EA A9 50 20
1060- B2 EB C9 FF F0 09 20 55
1068- EA EE 60 02 18 90 EC A0

```

```

1070- E9 A9 13 20 B2 EB C9 FF
1078- D0 09 20 39 EA CE 60 02
1080- 18 90 EC 20 39 EA 20 39
1088- EA 20 39 EA 20 A0 E7 20
1090- 23 EC 20 55 EA 20 55 EA
1098- 20 55 EA A0 EA A9 50 20
10A0- B2 EB F0 B7 20 34 ED 84
10A8- 72 85 71 A0 00 B1 71 F0
10B0- 08 99 51 02 C8 C0 05 D0
10B8- F4 AD 60 02 20 93 EB 20
10C0- 82 EB C9 FF D0 08 A9 AD
10C8- 8D 57 02 20 AF EB 20 34
10D0- ED 84 72 85 71 A0 00 B1
10D8- 71 99 58 02 C8 C0 02 D0
10E0- F6 C9 00 D0 0B AD 58 02
10E8- 8D 59 02 A9 B0 8D 58 02
10F0- 2C 58 FF 60

```

```

90 REM *****
92 REM * FORMATAGE DE NOMBRES *
94 REM * SCIENTIFIQUES *
96 REM * VERSION EN BASIC *
98 REM *****
100 CH$ = "+0.000E+00":CA% = 0
110 IF NB = 0 THEN 210
120 CH$ = "+": IF NB < 0 THEN CH$ = "-":
    NB = ABS (NB)
130 IF NB > = 10 THEN NB = NB / 10:CA%
    = CA% + 1: GOTO 130
140 IF NB < 1 THEN NB = NB * 10:CA% = C
    A% - 1: GOTO 140
150 NB = INT (NB * 1000 + .5)
160 IF NB = 10000 THEN NB = 1000:CA% =
    CA% + 1
170 CH$ = CH$ + LEFT$ ( STR$ (NB),1) +
    "." + RIGHT$ ( STR$ (NB),3) + "E
    +"
180 IF CA% < 0 THEN CA% = ABS (CA%):CH
    $ = LEFT$ (CH$,7) + "-"
190 IF LEN ( STR$ (CA%)) = 1 THEN CH$
    = CH$ + "0"
200 CH$ = CH$ + STR$ (CA%):NB = FRE (0
    )
210 PRINT CH$: RETURN

```

```

10 REM *** PROGRAMME SCIENT.DEM ***
20 PRINT CHR$(4)"BLOADSCIENT.LIB"
30 GOTO 1000
40 VTAB 20: PRINT "TAPER UNE TOUCHE ET
    DECLANCHER LE CHRONO": WAIT - 16
    384,128: POKE - 16368,0: RETURN
90 REM
    *** FORMATAGE ***
100 CH$ = "+0.000E+00":CA% = 0
110 IF NB = 0 THEN 210
120 CH$ = "+": IF NB < 0 THEN CH$ = "-":
    NB = ABS (NB)
130 IF NB > = 10 THEN NB = NB / 10:CA%
    = CA% + 1: GOTO 130
140 IF NB < 1 THEN NB = NB * 10:CA% = C
    A% - 1: GOTO 140
150 NB = INT (NB * 1000 + .5)
160 IF NB = 10000 THEN NB = 1000:CA% =

```

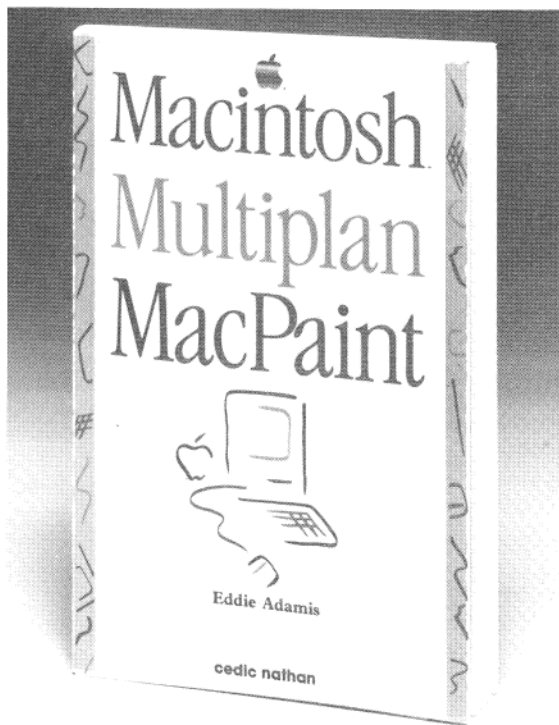
```

    CA% + 1
170 CH$ = CH$ + LEFT$ ( STR$ (NB),1) +
    "." + RIGHT$ ( STR$ (NB),3) + "E
    +"
180 IF CA% < 0 THEN CA% = ABS (CA%):CH
    $ = LEFT$ (CH$,7) + "-"
190 IF LEN ( STR$ (CA%)) = 1 THEN CH$
    = CH$ + "0"
200 CH$ = CH$ + STR$ (CA%):NB = FRE (0
    )
210 PRINT CH$: RETURN
990 REM
    *** PROG.PPAL ***
1000 TEXT : HOME
1005 PRINT "CHRONOMETRAGE COMPARATIF
    DES VERSIONS
    BASIC ET LANGAGE-MACHINE
    DE LA ROUTINE
    DE FORMATAGE
    "
1010 PRINT "<1>- BOUCLES A VIDE
    <2>- BOUCLES BASIC
    <3>- BOUCLES LANGAGE MACHINE
    <4>- FIN"
1020 GET A$:A = VAL (A$): IF A < 1 OR
    A > 4 THEN 1020
1025 IF A = 4 THEN END
1030 GOSUB 40: ON A GOSUB 1100,1200,130
    0
1040 GOTO 1000
1090 REM
    - BOUCLES VIDES -
1100 FOR I = 1 TO 1000: NEXT I: PRINT
    CHR$(7): RETURN
1190 REM
    - BOUCLES BASIC -
1200 FOR I = 1 TO 1000:NB = I: GOSUB 10
    0: NEXT I: PRINT CHR$(7): RETUR
    N
1290 REM
    - BOUCLES MACHINE -
1300 SCIENT = 4096: FOR I = 1 TO 1000: C
    ALL SCIENT,I: NEXT I: PRINT CHR$(
    7): RETURN

```

votre nouvelle bibliothèque Apple

Sélectionné par Apple
et Microsoft



pour découvrir Mac Paint et Multiplan sur le fabuleux Macintosh

Un livre qui vous permettra d'entrer de plain-pied dans ces deux logiciels en découvrant les infinies possibilités du Macintosh.

1C0556 89,00 F

Ecrivons un programme pour Apple

Pour vous amuser en apprenant la programmation sur un micro-ordinateur qui semble doté de pouvoirs magiques. Vous apprendrez à lui donner des ordres en BASIC et toutes sortes d'instructions. Vous pourrez vous en servir pour dessiner. Vous le comprendrez et vous saurez le faire fonctionner en ouvrant ce livre.

1C0544 99,00 F

Vous et l'ordinateur Apple

Un livre pour tous ceux qui veulent connaître l'APPLE et s'en faire un ami.

33 leçons pour apprendre à programmer en BASIC, chaque leçon est illustrée de commentaires, d'explications, de programmes et d'exercices complémentaires.

1C0542 129,00 F

Le Logo sur Apple

Un livre pour faire l'apprentissage de LOGO, un moyen d'expression très puissant. D'une façon très vivante, ce livre explique les étapes à suivre pour concevoir et réaliser toutes sortes de projets... à partir de la tortue LOGO, ce célèbre petit animal cybernétique qui se déplace sur l'écran en laissant une trace derrière lui.

1C0543 129,00 F



Envoi de notre catalogue sur simple demande
cedic/nathan
32, bd Saint Germain 75005 PARIS - Tél. (1) 326.42.71

Garantissez votre succès Professionnel

Oui, vous serez assurés d'enrichir votre savoir-faire « micro-informatique » indispensable aujourd'hui à la vie professionnelle en suivant les conférences de MICRO-EXPO.

Du 22 au 26 mai, dans le cadre de ce 9^e Congrès Exposition, à Paris au Palais des Congrès, Porte Maillot, pour 100 F seulement vous aurez accès à 40 heures de conférences sur :

Mardi 22 mai : Choisir son micro ou ce qu'il faut savoir avant de l'acheter; Pionniers US.

Mercredi 23 mai : Les bases de données et leur utilisation (compatibilité des PME et PMI sur micro avec dBASE 2); Un micro-ordinateur, à quoi ça sert? Visi On; Initiation au langage Basic; Pionniers France.

Jeudi 24 mai : Informatique et Télématique à la ferme organisée par l'ISAB; La micro dans l'officine; La micro et les médecins; Gestion financière avec un ta-

bleur électronique; Un micro-ordinateur, ça s'entretient; La programmation structurée (Pascal et Modula II); MS WIN; Initiation au langage ADA.

Vendredi 25 mai : Les enjeux du contrat informatique; Micro et ordinateur personnel : les postes de travail de demain; Décisionnel graphique, représentation, communication d'analyse et de résultats; Comprendre la téléma-

tiq ue.

Samedi 26 mai : choisir son micro ou ce qu'il faut savoir avant de l'acheter; Introduction aux microprocesseurs.

Participez à ces conférences, vous garantirez votre succès professionnel et deviendrez l'interlocuteur recherché à plus d'un titre.

**MICRO
EXPO**

COUPON-RÉPONSE A RETOURNER A SYBEX : 6-8, impasse du Curé - 75018 PARIS, ou à déposer dans un magasin portant l'affiche MICRO-EXPO. Celui-ci est votre titre de participation au tirage au sort du voyage à gagner pour Silicon Valley qui aura lieu en présence de Maître P. CHALE, Huissier à Paris. Merci de m'adresser :

- une entrée gratuite et le programme détaillé des conférences.
 un passeport valeur 100 F me donnant droit à l'entrée permanente, au guide de la micro, et de suivre toutes les conférences.

NOM
 Prénom
 Société
 N° Rue
 Code postal Ville
 Activité de l'entreprise
 Fonction
 Ci-joint chèque de 100 F.

po

Un mois avec le Macintosh

Hervé Thiriez

Cela fait maintenant environ un mois que j'utilise régulièrement le Macintosh, avec en fait trois programmes seulement : MacPaint, MacWrite et Multiplan, sur un appareil en version U.S. J'ai eu l'occasion de le montrer à des amis, à des étudiants, à des cadres et à des dirigeants d'entreprise. Le plus frappant est que, quel que soit mon interlocuteur, chacun aurait bien voulu en avoir un. On comprend pourquoi les américains, malgré l'usine super-robotisée de Fremont, n'arrivent pas à satisfaire la demande. Il est question que la chaîne publicitaire exploite le thème "On savait qu'on en vendrait un million, mais pas en deux semaines ...". Ceci dit, nous faisons un peu les frais de cette réussite, puisqu'il faudra attendre le mois de juin pour pouvoir en acheter un en France.

La souris

Tout d'abord quelque peu sceptique quant à l'intérêt pratique de la souris, je dois avouer qu'elle m'a convaincu. On s'habitue très vite au confort qu'elle apporte, et cela devient d'autant plus frustrant d'obtenir ensuite sur son brave Apple //e le message "SYNTAX ERROR" parce qu'on a tapé trop vite "CQTQLOG"! La souris est agréable car elle permet une sélection extrêmement rapide d'options dans un menu, mais aussi parce que les logiciels ont été conçus pour en tirer parti. Ainsi, toutes les instructions du "DOS", tout ce qui revient à demander un catalogue, lire un fichier, le sauvegarder, exécuter un programme, ..., tout cela s'obtient exactement de la même façon avec tous les logiciels tournant sur Macintosh. On se demande si cela a encore un sens de réaliser pour ces logiciels des documentations de plus de 10 pages. C'est vraiment très agréable de voir tous les logiciels utiliser le même vocabulaire, la même syntaxe (si on peut encore parler de syntaxe, tellement c'est simple).

En quoi le Macintosh change-t-il la façon de travailler sur des logiciels spécifiques tels que le traitement de texte ou un tableur ?

MacWrite

Avec MacWrite, on peut réaliser des textes impossibles à obtenir avec un

autre micro-ordinateur (à part Lisa). Parvenir à sortir un texte en justification totale avec, dans une même ligne, des polices et des corps (tailles) de caractères différents, pouvoir en outre imprimer tel quel ce qu'on voit à l'écran, cela tient du miracle. Bien entendu, contrairement à ce que l'on voit même sur certains programmes de traitement de texte implantés sur de gros ordinateurs, on ne perd pas la justification avec les trémas, ou les accents circonflexes.

Autre particularité originale de MacWrite : la possibilité de récupérer n'importe quel texte, dessin ou tableau provenant d'un autre logiciel pour l'insérer dans le texte. On peut même déformer à loisir chacune de ces insertions, en les tirant vers le bas, vers la droite, ... comme si elles étaient en caoutchouc.

Seul inconvénient, relevé d'ailleurs par Daniel Garric dans son article (Pom's 11), la taille relativement restreinte (8 pages) du fichier maximal actuellement entrable sous MacWrite. Mais cela devrait s'améliorer avec l'extension de la mémoire centrale au-delà des 128K de RAM actuels (en plus des 64K de ROM).

Multiplan

Par rapport à Multiplan sur Apple //e, on remarquera les originalités : possibilité d'imprimer un tableau simultanément au travail, impression de travers pour les tableaux très larges (aucun problème, puisque l'ImageWriter est une imprimante graphique). Sur le plan pratique, on constate un grand gain de temps pour les déplacements dans le tableau (à l'aide des curseurs horizontal et vertical). Il devient enfantin de modifier la largeur d'une colonne et de la mettre juste à la bonne dimension, ce qui demande normalement des tâtonnements pénibles sur tout autre matériel. Il n'est plus nécessaire de feuilleter la documentation pour retrouver le nom exact ou la syntaxe d'une fonction donnée : on fait rapidement défiler les noms de fonctions. On peut définir une largeur de colonne nulle, ce qui évite enfin ces dangereux "Mouvements" auquel on était obligé d'avoir recours pour bien présenter certains tableaux. Et, finalement, la récupération de tout ou par-

tie d'un tableau pour insertion dans un texte s'effectue sans douleur.

Il existe aussi un mode "entrée de données", dans lequel la grille lignes/colonnes n'est plus visible, et où les données à entrer sont représentées par des pointillés. Cela réduit le risque d'erreur. A ce mode est associé une possibilité de protection par un mot de passe : les cellules protégées ne peuvent alors être modifiées ni même consultées.

MacPaint

Faut-il encore décrire MacPaint qui est probablement le plus montré de tous les logiciels car il est "bluffant". Je le trouve excellent, mais j'estime cependant les performances obtenues avec MacWrite (et, à moindre titre, avec Multiplan) tout aussi surprenantes.

Les outils de développement

Ce n'est pas le moment d'oublier l'excellent Basic Microsoft du Macintosh, d'autant plus puissant qu'il a accès à toutes les instructions propres au Macintosh et relatives à la gestion de la souris, des écrans et des menus. Le Macintosh Pascal, interprété, avec la possibilité de faire du pas à pas et de voir le programme sur une partie de l'écran et l'exécution se réaliser à côté du listage. Et aussi l'assembleur 68000, Logo et Prolog. Il y a de quoi faire !

Les autres logiciels

Chart (graphiques) est déjà sorti, pour être utilisé en relation avec Multiplan ou Microsoft File. Parmi les célébrités, on attend PFS, Lotus 1-2-3, Microsoft Word, CX Base 200, Omnis... Sans oublier bien entendu les autres produits "naturels" MacTerm (communication), MacDraw et MacProject. Steven Jobs prévoit que 500 logiciels sur Macintosh seront disponibles avant la fin de l'année. Pour sa part, Bill Gates, le patron et fondateur de Microsoft (un gamin, lui aussi), s'attend à faire en 1985 la moitié de son chiffre d'affaires, pour tant impressionnant, avec des produits sur Macintosh.

La découverte de l'Apple //c

Guy Mathieu

A l'invitation d'Apple, j'ai eu le plaisir de visiter avec un groupe de journalistes les installations Apple en Irlande et de découvrir le nouveau-né de la famille, le //c. Tout cela en tant que représentant de Pom's et dans d'excellentes conditions d'organisation. Bravo, Apple !

Le //c est petit (6,35 * 30,48 *, 29,21 en cms.), pas bien lourd (3,4 kgs) et vous sera vendu dans une boîte en carton jetable. Ne croyez quand même pas qu'il se suffit à lui-même. L'alimentation (12 volts à l'entrée de la machine) est un boîtier séparé inclus dans le « paquet ». Notez toutefois que rien ne vous empêche de vous brancher sur l'allume-cigare de votre voiture. Une batterie faite sur mesure est enfin en préparation chez Apple.

Il vous faudra bien sûr un écran. Rien à voir avec celui du grand frère Macintosh. Ses parents ont sorti en son honneur un moniteur compact 9 pouces de très bonne définition. Vous pouvez aussi utiliser directement votre téléviseur pour peu qu'il accepte une prise Péritel. Un téléviseur Thomson, nouveau, compact lui aussi et fonctionnant sous 12 volts, présente de même bien des avantages. En préparation aussi, un écran plat de 24 lignes et 80 colonnes (fin 84 ?) ainsi qu'un moniteur couleur d'excellente définition.

A l'intérieur du //c (attention, vous perdez le bénéfice de la garantie si vous l'ouvrez — tant pis pour les bricoleurs), 128K de RAM (le Basic Applesoft, complété par des instructions « souris », se trouve dans les 16K de ROM). En standard, 80 colonnes, ou 40 obtenables par une astuce (le contraire de l'Apple //c). Un lecteur de 143K hélas intégré (bon point pour la portabilité, mauvais point pour la maintenance). Un mode haute résolution simple de 80*192 en six couleurs. Et un mode super-résolution de 560*192 en 16 couleurs. Le microprocesseur : un 65C02 fonctionnant à 1 MHz, et possédant 27 instructions internes de plus que le 6502.

Au-dessus du //c, un clavier bien dessiné, mais sans bloc numérique, en Azerty pour la version française, mais pouvant simuler le Qwerty. Tiens, le « Shift Lock » verrouille aussi les chiffres ! On peut aussi régler le volume du haut-parleur, et même brancher un casque ou un haut-parleur externe.

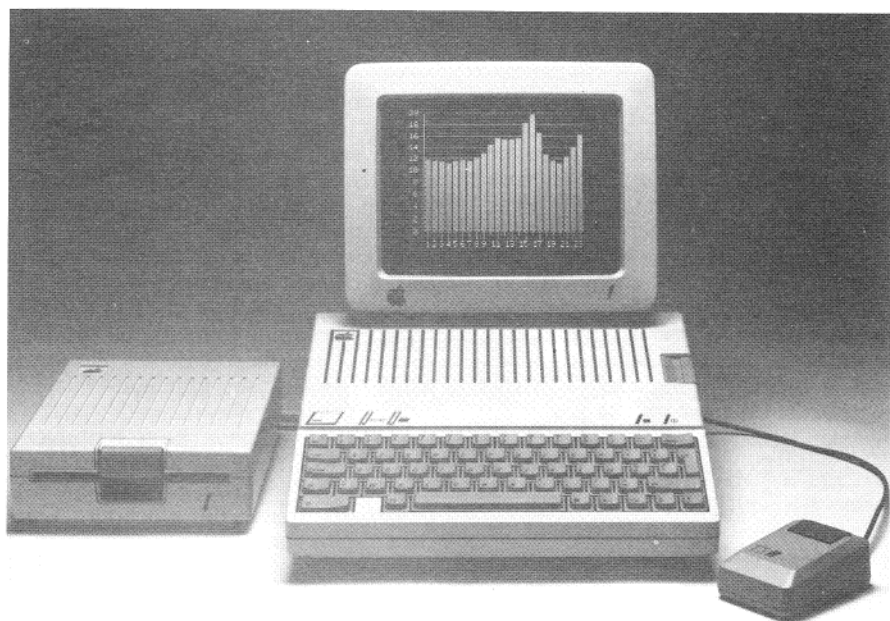
Derrière le //c, un lot de connecteurs : RS232 (imprimante 9600 bauds), connecteur pour second lecteur de disquettes (et pas plus...), prise vidéo standard et prise RVB Chat Mauve, RS232 (modem 300 bauds), prise à 9 broches pour manche à balai (joystick), manettes (paddles) ou souris. Pas de disque dur envisagé, mais peut-être un jour des disquettes 3,5 pouces comme pour le Macintosh. Remarque : les lecteurs de disquettes du //e ne sont pas compatibles, n'en déplaise aux « cannibales ».

A côté de tout cela, plusieurs disquettes d'initiation, y compris une initiation au langage Logo. Le système d'exploitation normal est le ProDOS, mais on peut travailler en DOS 3.3 et en Pascal.

Et le prix ? Pour l'unité centrale et l'alimentation, le tout dans la boîte en carton, 12 900 F TTC. Environ 2 000 FF pour l'écran Apple ou 3 900 FF pour le téléviseur Thomson. L'écran plat reviendra à 5 à 6 000 FF. Un peu plus de 3 000 FF pour le second lecteur. L'imprimante normale est l'ImageWriter.

Du côté des logiciels, compatibilité à peu près totale avec les logiciels de l'Apple //e. Le logiciel sélectionne l'option 80 ou 40 colonnes. Si cette sélection n'est pas prévue dans le logiciel, un inverseur manuel permet d'y remédier. Il semble que quelques problèmes de compatibilité subsistent encore au niveau d'une interférence avec l'option « souris », laquelle revient à environ 900 FF. Un nouveau venu, disponible en anglais, programmé pour cet été en français, AppleWorks : traitement de texte, tableur et gestionnaire de fichiers. Deux fenêtres, une dans laquelle on travaille, l'autre indiquant avec quoi on peut travailler et ce qu'on peut faire. Prix annoncé : de 2 000 à 2 500 FF.

Autre nouvelle : Applebus accueillera bientôt des modèles variés de microordinateurs en mettant en commun une imprimante laser (50 000 FF) et un serveur disque de 20 à 70 mégaoctets. Vitesse de transfert : 230 kilobauds. La connexion à 500 FF par appareil. Présentation au SICOB pour une disponibilité en fin 84.



PEEKs et POKEs en Pascal

Jacques Pino

Voici une petite application inspirée par le programme LINEFEED de la disquette APPLE3. Pour les irréductibles du Basic, il manquait en Pascal les instructions PEEK et POKE. Ce n'est plus qu'un mauvais souvenir aujourd'hui avec PEEPO, un programme introduisant la fonction PEEK et la procédure POKE.

Le programme

MOT représente une unité de mémoire Pascal, c'est-à-dire 16 bits sur deux octets.

BIFACE est un enregistrement dont les deux champs ADRESSE et CONTENU sont confondus.

ADRESSE est un entier sur 16 bits.

CONTENU est un pointeur, c'est-à-dire une adresse sur 16 bits. Les deux champs sont à la même place en mémoire.

Une référence à ADRESSE donne donc accès à cette zone. Une référence à CONTENU -> !O! donne accès au premier des deux octets du mot que pointe CONTENU. C'est donc l'octet dont l'adresse en mémoire est ADRESSE.

Utilisation

La dernière partie du programme illustre son utilisation. On commence par POKer une ligne blanche et écrire un texte. Il suffit de choisir les adresses à partir de 1024 (mémoire écran texte) pour les PEEKs, pour constater que le PEEK fonctionne bien et retourne effectivement les codes ASCII de l'écran ligne par ligne; en 40 colonnes, tout se passe bien; en 80 colonnes, cela renvoie les codes correspondant à une colonne sur deux.

Précaution

L'instruction POKE est à utiliser avec prudence. Le système Pascal réside en mémoire vive; il est donc assez facile à détruire. ■

Les recueils de Pom's

Le Recueil N° 1 de Pom's étant épuisé, nous avons décidé de le retirer pour ne pas léser les possesseurs récents d'Apple. Le papier ayant fortement augmenté depuis le tirage initial, le prix est passé à 130 FF. Il en est de même pour le Recueil N° 2 qui vient de sortir — veuillez nous pardonner l'erreur de la première annonce.

```
PROGRAM PEEPO;

VAR I,J,X,Y : INTEGER;
FUNCTION PEEK(ADRS : INTEGER) : INTEGER;
  TYPE MOT = PACKED ARRAY[0..1] OF 0..255;
  BIFACE = RECORD CASE BOOLEAN OF
    TRUE : (ADRESSE : INTEGER);
    FALSE : (CONTENU : ^MOT)
  END;
  VAR PEEKPOKET : BIFACE;

BEGIN
  PEEKPOKET.ADRS := ADRS;
  PEEK := PEEKPOKET.CONTENU^[0]
END;

PROCEDURE POKE(ADRS,VAL : INTEGER);

  TYPE MOT = PACKED ARRAY[0..1] OF 0..255;
  BIFACE = RECORD CASE BOOLEAN OF
    TRUE : (ADRESSE : INTEGER);
    FALSE : (CONTENU : ^MOT)
  END;
  VAR PEEKPOKET : BIFACE;

BEGIN
  PEEKPOKET.ADRS := ADRS;
  PEEKPOKET.CONTENU^[0] := VAL
END;

(*      Petits exemples d'utilisations de PEEK et POKE      *)

BEGIN PAGE(OUTPUT); WRITELN; WRITELN;          (*****
FOR I:=0 TO 39 DO POKE(1024+I,32);             (* )) LIGNE BLANCHE *)
                                              (*****
WRITELN('UN PETIT DUMP'); WRITELN('-----'); WRITELN;
                                              (*****
FOR I:= 1 TO 100 DO X:=PEEK(-16336);           (* )) HAUT-PARLEUR *)
                                              (*****

REPEAT WRITE('      ADRESSE-- '); READLN(X);
  WRITE('NOMBRE D'OCTETS-- '); READLN(Y);
  WRITELN;
  IF Y=0 THEN BEGIN WRITELN; WRITELN('QUE PEEPO SOIT AVEC VOUS.....');
    EXIT(PROGRAM)
  END
  ELSE Y:=X+Y;
  REPEAT WRITE(X:6,' ');
    FOR J:=0 TO 7 DO WRITE(PEEK(X+J):4);
  WRITELN; X:=X+8
  UNTIL X>Y;
  WRITELN
  UNTIL FALSE

END.
```

Disquettes de jeux

Oliver Herz

Pom's est heureux de pouvoir vous proposer deux disquettes de jeux au prix modique de 80 francs pièce. Ces disquettes sont composées avec les contributions de quatre fidèles lecteurs. Ces jeux, dont l'intérêt n'a rien à envier à l'ergonomie, sont variés, allant du labyrinthe au jeu d'adresse en passant par le jeu de réflexion. Signalons que, de plus, ces jeux sont écrits en Applesoft (certains d'entre eux sont aussi fournis sous forme compilée), et qu'ils présentent donc un grand intérêt didactique.

Disquette de jeux A

Labyrinthes - Jackpot - Paires - Awari - Chiffres en lettres - Multi - Pendu - Souris - Eve - Adam et Eve - Jeux à deux - Traclab

Les sept premiers programmes viennent de Jacques Duma, les autres de Joël Lecoœur.

Labyrinthes

Ce petit programme est présenté comme un jeu, mais son originalité principale réside dans l'algorithme de génération du labyrinthe. On voit se construire dans un rectangle un labyrinthe aléatoire qui, de plus, possède une propriété remarquable : si l'on prend deux cases quelconques du rectangle, il n'y a qu'un seul chemin possible pour les relier.

Il suffit de suivre les instructions pour jouer. Il y a deux versions du jeu, LABYRINTHE 2D en deux dimensions et 3D en trois dimensions.

Avec la version 2D, on voit le plan du labyrinthe et il suffit de trouver la sortie le plus rapidement possible. Si l'on fait "?", la solution est fournie par le programme.

Avec 3D, on est à l'intérieur du labyrinthe et seuls les couloirs sont visibles. Quelques indications d'orientation sont fournies. Si l'on fait "+", on tourne sur place et le plan devient visible quelques secondes. Le "?", cette fois, termine le programme sans donner la solution.

NDLR : Jacques Duma nous a fourni une troisième version qui est la même que la version 2D, mais compilée avec Expediter II. Elle est donc beaucoup plus rapide. Rien de vous empêche d'essayer de compiler d'autres jeux avec l'un des compila-

teurs Applesoft existant sur le marché.

Jackpot : il s'agit du jeu de casino, distrayant et utilisant au mieux les possibilités graphiques en basse résolution.

Paires : aussi appelé "jeu de la mémoire". Il s'agit de retourner des paires de cartes. On peut jouer seul contre l'ordinateur (en réglant sa mémoire ...) ou à plusieurs.

Awari : c'est une version particulière d'un jeu africain classique. La stratégie programmée est simple : à chaque coup, l'Apple prend le plus grand nombre de pions possible. Cette stratégie est faible, on peut battre le programme. Qui peut trouver mieux ?

Chiffres en lettres : ce programme écrit en lettres un nombre donné en chiffres.

Multi : un jeu réécrit pour l'Apple d'après une idée trouvée dans un des premiers numéros de l'Ordinateur Individuel.

Le Pendu est un grand classique. Dans la version facile, on a dix coups pour trouver le mot inconnu; dans la version difficile, on n'a que cinq coups. Les mots sont choisis parmi 333 d'une liste donnée en DATA qu'il est facile de modifier, à condition de modifier aussi le nombre de mots de la liste, indiqué au début de celle-ci.

Souris, Eve, Adam et Eve, Jeux à deux, Traclab

Ces programmes, basés sur les labyrinthes, sont fort didactiques. Ce sont des jeux avec divers degrés de difficulté, ainsi qu'un défi que les amateurs relèveront sans doute (du moins le tenteront-ils).

Il y a en fait 4 programmes et 2 tables de formes; et si votre Apple a déjà ingurgité la disquette, vous pouvez voir également une dizaine de "LABYRINTHE N" qui sont des images basse résolution de labyrinthes servant d'exemples à la démonstration. Ils ont été dessinés avec le programme TRACLAB, assez court, simple à utiliser et surtout bien utile pour le défi que lance EVE.

Le défi d'EVE est le suivant : « nul tracé ne m'arrête, je trouve toujours la Pomme ». Le jeu vous permet de tracer avec TRACLAB un labyrinthe et de voir EVE relever son défi ; vous pouvez aussi la voir se faufiler à travers l'un des 10 labyrinthes proposés sur la disquette. Dans la formule de jeu « Adam et Eve », vous vous trouvez en concurrence avec EVE pour la course à la Pomme. La formule « Jeux à deux », enfin, autorise le jeu à deux joueurs.

Disquette de jeux B

Draw Poker - Quandary - Simon II - Poker

Les trois premiers programmes ont été réalisés par Marc Bideault, le dernier par Alban Liger.

Draw Poker est un jeu de hasard, basé sur le principe des combinaisons du poker. Heureusement pour lui, le joueur peut, à l'aide d'un paramètre, se donner plus ou moins de chance. On mise, on distribue une main de 5 cartes, on sélectionne les cartes à remplacer (si nécessaire). Quand on gagne, on peut jouer le gain à quitte ou double en pariant, après avoir vu une carte tirée par l'ordinateur, sur le fait que la carte suivante (tirée elle aussi au hasard) sera plus petite ou plus grande.

Quandary est un jeu de stratégie dans lequel chaque adversaire possède quatre pions dans un échiquier coloré. En suivant certaines règles de déplacement, il faut être le premier à faire atteindre le bord opposé à l'un de ses pions. Ce jeu possède trois niveaux de jeu et est loin d'être imbattable. La couleur (ou de très bons yeux) est indispensable. Une version compilée par Tasc est fournie sur la disquette, afin d'accélérer le processus.

Simon II, quant à lui, ne se présente plus. Il s'agit du célèbre jeu de séquences sonores et colorées à mémoriser.

Poker est un programme de jeu qui reproduit sur l'Apple le jeu de Poker commercialisé sous forme de jeu électronique de poche. Il s'agit d'un poker très rapide, sans changement de cartes.

Apple-Tell est un système qui fait d'un simple Apple (votre Apple de tous les jours) un minitel intelligent

Apple -Tell comprend :

- une carte Modem incluant un décodeur Teletel
- un logiciel d'Emulation de Terminal Minitel enrichi de trois éblouissantes fonctions (celles qui faisaient le plus défaut jusqu'à présent sur votre minitel) :

IMPRESSION : l'imprimante *de votre Apple* est exploitée pour sortir les copies papier dont vous avez besoin lorsque vous consultez un serveur.

STOCKAGE : les disquettes *de votre Apple* sont utilisées pour enregistrer les pages dont la consultation vous est nécessaire :

- au format Teletel (c'est-à-dire telles que vous les avez reçues),
- en mode Texte pur (ASCII) pour exploitation locale ultérieure.

AUTOMATISME : l'intelligence *de votre Apple* est mobilisée pour accomplir l'**interrogation automatique** du serveur que vous avez désigné (appel téléphonique, orientation TRANSPAC, identification, choix successifs), **enregistrer** sur papier et/ou sur disque les données consultées, puis pour **traiter** celles-ci, en les incorporant dans votre application.

(Les Procédures d'interrogation sont créées par l'utilisateur, sans aucun langage de programmation, grâce au mode **Apprentissage** d'Apple-Tell.)

Événement du dernier SICOB, salué par toute la presse, consacré Pomme d'Or 1983 par le Jury Apple, le système Apple-Tell marque une mutation décisive dans l'évolution des techniques videotex en environnement professionnel :

- point d'arrêt à la prolifération des matériels sur votre bureau (effet "mini-Sicob")
- utilisation optimale des ressources dont vous disposez déjà (disques, imprimante, logiciels, etc.),
- utilisation possible en mode Terminal autant qu'en mode **Serveur** (jusqu'à six portes),
- enfin (et c'est sans doute le point le plus important) **JONCTION** entre le monde extérieur et les outils standard *de votre Apple* : l'incorporation de données dans Apple-Writer, Visicalc, Multiplan, PFS, Quick-File, etc., et même dans vos applications personnelles (comptabilité, suivi de commandes, fichier ...) devient possible.

Caractéristiques générales :

- Modem multimodes :
-1200/75, 1200 (half-duplex),
600 (half-duplex), 300 (full-duplex),
-standards CCITT et BELL (cette caractéristique **unique** rend accessibles les serveurs nord-américains, y compris par réseau téléphonique commuté),

- Sorties : video composite (N&B) et Peritel **couleur**.

- Compatible Apple 2, Apple 2+, Apple 2e (48k, une disquette)

- Enfilable dans n'importe quel slot libre **de votre Apple 2**.

- Transparence totale vis-à-vis du système.

BON DE COMMANDE

HELLD Informatique - 1, rue de Metz 75010 PARIS
Tél : (1) 523 30 34 - Telex : FLASH 210500F

Nom _____ Société _____

Adresse complète _____

Ville _____ Code Postal _____ tél. _____

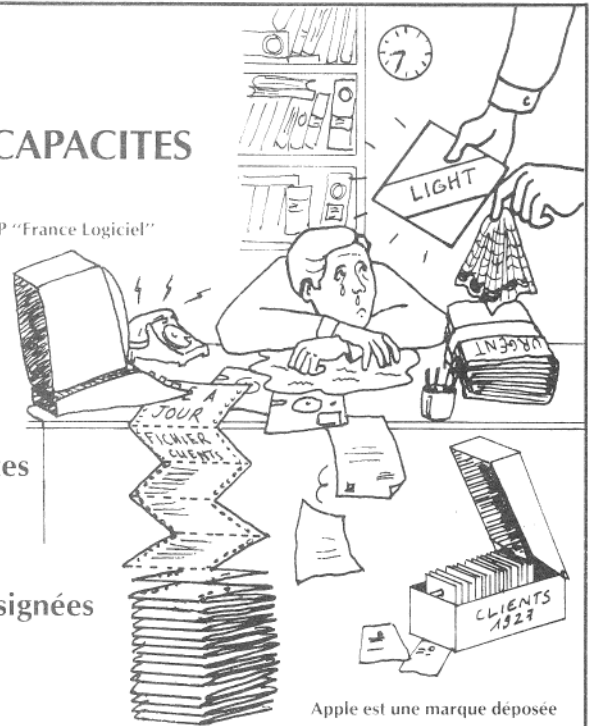
- Souhaite recevoir une documentation sur le système Apple-Tell
- Commande système(s) Apple-Tell au prix de F. 5900 H.T.
(règlement ci-joint par : Chèque bancaire CCP)

**VOUS PROGRAMMEZ SUR APPLE II
VOUS VOULEZ GAGNER DU TEMPS
VOUS CHERCHEZ A AMELIORER VOS CAPACITES**

LIGHT EST FAIT POUR VOUS Deposé APP "France Logiciel"

LIGHT est un ensemble de puissants logiciels d'aide à la programmation conçu pour :

- mieux gérer les fichiers et les disquettes
mise à jour d'un catalogue général
état des secteurs libres sur un disque
- faciliter l'exploration et l'exploitation des disquettes
accès direct, lecture, écriture des secteurs
transfert et recherche des séquences d'octets
- accélérer l'analyse de vos programmes
en Basic: recherche des tokens et des suites désignées
en Binaire: impression en désassemblé
de séquences d'octets; pistage en tout ou
partie du déroulement



Apple est une marque déposée

POUR EN SAVOIR PLUS SUR LIGHT :

M. _____

adresse _____

demande une documentation complète, sans engagement de sa part, aux Années Nouvelles, 70, rue du Javelot, 75013 Paris.

MODEMS TELSAT

**L'ACCES DE VOTRE MICRO ORDINATEUR AUX BASES DE DONNEES
PAR LE RESEAU TELEPHONIQUE**



TELSAT 440
Modem 300 bit/s full duplex



TELSAT 640
Modem 600/1200 bit/s full duplex

Ces produits font partie de la gamme complète de modems, multiplexeurs et équipements de réseaux de transmission de données TELSAT distribués par

SATELCOM international 69-71, rue du CHEVALERET 75013 PARIS

Tél. 584 14 75

Telex 204120F

Imprimez les codes ESC de votre PLE

Pascal Fontanier

Vous connaissez la phrase traditionnelle APRES avoir découvert PLE (le Program Line Editor, voir Pom's numéros 1 et 7) : "Comment pouvais-je faire AVANT ?". Pour ma part, je me suis découvert le besoin de plusieurs PLEs : un PLE standard, un PLE pour les programmes graphiques, un PLE pour les travaux sur fichiers, etc.

Mais, comment faire pour s'y retrouver plusieurs mois après la création d'un nouveau PLE ? La méthode des essais est bien sûr efficace, mais rustique et fastidieuse. J'ai donc écrit le programme suivant, qui affiche les caractères tapés après la touche ESC et leur signification, dans l'ordre où ils sont en mémoire, puis rangés par ordre ASCII.

Pour utiliser ce programme, chargez PLE, puis lancez le programme PRINT PLE.ESCAPE.

- Le programme détecte l'absence de fichier PLE en mémoire.
- Puis vous demande le nom de votre PLE sur votre disquette, la date, et vous rappelle d'allumer votre imprimante. Cela fait, appuyez sur n'importe quelle touche.
- Certains items sont suivis de <RETURN> : ce qui s'affiche à l'écran est alors suivi d'un "retour chariot" qui vous évite d'avoir à le taper.
- D'autres items sont suivis de * RECURSIF *, ce qui signifie qu'il n'est pas nécessaire de retaper ESC pour obtenir de nouveau le résultat désiré. Une application évidente est disponible d'origine sur l'Apple : ESC I,J,K,M permettent les mouvements du curseur dans les quatre directions.
- Certains items, enfin, comprennent des caractères de contrôle qui sont alors imprimés comme suit : C(X) correspond à CTRL-X.

Le programme est suffisamment documenté pour être compris facilement. Il suffit de savoir que les codes ESC sont rangés en mémoire à partir de l'adresse 39858 (\$9BB2). La liste des codes se termine par 0, ce qui

permet de la détecter. En mémoire, les codes sont composés de la façon suivante :

- Premier caractère = touche frappée après ESC.
- Caractères suivants = ce qui apparaîtra à l'écran, tous les octets étant supérieurs à 128 (\$80).
- Le caractère de fin de code est inférieur à 128.

Le programme a été écrit pour fonctionner sur Apple II Plus 48K avec une imprimante Epson MX/82. Si vous utilisez une autre imprimante, vous aurez peut-être à modifier les lignes 250 (qui sélectionne les caractères U.S.), 270, 480, 620 et 700.

Si vous ne désirez pas avoir la liste imprimée dans le désordre, supprimez les lignes 270 et 280. ■

Exemple de sortie dans l'ordre ASCII

```
<ESC> = * RECURSIF *
ESC C(E) = C(G)C(G)C(X)
ESC ! = ?TAB(20-LEN(ZY$))/2)
ESC " = ?D$"
ESC * = CALL-151 <RETURN>
ESC 0 = SPEED=255
ESC 1 = LEFT$(
ESC 2 = MID$(
ESC 3 = RIGHT$(
ESC 4 = CHR$(
ESC 5 = STR$(
ESC 6 = SPEED=
ESC 7 = CALL
ESC 8 = PEEK(
ESC 9 = POKE
ESC : = APPEND"
ESC < = OPEN"
ESC = = H=PEEK(36):V=PEEK(37)
ESC > = CLOSE"
ESC ? = ?TAB(
ESC A = CATALOGD2 <RETURN>
ESC D = READ"
ESC G = GOTO
ESC H = FLASH:
ESC I = <ESC> D * RECURSIF *
ESC J = <ESC> B * RECURSIF *
ESC K = <ESC> A * RECURSIF *
ESC L = LIST <RETURN>
ESC M = <ESC> C * RECURSIF *
ESC N = NEXT
ESC O = ONERRGOTO
ESC P = INPUT
ESC Q = CATALOGD1 <RETURN>
ESC R = RETURN <RETURN>
ESC S = GOSUB
ESC T = TEXT:HOME:SPEED=255:NORMAL:D$=CHR$(4) <RETURN>
ESC U = NORMAL:
ESC V = INVERSE:
ESC W = WRITE"
ESC X = RESUME <RETURN>
ESC Z = THEN
```

```
10 REM *****
11 REM *
12 REM * PASCAL FONTANIER *
13 REM *
14 REM * PRINT PLE.ESCAPE *
15 REM *
16 REM * VERSION 1.2 *
17 REM *
```

```
18 REM *****
100 TEXT : HOME : SPEED= 255: NORMAL :D
    $ = CHR$(4)
110 INVERSE : PRINT " IMPRESSION DES
    CODES 'ESC' DU PLE PASCAL
    FONTANIER - OCTOBRE 1983 ====
    =====
```

```

      ==": POKE 34,4
120 REM
      *** PLE IN SYSTEM ? ***

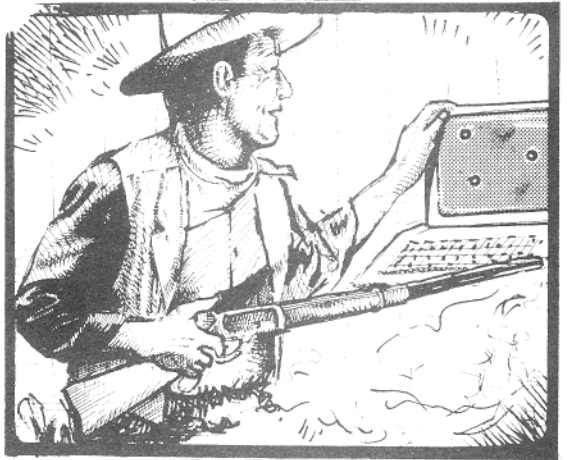
130 Q = PEEK (1011) + 4: IF Q < = PEE
      K (978) THEN 170
140 PRINT CHR$( 7); CHR$( 7); CHR$( 7)
      : PRINT : INVERSE : PRINT "*** PL
      E N'A PAS ETE CHARGE ***": PRINT
      : PRINT : NORMAL
150 FLASH : PRINT "CHARGEZ LE": PRINT :
      PRINT : NORMAL : PRINT " PUIS RE
      CHARGEZ CE PROGRAMME": PRINT : EN
      D
160 REM
      *** PLE EST BIEN PRESENT ***

170 NORMAL
180 DIM S$(100)
190 INPUT "NOM DE LA VERSION DU PLE : "
      ;NOMS
200 PRINT : PRINT : PRINT
210 INPUT "DATE : ";DA$
220 ADR = 39858: REM ADR CODES ESC
230 N = N + 1: REM NOMBRE D'ELEMENTS
240 PRINT : PRINT : INVERSE : PRINT "A
      LLUMEZ L'IMPRIMANTE ";: NORMAL :
      POKE - 16368,0: GET ZZ$: PRINT
250 PRINT D$"PR$1": PRINT CHR$( 27); C
      HR$( 82); CHR$( 0): PRINT : PRINT
      : PRINT D$"PR$0": REM 3 CHR$ =
      PRINT USA
260 NOMS = NOMS + " " + DA$:NO = LE
      N (NOMS)
270 PRINT D$"PR$1": PRINT NOMS;"
      DANS LE DESORDRE"
280 FOR X = 1 TO NO + 25: PRINT "=";: N
      EXT X: PRINT : PRINT
290 REM
      *** ANALYSE DU CODE 'ESC' ***

300 IF PEEK (ADR) = 0 THEN PRINT "---
      -----":
      PRINT : PRINT : GOTO 480: REM T
      RI
310 Y = PEEK (ADR)
320 IF Y < 128 THEN 390: REM FIN DE COD
      E
330 IF Y = 155 THEN S$(N) = S$(N) + "<E
      SC> ": GOTO 370
340 IF Y > = 240 AND Y < 250 THEN Y =
      Y - 240:S$(N) = S$(N) + STR$( Y)
      : GOTO 370
350 IF Y > 127 AND Y < 159 THEN S$(N) =
      S$(N) + "C(" + CHR$( Y - 64) +
      ")": GOTO 370
360 S$(N) = S$(N) + CHR$( Y)
370 ADR = ADR + 1: GOTO 310
380 REM
      *** FIN DE CODE ***

390 IF Y = 13 THEN S$(N) = S$(N) + " <
      RETURN>": GOTO 430
400 IF Y = 127 THEN S$(N) = S$(N) + "
      * RECURSIF *": GOTO 430
410 IF Y < 26 THEN S$(N) = S$(N) + "C("
      + CHR$( Y + 64) + ") ": GOTO 43
      0
420 S$(N) = S$(N) + CHR$( Y + 128)
430 IF LEN (S$(N)) > 5 THEN IF LEFT$

```



```

      (S$(N),5) = "<ESC>" THEN S$(N) =
      "<ESC> = * RECURSIF *": GOTO 460
440 IF LEFT$( S$(N),2) = "C(" THEN S$(
      N) = "ESC " + LEFT$( S$(N),4) +
      " " + MID$( S$(N),5): GOTO 460
450 S$(N) = "ESC " + LEFT$( S$(N),1) +
      " " + MID$( S$(N),2)
460 PRINT S$(N):ADR = ADR + 1:N = N + 1
      : GOTO 300
470 REM
      *** TRI ***

480 PRINT D$"PR$0"
490 FLASH : PRINT "PATIENTEZ, S.V.P.,":
      INVERSE : PRINT : PRINT "JE TRIE
      LES CODES PAR ORDRE A.S.C.I.I.":
      NORMAL
500 M = N
510 M = INT (M / 2)
520 IF M = 0 THEN 620
530 P = N - M:J = 1
540 I = J
550 L = I + M
560 IF S$(I) < = S$(L) THEN 590
570 W$ = S$(I):S$(I) = S$(L):S$(L) = W$
580 I = I - M: IF I > = 1 THEN 550
590 J = J + 1: IF J > P THEN 510
600 GOTO 540
610 REM
      *** IMPRESSION TRIEE ***

620 PRINT D$"PR$1"
630 PRINT NOMS;" DANS L'ORDRE ASCI
      I"
640 FOR X = 1 TO NO + 24: PRINT "=";: N
      EXT X: PRINT : PRINT
650 FOR X = 1 TO N
660 PRINT S$(X)
670 NEXT X
680 PRINT "-----
      -----": PRINT : PRINT "FIN DE ";
      NOMS: PRINT : PRINT : PRINT
690 PRINT D$"PR$0"
700 PRINT : PRINT : INVERSE : PRINT "UN
      DOUBLE ? (O/N) ";: GET ZZ$
710 NORMAL
720 IF ZZ$ = "O" THEN PRINT : GOTO 620
750 TEXT : NORMAL : CALL 38656: END
65535 REM
      =====
      PASCAL FONTANIER
      PRINT PLE.ESCAPE
      OCTOBRE 1983
      =====

```


Tracé rapide de cercles

Dominique Bernardi

Contrairement à ce qu'affirme Olivier Herz dans le *Courrier des Lecteurs* du numéro 10 (je ne suis pas peu fier de le prendre en défaut), il est possible de tracer des cercles sans calculer de sinus ni de cosinus. Le principe est le suivant :

Si les deux suites de nombres $X(n)$ et $Y(n)$ sont définies par récurrence par les formules :

$X(0)=R$ (par exemple)

$Y(0)=0$

$Y(n+1)=Y(n)+e*X(n)$

$X(n+1)=X(n)-e*Y(n+1)$

les points de coordonnées $X(n)$ et $Y(n)$ s'inscrivent sur une ellipse d'équation :

$$X*X + e*X*Y + Y*Y = R*R$$

Il suffit de donner une petite valeur à e pour que l'ellipse soit indiscernable d'un cercle. L'intérêt de cette méthode vient de ce qu'elle est stable : les erreurs de calcul ne s'accumulent pas.

Pour des raisons évidentes, j'ai choisi la valeur $e=1/256$, qui donne un excellent résultat au point de vue graphique, mais peut-être pourrait-on accélérer un peu le programme en prenant $e=1/128$ ou $1/64$? Je ne suis pas un spécialiste de l'assembleur; essayez donc vous-mêmes si le cœur vous en dit. Dans le même ordre d'idées, la routine actuelle n'accepte que les centres sur l'écran, et les rayons inférieurs à 256...

Vous trouverez ci-après, comme sur la disquette d'accompagnement de ce numéro, le programme source CERCLE en DOS Tool Kit, la version code CERCLE.OBJ0 et un programme de démonstration en Basic.

Programme CERCLE DOS ToolKit

```

1 RAYON EQU $300
2 CX EQU $301
3 CY EQU $303
4 COS EQU $6
5 SIN EQU $8
6 PLOT EQU $F457
7 *
8 *
9 ORG $304
10 *
11 *
12 INIT LDA RAYON
13 STA COS+1
14 LDA #$0
15 STA SIN+1
16 STA SIN
17 STA COS
18 JSR ECRIS
19 *
    
```

```

20 *
21 VARY CLC
22 LDA SIN
23 ADC COS+1
24 STA SIN
25 BCC VARX
26 INC SIN+1
27 JSR ECRIS
28 VARX SEC
29 LDA COS
30 SBC SIN+1
31 STA COS
32 BCS VARY
33 DEC COS+1
34 JSR ECRIS
35 LDA COS+1
36 BNE VARY
37 FIN RTS
38 *
39 ECRIS LDA CY
40 CLC
41 ADC SIN+1
42 BCS T1
43 CMP #192
44 BCS T1
45 JSR ECRIS2
46 T1 SEC
47 LDA CY
48 SBC SIN+1
49 BCC RETOUR
50 ECRIS2 PHA
51 LDA CX
52 CLC
53 ADC COS+1
54 BCS T6
55 LDY CX+1
56 BEQ T7
57 CMP #24
58 BCS T8
59 T7 TAX
60 PLA
61 PHA
62 JSR PLOT
63 JMP T8
64 T6 LDY CX+1
65 BNE T8
66 CMP #24
67 BCS T8
68 TAX
69 INY
70 PLA
71 PHA
72 JSR PLOT
73 T8 SEC
74 LDA CX
75 SBC COS+1
76 BCC T9
77 LDY CX+1
    
```

```

78 TAX
79 PLA
80 JSR PLOT
81 JMP RETOUR
82 T9 TAX
83 PLA
84 LDY CX+1
85 BEQ RETOUR
86 DEY
87 JSR PLOT
88 RETOUR RTS
    
```

CERCLE.OBJ0

```

0304- AD 00 03 85
0308- 07 A9 00 85 09 85 08 85
0310- 06 20 35 03 18 A5 08 65
0318- 07 85 08 90 05 E6 09 20
0320- 35 03 38 A5 06 E5 09 85
0328- 06 B0 E9 C6 07 20 35 03
0330- A5 07 D0 E0 60 AD 03 03
0338- 18 65 09 B0 07 C9 C0 B0
0340- 03 20 4C 03 38 AD 03 03
0348- E5 09 90 49 48 AD 01 03
0350- 18 65 07 B0 12 AC 02 03
0358- F0 04 C9 18 B0 19 AA 68
0360- 48 20 57 F4 4C 77 03 AC
0368- 02 03 D0 0B C9 18 B0 07
0370- AA C8 68 48 20 57 F4 38
0378- AD 01 03 E5 07 90 0B AC
0380- 02 03 AA 68 20 57 F4 4C
0388- 95 03 AA 68 AC 02 03 F0
0390- 04 88 20 57 F4 60
    
```

Programme CERCLE DEMO

```

10 D$ = CHR$(4): PRINT
   D$;"BLOAD CERCLE.O
   BJO"
20 HGR2: HCOLOR=3: PRI
   NT CHR$(7);
30 FOR X = 0 TO 279 STEP
   55
33 XH = INT (X / 256):XL
   = X - 256 * XH
36 POKE 769,XL: POKE 770
   ,XH
40 FOR Y = 0 TO 191 STEP
   50
45 POKE 771,Y
60 FOR R = 2 TO 200 STEP
   40
70 POKE 768,R: CALL 772
80 NEXT : NEXT : NEXT
100 PRINT CHR$(7);: GE
   T A$
110 TEXT : END
    
```

La micro-informatique au Japon

Olivier Herz

Le développement de la micro-informatique au Japon prend l'allure d'une explosion. Le marché, quasi inexistant il y a un ou deux ans, évolue de manière exponentielle.

Les magasins

A proprement parler, il y a très peu de magasins exclusivement consacrés à l'informatique, mais chaque magasin d'articles électroniques possède un rayon TV-vidéo, un rayon calculatrices, un rayon Walkman... et un rayon micro-informatique.

Les trois quarts de ces magasins sont concentrés dans le quartier d'Akihabara, où ils pullulent par dizaines, voire centaines.

En gros, les ordinateurs sont classés en deux catégories :

- les pasocom (personal computers) destinés à l'usage domestique;
- les wordpro (word processors) à usage professionnel.

Il va sans dire que les ordinateurs étrangers sont quasiment absents : parmi les ordinateurs exposés, il y a moins de trois pour cent d'Apple, et encore, jamais en état de marche pour démonstration; quant à IBM, il n'utilise pas les circuits de distribution mentionnés ci-dessus, mais les "Computerland". Ceci n'a rien d'étonnant, Apple et IBM étant à peu près au prix français (voir ci-dessous les prix des ordinateurs japonais).

Le langage

La langue japonaise utilise un système d'écriture fort complexe. Il y a deux alphabets syllabiques (kana) :

- le katakana, qui sert à écrire les mots d'origine étrangère, et que les lecteurs de Pom's connaissent bien (voir HAIFA Pom's 5);
- le hiragana, qui sert à écrire les terminaisons grammaticales, les post-positions...

Il y a aussi des idéogrammes d'origine chinoise (kanji) qui servent à écrire les racines des noms, verbes, adjectifs... et dont on ne peut se passer, compte tenu du très grand nombre d'homonymes.

Cet état de fait pose un énorme problème au niveau du software de base, qui peut être résolu de la façon suivante :

- pour les ordinateurs bas de gamme, on peut écrire en kana (généralement katakana) grâce à

un générateur de caractères et à une commutation kana/romaji (romaji est notre alphabet, parfois utilisé pour écrire certains mots);

- pour les ordinateurs professionnels, le kanji est intégré au système d'exploitation, c'est-à-dire que l'on rentre les mots en syllabaire hiragana et que le DOS offre

un choix de kanji dont on tire celui cherché. Les logiciels de traitement de texte tournent avec un tel DOS. Au Japon, c'est une révolution !

Tout ceci explique les efforts considérables portés par le Japon en matière de ROM et RAM (on en est au mé-

Marque	Type	Microprocesseur	ROM	RAM	Disquettes	DOS	Prix
NEC	N5200	8086 8MHz	/	256K	1 * 8p.	FTDOS	25200
NEC	PC100	8086 7MHz	32K	128K	1 * 720K	MS-DOS	14400
NEC	PC6601	PD780C 4MHz	96K	64K	/	/	3060
NEC	PC8801	PD780C 4MHz	128K	?	?	?	?

SHARP	X1	Z80A 4MHz	6K	64K	Cassettes	/	4320
SHARP	MZ2200	Z80A 4MHz	4K	64K	Option :	/	4608
SHARP	MZ5500	8086 5MHz	16K	128K	1 * 320K	MS-DOS et	10368
Pocket	PC1245	CMOS 8bits	24K	2.2K	/	CP/M 86	648
Pocket	PC1255	CMOS 8bits	24K	10.2K	/	/	1296
Pocket	PC1401	CMOS 8bits	40K	4.2K	/	/	1080
Pocket	PC1501	CMOS 8bits	16K	8.5K	/	/	2340

FUJITSU	FM 7	6809 8MHz Z80A 4MHz	44K	64K	1 * 320K	CP/M 80 ou UCSD	4536
FUJITSU	FM 11	8088 8MHz 6809 2MHz Z80A 4MHz	4K	128K	1 * 320K	CP/M 80 ou 86, MS-DOS et UCSD, OS-9 plus	9720
FUJITSU	FM X	Z80A 3.58MHz	16K	16K	Cassettes	/	1800

	OASYS	?	?	?	1 drive	Tr. texte	27000

HITACHI	H1	Z80A	40K	32K	Cassettes	/	2268

TOSHIBA	PASOPIA 10	Z80A	32K	64K	Cassettes	/	2376
TOSHIBA	PASOPIA 7	Z80A 4MHz	16K	64K	Cassettes	/	4284

CANON	V-10	Z80A 3.58MHz	32K	16K	Cartouches	/	1980

NATIONAL	CF 200	?	?	32K	Cartouches	/	1980

OKI	IF 800	8086 8MHz	?	256K	2 floppies	MS-DOS	?

EPSON	HC 20	?	?	?	Micro-cass.	/	5760
EPSON	GC 10	?	?	?	?	CP/M	?

CASIO	FP-3000	8086 8MHz	?	128K	?	MS-DOS	5400

SANYO	MBC-55	8086	?	128K	?	MS-DOS, CPM	6408
SANYO	MBC 200	Z80A 4MHz	?	?	1 floppy	CP/M	16200

Apple	//e	(prix au Japon, à titre de comparaison)					13608

Le tableau a été calculé avec le taux de change : 1 000 yens = 36 FF

Certains prix n'ont pu être découverts par Olivier Hertz, d'où les "?"

gabit sur une puce), la recherche d'un kanji se faisant actuellement en partie sur disquette.

Il va sans dire que le code ASCII ne marche pas dans une telle situation; on définit un code sur deux octets.

Livres et revues

Chaque ordinateur ou presque a sa revue et ses livres (je n'ai pas vu de revue consacrée à l'Apple, seulement quelques livres) et il est étonnant de constater que les livres américains sont quasiment absents. Pour les trouver, il faut chercher dans les librairies ayant des rayons "étranger". Il y a en outre quelques revues d'intérêt général, avec articles et publicités.

Imprimantes

Il n'y a pas tellement plus d'imprimantes qu'en France. On retrouve toujours les mêmes (Seikosha, Epson, Oki, NEC...), mais souvent habillées d'une autre marque : celle

de l'ordinateur avec lequel est vendu l'imprimante.

Ordinateurs familiaux

C'est la pléthore; on en trouve à tous les prix, y compris des seize bits aux possibilités graphiques exceptionnelles.

Le Sinclair QL et autres n'ont plus qu'à trembler (voir le tableau ci-joint).

Ordinateurs professionnels

Là aussi, il y a pléthore de modèles seize bits (avec 68000, 8086...) fonctionnant sous CP/M 86, MS-DOS... Le problème est que je n'ai pas encore trouvé de compatible IBM, ce qui serait très sympathique, compte tenu des prix.

IBM 5550

Cet ordinateur, fabriqué au Japon par IBM-Japan, a un succès extraor-

dinaire auprès des sociétés, bien qu'il soit très cher (près de 100.000 F avec trois drives).

Construit autour d'un 8086, il possède deux systèmes d'exploitation :

- le premier sert uniquement au traitement de texte en japonais;
- le Nihongo DOS, enfin, ressemble au MS-DOS pour les mots-clés, mais possède en outre une conversion kana/kanji. Sous ce système, on peut faire tourner des langages (Basic, Pascal, Fortran, avec affichage des messages en japonais si nécessaire), et des logiciels comme Multiplan, Multitool Chart (graphiques), Multitool File (Data Base)...

Je n'en dirai pas plus, les lecteurs intéressés par les détails pouvant se reporter à l'article paru dans la revue Byte (novembre 83) : "Big Blue goes Japanese".

Imagol s'est fait depuis longtemps déjà une place un peu particulière en devenant le champion du MEM/DOS, un système d'exploitation particulièrement adapté à la gestion de fichiers sur Apple (voir "La carte M/DOS 6502 à l'essai", Pom's No 4).

Imagol lance maintenant, toujours dans la famille MEM/DOS, un nouveau service : MEM/TERM, logiciel de communication transformant un Apple en terminal d'un autre Apple. La communication peut se faire par ligne directe, ou par l'intermédiaire d'un modem sur une ligne téléphonique. Notre but ici est de vous présenter sa genèse et les projets de développement.

Les réalisations

MEM/TERM est né avant tout d'une constatation : MEM/DOS est surtout répandu dans les PME et les SSCI (80% des installations se trouvent en province). L'isolement géographique de nombreux utilisateurs a conduit à leur offrir un service de télé-mise au point et de télémaintenance des logiciels.

Dans le cas d'installations multipostes (fréquentes sous MEM/DOS), les diagnostics de dysfonctionnement peuvent parfois s'avérer délicats : MEM/TERM s'est alors vu doter d'une fonction de "RUN" à distance permettant le test des fonctions de l'installation réelle de l'utilisateur. Sous MEM/DOS, les masques sont stockés indépendamment des programmes, ce qui permet de ne pas les trans-

mettre. Ils sont dupliqués sur l'installation du "mainteneur", et seules les informations utiles transitent par les lignes de communication.

En "prime", MEM/TERM permet aussi d'exécuter un programme à distance chez l'utilisateur lui-même, par exemple des transactions en temps réel sur un poste dans la journée, et une exploitation en temps différé sur un autre poste (toujours en ne transmettant que les données utiles).

Les projets

La première voie de recherche vise une amélioration du produit de base, en le dotant d'un système de protection permettant d'utiliser MEM/TERM avec CERTAINS programmes et fichiers, mais empêchant de le faire pour d'autres, logés sur le MEME disque (ceci en vue d'assurer la confidentialité de certains fichiers et traitements).

La seconde voie consiste à créer une sorte de "banque de démos" éclatée, les divers abonnés à ce service (revendeurs notamment) pouvant de leur poste utiliser à des fins commerciales les "démos" proposées, en particulier par des producteurs ou distributeurs de logiciels.

La troisième voie est la création d'un réseau avec un serveur et des Apple

Pom's a vu MEM/TERM

Guy Lapautre

(en "concurrence" avec Minitel). Toutefois, d'autres solutions possibles sont à l'étude, notamment celles incluant des Minitel dans le réseau. À faire à suivre.

Coût du service

MEM/TERM est-il un service coûteux ? Rappelons tout d'abord qu'il ne fonctionne que sous MEM/DOS, et le plus souvent sur des installations relativement musclées. Les prix qui suivent ne comprennent pas l'acquisition du MEM/DOS.

1) Installation simple pour télémaintenance : 5.000 F environ pour le modem et la carte Super Série, plus 1.000 F pour le logiciel, soit un coût total d'environ 6.000 F HT.

2) Communication entre deux Apple chez l'utilisateur : l'installation ci-dessus en double (1 des modems avec porteuse) et 1 logiciel à chaque extrémité, soit environ 12.000 F HT. Par contre, si la distance est courte, au maximum 100 mètres sans ampli, le coût est d'environ 4.500 F HT, plus celui d'un câble 3 brins entre les postes.

Pour tous renseignements, téléphoner à Monsieur Ratié, Imagol, 1-5 rue Gutenberg, 75015 Paris - Tel 577.59.39.

FAITES VIVRE VOTRE APPLE II

DES LIVRES ET DES DISQUETTES POUR VOTRE APPLE

Les disquettes d'accompagnement des Editions du P.S.I. sont destinées aux personnes qui n'ont pas le temps de taper elles-mêmes les programmes. Chaque disquette constitue la fidèle adaptation sur Apple II des listings proposés dans l'ouvrage de référence. Ces disquettes doivent être considérées comme une aide au lecteur et non comme un progiciel.

Attention! les consignes d'utilisations des programmes enregistrés sur les disquettes sont précisées dans les ouvrages de référence. Pour chaque disquette, l'achat du livre est donc indispensable.

La disquette seule: 195,00 FF (port et emballage compris)

Pour Apple II, II plus, //e - Dos 3.3 - version 48 K ou plus :

- Disquette: "Nouvelle comptabilité sur Apple II" tome 1
- Disquette: "Nouvelle comptabilité sur Apple II" tome 2
- Disquette: "La paie et ses annexes"
- Disquette: "Outils financiers et comptables"
- Disquette: "Modèles pratiques de décision" tome 1
- Disquette: "Modèles pratiques de décision" tome 2
- Disquette: "L'Apple et ses fichiers" tome 1
- Disquette: "Modèles d'expression graphique"
- Disquette: "Mathématiques et Statistiques"
- Disquette: "Méthodes de calcul numérique" tome 1 et 2
- Disquette: "Le Basic et l'école" tome 1
- Disquette: "Le Basic et l'école" tome 2
- Disquette: "Visicalc sur Apple II"

Attention: Disquette maîtresse Visicalc indispensable.

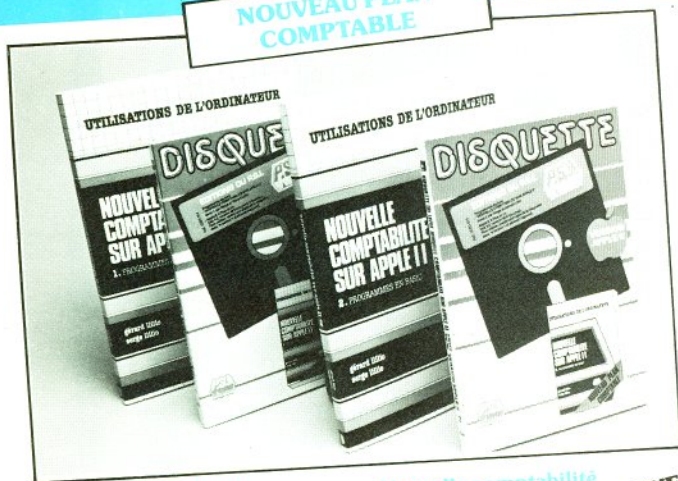
Pour Apple II plus, //e - Dos 3.3 - version 64 K ou plus

- Disquette: "Multiplan pour Apple II"
- Attention: Disquette maîtresse Multiplan indispensable.

Pour Apple II, II plus, //e avec Système Pascal

- Disquette: "Bibliothèque scientifique en Pascal"

NOUVEAU PLAN COMPTABLE



Nouvelle comptabilité sur Apple II - tome 1

par Serge et Gérard Lillio

Voici un ensemble complet de programmes de Comptabilité sur Apple II adapté au Nouveau Plan Comptable, pour petites entreprises, professions libérales, artisans, commerçants. On y trouve des programmes permettant l'édition des livre-journal, grand-livre, balance, bilan; la possibilité d'éditer les livres de banque, de TVA, de recettes, ainsi qu'un programme de détection des erreurs de frappe. Capacité du système: compte à 6 chiffres, 400 lignes par plan.

Le livre: 184 pages - 120,00 FF
La disquette: 195,00 FF

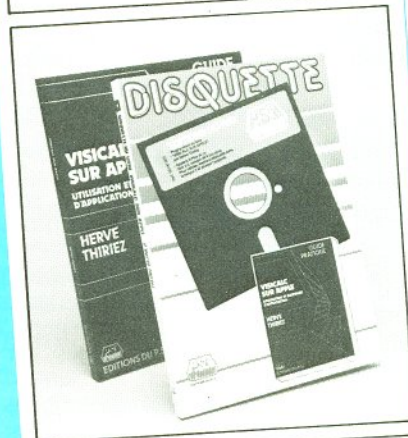
Nouvelle comptabilité sur Apple II - tome 2

par Serge et Gérard Lillio

Ce second tome contient trois programmes complémentaires des programmes de comptabilité du tome 1: - visualisation graphique tridimensionnelle des recettes et dépenses avec calcul des ratios. - tableaux des amortissements. - calcul d'impôts, ainsi qu'un programme de comptabilité générale pour les utilisateurs de disques 8 pouces, disques durs ou d'une seule unité 5 pouces 1/4.

Le livre: 144 pages - 110,00 FF
La disquette: 195,00 FF

NOUVEAU



La paie et ses annexes

par Jean-Michel Jégo

Cet ouvrage présente l'analyse et la programmation des problèmes de la paie et des charges salariales: fichiers de personnel, calcul des cotisations, élaboration d'un bulletin de salaire avec congés payés. En fin d'ouvrage on aboutit à la tenue d'un cahier des charges salariales tel qu'il peut se présenter dans une petite entreprise. Attention: beaucoup de simplifications ont été apportées par rapport aux situations très diverses rencontrées par les professionnels de la paie.

Le livre: 136 pages - 110,00 FF
La disquette: 195,00 FF

Outils financiers et comptables pour l'entreprise

par Bernard Sulmon

Sont traités dans ce livre: l'analyse des coûts marginaux, le calcul de seuils de rentabilité, le Direct Costing, le calcul des ratios ou du Fond de Roulement minimum, la rentabilité des investissements et la gestion des stocks. Chaque programme est proposé avec un exemple d'application.

Le livre: 156 pages - 100,00 FF
La disquette: 195,00 FF

Multiplan pour Apple II plus et //e

NOUVEAU

par Hervé Thiriez

Multiplan est un progiciel qui permet de gérer plusieurs tableaux simultanément: cet ouvrage sera pour les possesseurs d'ordinateurs Apple II Plus ou //e un véritable guide d'utilisation de Multiplan grâce à des exemples progressifs et à de nombreux cas d'application (gestion de portefeuilles, de copropriété, feuille de paie, impôts, tableaux de bord, etc.).

Le livre: 216 pages - 100,00 FF
La disquette: 195,00 FF

Visicalc sur Apple

par Hervé Thiriez

Après une présentation progressive du modèle Visicalc, l'ouvrage étudie de nombreux cas d'application: feuille d'impôt, gestion de copropriété, paie, facturation... permettant d'introduire les différentes instructions et astuces d'utilisation.

Le livre: 176 pages - 90,00 FF
La disquette: 195,00 FF

L'Apple et ses fichiers

par Jacques Boisgontier

Pour apprendre progressivement la programmation des applications utilisant les fichiers l'ouvrage commence par une présentation concise et illustrée des com. andes du Système d'Exploitation Disque et des instructions du Basic Applesoft. Les instructions des fichiers séquentiels et à accès direct sont ensuite décrites ainsi que leur utilisation. Des méthodes pratiques, souvent mal connues, montrent comment utiliser au mieux des fichiers à accès direct: accès indexé, liste inverse. Une vingtaine de programmes illustrent l'utilisation de ces techniques.

Le livre: 176 pages - 90,00 FF
La disquette: 195,00 FF



Modèles pratiques de décision

Tome 1

par Jean-Pierre Blanger

Cet ouvrage vise l'automatisation du processus de la prise de décision. Les différentes techniques exposées sont complétées d'un exemple et d'un programme en Basic pour permettre au lecteur une rapide maîtrise des modèles présentés et leur intégration à de nombreuses applications (simulation, gestion, intelligence artificielle...).

Le livre : 144 pages - 90,00 FF
La disquette : 195,00 FF



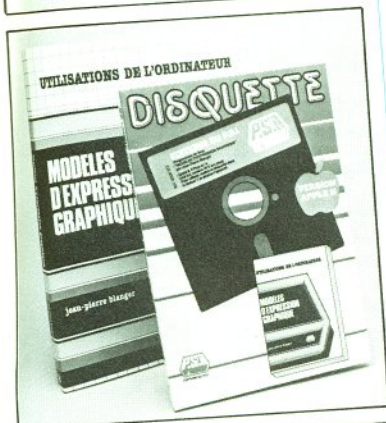
Modèles pratiques de décision

Tome 2

par Jean-Pierre Blanger

Ce tome 2 de Modèles pratiques de décision offre un nouvel éventail de techniques visant l'automatisation du processus de la prise de décision. Chaque des vingt modèles présentés donne lieu à un bref exposé, un exemple et un programme en Basic standard.

Le livre : 176 pages - 90,00 FF
La disquette : 195,00 FF



Modèles d'expression graphique

par Jean-Pierre Blanger

Cet ouvrage expose un ensemble de techniques visant la mise en œuvre des possibilités graphiques des ordinateurs individuels. Il permet au débutant comme à l'amateur chevronné d'aborder la résolution de problèmes de plus en plus complexes (tracés d'ellipse, rotation de polygone, hachurage de surface...). Les modèles d'expression graphique, écrits en Basic Applesoft, sont abondamment commentés et facilement adaptables à d'autres ordinateurs individuels.

Le livre : 232 pages - 130,00 FF
La disquette : 195,00 FF



Mathématiques et statistiques

par Hervé Haut

Cet ouvrage est un recueil de 16 logiciels de base (niveau supérieur) tant en mathématiques qu'en statistiques. Chaque problème traité comporte une introduction numérique, un exposé de la technique de programmation utilisée, un organigramme détaillé et un programme complet en Basic suivi d'un exemple d'utilisation.

Le livre : 272 pages - 100,00 FF
La disquette : 195,00 FF

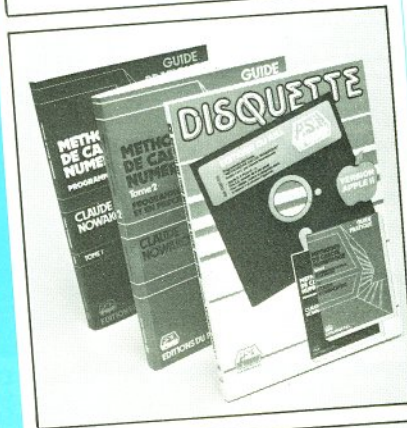


Bibliothèque scientifique en Pascal

par Hervé Haut

Les procédures proposées dans cet ouvrage sont conçues pour être implantées dans le System Library du langage. L'utilisateur disposera d'une bibliothèque enrichie, permettant une résolution aisée et performante d'un grand nombre de problèmes mathématiques et statistiques souvent rencontrés dans les programmes scientifiques.

Le livre : 152 pages - 90,00 FF
La disquette : 195,00 FF



Méthodes de calcul numérique

par Claude Nowakowski

Tome 1 : Equations non linéaires, polynômes, calcul matriciel, interpolation, intégration et équations différentielles; pour chaque problème les différentes méthodes de calcul numérique sont étudiées. Ces algorithmes sont illustrés par un organigramme, un programme en Basic et un exemple d'exécution. Le Tome 2 donne : des algorithmes plus élaborés ou plus subtils et aborde de nouveaux thèmes : approximation des fonctions, problèmes aux limites, équations aux dérivées partielles. Chaque algorithme est programmé en Basic et en Pascal.

Tome 1 : 144 pages - 90,00 FF
Tome 2 : 184 pages - 110,00 FF
La disquette pour les 2 tomes : 195,00 FF

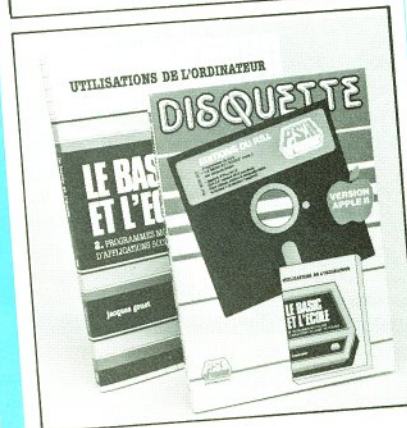


Le Basic et l'école - Tome 1

par Jacques Gouet

Le livre décrit un ensemble de programmes en Basic, destinés aux professeurs, aux parents et aux élèves. Grammaire, mathématiques, conjugaisons française, anglaise, sont autant d'exemples d'application qui font de cet ouvrage un véritable outil d'enseignement et d'initiation, à l'école ou à la maison.

Le livre : 192 pages - 120,00 FF
La disquette : 195,00 FF



Le Basic et l'école - Tome 2

par Jacques Gouet

Pourvu que vous possédiez un lecteur de disquette, vous allez pouvoir créer et gérer vous-même vos Questionnaires à Choix Multiples et vos exercices, suivre les notes obtenues, et concevoir des carnets de vocabulaire étranger, à l'école comme à la maison.

Le livre : 160 pages - 110,00 FF
La disquette : 195,00 FF



P.S.I. DIFFUSION
 BP 86 - 77402 Lagny-S/Marne Cedex
 FRANCE
 Téléphone (6) 006.44.35

P.S.I. BENELUX
 5, avenue de la Ferme Rose
 1180 Bruxelles
 BELGIQUE
 Téléphone (2) 345.08.50

P.S.I. SUISSE
 Case postale
 Route neuve 1
 1701 Fribourg
 SUISSE
 Tél. : (037) 23.18.28
 CCP 17.5684

au Canada
 SCE Inc
 65, avenue Hillside
 Montreal (Westmount)
 Quebec H3Z1W1
 Tél (514) 935 13 14

Table de conversions en Francs belges et Francs suisses	
90 FF = 695 FB - 26.40 FS	
100 FF = 770 FB - 31.50 FS	
110 FF = 850 FB - 34.60 FS	
120 FF = 925 FB - 37.60 FS	
130 FF = 1000 FB - 40.60 FS	
DISQUETTES	
195 FF = 1500 FB - 61.50 FS	

Envoyer ce bon accompagné de votre règlement à P.S.I. DIFFUSION ou, pour la Belgique et le Luxembourg à P.S.I. BENELUX ou, pour la Suisse à P.S.I. SUISSE.

Paiement par chèque joint Paiement en FF par carte bleue VISA (à P.S.I. DIFFUSION uniquement) paiement supérieur à 50 FF

N° _____ Date d'expiration _____

NOM _____ PRENOM _____

rue _____ n° _____

Code postal _____ Ville _____

IG 4

DESIGNATION	NOMBRE	PRIX
TOTAL		

par avion : ajouter 8 FF (75 FB) par livre

Signature (obligatoire pour paiement par carte de crédit)



AG 498

La famille Apple //

Daniel Garric

Une année, on change le manche. L'année suivante, la lame. Au bout de huit ans, c'est toujours le même couteau. Mais comme, entre temps, les techniques de fabrication du manche et de la lame se sont beaucoup améliorées, vous avez un couteau qui peut toujours faire les mêmes choses. Mais tellement mieux ! Et, ces techniques, elles n'ont pas fini de progresser. Alors, votre couteau, il va durer longtemps. Et puis, si deux ou plusieurs années de suite, vous ne pouvez ou ne voulez rien changer, quelle importance ? Vous continuez de vous en servir : quand vous mettez une lame neuve, elle est deux fois meilleure.

Le présent du //c

Voici donc l'Apple //c (pour "compact"), énième version de l'Apple I, ce grand ancêtre, né dans un garage de Californie il y aura un siècle et demi : en informatique, on divise le temps par 20; 100 ans = 5 années de la vie d'un humain. Les différences du micro avec "l'humain" sont d'ailleurs curieuses : plus il vieillit, plus il prend de la bouteille (pardon, de la puce) et plus il se dégraisse. Cet Apple //c ne pèse plus que 3,4 kilos, lecteur de disquettes incorporé compris.

S'il grandit, c'est par sa matière grise : 128 K-octets pour cet Apple-plume de 30 cms sur 30. Il n'en aura jamais davantage d'ailleurs puisque - grande nouveauté - il a perdu au passage ces slots qui faisaient la joie des bricoleurs. Faut-il pleurer ? Non. Puisque son grand frère //e les conserve. Et qu'il est tout de même bien agréable de prendre son ordinateur à la main, comme jadis papa son chapeau, pour aller passer le week-end ailleurs ou, tout simplement, le ramener le soir à la maison, après une journée studieuse sur les bancs de la fac. L'écran ? Bof. Pendant encore trois ou quatre mois, c'est vrai, il y aura un demi-problème : il vous faudra emporter un moniteur, à moins que vous ne trouviez sur place un poste de télé avec prise Péritel - puisque la carte Chat Mauve est intégrée (tiens, vous mettez les slots à la porte, ils reviennent se coller derrière); il vous suffit de brancher le fil du //c sur le récepteur. Cela veut dire également que les 80 colonnes sont là, d'origine. Un petit bouton bien visible, sur le capot couleur beurre frais, et vous passez à volonté en 40 colonnes. Et vice versa.

Vers la fin de l'année - ou peut-être avant -, les choses vont changer : vous pourrez lui mettre un couvercle, au //c. Un couvercle pas comme les autres puisque, dès que vous l'ouvrez, il se transformera en...écran plat à cristaux liquides, 80 et 40 colonnes. Or, le processeur du //c est bien un 6502 - mais avec un C, 65C02, le C de CMOS, bien sûr, cette technologie qui diminue fortement la consommation d'énergie. Ce 65C02 ajoute une vingtaine d'instructions aux 60 que possédait le 6502. D'où une mémoire plus rapide d'accès et une écriture de code plus rapide. Si donc vous ajoutez à la machine une batterie - d'origine, Apple-plume fonctionne sur 12 volts, avec un mini-transformateur - que se passe-t-il ? Vous avez deviné : vous voici en possession d'un ordinateur portable, un vrai, l'un des plus légers de la catégorie.

Intéressant, cette fois, pour les week-ends que vous pouvez transformer en pique-niques. Encore faut-il que le Multiplan sur lequel vous avez peiné toute la semaine veuille bien aligner ses chiffres dans une clairière. Et là, il y aurait comme une histoire de souris qui a déniché un fromage, lequel fromage n'est autre que l'emplacement traditionnel de la vidéo inversée de certains caractères secondaires. Car le //c y a droit, à sa souris. C'est bien, c'est pratique ! Mais il faut bien la mettre quelque part. Alors, chez Microsoft, on donne un dernier petit coup de patte à une version de Multiplan qui mettra sa vidéo inversée ailleurs. Ceci est vrai pour la série des PFS en particulier, de tous les programmes qui font un usage étendu de l'inverse en général. A des degrés divers : prenez Apple Writer. Oui, la ligne du haut, celle qui affiche la mémoire et les titres, grignotée par des caractères souris, ressemble à un morceau de gruyère sur le //c. Mais - c'est mon cas, j'aime les écrans bien vides - on peut toujours s'en échapper avec deux fois Esc. Alors se produit l'inattendu : disparaît l'un des rares inconvénients du //e, sa lenteur de frappe qui me supprime environ un caractère sur cinq. La relation entre le clavier de l'Apple-plume et sa mémoire est différente. La conséquence, c'est toute la rapidité de frappe retrouvée. D'autant plus surprenant, justement, que le clavier, d'apparence mi-chewing-gum mi-mécanique se révèle, à l'usage, plus mécanique que chewing-gum, très agréable au toucher.

Cela dit, tous les programmes de jeu de type Zaxxon et autres arcadiens s'accrochent à la perfection d'Apple-plume. Sans parler des jeux de rôle ("Sorcellerie" dans un château hanté, pour les amateurs de sensations, cela doit avoir son charme) et des jeux d'aventures. Essayez le dernier-né de chez Vifi-Nathan, "La Belle au Bois dormant", tiré de Charles Perrault par des américains, traduit en français par des canadiens, dans un grand parc du Berry sur fond de tours et de créneaux, et je suis sûr que les enfants apprécieront le réalisme du conte.

Notre //c, avec ses deux sorties série, modem et imprimante, par exemple (l'ImageWriter lui va comme un gant) et sa connexion pour un possible second drive, est un petit bijou. Mais il y a gros à parier que, dans l'échelle de l'évolution Apple, il ne soit tel qu'en lui-même; l'éternité l'a déjà changé. Ce qui est très bien ainsi. Puisque son frère aîné, le //e, est, lui, en pleine mue.

L'avenir du //e

La souris arrive. Par le biais de Mousepaint, d'abord : c'est un programme de dessin et de peinture (vous l'auriez deviné) furieusement tiré du Macintosh. Même avec les possibilités plus limitées de l'Apple //e, c'est une micro-révolution que de pouvoir utiliser un écran comme une feuille de papier. D'autant plus que les couleurs demeurent. Et que l'on n'a plus à manipuler de routines de sauvegarde ou d'affichage.

Mousepaint, c'est l'avant-garde de tout un bataillon d'applications que l'on pourra très bientôt mener à la baguette - pardon, à la souris.

Mais il y a aussi Jane, oeuvre et/ou chef d'oeuvre d'un petit génie de 20 ans presque de chez nous puisque, si Howard Marks - c'est son nom et son prénom - est né britannique, il a fait ses études en France avant de partir les poursuivre aux Etats-Unis. L'Apple, il connaît depuis toujours (pour lui). Il a donc tout démonté, dans la machine. Et tout reconstruit, le DOS en tête, en utilisant une carte à enficher dans un slot, l'assembleur et "un peu", dit-il, d'intelligence artificielle. La première version, sur les 64K du //e ou du II Plus, est intéressante : les commandes ont disparu, la souris se balade librement, les fenêtres s'éclairent, le tout-graphique

explose. Mais on est un peu à l'étroit dans la mémoire. Attendons la route ou la RAM 128K avant de se décider.

De toute façon, il y a le choix. L'Apple //e, pour les usages professionnels, est comblé. Du "soft", il lui en arrive de tous les côtés (voir l'article de Jean-Michel Gourévitch). Dont un nouveau langage, Modula II, version modulaire et bien plus intéressante du Pascal, mis au point par le génie helvète de Nicklaus Wirth soi-même, et distribué par Volition System, à San Diego. Ce Modula II, dit-on, sera le grand rival du C (non, ce n'est plus pour "compact", cette fois).

Le "hard" ne se fait pas prier non plus, d'ailleurs, avec les cartes comme celle d'Act-Logo, en France, qui vous autorise à animer quelque 32 sprites en couleurs sur votre écran. L'esprit est vraiment là, qui peut même déjà s'exprimer à haute et presque intelligible voix avec des synthétiseurs variés.

Drôle de "couteau" que cet Apple //, qui n'en finit pas de renaître avant même d'avoir rendu sa puce. Celle-ci finira bien par y passer, d'ailleurs. Mais pas avant d'avoir assuré, non

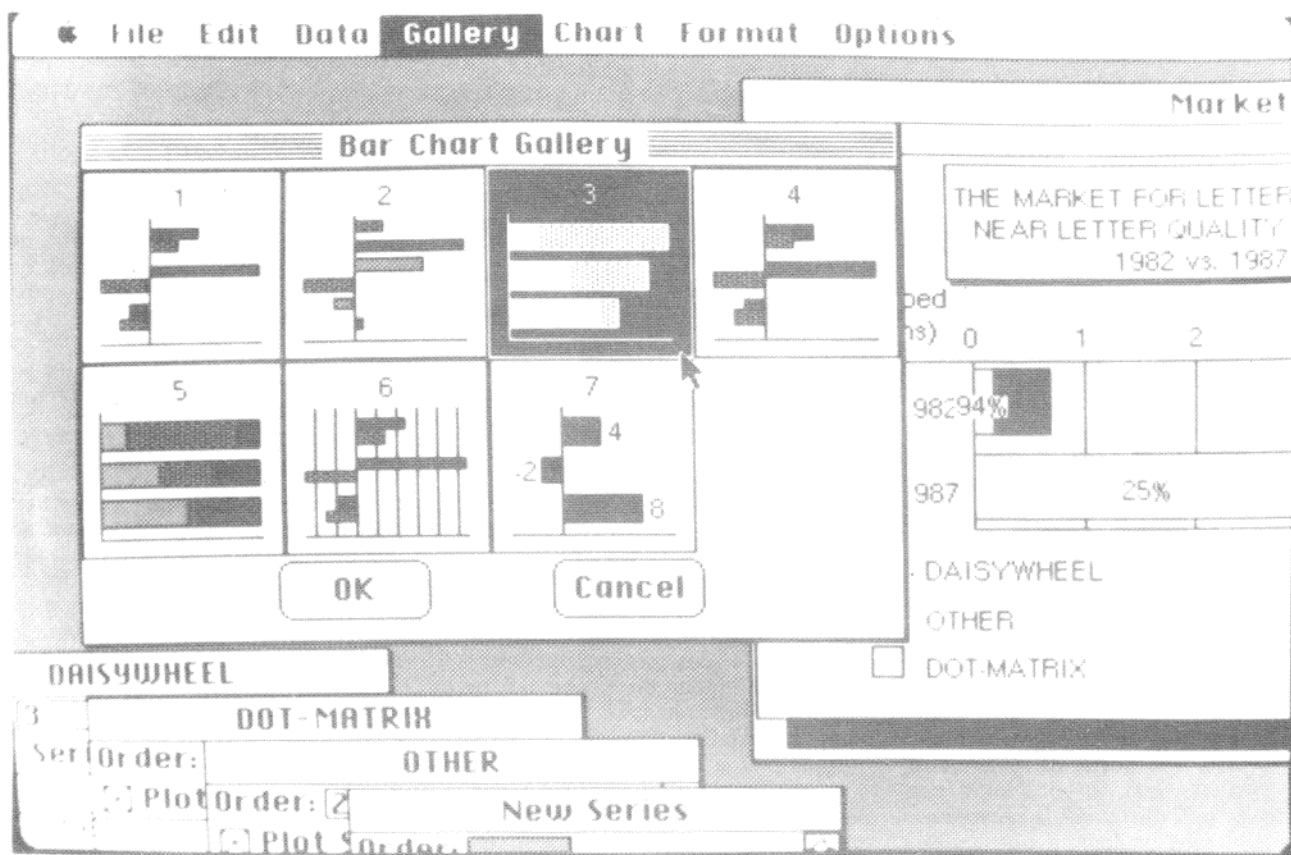
seulement sa postérité, mais aussi, et peut-être surtout, son passé, un passé de 16.000 programmes environ. Car il fallait bien qu'arrivât le processeur 16 bits entièrement compatible avec le 6502. Son matricule est quelque chose comme 65816 et il vient tout droit d'une petite boîte californienne spécialisée dans la technologie avancée. Qu'est-ce qu'Apple en fera, de son 16 bits ? Un Apple //x, comme on le murmure ? La "bande à Steve Jobs" l'avait jusqu'à présent plutôt snobé, le 16 bits, préférant passer directement du 8 au 32 avec le 68000 de Lisa et du Macintosh.

Mais justement : il y a eu rupture. Tandis que le 65816 assure la continuité dans le changement. La tentation est grande de prendre appui sur le passé pour se propulser en avant. D'autant plus que la stratégie d'Apple est d'une grande clarté : devenir, ou plutôt demeurer "l'Autre", un défi de Cinq lettres aux Trois lettres magiques, avec de grands pans d'une politique cohérente à ses 8 et 32 bits : tout sur la communication en série (adieu au parallèle), tout sur Pascal et ses modules (tant pis pour le Basic, lequel a d'ailleurs ten-

dance à partir dans tous les sens). Beaucoup sur le soft.

De là à vous dire ce que sera l'Apple // dans un an, je m'y risquerai d'autant moins que Steve Wozniak, l'autre "père" de l'Apple est entré en loge, comme on dit, quelque part à Cupertino, pour "travailler à quelque chose" dont rien ne transpire vraiment. Et ce n'est pas en regardant la concurrence que l'on en saura davantage puisque, par principe, Apple ne fait rien comme les autres.

Mais on peut être certain que, du côté de San Francisco, il y aura du nouveau, dans la taille de l'appareil lui-même, la capacité de sa mémoire, de ses gestions d'I/O (n'importe quoi). D'autant plus que l'intelligence artificielle commence à rôder dans les parages de la micro. Ce nouveau produit aura pour fonction essentielle de projeter très loin le passé vers l'avenir. En clair : il y a très peu de chances pour que vos Apple II Plus, //e ou //c deviennent stériles avant une bonne décennie. Le cercle de famille va, au contraire, s'élargir. Vers 1995, l'Apple II sera toujours là, plus tout à fait le même, pas vraiment un autre.



Microsoft Chart for Apple Macintosh

Exemple de Chart avec le Macintosh.

NLDR : tous les prix en francs indiqués dans cet article sont TTC. Chaque fois que l'adresse d'un fournisseur nous est connue, nous l'indiquons en fin d'article. Inutile par conséquent de nous appeler pour demander les coordonnées des autres fournisseurs.

On a sans doute enterré un peu vite le best-seller d'Apple et son "vieux" 6502. Voici que non seulement l'Apple se donne des airs de Macintosh (voir l'article de Daniel Garric), mais que les programmes, les interfaces, les périphériques et les gadgets arrivent comme mars en carême. Oh, certes, cette marée fait des jaloux, et suscite même de vilaines vocations de copieurs. Ces derniers prendront toutefois garde après la mésaventure de Franklin, un fabricant américain de simili-Apple : ayant peu de chances de gagner le procès que lui intentait Apple, il a préféré payer à la société de Steve Jobs la bagatelle de deux millions et demi de dollars, s'engager à ne plus copier les ROMs de Cupertino et même accepter d'Apple sur ses ordinateurs à venir... Un arrangement qui freinera peut-être d'autres vocations et renforcera la prééminence de l'Apple II, le seul, le vrai. Un autre évènement qui risque bien de conforter l'Apple, ses programmeurs et ses fabricants d'interface, c'est le décollage laborieux du petit dernier d'IBM, le PC junior. Tandis que ce dernier restait sur les étagères des revendeurs, Apple vendait pendant le seul mois de décembre 110.000 //e. Alors, si ses perfectionnements sont si nombreux, il ne faut pas, à coup sûr, y voir l'effet du hasard.

Car voici l'Apple accommodé à toutes les sauces : en plat de résistance, une nouvelle imprimante lui donne des capacités de traitement de texte bien augmentées; il peut maintenant recevoir le célèbre tableau-gestionnaire de fichiers "1-2-3" de Lotus; ou dater et estampiller de l'heure n'importe quel fichier, comme un pro. Un programme le transforme en "processeur d'idées", des mémoires virtuelles lui donnent des ailes. En guise de dessert, c'est l'utilisateur qui croit avoir des ailes, installé devant son écran-cockpit, aux commandes d'un Piper, grâce à l'un des meilleurs simulateurs de vol du marché. Et pour finir en musique, comme il se doit, un programme et une carte transforment l'Apple en



instrument musical. Et ce n'est qu'un début, l'année 1984 sera peut-être celle de Macintosh, mais à coup sûr aussi celle du //e.

Hard : place à l'Apple "pro"

Elle s'appelle "Imagewriter", débite les textes à 180 cps, au son d'un simple "crachouillement". C'est la nouvelle imprimante matricielle d'Apple. Bidirectionnelle, graphique, à traction et friction, elle utilise donc n'importe quelle forme de papier et même des liasses. Elle dispose de 7 alphabets (dont le français), de 8 jeux de caractères (avec notamment l'espacement proportionnel), d'un choix de taille des caractères, de possibilités de soulignement et d'impression en caractères gras. L'intéressant, c'est que l'on peut fixer un choix "habituel" (par défaut) grâce aux micro-interrupteurs de l'imprimante, mais que l'on peut aussi, grâce à des codes de contrôle, facilement modifier son choix en cours d'utilisation. Bien évidemment, l'Imagewriter permet d'imprimer les textes saisis avec Apple Writer II et Visicalc. L'imprimante coûte moins de 5000 francs. Il faut encore lui rajouter un kit d'accessoires vendu un peu moins de 300 francs pour disposer des deux manuels (d'ailleurs détaillés et bien conçus) et d'un câble. Attention, l'Imagewriter est prévue pour une interface série, et non plus parallèle, comme l'ancienne DOT Matrix.

Côté interface, on peut donc acquérir la carte super-série d'Apple. MID propose déjà une autre carte "made in France", munie de divers perfectionnements, dont notamment la copie d'écran haute résolution par commande directe : c'est la SG2E vendue un peu plus de 2000 francs (mais avec un câble). Aux Etats Unis, Practical Peripherals vient lui aussi de sortir une nouvelle carte série aux possibilités graphiques, disposant notamment de 27 commandes permettant le formatage du texte ou la copie d'écran. C'est la "SeriALL", vendue outre Atlantique 159 dollars (moins cher que la super-série d'Apple). En attendant la nouvelle imprimante laser qu'Apple pourrait commercialiser avant la fin de l'année.

En temps et en heure

C'est la nouvelle possibilité offerte par le ProDOS pour estampiller les fichiers. Voilà de quoi donner un regain de faveur aux cartes horloge. Les fabricants ne se privent d'ailleurs pas d'en tirer argument. Thunderware souligne lourdement qu'on mentionne sa "Thunderclock" dans le manuel du ProDOS, Practical Peripherals affirme que sa "Proclock" a été prévue pour tous les DOS, et Applied Engineering souligne que sa "Timemaster II" est "officiellement" ProDOS, en mettant l'accent sur le prix (129 dollars).

Toujours plus !

C'est de mémoire qu'il s'agit, ou plus exactement de mémoire virtuelle. Grâce aux pseudo-disques. Leur caractéristique, c'est bien sûr de ne pas en avoir. De quoi ? Mais, de disquette, bien sûr. La "Flashcard" de Syntex contient jusqu'à 288K de mémoire, et est livrée avec les disquettes pour le Dos 3.3, le CP/M ou le Pascal. Et en prime les disquettes de "Wordstar" (le tout pour 629 dollars). Un logiciel, le "Ramdrive //e" de Precision Software, permet d'utiliser toute carte 80 colonnes étendue comme pseudo-disque (30 dollars, mais l'intérêt de la chose est réduit car ce sera précisément un des avantages du ProDOS). Par contre, le "QuickLoader" de Southern California Research Group permet, lui, de charger immédiatement des programmes sans attendre le fonctionnement du lecteur de disquettes. Le miracle ? Les programmes ont préalablement été inscrits en PROM. Mieux, le DOS

est déjà livré sur une PROM de la carte, d'où la possibilité d'accès direct, par un simple CTRL-C Reset, à la fonction Catalog ou à d'autres commandes comme le lancement du HELLO. La carte est aussi livrée avec le COPYA. Plus besoin de l'entrer avant de recopier un programme. Le QuickLoader, d'une capacité de 256K, est vendu 180 dollars.

Plus vite. . .

Grâce au Spee Demon de M.C.T. Une carte accélérant l'Apple //e ou II Plus, qui travaille alors trois fois et demi plus vite. Il en existe d'autres. Celle-ci présente un "plus": son prix de 295 dollars. Qu'en adviendra-t-il si jamais on l'importe chez nous ?

Communiquer

Avec un modem bien sûr. Comme le Buzz Box vendu chez Micro Périph, lui aussi à un prix intéressant: 1100 francs. Le hic? Il n'a pas encore obtenu son homologation PTT. Cependant, il fonctionne aux normes françaises, à 300 bauds en full et half duplex, et est homologué en Grande Bretagne.

Musique

Transformer l'Apple en instrument de musique? Avec la carte Mockingboard importée par BIP, l'Apple génère des sons en stéréophonie, avec possibilité de jouer simultanément 6 notes (avec un programme comme Music Construction Set). Et de nombreux jeux (plus de quarante sociétés de logiciel l'ont adoptée) sont étudiés pour la Mockingboard, comme Zaxxon, Exodus, Maze Craze, etc. Attention, les prix de la Mockingboard peuvent varier de 1500 à 2400 francs selon les revendeurs. On la trouve notamment à la Fnac, à Micrishop et chez Sideg.

Logiciels

De la mémoire et du caractère

Videx vient de sortir Ultraplan, un tableur capable de tirer profit de nombreuses cartes d'extension mémoire (jusqu'à 512K), ainsi que de la carte Videx, pour obtenir des écrans ayant jusqu'à 48 lignes et 128 colonnes, avec des caractères très lisibles et différents niveaux lumineux pour mettre en relief telle ou telle partie du tableau.

Un plan de travail électronique

C'est Think Tank. Un logiciel américain dont on n'a pas fini de parler. Aux Etats-Unis, il fait fureur. Le magazine A+, qui publie un banc d'essai de Think Tank dans son numéro de mars, cite certains de ses utilisateurs:

Steve Wozniak, d'Apple, un avocat, le vice-président de la Chase Manhattan Bank, entre autres. Concrètement, Think Tank est un outil idéal pour mettre en forme ses idées. On y entre des titres, des sous-titres, des paragraphes, qu'on peut classer. Les idées brutes se structurent, s'organisent pour former un plan cohérent. Bien pratique pour établir le plan d'un article, les idées force d'un mémo. Un outil diablement utile à tous les professionnels de l'écriture. Mais les utilisations sont illimitées. Wozniak s'en est servi pour faire le planning d'un festival de rock, le banquier pour classer les notes prises au cours de réunions. Le programme peut même classer automatiquement par ordre alphabétique les sous-chapitres, numéroter les paragraphes, etc. On peut ensuite imprimer sous diverses formes le résultat de son travail. Ce programme écrit par Living Videotex nécessite deux lecteurs et un Apple II pour fonctionner. Son prix outre Atlantique est de 150 dollars.

Lotus, es-tu là ?

A quand Lotus 1-2-3 sur Apple, demandait Pom's dans son dernier numéro? Le voici! Il tournera sur un Apple II Plus ou //e, à condition d'utiliser le lecteur de disquettes Rana 8060/2 (incorporant une carte MS-DOS et vendu, hélas, pratiquement le prix d'un IBM PC...). Prix de Lotus: 495 dollars. Alléluia ?

Des fenêtres

En attendant, les amateurs de programmes intégrés pourront se contenter de Jane (voir l'article de Daniel Garric). Jane est vendu en France par Ordinateur Express pour moins de 4.000 francs avec carte et souris. Il est en français: il a même été conçu dans notre langue. Cocorico! Aux Etats-Unis, un autre programme intégré commence à faire parler de lui, c'est Jack 2 (la nouvelle version d'Incredible Jack) de Business Solutions, permettant de faire du traitement de texte, des calculs ou des graphiques, sur la même page et sans changer de disquette. Avec impression de ce qu'on voit sur l'écran. Jack 2 n'a pas de fenêtres, mais dispose de dessins d'enveloppes qu'on peut ouvrir (50 dossiers simultanés maximum). A voir.

Imprimer Visicalc en travers ?

C'est possible avec un logiciel sud-africain, Sidewise, disponible par l'intermédiaire de Pace Software Supplies en Grande Bretagne. Utile pour les tableaux larges, qui dépassent la plus grande largeur des imprimantes.

NDLR: nous rappelons à cet égard que le programme Multiplan du Ma-

cintosh possède d'origine la possibilité d'imprimer un tableau en travers si on le souhaite.

Vol au dessus d'un nid d'Apple

Après une bonne séance de travail, décollez aux commandes de votre Apple. Avec Flight Simulator II de SubLogic (en vente notamment chez Sideg). On n'avait encore pas volé ainsi sur Apple. En couleurs. D'un bout à l'autre des Etats-Unis (le programme permet d'utiliser une soixantaine d'aéroports, dont les pistes sont reproduites le plus fidèlement possible, d'après des cartes aériennes). On peut définir soi-même les conditions de vol (heure, saison, nuages, force du vent), voler de jour ou de nuit dans un cockpit parfaitement équipé: VOR pour la radio-navigation, ILS pour les atterrissages aux instruments, instruments numériques, horizon artificiel en trois couleurs, etc. Le programme est livré avec un manuel de 88 pages et quatre cartes de radio-navigation. Un simulateur subtil dont l'utilisation nécessite un doigté certain (Bruce Artwick, son auteur, essaierait actuellement de le faire certifier par la FAA, l'autorité aéronautique américaine). Que ceux qui ont réussi à poser leur Piper, sur la piste et en bon état, m'écrivent. Je ne suis pas encore parvenu à éviter le crash en finale...

Adresses

Apple Seedrin - av. de l'Océanie - ZA de Courtaboeuf - B. P. 131 - 91944 Les Ulis Cedex - Tél (6) 928 0139

Applied Engineering - P. O. Box 798 - Carrollton TX 75006 - USA

Business Solutions - 60 East Main Street - Kings Park - NY 11754 - USA

Living Videotext Inc. - 450 San Antonio Road no. 56 - Palo Alto CA 94306 - USA

MID - 96 bd. Richard Lenoir - 75011 Paris - Tél 357.82.30

Micro Periph - 62 rue Ducouédic - 75014 Paris - Tél 321.53.16

Ordinateur Express - 3 rue de Pelouze - 75008 Paris - Tél 522.15.15

Pace Software Supplies - 92 New Cross Street - West Bowling - Bradford BD58BS - UK

Practical Peripherals - 131245 La Baya Dr. - Westlake Village - CA 91362 - USA

Sideg - 170 rue Saint Charles - 75015 Paris - Tél 557.79.12

Southern California Research Group - P. O. Box 2231 - S Goleta CA 93118 - USA

Synetix - 10635 NE 38th Place - Kirkland WA 98033 - USA

Thunderware Inc. - 44 Hermosa Av. - Oakland CA 94618 - USA

Courrier des lecteurs

Alexandre Duback

Je me suis servi du logiciel Omnis (gestion de fichiers) une dizaine de fois de manière satisfaisante. Hier, je le remets en route et voilà que l'Apple devient littéralement "fou". Il n'y a plus d'affichage possible, des caractères désordonnés et inversés apparaissant à l'écran. Et plus aucun logiciel ne fonctionne. Ceci m'étonne d'autant plus que, lors du dernier emploi, tout était parfaitement normal. Qu'en pensez-vous ? Pourriez-vous me dire ce qui se passe ?

Ayant mis en route l'autotest de l'Apple //e, j'ai vu apparaître à l'écran "Kernel OK" puis, l'ayant recommencé, est apparu "RAM" suivi de quelque chose d'incompréhensible, puis "ROM".

Jean-Albert Jouve - BP 4169 - Tananarive - Madagascar

Après des essais intensifs sur Apple //e d'Omnis dans toutes ses options, il nous a été impossible de provoquer une panne analogue à celle que vous décrivez dans votre lettre. Nous ne pouvons donc faire que des hypothèses.

Ce genre de panne se produit parfois sous le système d'exploitation CP/M à la suite de fausses manoeuvres, en particulier quand on tente de "booter à chaud" (par CTRL-C) à partir d'une disquette ne contenant pas le système. Bien qu'Omnis soit écrit en Pascal, rien n'interdit de penser qu'une manoeuvre anormale avec les disquettes ait pu être la cause de vos ennuis.

Une autre cause possible réside dans l'alimentation électrique de l'ordinateur. Il arrive que des micro-coupures extrêmement brèves ne fassent pas décrocher complètement et produisent des résultats surprenants. Il en est de même avec certaines brusques sautes de tension. Si vous vous trouvez dans des conditions d'alimentation électrique instables, un régulateur (ondulateur) pourrait PEUT-ETRE améliorer la situation. Même problème avec les décharges d'électricité statique. Bannissez toute moquette au sol, et essayez d'avoir une bonne terre dans votre installation électrique.

Enfin, il est arrivé également, y compris sous DOS, qu'une température de fonctionnement exagérée ait des répercussions bizarres. Ceci peut se produire, outre le cas d'une température extérieure élevée, quand de nombreuses cartes sont installées simultanément. Certaines cartes d'in-

terface imprimante, 80 colonnes ou MEM/DOS, ont en particulier tendance à se conduire en radiateurs. Vous pouvez fonctionner capot ouvert (attention, vous supprimez alors l'effet de cage de Faraday du //e, gare à l'électricité statique), ou installer un petit ventilateur (il y en a de spécialement étudiés).

Et puis...sauvegardez souvent une copie de vos fichiers...

Contrairement à la réponse d'Olivier Herz au Docteur Selosse (Pom's 11 page 69), il est assez simple d'augmenter le nombre de titres du catalogue, au-delà de 105. J'ai d'ailleurs réalisé une disquette préparée pour recevoir 217 titres en face 1 et 161 titres en face 2. Elle est facilement copiable par COPYA, et les fichiers sont transférables par FID sans problème. Les lecteurs de Pom's qui le souhaitent pourront en recevoir, après envoi d'un chèque de 45 francs à l'adresse indiquée en fin de mon courrier, une copie sur disquette Maxell MD1-D.

Voici comment il faut s'y prendre pour constituer une disquette à catalogue étendu. Prendre tout d'abord un utilitaire permettant la lecture et l'édition directe de pistes/secteurs sur la disquette. Le catalogue est normalement rangé de \$11-0F (piste 17, secteur 15) à \$11-01. Au début de chaque secteur se trouve l'adresse du secteur suivant : on lit ainsi "00 11 0C" au début de \$11-0D, ce qui signifie que la suite du catalogue est en \$11-0C.

Si l'on entre "00 10 0F" au début de \$11-01, on indique au DOS que le catalogue se poursuit en piste 16, secteur 15. Il suffit de continuer ainsi autant que nécessaire. Bien entendu, après chaque modification, il ne faut pas oublier de recopier le secteur modifié sur la disquette. Il reste enfin à modifier la table d'occupation, pour indiquer au DOS que la piste \$10 est occupée et éviter ainsi la destruction du catalogue lors de la sauvegarde de fichiers : mise à 0 des octets en position \$78 à \$7D de la piste \$11, secteur \$00.

A l'inverse, on peut réduire la taille du catalogue pour sauver des programmes importants. Cela permet de récupérer 13 ou 14 secteurs par disquette, soit plus de 3K. Dès lors, après libération des pistes \$00 à \$02 (si on ne met pas le DOS sur la disquette), une disquette peut contenir 140K de fichiers.

S. Thélér - SODEC - 1 rue de la Lys - 59253 La Gorgue

Merci pour cette lettre intéressante. Permettez-moi par la même occasion de blanchir Olivier : j'avais complété le Courrier après son départ, et laissé son nom car il en avait traité la majeure partie.

1) Je tiens à faire profiter vos lecteurs d'une information que j'ai mis trois mois à obtenir. Ayant la carte Chat Mauve et une imprimante Epson, j'ai eu la désagréable surprise de constater que les logiciels écrits en Pascal (Omnis et Quickfile, et Pascal lui-même) refusent obstinément de reconnaître l'imprimante connectée au slot 1. Quelqu'un m'a finalement donné la solution : il suffit tout bêtement d'enlever la puce située juste au-dessus du "T" de CHAT MAUVE (il s'agit d'un buffer), et ça marche ! Le plus extraordinaire dans cette histoire est que l'absence dudit buffer ne change apparemment rien dans le fonctionnement de la carte, aussi bien pour ses fonctions graphiques que pour l'extension 64K. Apple et le Chat Mauve n'ont, paraît-il, pas encore trouvé l'explication.

2) Je tiens à vous féliciter par ailleurs pour la qualité de votre revue et le haut niveau des programmes que vous proposez. J'ai cependant eu bien des difficultés avec certains d'entre eux : je n'ai en particulier jamais réussi à faire fonctionner correctement le programme de copie écran texte HCT.OBJ (sur les disquettes à partir de la 7). Il n'imprime qu'une ligne sur 3, même si j'inhibe l'impression écran par un POKE 1657,80.

3) Une suggestion : j'aimerais des programmes utilisant les possibilités des cartes 80 colonnes (par exemple pour la gestion de masques, les menus à barre, les catalogues). C'est vraiment déprimant de revenir à ce vieil affichage étriqué en 40 colonnes après avoir travaillé sur Visicalc ou Apple Writer en 80 colonnes !

Jean-Pierre Januel - 5 rue Duperré - 75009 Paris

1) Nous vous remercions pour votre lettre au sujet de la carte RVB Chat Mauve. Nous avions en effet parlé du problème dans le Courrier du Pom's 10, page 72.

2) Le programme de copie d'écran texte ne fonctionne hélas pas avec

tous les modèles de cartes parallèles et d'imprimantes. Nous n'avons pas encore trouvé le moyen de le faire fonctionner de façon universelle.

3) En ce qui concerne les cartes 80 colonnes, on retombe immédiatement sur le problème suivant : elles utilisent en général des codes différents. On ne pourrait donc proposer des programmes que pour la 80 colonnes Apple //e ou Chat Mauve, qui sont en théorie gérées de la même façon, ce qui semble d'ailleurs peu évident vu les problèmes du Pascal avec la Chat Mauve et une Epson. Nous devons déjà tester les programmes pour voir à chaque fois s'ils marchent sur II Plus et //e ! Si l'on devait tester avec des cartes 80 colonnes ou de multiples configurations d'imprimante et de carte, on ne s'en sortirait plus.

Pom's paraît dorénavant tous les deux mois. Bravo ! Peut-on espérer une parution mensuelle ? Est-ce un manque d'articles ou une diffusion trop restreinte, ou encore la probable concurrence du "petit" frère Macintosh ou enfin la suprématie d'IBM qui vous empêchera de franchir le pas ?

La lecture de POKEs à gogo (Pom's 11) m'a fait découvrir quelques im-

perfections qui m'avaient échappé. Mea culpa. Je rectifie le tir.

*Adresse 48 : faire un POKE 48,C*17 avec C = code de la couleur GR (0-15).*

Adresses 55/56 et 57/58 : le DOS remet automatiquement les anciennes valeurs. Il vaut mieux poker dans les adresses d'entrée/sortie du DOS (43603-4 et 43605-6).

Adresses 60/67 : avant l'appel de MOVE ou de VERIFY, le registre Y doit être mis à zéro. Pour réaliser ceci, faire précéder la série de POKEs par un CALL -610.

*L'adresse du tampon utilisé par RWTS est donnée par POKE IOB + 9, A/256 : POKE IOB + 8, A - 256 * PEEK(IOB+9).*

*Les POKEs qui permettent d'activer ou de désactiver une carte d'extension ne sont valables que si la carte est dans le port (slot) 0. De façon plus générale, il faudra faire POKE XXXXX + 16*S pour une carte insérée dans le slot S.*

Page 18 du Pom's 11 : corriger les POKEs permettant d'activer les 12K du bloc 1 :

POKE 49288,0 (-16248)

POKE 49289,0 (-16247)

POKE 49290,0 (-16246)

POKE 49291,0 (-16245)

Roland Jost - 4 allée Spach - 67000 Strasbourg

Erratum

Dans un souci de présentation, nous avons ajouté au programme de Gilles Mauffrey (Pom's 11, page 71) des REMs (1 à 3), oubliant que le programme ne marcherait alors plus. Il faut donc éliminer ces remarques, qui d'ailleurs ne figurent pas dans le programme de la disquette d'accompagnement du numéro 11.

De nouveaux développements attendent les lecteurs intéressés par la méthode « Aliénor » parue dans le Pom's 11. Les personnes concernées peuvent contacter A. Guillez au 128 avenue de la République - 75011 Paris - Tél Labo MEDIMAT 329.79.26.

Multiplan pour Apple

Une légère confusion au moment de l'impression du livre « Multiplan pour Apple II Plus et Apple //e » a fait que le premier tirage comporte, en page 5, un Sommaire manifestement trop sommaire (une page seulement). Les acheteurs de cette première édition peuvent envoyer à Pom's une lettre comportant une enveloppe timbrée à leur adresse et un mot demandant le sommaire détaillé : ils recevront gratuitement en retour de courrier ce sommaire en quatre pages.

LILLE..... LILLE..... LILLE..... LILLE

m.b.d.c. Apple //e, Apple ///, Mac-Intoch,
BFM 186. dragon, oric, alicé, casio

m.b.d.c. disquettes, Flexettes, 3M, Verbatim
listing, rubans encreurs.

m.b.d.c. le conseil, le matériel, les périphériques
les logiciels, les consommables,

m.b.d.c. le S.A.V. sur place.
le Service Complet.



m.b.d.c.

172, RUE SOLFERINO. 59800 LILLE — TEL. (20) 57.91.87
OUVERT DU MARDI AU SAMEDI DE 9h30 à 12h ET DE 14h30 à 19h

Courrier des clubs

Depuis que nous avons accès à un Apple, nous souffrons dès que nous en sommes éloignés. Pensant que cette étrange machine a fait d'autres victimes, nous venons de fonder un club pour les rassembler. Il est ouvert à tous, du débutant au docteur en informatique.

CTRL RESET - Club Informatique - 172 Bd. Haussmann - 75008 Paris
Richard Chalet - Secrétaire Général - 20 rue Nélaton - 75015 Paris

Nous avons créé depuis quelques mois l'association Info-Maniaques. Son but est d'abord de fournir un lieu de réunion et d'échanges à des

personnes qui ont en commun d'aborder ou d'approfondir les divers domaines de la micro-informatique. Très rapidement, des personnes de tous âges souhaitant recevoir une initiation sont venues nous voir. Nous comptons actuellement 60 adhérents et chaque mois une dizaine de nouvelles adhésions.

L'association met à la disposition de ses adhérents divers matériels : Apple, Multitech MPF II, Jupiter Ace, Alice, Thomson TO7.

Info-Maniaques - 62 avenue Paul Santy - 69008 Lyon - Tel (7) 801.43.19

Nos prochaines réunions sont les suivantes : "le point sur les tableurs", animé par Hervé Thiriez le 16 mai avec présentation de Multiplan et Multichart sur Macintosh; "les techniques graphiques" avec démonstration de vision en relief, animation par sprites, découpage d'écran et incrustations, carte haute résolution 1024*1024, le 20 juin. Les réunions ont lieu à 19H au 110 avenue du Général Leclerc, à Paris.

Ma Pomme - 6 rue Paul Saunière - 75016 Paris - Jean-François Duvivier (1) 558.05.78 le soir.



Bibliographie

The Apple House (how to computerize your home using your Apple Computer) de John Blankenship, Prentice Hall - 66 Wood Lane End, Hemel Hempstead, Herts, HP2 4RG, Royaume Uni - 153 pages - \$19,45.

Voici un livre qui décrit comment concevoir un système qui fasse exécuter à votre Apple un ensemble de tâches liées à l'habitation telles que le contrôle de la sécurité, des lumières ou du téléphone. Une des originalités de ce système est qu'il accepte des ordres verbaux et réagit en fournissant une réponse du même type. Mais une certaine connaissance de l'électronique et l'achat de composants sont nécessaires à la réalisation de tout ou partie du système. Cet ouvrage est très bien structuré et comporte en appendice les listings de tous les programmes. Il est possible aussi d'obtenir une disquette les contenant au prix de \$12,50.

Jean-Luc Boyer

Exercices pour Apple II, de Frédéric Lévy, Editions du PSI - 148 Pages - 90 FF.

Il est agréable de voir que le listage de programmes n'est pas le seul but de ce livre. Bien poser le problème et réaliser une analyse sont considérées comme des étapes importantes et mises en valeur par l'auteur. De même, on apprécie beaucoup les programmes commentés d'une façon précise. Cet ouvrage comprend une première partie donnant les énoncés des exercices ainsi que leur analyse, la deuxième partie étant réservée aux programmes proposés comme solutions (en Basic). Les problèmes abor-

dés correspondent à quatre thèmes principaux (calcul, manipulation de tableaux, traitement de chaînes et graphiques). En conclusion, ce livre apporte une excellente formation à la programmation.

Jean-Luc Boyer

36 programmes Apple II pour tous, de Jacques Boisgontier, Editions du PSI - 126 pages - 90 FF.

Après un bref rappel des instructions graphiques spécifiques à l'Apple II, l'auteur fournit un ensemble de programmes (en Basic) sur 3 thèmes différents : l'éducation, la gestion et les jeux. Nous regrettons le manque d'explications pour certains programmes, ainsi que le peu d'originalité des sujets traités, que l'on retrouve dans beaucoup d'ouvrages déjà parus. Malgré cela, il y a une bonne diversité dans le choix des énoncés.

Jean-Luc Boyer

Apple Graphics - Activities Handbook de Harold J. Bailey et J. Edward Kerlin - Prentice Hall International - 422 pages - \$ 11.95.

Ce livre est agréablement présenté, et sa pédagogie plaisante : tests de compréhension, exercices, discussions, exemples. Il contient de très nombreux exemples de programmes. Nombre d'entre eux sont de bonne taille (allant jusqu'à plusieurs centaines d'instructions). Le revers de la médaille est que, lors de la retranscription au clavier, il est presque impossible de ne pas commettre quelques erreurs de frappe. Au niveau d'un apprentissage, c'est d'autant plus réhébitorique que les résultats à

obtenir ne sont en général pas décrits : il s'agit pour le lecteur d'exercer sa capacité de compréhension. Moralité : la disquette "optionnelle" (à \$26) est de fait quasiment obligatoire.

D'une façon plus générale, pour un ouvrage consacré au graphique, celui-ci manque nettement d'illustrations. On peut aussi regretter un manque de distinction entre l'essentiel et l'accessoire. Deux exemples à l'appui de cette affirmation :

- les graphiques dits "alphanumériques", créés en basse résolution à partir de chaînes de caractères, ne sont sans doute pas d'usage quotidien; c'est une "activité" parmi d'autres d'un "module" basse résolution;
- à l'inverse, il est d'usage courant d'utiliser les "shapes" en graphique haute résolution; or ce point n'est traité que dans un chapitre sur les techniques avancées.

Mais l'ensemble est intéressant et l'on trouvera dans cet ouvrage de quoi satisfaire le débutant ainsi que le programmeur expérimenté qui souhaite encore progresser. Environ 100 pages s'adressent au débutant. Les 100 pages suivantes abordent les techniques avancées, cependant que les 200 dernières pages sont consacrées aux problèmes de géométrie plane, puis dans l'espace (c'est peut-être un peu trop). Quelques annexes, dont on peut fort bien se passer, complètent l'ouvrage.

Guy Mathieu



pom's

	Montant TTC
• la disquette HAIFA Source <input type="checkbox"/> au prix de 55 F la disquette (cf. Pom's n° 5)	
• le logiciel H-BASIC <input type="checkbox"/> au prix de 150 F (cf. Pom's n° 8)	
• le logiciel MUSIC <input type="checkbox"/> au prix de 80 F (cf. Pom's n° 10)	
• le Disk Manager <input type="checkbox"/> au prix de 450 F (cf. Pom's n° 11)	
• DBSTAG (CP/M) <input type="checkbox"/> au prix de 450 F (cf. Pom's n° 11)	
• Disquette de jeux A <input type="checkbox"/> au prix de 80 F (cf. Pom's n° 12)	
• Disquette de jeux B <input type="checkbox"/> au prix de 80 F (cf. Pom's n° 12)	
• Recueil n° 1 de Pom's (n° 1 à 4) <input type="checkbox"/> avec ses 3 disquettes au prix de 280 F _____ <input type="checkbox"/> sans disquette au prix de 130 F _____ <input type="checkbox"/> les 3 disquettes seules au prix de 150 F _____	
• Recueil n° 2 de Pom's (n° 5 à 8) - Parution 15 avril <input type="checkbox"/> avec ses 4 disquettes au prix de 320 F _____ <input type="checkbox"/> sans disquette au prix de 130 F _____ <input type="checkbox"/> les 4 disquettes seules au prix de 190 F _____	
Je désire recevoir :	
• les numéros de la revue Pom's <input type="checkbox"/> 4 <input type="checkbox"/> 7 <input type="checkbox"/> 8 au prix de 35 F le numéro _____	
• les numéros de la revue Pom's <input type="checkbox"/> 9 <input type="checkbox"/> 10 <input type="checkbox"/> 11 <input type="checkbox"/> 12 au prix de 40 F le numéro _____	
• les disquettes d'accompagnement des numéros <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/> 6 <input type="checkbox"/> 7 <input type="checkbox"/> 8 <input type="checkbox"/> 9 <input type="checkbox"/> 10 <input type="checkbox"/> 11 <input type="checkbox"/> 12 au prix de 55 F par disquette _____	
Je désire m'abonner pour 6 numéros à partir du n°	
<input type="checkbox"/> sans disquette au prix de 200 F _____	
<input type="checkbox"/> avec disquettes au prix de 480 F _____	

TOTAL :

Envoyez ce bon de commande et votre règlement à :

Éditions MEV — 64-70, rue des Chantiers — 78000 Versailles

Nom _____

Adresse _____

Ces tarifs comprennent l'envoi postal en France Métropolitaine, CEE et Suisse (voie aérienne exceptée)
 Supplément avion : 10 F par numéro et / ou disquette

Souris créative cherche amitié passionnée pour fonder club.

Pour tous les passionnés de l'ordinateur personnel,
les branchés et ceux qui le sont moins, Apple lance le Club Apple.
Enfin un club avec des idées et des services
pour comprendre, pour gagner, pour s'évader.
Le Club Apple, c'est l'esprit Apple.
C'est le fruit de la passion.



PROMO 2000



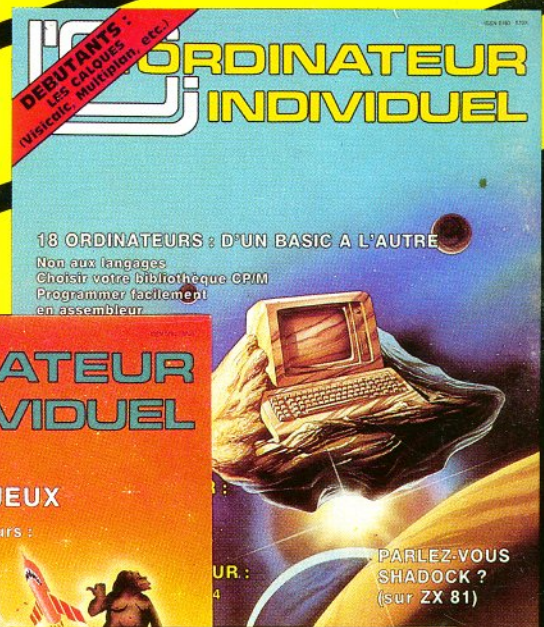
Pour en savoir plus sur le Club Apple - le fruit de la passion - et connaître tous les avantages que nous réservons aux membres du club, découpez dès aujourd'hui le bon et retournez-le à Club Apple, avenue de l'Océanie - ZA de Courtabœuf - BP 131, Les Ulis cedex 91944. Vous recevrez sans engagement de votre part toutes les informations pour devenir membre du club.

Nom _____ Prénom _____

Adresse _____

Code postal _____

**POUR
MIEUX CHOISIR
VOTRE ORDINATEUR
ET POUR MIEUX
L'UTILISER :**



Vous y trouverez :

- l'actualité et les tendances de l'informatique individuelle
- les bancs d'essais 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

22 FF chez votre marchand de journaux

L'ORDINATEUR INDIVIDUEL

La Référence