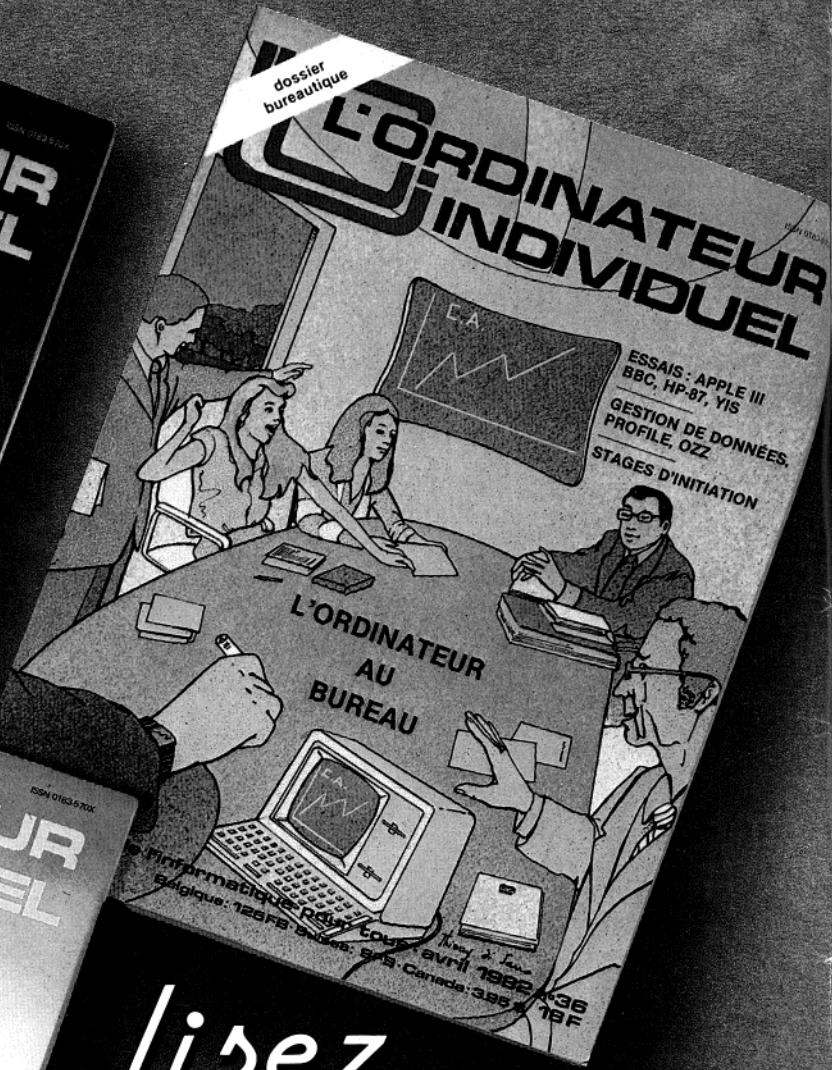


P O M M S



**NUMERO 8
JUN 1983
35 F**

pour mieux choisir votre ordinateur et pour mieux l'utiliser:



lisez

L'ORDINATEUR INDIVIDUEL

Vous y trouverez:

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

L'ORDINATEUR INDIVIDUEL, chez votre marchand de journaux

pom's n° 8

Sommaire	Page	Langage *	Niveau **
Editorial par Hervé Thiriez	5		
Mini-base de données par Yves Volpi	7	B	T
Calculs en format gestion par Gérard Michel	13	A	M-T
Bibliographie par Alexandre Duback	14	/	T
Recherche de codes binaires par Guido Bettiol	15	B-A	P
PILOT et SuperPILOT à l'essai par Olivier Herz	17	/	T
Accélérez vos programmes en BASIC par Hervé Thiriez	19	(B)	T
Banc d'essai du Basis 108 par Hervé Thiriez	19	/	T
Création de polices de caractères par Guy Mathieu	21	B	M-T
Allo, Questel par Pom's	24	/	T
Calcul à 12 chiffres en Pascal par Michel Crimont	25	P	M
Deux ans de Pom's : sommaire par Alexandre Duback	33	/	T
Conseils aux débutants par Alexandre Duback	34	/	T
Le H-BASIC : un BASIC Pascalien par Olivier Herz	35	B-P	M-T
Moniteur étendu par Thierry Le Tallec	39	A	M-P
Notions de base : gestion de fichiers par Gérard Michel	51	B	T
Francisez le DOS par Philippe Krepper	55	B	M-T
Saisie de réels en Pascal par Philippe Chemin	57	P	M
Micro-informations par Jean-François Duvivier	59	/	T
Courrier des lecteurs par Olivier Herz	63	/	T
Courrier des clubs	65	/	T
Trucs et astuces	20-37-61	/	T

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

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

Il s'agit là du niveau nécessaire pour comprendre le fonctionnement du programme décrit dans l'article, ou pour digérer l'article lui-même. Ceci dit, un programme peut être utilisé sans compréhension de son fonctionnement interne (sinon, qui jouerait aux échecs sur ordinateur ?)

Les annonceurs

ALTI : p. 9 - B.F.I. : p. 6 - BIP : p. 62 - CEMIA : p. 41 - DIF : p. 4 - EUROTRON : p. 61 - ANDRE F. FINOT : p. 22 - IEF : p. 56 - JCR : p. 68 - KA : p. 16 - M.B.D.C. : p. 37 - MICRO-PERIPH : p. 67 - MILOG : p. 54 - MINIGRAPHE : p. 62 - MICRO INFORMATIQUE SERVICE : p. 12 - L'ORDINATEUR INDIVIDUEL : p. 2 - P.S.I. : p. 58 - PFS : p. 50 - SAARI : p. 38 - SACASA : p. 11 - SOFITEC : p. 32 - TOTALE FORMATION : p. 65.

Éditions MEV - 49, rue Lamartine - 78000 Versailles

Directeur de la publication : Hervé Thiriez. Imprimerie Rosay, 94300 Vincennes. Imprimé en France. Dépôt légal : 2^e trimestre 1983.



ETUDES ET
REALISATIONS
INFORMATIQUES

DIF ELECTRONIC

LE NOUVEL OUTIL DE LA COMPETITIVITE.



MATERIEL ET LOGICIELS
GARANTIS UN AN
AVEC INSTALLATION

- **Comptabilité**
6 et 8 chiffres
interfacée avec
VISICALC - EDICALC
- **Facturation**
- **Stock**
- **Paye**

- **VISICALC version avancée**
- **Traitement de texte**
- **Gestion de fichier**
- **Programme spécifique pour
statistique et analyses**



PARIS 28, rue Miollis 75015 Paris Tél. : (1) 566.68.38

CAMBRAI 6, rue des Ratelots 59400 Cambrai Tél. : (27) 81.03.24

BOULOGNE S/MER 71, rue du Camp de Droite 62200 Boulogne S/Mer
Tél. : (21) 30.79.46



1979-1983 4 ANS D'EXPERIENCE AU SERVICE DES ENTREPRISES

DIF ELECTRONIC

Editorial

Longtemps après les Etats-Unis et nos voisins d'Outre Manche, nous voyons enfin arriver sur le Continent les matériels compatibles Apple. Nous avons réalisé pour nos lecteurs un premier test du Basis 108 dans ce numéro; le numéro de Septembre verra, dans la même veine, un banc d'essai du Golem, apparu encore plus récemment.

Chez Apple, l'existence du McIntosh est maintenant confirmée : nous pouvons espérer voir ce petit frère du Lisa débarquer en France dans moins d'un an. A quel prix ? De plus, à un horizon pareillement imprécisé et à un prix tout aussi inconnu, nous attendons la version portable de l'Apple II ou //e.

Ce numéro est encore très riche et, comme les précédents, essaie de réaliser la quadrature du cercle : fournir aux débutants de nombreux articles à leur niveau, tout en proposant beaucoup de matière aux utilisateurs de niveaux intermédiaire et professionnel.

Ceux qui désirent travailler sur leur micro avec des grands nombres pourront choisir entre le BASIC et le Pascal, avec les articles de **Philippe Chemin**, **Michel Crimont** et **Gérard Michel**.

Les débutants et les nouveaux venus trouveront des renseignements utiles dans le sommaire thématique des huit premiers numéros de Pom's et les conseils aux débutants d'**Alexandre Duback**, ainsi que dans mon article sur l'accélération des programmes en BASIC. Ils pourront aussi analyser et utiliser la mini-base de données que nous a proposée **Yves Volpi**. Enfin, **Gérard Michel** consacre un nouvel article à la gestion de fichiers, dans la série fort appréciée des "Notions de base".

Les fanatiques de l'assembleur pourront utiliser, en guise d'apéritif, la recherche de codes binaires de notre fidèle lecteur **Guido Bettiol**. Après cela, il leur restera à s'offrir la pièce de résistance, avec le moniteur étendu de **Thierry Le Tallec**; encore un programme de Pom's qui, à lui seul, vaut plus que le prix de l'abonnement annuel !

Guy Mathieu nous montre comment créer facilement des caractères graphiques. **Philippe Krepper** nous donne le moyen de "franciser" enfin les messages d'erreur du DOS (ou SED) : il ne reste plus qu'à faire des erreurs pour pouvoir apprécier...

Enfin, dans un registre plus général, vous trouverez les bancs d'essai du Basis 108 et du langage PILOT, ainsi que du plus récent SuperPILOT. Les nouveautés sont épluchées par **Jean-François Duvivier**, et **Olivier Herz** nous présente sa dernière invention, le HBASIC. Le HBASIC est un Apple-soft enrichi de nombreuses structures inspirées des langages Pascal et Ada. Comme il aurait fallu consacrer quasiment un numéro au HBASIC à lui tout seul, nous avons décidé de commercialiser celui-ci directement au prix de 150 francs, manuel d'utilisation compris.

Il s'agit là du premier progiciel que nous commercialisons séparément, et sur lequel l'auteur percevra des droits bien mérités de 10% du chiffre d'affaires TTC. Avis aux autres créateurs !

Nous commençons à vous parler de communication avec l'interview baptisée "Allo, Questel ?". Nous vous en reparlerons de façon plus étendue dans les numéros suivants.

Etant donné la croissance sensible du courrier des lecteurs, nous prions les lecteurs de joindre à leurs lettres une enveloppe timbrée pour la réponse : cela nous simplifie la tâche et permet de répondre plus vite.

Puisque nous en sommes aux envois postaux, nous informons les lecteurs qui ont reçu en retard leurs envois par avion que nous n'y sommes malheureusement pour rien. Nous avons correctement timbré ces envois, mais les Postes ont barré la mention "Par Avion" - apparemment du personnel peu au courant des tarifications. Nous espérons que cela ne se reproduira pas.

Le nombre d'abonnés continue à augmenter. Nous en profitons pour vous rappeler que ce sont les abonnés qui permettent à une revue de vivre. Si donc vous êtes en train d'hésiter à vous abonner, ne tardez plus. Plus vous serez nombreux, meilleur sera le service que nous pourrons vous rendre.

Hervé Thiriez

Ont collaboré à ce numéro : Guido Bettiol - Philippe Chemin - Michel Crimont - Alexandre Duback - Jean-François Duvivier - Olivier Herz - Philippe Krepper - Thierry Le Tallec - Guy Mathieu - Gérard Michel - Hervé Thiriez - Yves Volpi.

Rédacteur : Olivier Herz - **Dessins :** Laurent Bidot.

Directeur de la publication - rédacteur en chef : Hervé Thiriez - **Siège social et abonnements :** Editions MEV - 49, rue Lamartine - 78000 Versailles - **Rédaction :** 59, bd de Glatigny - 78000 Versailles - Tél. : (3) 918.13.07 - **Courrier des lecteurs (logiciel) :** Olivier Herz - 17, rue du Gros Chêne - 44300 Nantes. **Régie publicitaire :** Force 7 - Anne Jourdan - 39, rue de la Grange-aux-Belles - 75483 Paris Cedex 10 - Tél. : (1) 238.66.10. **Diffusion auprès des boutiques et librairies :** PSI Diffusion - BP 86 - 77402 Lagny-sur-Marne Cedex.

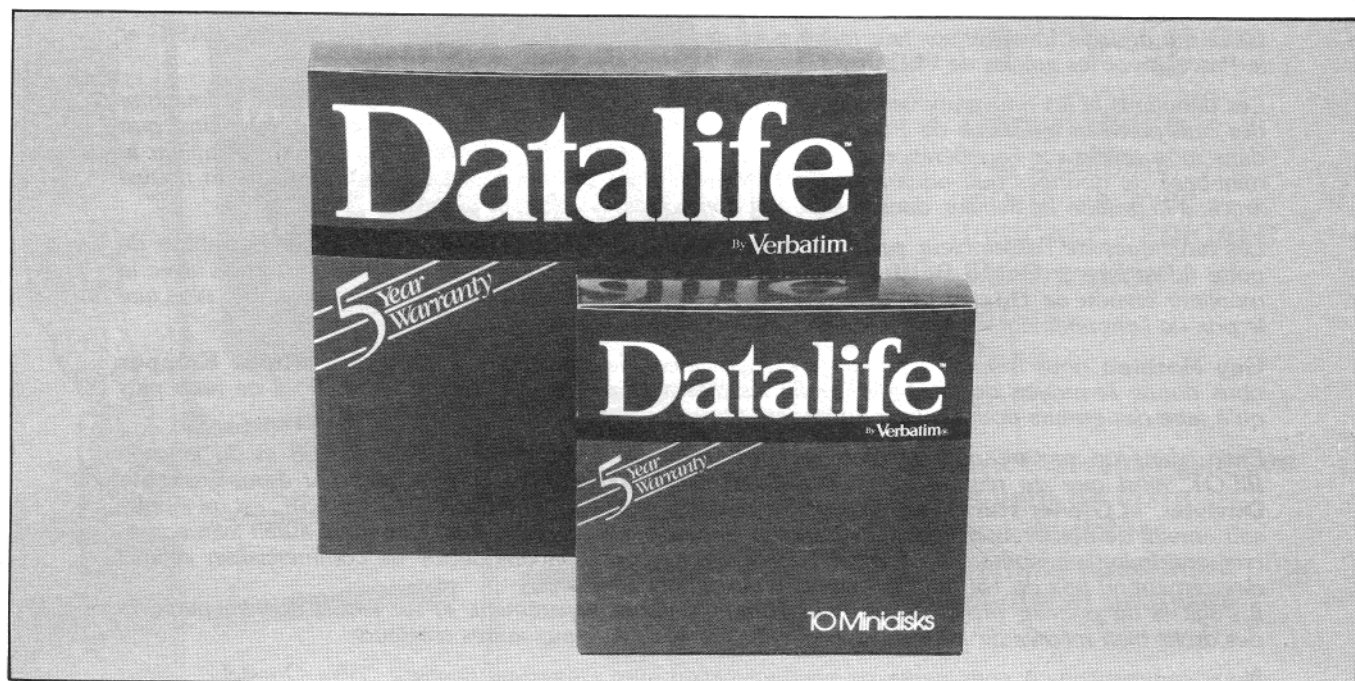
Composition : Télécompo - 13-15, avenue du Petit Parc - 94300 Vincennes - Tél. : 328.18.63.

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.

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

Mini-base de données

Yves Volpi

Le but de ces trois programmes est de permettre la constitution rapide d'une petite base de données, telle qu'un annuaire téléphonique, une collection de disques, une bibliothèque, ...

On commence par définir la forme de la fiche, après quoi il ne reste plus qu'à remplir le fichier.

Création de la fiche

Chaque fiche est constituée par un certain nombre d'items (10 au maximum pour des raisons de présentation sur écran). A chaque item devra correspondre l'entrée d'une donnée ayant des caractéristiques bien définies :

Caractéristiques de type possibles :

- 1) Alphanumérique avec une longueur maximale autorisée
- 2) Numérique de longueur limitée (maximum autorisé)
- 3) Numérique entier de longueur limitée
- 4) Date de la forme "JJ/MM/AA" ou bien "inconnue"
- 5) Longueur imposée
- 6) Alphanumérique de longueur imposée
- 7) Numérique de longueur imposée
- 8) Numérique entier de longueur imposée
- 9) Type à définir au sein du programme de création du fichier.

Caractéristiques de longueur : c'est le nombre maximum de caractères autorisé pour la donnée (types 1 à 3; pour la date, elle est imposée à 8),

ou bien une longueur imposée à l'entrée (types 5 à 8).

Pour des raisons de mise en page à l'écran, la longueur de l'item (libellé) augmentée de la longueur de la réponse ne doit pas dépasser 38 caractères (tout dépassement est contrôlé et signalé par le programme).

Le programme demande en fin de traitement le nom du fichier dans lequel seront stockées les caractéristiques de la fiche. Répondre ainsi XXX, par exemple, entraîne la création d'un fichier "XXX MEMO". Après cela, le programme de création du fichier peut être utilisé.

Création du fichier

1) Charger le programme

2) Mettre en ligne 3 le nom du fichier à traiter, soit XXX dans notre exemple. Le programme créera le fichier des données sous le nom "XXX FICHER".

3) Sauver le programme, ainsi rendu opérationnel, sous le nom "XXX" (il n'est pas obligatoire de reprendre le même nom, mais cela soulage la mémoire...).

Le programme peut bien sûr être modifié, notamment en rajoutant d'autres contrôles de type (emplacement prévu aux lignes 650 à 699).

Le menu peut également être complété pour d'autres traitements : il suffit d'intercaler à la ligne 20, avant le ">", les items du menu (10 au maximum). Chaque item supplémentaire renvoie respectivement aux lignes 6000, 7000, 8000, 9000 et 10000.

Les principales variables utilisées dans ce programme sont :

ME\$()	: menu
M\$()	: liste des mois (inutilisée)
F\$(N,I)	: fiches de dimension DI,NL
FI\$()	: intitulés des lignes
DO\$()	: liste des types
LO()	: longueurs permises pour les entrées
TY()	: types des différentes entrées
HT	: tabulation horizontale
VT	: tabulation verticale
VV	: idem
NF	: nombre de fiches
MF	: nombre maximum de fiches possible (calculé par le programme)
NL	: nombre de lignes par fiche
NM	: idem

Utilisation du fichier

1) Charger le programme correspondant.

2) Donner en ligne 3 le nom du fichier et ... à vous de travailler. Le programme listé ici se contente de lire le fichier; à vous de programmer les traitements (tris, impressions ...) que vous désirez.

Attention à la longueur de votre programme, qui doit rester compatible avec le nombre maximum de fiches traitées : si des arrêts de traitement se produisent, ils seront certainement dus au fameux nettoyage de la mémoire FRE(0), alors patience... (ou reportez-vous au numéro 2 de Pom's pour remplacer cette routine de nettoyage).

```
1  REM *** MINI-BASE - Y VOLPI ***
2  REM *** CREATION DU FICHER ***
3  FFS = "XXXXXXX"
4  KOCTET = 10000: REM RESERVE MEMOIRE
5  DIM DOS(11)
6  DS = CHR$(13) + CHR$(4)
7  F1$ = FFS + " MEMO": FFS = FFS + " FICHER"
8  O1$ = DS + "OPEN" + F1$: R1$ = DS + "RE
AD" + F1$: C1$ = DS + "CLOSE" + F1
$: OS = DS + "OPEN" + FFS: RS = DS
+ "READ" + FFS: CS = DS + "CLOSE"
+ FFS: WS = DS + "WRITE" + FFS
9  DOS(2) = "NUMERIQUE": DOS(1) = "ALPHA"
+ DOS(2): DOS(3) = "NUMERIQUE ENTI
ER": DOS(4) = "DATE'JJ/MM/AA' OU
INCONNUE'": DOS(5) = " DE LONGUEUR
IMPOSEE": DOS(6) = DOS(1) + DOS(5
): DOS(7) = DOS(2) + DOS(5): DOS(8)
= DOS(3) + DOS(5)
10 DOS(9) = " A DEFINIR ET A PROGRAMMER"
: DOS(10) = DOS(9): DOS(11) = DOS(1
0)
11 MUS = "300:02 02 AD 30 C0 88 D0 05 C
E 01 03 FO 09 CA D0 F5 AE 00 03 4
C 02 03 60 ND9C6G": FOR I = 1 TO
LEN(MUS): POKE 511 + I, ASC(M
ID$(MUS,I,1)) + 128: NEXT: POKE
72,0: CALL - 144
13 DATA JANVIER ,FEVRIER ,MARS
,AVRIL ,MAI ,JUIN
,JUILLET ,AOUT ,SEPTEMBRE,O
CTOBRE ,NOVEMBRE ,"DECEMBRE ": D
IM M$(12): FOR I = 1 TO 12: READ
M$(I): NEXT
19 REM
20 DATA ENTRER LES DONNEES,LIRE LES DON
```

```

NEES,MODIFIER LES DONNEES,SAUVER
LES DONNEES,FINIR,>
21 DIM MES(10)
22 MN = MN + 1: READ MES(MN): IF MES(MN)
  < > ">" THEN 22
23 MN = MN - 1: IF MN > 10 THEN INVERSE
  : PRINT "TROP DE LIGNES ": NORMA
  L : END
29 REM

35 ONERR GOTO 39
36 PRINT D$"VERIFY"Fl$
37 GOTO 40
39 HOME : PRINT " IL FAUT CREER LA FICH
  E ": END
40 POKE 216,0
41 PRINT O1$: PRINT R1$: INPUT NL: INPU
  T HT: FOR I = 1 TO NL: INPUT FI$(
  I): INPUT TY(I): INPUT LO(I): NEX
  T : PRINT C1$
45 REM CALCUL N MAX FICHES
46 PRINT "UN INSTANT SVP ":A = ( PEEK (
  111) + PEEK (112) * 256) - ( PEE
  K (109) + PEEK (110) * 256): FOR
  I = 1 TO NL:B = LO(I) + B: NEXT
  :MF = INT ((A - KOCTET) / (B + 7
  * NL)): REM CALCUL MAX FIC
  HES
47 IF NF > MF THEN MF = NF
48 DIM F$(MF,NL)
50 REM LIRE LE FICHER EXISTANT
51 ONERR GOTO 58
52 PRINT D$"VERIFY"FF$
53 PRINT O$: PRINT R$: INPUT NF
56 FOR I = 1 TO NF: FOR J = 1 TO NL: IN
  PUT F$(I,J): NEXT : NEXT : PRINT
  C$: POKE 216,0
57 REM
58 POKE 216,0
59 REM
  *** MENU ***
60 TEXT : HOME : INVERSE : HTAB 20 - L
  EN (FF$) / 2: PRINT FF$: NORMAL :
  POKE 34,2: VTAB 10
61 FOR I = 1 TO MN: INVERSE : PRINT I;"
  -";: NORMAL : PRINT MES(I): NEXT
  : PRINT :VM = PEEK (37) + 2
62 VTAB VM: PRINT " VOTRE CHOIX ";
63 VTAB VM: HTAB 17: CALL - 868: GET A
  $:A = VAL (A$): IF A < 1 OR A >
  MN THEN 63
64 ON A GOTO 1000,2000,3000,4000,5000,6
  000,7000,8000,9000,10000
65 VTAB 4: HTAB 5: PRINT "FICHE NO ";:
  INVERSE : PRINT NF;: NORMAL : PRI
  NT " MAX = ";: INVERSE : PRINT
  MF;: NORMAL : PRINT " FICHES": R
  ETURN
66 VTAB 4: HTAB 15: INVERSE : PRINT NF;
  : NORMAL : PRINT " FICHE";: IF NF
  > 1 THEN PRINT "S"
67 PRINT : RETURN
70 POKE 768,DU: POKE 769,HZ: CALL 770:
  RETURN
71 DU = 5:HZ = 10: GOSUB 70: RETURN
72 POKE 777,0:DU = 45:HZ = 30: GOSUB 70
  : POKE 777,1: RETURN
99 REM

100 REM VERIF SI = AU TYPE 1 ALPHANUM
110 FOR II = 1 TO LEN (A$):R$ = MID$(
  A$,II,1): IF R$ = "," THEN ER =
  1: GOTO 140
130 NEXT
140 RETURN
199 REM

200 REM VERIF SI = AU TYPE 2 NUMERIQUE
210 NV = 0: FOR II = 1 TO LEN (A$):A =
  ASC ( MID$( A$,II,1)): IF (A < 4
  8 AND A < > 46) OR A > 57 THEN E
  R = 1
220 IF A = 46 THEN NV = NV + 1: IF NV >
  1 THEN ER = 1
230 NEXT II
250 RETURN
299 REM
300 REM VERIF SI = AU TYPE 3 NUMERI E
  NTIER
310 FOR II = 1 TO LEN (A$):A = ASC (
  MID$( A$,II,1)): IF A < 48 OR A >
  57 THEN ER = 1
315 NEXT
320 RETURN
399 REM

400 REM VERIF SI = AU TYPE 4 DATE
410 IF LEN (A$) < > 8 THEN ER = 1: GO
  TO 490
415 IF A$ = "INCONNUE" THEN GOTO 490
420 BI = 0: IF MID$( A$,3,1) < > "/" O
  R MID$( A$,6,1) < > "/" THEN ER
  = 1: GOTO 490
430 AN = VAL ( STR$( 19) + MID$( A$,7,
  2)): IF MID$( A$,7,2) = "00" THE
  N BI = 1
440 IF ( INT ( AN / 4) ) * 4 = AN THEN BI
  = 1
450 JO = VAL ( MID$( A$,1,2)):MO = VAL
  ( MID$( A$,4,2))
460 IF JO > 31 OR JO < 1 THEN ER = 1: G
  OTO 490
461 IF JO > 29 AND MO = 2 THEN ER = 1:
  GOTO 490
462 IF JO > 28 AND MO = 2 AND BI = 0 TH
  EN ER = 1: GOTO 490
463 IF JO > 30 AND (MO = 4 OR MO = 6 OR
  MO = 9 OR MO = 11) THEN ER = 1:
  GOTO 490
464 IF MO < 1 OR MO > 12 THEN ER = 1
490 RETURN
499 REM

500 REM VERIF SI = AU TYPE 5 LONGUEUR
  IMPOSEE
510 IF LEN (A$) < > LO(I) THEN ER = 1
520 RETURN
599 REM

600 REM VERIF SI = AU TYPE 1 ALPHANUM
  DE LONG IMPOSEE
601 GOSUB 100: GOSUB 500: RETURN
610 REM VERIF SI = AU TYPE 2 NUMERIQUE
  EDE LONG IMPOSEE
611 GOSUB 200: GOSUB 500: RETURN
620 REM VERIF SI = AU TYPE 3 NUMERI
  ENTIER DE LONGUEUR IMPOSEE
621 GOSUB 300: GOSUB 500: RETURN
700 REM LECTURE 1 FICHE
710 PRINT "FICHE NO ";: INVERSE : IF I
  < 100 THEN PRINT "0";: IF I < 10

```



```

        THEN PRINT "O";
711 PRINT I: NORMAL
720 FOR J = 1 TO NL: PRINT F$(J);: HTA
    B HT: PRINT F$(I,J): NEXT
730 RETURN
740 FOR J = 1 TO NL: PRINT F$(J);: HTA
    B HT: PRINT F$(I,J): NEXT : RETUR
    N
900 REM
        *** ENTREE DES DONNEES ***
910 VTAB I + VV: HTAB HT: CALL - 868:
    FOR II = 1 TO LO(I): PRINT "-";:
    NEXT : HTAB HT: GOSUB 950
911 IF A$ = "" AND I = 1 THEN 920
912 IF ( LEN (A$) > LO(I)) OR (A$ = ""
    AND I > 1) THEN GOSUB 72:A$ = ""
    : FOR II = I TO NL: VTAB II + VV:
    CALL - 958: PRINT F$(II): NEXT
    : GOTO 910
913 ON TY(I) GOSUB 100,200,300,400,500,
    600,610,620,650
914 IF ER < > 0 THEN GOSUB 72:ER = 0:
    A$ = "": GOSUB 960: GOTO 910
920 RETURN
950 REM GET DES ENTREES
952 A$ = ""
954 GET R$: GOSUB 71: PRINT R$;: IF R$
    = CHR$(13) THEN 959
956 IF R$ = CHR$(8) THEN ER = 2: GOTO
    959
958 A$ = A$ + R$: GOTO 954
959 RETURN
960 REM MESSAGES ERREUR
961 VTAB 23: INVERSE : PRINT DO$(TY(I))
    ;: NORMAL : GET A$: VTAB 23: HTAB
    1: CALL - 958: RETURN
999 END
1000 REM
        *** ENTREE DES FICHES ***
1001 HOME
1002 VV = 6
1010 NF = NF + 1
1011 IF NF > MF THEN NF = NF - 1: INVER
    SE : PRINT "CAPACITE MAXIMUM ATTE
    INTE ": NORMAL : GET A$: GOTO 60
1020 GOSUB 65
1030 VTAB 6: FOR I = 1 TO NL
1040 VT = VV + I: VTAB VT: HTAB 1: PRINT
    F$(I): CALL - 958: NEXT
1050 FOR I = 1 TO NL: GOSUB 900: IF A$
    = "" THEN 1100
1060 F$(NF,I) = A$: NEXT : GOTO 1010
1100 NF = NF - 1: GOTO 60
2000 REM LECTURE FICHES
2001 A$ = ""
2010 HOME : GOSUB 66
2020 VTAB 16: HTAB 1: PRINT " <RTN> RE
    VENIR AU MENU ": PRINT : PRINT "
    =
    TOUTES LES FICHES"
2030 VTAB 20: HTAB 1: CALL - 868: INPU
    T " DONNEZ LE NUMERO ";A$
2031 IF A$ = "" THEN GOTO 2100
2032 IF A$ = "=" THEN : HOME : FOR I =
    1 TO NF: GOSUB 700: GET A$: PRINT
    : NEXT : GOTO 2100
2040 NA = VAL (A$): IF NA < 1 OR NA > N
    F THEN 2010
2050 I = NA: HOME : GOSUB 700: GET A$: G
    OTO 2010
2100 GOTO 60
2999 REM

```

à propos de logiciels ...

VENEZ POSER LES QUESTIONS QUE VOUS VOUS POSEZ ...

- Les performances
- Leurs différences
- Les matériels sur lesquels ils tournent
- Les configurations nécessaires ou souhaitables

● LOGICIELS OUTILS - GÉNÉRATEURS D'APPLICATION

DÉCISION	VISICALC - VISICAL 80 COL - VISICAL VF VISICALC VERSION ANANCÉE - MULTI PLAN CALCSTAR-SUPERCALC-MAGICALCLISACALC
TRAITE- MENT DE TEXTE	APPLEWRITER : 2 φ -][e - // - MAGICWINDOW - PLUME - LISAWRITE WORDSTAR - MOTUS ...
LES FICHIERS	PFS - PFS REPORT - DATAPLAN - DESKTOPPLAN QUIQFILE - VISIFILE - VISIDEX DBASE JC - LISALIST - DATASTAR - INFOSTAR
LE GRAPHIQUE LE DESSIN	COPYGRAPH EPSONGRAPH CEEMAC APPLEGRAPHICS - BUSINESS GRAPHICS PFS GRAPH - LISAGRAPH - LISADRAW ...
LA COMMU- NICATION	COMWARE - CALVADOS (banque données) ACCES // - IBM REFORMATEUR

● LOGICIELS D'APPLICATION

COMPTA BILITÉ	OMEGAS - COMPTA // - COMPTA // intégrée MACOMPTA - MICROCIEL - ORDIOUEST
FACTURA- TION STOCK	FAC // - GESTER MASTOCK - STOCK // CONTAINER
PAYE	PEGASE - MAPAYE - GIPSI
BIBLIO	« Le documentaliste » Logiciel de gestion documentaire permettant la recherche rapide d'une fiche bibliogra- phique faisant référence à un livre, un volume, un magazine, une publication et comportant 7 clés d'accès (2 noms d'auteurs 1 nom d'ouvrage, 4 mots-clés).
ENQUETE	Dépouillement d'enquêtes, questionnaires, sondages, études marketing. Déroulement du questionnaire - tri à plat - tris croisés. Corrections questionnaires et réponses.
ASSUR	Logiciel complet de gestion d'un cabinet d'assurances - trésorerie - police - risques fichiers assurés.

NOUVEAU

NOUVEAU

● LOGICIELS SYSTEME & LANGAGE

DOS - MEM/DOS - CP/M
BASIC - PASCAL - FORTRAN - COBOL

STAGES PASCAL : nous consulter

ALTI 39, rue Barrier
69006 LYON (7) 824 00 03

```

3000 REM *** MODIFICATION FICHE ***
3001 A$ = ""
3010 TEXT : HOME : GOSUB 66
3020 VTAB 16: HTAB 1: PRINT " <RTN> RE
VENIR AU MENU ": PRINT
3030 VTAB 20: HTAB 1: CALL - 868: INPU
T " NUMERO DE LA FICHE A MODIFIER
";A$
3031 IF A$ = "" THEN GOTO 60
3040 NA = VAL (A$): IF NA < 1 OR NA > N
F THEN 3010
3050 I = NA: HOME : GOSUB 740: PRINT :VV
= PEEK (37)
3060 VTAB VV: HTAB 5: CALL - 868: PRIN
T "ON MODIFIE CETTE FICHE NO ";NA
;" "": INVERSE : PRINT "O/N";: NO
RMAL : PRINT " "": GET A$: IF A$
< > "O" AND A$ < > "N" THEN 306
0
3070 IF A$ = "N" THEN 3000
3080 VV = PEEK (37) + 1: IF NL < 10 THE
N VV = VV + 2
3130 FOR I = 1 TO NL
3140 VT = VV + I: VTAB VT: HTAB 1: PRINT
FIS(I): CALL - 958: NEXT

```

```

3150 FOR I = 1 TO NL
3152 GOSUB 900
3155 IF A$ = "" THEN 3152
3160 F$(NA,I) = A$: NEXT : PRINT : PRINT
"PRESSEZ UNE TOUCHE "": GET A$:
GOTO 3000
3999 REM
4000 REM *** SAUVER LE FICHIER ***
4001 HOME : VTAB 20: PRINT " UN IN
STANT SVP "
4010 PRINT OS: PRINT WS: PRINT NF: FOR
I = 1 TO NF: FOR J = 1 TO NL: PRI
NT F$(I,J): NEXT : NEXT : PRINT C
$
4020 GOTO 60
5000 REM FINIR
5010 HOME : VTAB 20: HTAB 1: CALL - 95
8: PRINT " AVEZ VOUS SAUVE LE FIC
HIER "": INVERSE : PRINT "O/N";:
NORMAL : PRINT " "": GET A$: IF A
$ < > "O" AND A$ < > "N" THEN 5
010
5020 IF A$ = "N" THEN 60
5030 VTAB 20: HTAB 1: CALL - 958: PRIN
T " AU REVOIR.....": END

```

Création de la fiche

```

1 DIM FIS(10),LO(10),TY(10)
2 HOME : VTAB 20: INPUT "NOM DU FICHIER
";FF$
3 FF$ = FF$ + " MEMO"
4 DIM DOS(11)
5 D$ = CHR$(13) + CHR$(4)
7 OS = D$ + "OPEN" + FF$:WS = D$ + "WRIT
E" + FF$:RS = D$ + "READ" + FF$:C
$ = D$ + "CLOSE" + FF$
8 HOME : INVERSE : PRINT " CREAT
ION DE LA FICHE": NORMAL : POKE 3
4,2
9 DOS(2) = "NUMERIQUE":DOS(1) = "ALPHA"
+ DOS(2):DOS(3) = "NUMERIQUE ENTI
ER":DOS(4) = "DATE'JJ/MM/AA' OU '
INCONNUE'":DOS(5) = " DE LONGUEUR
IMPOSEE":DOS(6) = DOS(1) + DOS(5
):DOS(7) = DOS(2) + DOS(5):DOS(8)
= DOS(3) + DOS(5)
10 DOS(9) = " A DEFINIR ET A PROGRAMMER"
:DOS(10) = DOS(9):DOS(11) = DOS(1
0)
20 VTAB 10: HTAB 1: CALL - 958:I = I +
1: PRINT " DONNEZ LE NOM DE LA L
IGNE NO ";I
25 PRINT : INPUT " ";F$(I)
26 IF F$(I) = "" THEN NL = I - 1: GOTO
30
27 IF I < 10 THEN GOTO 20
28 IF I = 10 THEN NL = 10
30 HH = 0:HT = 0: HOME : FOR I = 1 TO NL
: PRINT F$(I):HH = LEN (F$(I)):
IF HH > HT THEN HT = HH
31 NEXT :HT = HT + 2
32 PRINT : PRINT "LONGUEUR MAX DISPONIB
LE POUR L' ENTREE": PRINT : PRINT
" DES DONNEES "": INVERSE : PRI
NT 39 - HT;: NORMAL : PRINT " CAR
ACTERES"

```

```

40 VTAB 20: HTAB 1: CALL - 868: PRINT
" EXACT";: INVERSE : PRINT "O/N";
: NORMAL : PRINT " "": GET A$: IF
A$ < > "O" AND A$ < > "N" THEN
40
45 IF A$ = "N" THEN HOME : I = 0: GOTO
20
46 HOME : VTAB 3: PRINT "DONNEZ LE TYPE
DES ENTREES ": PRINT : PRINT : P
RINT
50 FOR I = 1 TO 11: IF I < 10 THEN PRI
NT "O";
55 PRINT I;" ";DOS(I): NEXT
60 FOR I = 1 TO NL
61 VTAB 20: HTAB 1: CALL - 868: INVERS
E : PRINT F$(I);: NORMAL : PRINT
" "": INPUT " TYPE NO ";NO
65 IF NO < 1 OR NO > 11 THEN 61
66 TY(I) = NO
67 NEXT
70 HOME : PRINT : PRINT : HTAB 10: PRIN
T "FORMAT DES ENTREES": PRINT : P
RINT : FOR I = 1 TO NL: PRINT F$(
I);: HTAB HT: PRINT DOS(TY(I)): N
EXT
71 VTAB 20: HTAB 1: CALL - 868: PRINT
" EXACT";: INVERSE : PRINT "O/N";
: NORMAL : PRINT " "": GET A$: IF
A$ < > "O" AND A$ < > "N" THEN
71
72 IF A$ = "N" THEN 46
80 HOME : PRINT "DONNEZ LA LONGUEUR MAX
DES ENTREES ": PRINT : INVERSE :
PRINT 39 - HT;: NORMAL : PRINT "
CARACTERES AU MAXIMUM"
90 VV = 6: FOR I = 1 TO NL
92 VTAB VV + I: HTAB 1: CALL - 868: PR
INT F$(I);: HTAB HT: INPUT " LONG
UEUR ";NO: IF NO < 1 OR NO > 39 -
HT THEN GOTO 92
93 IF TY(I) = 4 THEN NO = 8
96 VTAB VV + I: HTAB HT: CALL - 868: F

```

```

OR Q = 1 TO NO: PRINT "-";: NEXT
:LO(I) = NO
97 NEXT
98 VTAB 20: HTAB 1: CALL - 868: PRINT
" EXACT";: INVERSE : PRINT "O/N";
: NORMAL : PRINT " ";: GET A$: IF
A$ < > "O" AND A$ < > "N" THEN
98

```

```

99 IF A$ = "N" THEN 80
100 HOME : VTAB 20: PRINT "UN INSTANT..
..
110 PRINT O$: PRINT W$: PRINT NL: PRINT
HT: FOR I = 1 TO NL: PRINT F$(I)
: PRINT TY(I): PRINT LO(I): NEXT
: PRINT C$

```

Utilisation du fichier

```

1 REM *** UTILISATION DU FICHIER ***
2 REM *** NOM DU FICHIER LIGNE 3 ***
3 FF$ = "XXXX":KOCJET = 10000
4 TEXT : HOME
5 D$ = CHR$(13) + CHR$(4)
6 F1$ = FF$ + " MEMO":FF$ = FF$ + " FICH
IER"
7 O1$ = D$ + "OPEN" + F1$:R1$ = D$ + "RE
AD" + F1$:C1$ = D$ + "CLOSE" + F1
$
8 O$ = D$ + "OPEN" + FF$:R$ = D$ + "READ
" + FF$:C$ = D$ + "CLOSE" + FF$:W
$ = D$ + "WRITE" + FF$
10 DATA JANVIER ,FEVRIER ,MARS
,AVRIL ,MAI ,JUN
,JUILLET ,AOUT ,SEPTEMBRE ,OC
TOBRE ,NOVEMBRE ,"DECEMBRE ": DI
M M$(12): FOR I = 1 TO 12: READ M
$(I): NEXT
11 MUS$ = "300:02 02 AD 30 CO 88 DO 05 C
E 01 03 FO 09 CA DO F5 AE 00 03 4
C 02 03 60 ND9C6G": FOR I = 1 TO
LEN (MUS$): POKE 511 + I, ASC ( M
ID$(MUS$,I,1)) + 128: NEXT : POKE
72,0: CALL - 144
19 REM
20 DATA TRAITEMENT NO 1,TRAITEMENT NO 2
,TRAITEMENT NO 3,TRAITEMENT NO 4,
ETC.....,>
21 DIM ME$(10)
22 MN = MN + 1: READ ME$(MN): IF ME$(MN)
< > ">" THEN 22
23 MN = MN - 1: IF MN > 10 THEN INVERSE
: PRINT "TROP DE LIGNES ": NORMA
L : END
29 REM
35 ONERR GOTO 39
36 PRINT D$"VERIFY"F1$
37 GOTO 40
39 HOME : PRINT " IL FAUT CREER LA FICH
E ": END
40 POKE 216,0
41 PRINT O1$: PRINT R1$: INPUT NL: INPU
T HT: FOR I = 1 TO NL: INPUT FI$(
I): INPUT TY(I): INPUT LO(I): NEX
T : PRINT C1$
46 PRINT "UN INSTANT SVP ":A = ( PEEK (
111) + PEEK (112) * 256) - ( PEE
K (109) + PEEK (110) * 256): FOR
I = 1 TO NL:B = LO(I) + B: NEXT
:MF = INT ((A - KOCJET) / (B + 7
* NL)): IF NF > MF THEN NF = MF
47 IF NF > MF THEN MF = NF
48 DIM F$(MF,NL)
50 REM LIRE LE FICHIER EXISTANT
51 ONERR GOTO 57

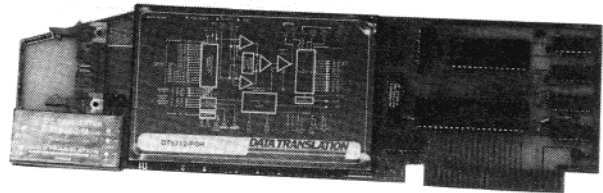
```

```

52 PRINT D$"VERIFY"FF$
53 PRINT O$: PRINT R$: INPUT NF
56 FOR I = 1 TO NF: FOR J = 1 TO NL: IN
PUT F$(I,J): NEXT : NEXT : PRINT
C$: GOTO 58
57 PRINT "IL N'Y A PAS DE ";FF$: PRINT
" SUR LA DISQUETE": END
58 POKE 216,0
59 REM
*** MENU ***
60 TEXT : HOME : INVERSE : HTAB 20 - L
EN (FF$) / 2: PRINT FF$: NORMAL :
POKE 34,2: VTAB 10
61 FOR I = 1 TO MN: INVERSE : PRINT I;"
-";: NORMAL : PRINT ME$(I): NEXT
: PRINT :VM = PEEK (37) + 2
62 VTAB VM: PRINT " VOTRE CHOIX ";
63 VTAB VM: HTAB 17: CALL - 868: GET A
$:A = VAL (A$): IF A < 1 OR A >
MN THEN 63
64 ON A GOTO 1000,2000,3000,4000,5000,6
000,7000,8000,9000,10000: REM

```

CARTES D'ENTRÉES ANALOGIQUES COMPATIBLES - APPLE®



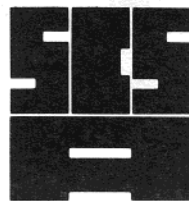
- * Nombre d'entrées : 16SE - 8DI ou 4 entrées isolées à 250 V
- * Entrée $\pm 10\text{mV}$ à $\pm 10\text{V}$
- * Résolution 12 - 14 - 16 bits
- * Vitesse de 15 kHz à 135 kHz avec DMA

- Options :
- Gain programmable
 - Contrôleur de DMA
 - Programme de tests
 - Logiciels

**FABRICATION
FRANÇAISE**

En standard : Horloge temps réel programmable

**DES CENTRALES DE MESURE COMPLETES
ÉTUDIÉES SUR CAHIER DES CHARGES**



Catalogues et notices techniques :

sacasa

société applications composants actifs
et systèmes associés

2, rue de l'Avenir
92360 Meudon-La-Forêt
Tél. : (1) 630.68.39 - Télex : 205157

CB 01xépob 448

dis monsieur,
apprends-moi
à gérer un fichier.



apple II apple III

Carte MEM/DOS 6502

LE SYSTEME D'EXPLOITATION
DU 6502 - MONOPOSTE/MULTIPOSTE

UNE EXTRÊME SIMPLICITÉ DE PROGRAMMATION.

- La division de la longueur des programmes par 20.
- La possibilité réelle de dessiner ses masques de saisie ou d'impression.
- Une indépendance totale de la périphérie choisie par rapport au système.
- L'intégralité du système contenu sur une carte mémoire de 20 K.
- Une gestion de mémoire de 140 K à 120 mégas.
- Des utilitaires déterminants
 - un générateur de programmes de gestion de fichiers permettant même le séquentiel indexé multiclé
 - un générateur d'écrans.
- CALL FN, une nouvelle commande basic, très puissante, intégrée au système permettant l'appel des sous-programmes par noms avec passage de paramètres et variables locales.
- Une version multiposte assurant la mise en commun totale des ressources sans conflit et l'autonomie des postes intelligents disposant de leur propre unité centrale.
- Des programmes compatibles APPLE II et APPLE III automatiquement transférables sur COMMODORE 8096.
- Et pour demain, des logiciels développés aujourd'hui directement compatibles avec le réseau local memnet.



3, rue Meyerbeer - 06000 NICE - Tél. 461 916 F

DISTRIBUTEURS AGREES

D.S.A. INFORMATIQUE
5, bd Dubouchage
06000 NICE
Tél. (93) 85.15.96

MICRO ALPHA SOFT
11, impasse du Lacquet
25200 MONTBELIARD
Tél. (81) 97.16.46

SEEMI
61, rue Ch. Rivière - B.P. 0701
44401 REZE CEDEX
Tél. (40) 75.52.80

MICROMEGAS
22, rue des 3 Pierres
69007 LYON
Tél. (7) 861.19.52

G-B
C.I.C.C.
Grove house
the bordage
St Peter Port
GUERNSEY
(0481) 20155

BENELUX
MEGAVOLT S.A.
Rue de Bleurmont
32 B
B 4920 EMBOURG
(041) 43.01.28

Routines de calcul en "Format Gestion" *Gérard Michel*

Applesoft n'aime guère les nombres qui comportent trop de chiffres : si vous ne l'avez pas encore constaté, vous risquez fort d'en faire les frais un jour ou l'autre !

Ainsi, Applesoft n'affiche jamais plus de 9 chiffres et n'effectue pas davantage de calculs justes au-delà de cette limite. Jusqu'à un certain point, le problème se limite à des arrondis intempestifs et souvent curieux; ensuite tous les nombres sont représentés en notation scientifique (Cf pages 31 à 33 du manuel de référence Apple II Plus pour plus de détails).

Mais pour certaines applications, et typiquement dans les problèmes de gestion, il peut être nécessaire d'obtenir des calculs justes "au centime", et de voir "999.999.999,99" comme résultat d'une addition plutôt que "1E09".

L'ensemble des routines présenté ci-après permet de réaliser des additions et des soustractions au format gestion, en traitant les nombres comme des chaînes de caractères (ce qui autorise tout de même 254 chiffres par nombre !). Quelques explications doivent suffire pour en analyser le mécanisme et les principes d'utilisation.

- Lignes 20 à 26 : tout nombre rangé en M\$ et comportant au plus 2 décimales est ajusté à un format unique comportant exactement 2 décimales; si nécessaire, un zéro est ajouté devant le point décimal.

A noter que l'utilisation des routines suppose que tous les nombres traités sont en notation anglo-saxonne et ne comportent effectivement pas plus de 2 décimales. C'est évidemment dans la procédure de saisie des données,

par l'analyse des chaînes de caractères fournies, que le respect de cette règle doit être contrôlé.

- Lignes 30 à 42 : routine d'addition. Les deux nombres à additionner sont stockés dans Z1\$ et Z2\$. Il s'agit finalement de réaliser l'addition comme à la main, chiffre par chiffre, en tenant compte des retenues éventuelles. Ces retenues sont détectées lorsque la longueur de la variable re-



sultat M\$ est égale au nombre de chiffres traités Z1, y compris celui qui est en cours de traitement (lignes 38 et 40).

- Lignes 50 à 70 : routine de soustraction. Mêmes principes que ci-dessus, mais la procédure est un peu plus compliquée, au début comme à la fin.

Tout d'abord, on ne peut pas toujours utiliser la fonction VAL pour

déterminer quel est le plus grand nombre des deux. En effet, pour des grands nombres qui ne différeraient que par les décimales, par exemple, la valeur retournée par VAL serait identique. Il est donc parfois nécessaire d'examiner le problème chiffre par chiffre (lignes 50 à 53). Par ailleurs, il faut tenir compte des retenues en chaîne éventuelles lorsque le plus petit des deux nombres est très inférieur à l'autre (lignes 66 à 70).

- Lignes 80 à 84 : cette routine est utilisée pour l'affichage des résultats en notation française avec virgule de décimalisation et points pour les puissances de mille.

- Le début du programme (lignes 10 à 17) fournit quelques exemples de démonstration avec des nombres générés de façon aléatoire.

Exemple d'exécution du programme

Z1+Z2=8.28161808E+10
Z1-Z2=-3.87844131E+10

22015883851.16 60800296963.33
22015883851.16 60800296963.33

Z1\$+Z2\$=82816180814.49

82.816.180.814,49

Z1\$-Z2\$=-38784413112.17

-38.784.413.112,17

```

5  REM
   *** CALCUL FORMAT GESTION ***
   *** GERARD MICHEL - POM'S ***
10  FOR U = 1 TO 10:Z1 = INT ( RND (U)
   * 1000000 * RND (1)) / 100:Z1$ =
   STR$(Z1):Z1 = INT (Z1 * RND
   (1) * 1000):Z1$ = STR$(Z1) + Z1
   $:Z2 = INT ( RND (U) * 1000000 *
   RND (1)) / 100:Z2$ = STR$(Z2)
   :Z2 = INT (Z2 * RND (1) * 1000)
11  Z2$ = STR$(Z2) + Z2$
13  PRINT :Z1 = VAL (Z1$):Z2 = VAL (Z2
   $): INVERSE : PRINT "Z1+Z2="Z1 +
   Z2: PRINT "Z1-Z2="Z1 - Z2: NORMAL
   : PRINT
15  PRINT Z1$,Z2$:M$ = Z1$: GOSUB 20:Z1$
   = M$:M$ = Z2$: GOSUB 20:Z2$ = M$
   : PRINT Z1$,Z2$: PRINT : GOSUB 30
16  PRINT "Z1$+Z2$="M$: GOSUB 80: PRINT
   : PRINT M$: PRINT : GOSUB 50: PRI
   NT "Z1$-Z2$="M$: GOSUB 80: PRINT
   : PRINT M$: PRINT : GET Z$: NEXT
17  END
19  REM
   *** ROUTINE DE CALCUL ***
20  IO = 0: FOR I = LEN (M$) TO 1 STEP
   - 1:Z$ = MID$(M$,I,1): IF Z$ =
   "." THEN IO = I:I = 1
22  NEXT : IF IO = 0 THEN M$ = M$ + ".00
   ": RETURN
24  IF IO = LEN (M$) - 1 THEN M$ = M$ +
   "0"
25  IF IO = 1 THEN M$ = "0" + M$

```

```

26 RETURN
29 REM
    *** ADDITION ***
30 Z1 = 0:Z = 0:M$ = "":L = LEN (Z1$):M
1$ = Z1$:M2$ = Z2$: IF LEN (Z2$)
> L THEN L = LEN (Z2$):M1$ = Z2
$:M2$ = Z1$
32 FOR I = L TO 1 STEP - 1:Z = Z + 1:Z
1 = Z1 + 1:C1$ = MID$ (M1$,I,1):
IF C1$ = "." THEN Z1 = Z1 - 1:G
OTO 42
34 IF Z > LEN (M2$) THEN C2$ = "":GOT
O 38
36 C2$ = MID$ (M2$, LEN (M2$) - Z + 1,1
)
38 C$ = STR$ (VAL (C1$) + VAL (C2$)):
IF LEN (M$) < Z1 THEN M$ = C$ +
M$:GOTO 42
40 C$ = STR$ (VAL (C$) + VAL (LEFT$
(M$,1))):M$ = C$ + MID$ (M$,2)
42 NEXT :M$ = LEFT$ (M$, LEN (M$) - 2)
+ "." + RIGHT$ (M$,2):RETURN
49 REM
    *** SOUSTRACTION ***
50 Z = 0:Z1 = 0:M$ = "":L = LEN (Z1$):M
1$ = Z1$:M2$ = Z2$: IF LEN (Z1$)
< > LEN (Z2$) OR VAL (Z1$) <
> VAL (Z2$) OR Z1$ = Z2$ THEN 5
3
51 FOR I = 1 TO L: IF VAL (MID$ (Z2$,
I,1)) > VAL (MID$ (Z1$,I,1)) TH
EN M1$ = Z2$:M2$ = Z1$:I = L
52 NEXT :GOTO 54
53 IF VAL (Z2$) > VAL (Z1$) OR LEN (
Z2$) > L THEN L = LEN (Z2$):M1$
= Z2$:M2$ = Z1$
54 FOR I = LEN (M2$) TO 1 STEP - 1:C2
$ = MID$ (M2$,I,1):Z = Z + 1:Z1
= Z1 + 1: IF C2$ = "." THEN Z1 =
Z1 - 1:GOTO 58
55 C2 = VAL (C2$): IF LEN (M$) = Z1 TH
EN C2 = C2 + VAL (LEFT$ (M$,1))
:M$ = MID$ (M$,2)
56 C1$ = MID$ (M1$,L - Z + 1,1):C1 = V
AL (C1$):C1 = C1 + 20 * (C2 > C1)
:C$ = STR$ (C1 - C2):M$ = C$ + M
$
58 NEXT :M$ = LEFT$ (M$, LEN (M$) - 2)
+ "." + RIGHT$ (M$,2): IF L > Z
AND LEN (M$) = Z THEN M$ = LEF
T$ (M1$,L - Z) + M$:GOTO 60
59 IF L > Z THEN GOSUB 66
60 IF LEFT$ (M$,1) = "0" AND LEFT$ (M
$,2) < > "0." THEN M$ = MID$ (M
$,2):GOTO 60
62 IF M1$ = Z2$ AND Z1$ < > Z2$ THEN M
$ = "-" + M$
64 RETURN
66 C1 = VAL (MID$ (M1$,L - Z,1)):C2 =
VAL (LEFT$ (M$,1)):C = C1 + 20
* (C2 > C1) - C2:Z = Z + 1:M$ =
STR$ (C) + MID$ (M$,2): IF Z = L
THEN RETURN
68 IF C > 10 THEN M$ = LEFT$ (M1$,L -
Z) + M$:RETURN
70 GOTO 66
79 REM
    *** AFFICHAGE ***
80 Z = 1:M$ = LEFT$ (M$, LEN (M$) - 3)
+ "," + RIGHT$ (M$,2):S$ = "": I
F LEFT$ (M$,1) = "-" THEN S$ = "
-":M$ = MID$ (M$,2)
82 IF LEN (M$) < = 2 + 4 * Z THEN M$
= S$ + M$:RETURN
84 M$ = LEFT$ (M$, LEN (M$) - 2 - 4 * Z
) + "." + RIGHT$ (M$,2 + 4 * Z):
Z = Z + 1:GOTO 82

```

Bibliographie

Alexandre Duback

Pratique de l'Apple II, de H. Lilien, aux Editions Radio - 192 pages, 80 FF.

Comme de nombreux autres ouvrages de H. Lilien, il s'agit ici d'un ouvrage clair à la présentation soignée. Il est déjà sorti depuis quelque temps, mais nous n'avions pas encore eu l'occasion de le consulter. C'est un bon livre de démarrage, pour ceux qui débutent complètement. Malheureusement, l'Apple II n'est pas encore couvert par cet ouvrage : cela ne devrait rien changer pour l'Applesoft; par contre, la gestion du clavier est assez différente.

Éléments de programmation, de Jacques Schmit, aux Editions Eyrolles - 344 pages, 148 FF.

Ce pavé, à l'aide de nombreux exemples en BASIC, enseigne le découpage d'un programme à écrire en tâches, montre comment on parvient à la construction d'organigram-

mes. La réalisation par étapes d'un programme de comptabilité est un exemple approfondi présenté dans les 90 dernières pages.

Je reproche à ce livre, comme à d'autres d'ailleurs, de ne pas prévenir le lecteur qu'il y a plusieurs BASICs : nous ne savons même pas que le BASIC Microsoft est celui utilisé par cet ouvrage. Il vous faudra donc adapter les programmes pour les transcrire en Applesoft.

Clefs pour l'Apple II, de Nicole Bréaud-Pouliquen, aux Editions du PSI - 143 pages, 92 FF.

Avec le "BASIC Applesoft - Manuel de référence alphabétique" de Paul Merry édité par Mnémodyne et "Visi-calc sur Apple" d'Hervé Thiriez aux Editions du PSI, ce livre est pour moi le troisième "must" de tout Applesoft.

Comme le souligne sa publicité, il s'agit d'un ouvrage destiné à rester près de votre Apple : le répertoire tous les mots-clés du BASIC, du moniteur et du DOS, la syntaxe de leur utilisation et les messages d'erreur. Il vous fournit plein de bonnes adresses, et offre en prime de nombreux trucs et astuces, dont plusieurs sont déjà connus des lecteurs assidus de Pom's.

Je n'ai que deux remarques à faire : j'aurais aussi aimé voir un classement des erreurs par numéro, et non seulement par ordre alphabétique. Enfin, mes yeux auraient apprécié des sorties sur imprimante réalisées en qualité courrier, ou pour le moins avec un ruban neuf ...

En conclusion, il s'agit d'un excellent ouvrage que je recommande chaudement à tous ceux chez qui l'Apple est autre chose qu'un simple élément décoratif.

Recherche de codes binaires

Guido Bettiol

Il m'arrive souvent d'avoir à analyser un programme en langage machine que je n'ai pas écrit. Pour pouvoir s'y retrouver, il est nécessaire de repérer des codes connus. Ainsi, le code "20 58 FC" (HOME en Basic) est un bon repère, car on sait qu'à ce moment l'écran est effacé. Le

code "20" (JSR) n'est pas important en soi, mais "58 FC" (adresse moniteur \$FC58) doit être repéré.

Il est très long et fatigant d'éplucher le programme pour retrouver des codes binaires, alors qu'il suffit de laisser l'ordinateur travailler tout seul.

Le programme que j'ai écrit à cet effet et que je vous propose ici effectue la recherche d'un ou de deux octets, et imprime éventuellement le résultat de sa recherche.

Il est possible de fractionner la recherche en choisissant les adresses de début et de fin de celle-ci.

Recherche de codes binaires

```
1LIST
10 REM
   = RECHERCHE DE CODES BINAIRES =
   ** GUIDO B. BETTIOL - NOV 82 **

50 REM ** ON CHARGE LA SUBROUTINE **
60 PRINT CHR$(4)"BLOAD OBJ.BCERCHE"
100 TEXT : HOME
110 PRINT "***** RECHERCHE DE CODES BIN
    AIRES *****"
115 REM
    ** ADRESSE DE DEPART **
120 VTAB 4: CALL - 868: INPUT "ADRESSE
    DE DEPART: $";X$
130 GOSUB 1000
135 IF X$ = "" THEN 120
140 AH = INT (X / 256)
150 POKE 1,AH
160 AL = X - AH * 256
170 POKE 0,AL
200 REM
    ** ADRESSE DE FIN **
210 VTAB 5: CALL - 868: INPUT "ADRESSE
    FINALE: $";X$
220 GOSUB 1000
230 IF X$ = "" THEN 210
240 AH = INT (X / 256)
250 POKE 3,AH
260 AL = X - AH * 256
270 POKE 2,AL
300 REM
    ** CODE RECHERCHE **
310 VTAB 7: PRINT "CODE RECHERCHE"
320 VTAB 9
330 PRINT "1 OU 2 BYTES? (1/2) ";: GET
    Z$: PRINT Z$
340 IF Z$ = "1" THEN POKE 6,0: GOTO 40
    5
350 IF Z$ = "2" THEN POKE 6,128: GOTO
    405
360 CALL - 1059: GOTO 320
405 VTAB 11: CALL - 868
410 INPUT "CODE: $";X$
412 X1$ = X$
415 GOSUB 1000
420 IF X$ = "" THEN 405
425 IF Z$ = "1" THEN POKE 4,X: GOTO 47
    0
430 AH = INT (X / 256)
```

```
440 POKE 4,AH
450 AL = X - AH * 256
460 POKE 5,AL
470 VTAB 11: HTAB 20: PRINT "IMPRIMANTE
    ? (O/N) ";: GET ZZ$: PRINT ZZ$
480 IF ZZ$ = "O" THEN POKE - 16240,17
    : PR# 1: PRINT : PRINT : PRINT "C
    ODE: $";X1$: PRINT
500 VTAB 13: PRINT "ADRESSES:": PRINT
600 CALL 768
890 PR# 0
900 HTAB 1: VTAB 23: PRINT "PRESSEZ UNE
    TOUCHE POUR CONTINUER ";: GET XX
    $: GOTO 100
1000 REM
    ** CONVERSION HEX-->DEC **
1020 X = 0
1030 IF X$ = "" THEN CALL - 1059: RET
    URN
1040 A1$ = LEFT$(X$,1)
1050 X1 = ASC (A1$)
1060 X1 = X1 - 48
1070 IF X1 > 9 THEN X1 = X1 - 7
1080 IF X1 < 0 OR X1 > 15 THEN CALL -
    1059:X$ = "": GOTO 1000
1090 X = X * 16 + X1
1100 IF LEN (X$) = 1 THEN RETURN
1110 X$ = RIGHT$(X$, LEN (X$) - 1)
1120 GOTO 1030
```

Programme B.CHERCHE

*300.35A

```
0300- A0 00 A2 00 B1 00 C5 04
0308- F0 06 C8 F0 15 4C 04 03
0310- A5 06 10 1A C8 F0 0B B1
0318- 00 C5 05 F0 11 C8 F0 02
0320- D0 E2 E6 01 A5 01 C5 03
0328- F0 03 4C 04 03 60 A5 01
0330- 20 DA FD A5 06 10 09 88
0338- 98 20 DA FD C8 4C 44 03
0340- 98 20 DA FD A9 A0 20 ED
0348- FD C8 F0 D6 E8 E0 10 D0
0350- 07 A9 8D 20 ED FD A2 00
0358- 4C 04 03
```

Formation continue à la micro-informatique

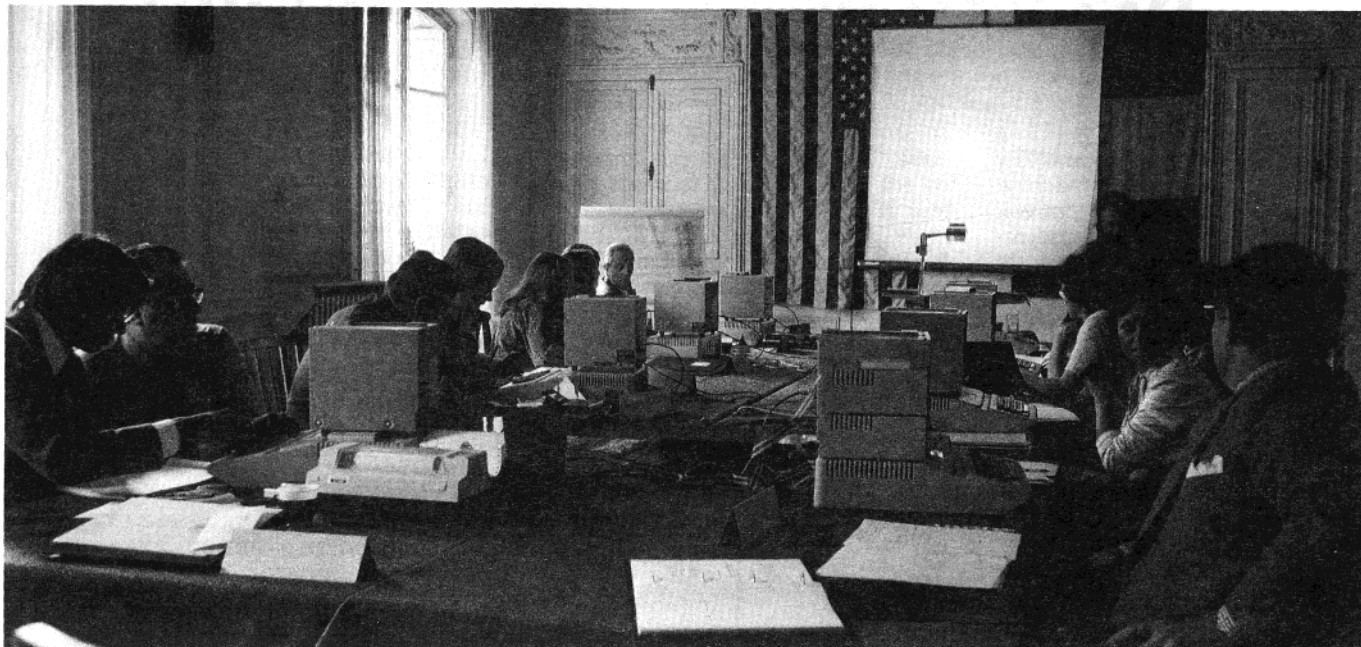


PHOTO GUNHILD BULL

Tous nos informaticiens viennent de l'informatique traditionnelle, et en maîtrisent totalement les langages classiques : Assembleur, COBOL, FORTRAN... Ils utilisent leur professionnalisme et les méthodes de l'informatique pour réaliser des **applications professionnelles en micro-informatique**. Nous vendons des micro-ordinateurs sans programme. Nous vendons aussi des micro-ordinateurs avec les programmes. Il s'agit de programmes réalisés par la société KA, dont nous **garantissons la qualité** et le bon fonctionnement.

Nous formateurs enseignons l'informatique. L'enseignement de la micro-informatique nécessite des **formateurs professionnels**, suffisamment de **matériel** pour que **chacun puisse pratiquer**, un **support de cours** couvrant non seulement l'enseignement diffusé, mais permettant au participant de **s'auto-former** après le stage. Nous avons déjà accueilli de nombreux stagiaires, d'horizons et de centres d'intérêts divers : chefs d'entreprise, universitaires, professions libérales, informaticiens, musiciens compositeurs, retraités, cadres de grandes entreprises, revendeurs de micro-ordinateurs...

Nous proposons 5 possibilités :

■ Stage de 2 jours bases de données.

Comment utiliser les progiciels :

- bases de données
 - manipulateurs de nombres et générateurs de tableaux
 - générateurs d'états imprimés
- Application pratique (un 48 K + un lecteur de disquettes pour deux participants).

Après ce stage, on peut générer, à partir de progiciels, un programme totalement adapté à son application en moins d'une journée de travail. Ce stage nécessite de connaître la manipulation de l'APPLE II, ou d'avoir suivi au minimum la journée d'initiation.

Dates 27-28 juin

29-30 août

Prix 2192 F h.t.

■ Stage de 1 semaine de programmation BASIC.

Il débute par la journée d'initiation.

Le stage permet d'assimiler la logique de programmation et de l'appliquer (un micro-système 48 K pour 2 participants). En fin de stage, on sait établir un programme de gestion de fichier avec consultation en temps réel. Ce stage ne nécessite pas de connaissance de départ en informatique.

Dates

du 20 au 24 juin

du 18 au 22 juillet

du 22 au 26 août

Prix 4245 F h.t.

■ Stage 3 jours disquettes.

Consacré à l'organisation, à la programmation et à l'exploitation de **fichiers sur disquettes magnétiques**, à travers l'étude du Disk Operating System APPLE II. Travaux pratiques sur **micro-systèmes** (un 48 K + un lecteur de disquettes pour deux participants). Ce stage nécessite :

- soit d'avoir suivi le stage de 1 semaine de programmation au préalable ;
- soit d'avoir une bonne connaissance théorique et une sérieuse pratique de BASIC de l'APPLE II.

Dates du 25 au 27 juillet

du 17 au 19 octobre

Prix 3378 F h.t.

■ Journées de sensibilisation et stages de formation à Paris et en Province.

Ils sont organisés à la demande

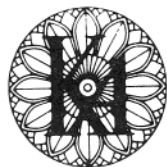
- d'une instance régionale telle, par exemple, une Chambre de Commerce ;
 - d'un organisme de formation dans le cadre d'un cycle plus vaste de formation ;
 - d'une entreprise.
- La société KA installe le matériel pour la durée de la formation, assure la formation et fournit les supports de cours.

■ **Journée d'initiation** - Dates : 20 juin, 18 juillet, 22 août - Prix 773 F h.t.

Le nombre de places pour chaque stage est strictement limité, à la fois pour la qualité de l'enseignement et par les contraintes du matériel. Deux animateurs sont présents pour aider les participants à la réalisation de leurs programmes.

Un support de cours très complet est remis à chaque participant.

Pour la journée d'initiation et pour les stages, les déjeuners sont pris en commun et compris.



l'informatique douce*

Renseignements et inscriptions à KA - Programme détaillé sur demande.

212 rue Lecourbe 75015 Paris - Tél. 533.13.50.

Le calendrier 83 est disponible.

*"L'informatique douce" est une marque déposée de la société KA

PILOT et SuperPILOT au banc d'essai

Olivier Herz

Le langage interprété PILOT (Programmed Inquiry, Learning Or Teaching), développé en 1968 à l'université de San Francisco, est destiné à être employé à la réalisation de programmes d'EAO (Enseignement Assisté par Ordinateur), appelés "leçons". Schématiquement, il s'agit de faire trois choses essentielles : informer l'élève, lire sa réponse à des questions et sauter à un endroit ou à un autre de la leçon selon cette réponse.

En 1980, Apple Computer Inc. a publié le système Apple PILOT, comprenant un PILOT basé sur la dernière version du langage, le Common PILOT, et quatre éditeurs :

- . l'éditeur de texte, pour écrire les programmes;
- . l'éditeur de dessins, éventuellement appelés par le programme;
- . le générateur de caractères. PILOT écrit en effet sur la page graphique haute résolution, ce qui nécessite des caractères générés par SOFT et non par HARD comme pour la page texte;
- . l'éditeur de sons et d'airs musicaux que le programme peut appeler.

Le tout a été développé grâce au système Pascal. Toutefois, pour utiliser Apple PILOT, ni la connaissance du langage Pascal, ni la carte langage ne sont nécessaires, bien que les possesseurs du Pascal Apple puissent, avec le système Pascal, lire les catalogues des disquettes PILOT et éditer directement les textes écrits en PILOT.

Puis, l'année dernière, Apple a publié le système SuperPILOT, version améliorée du précédent : des instructions ont été ajoutées au langage, les éditeurs sont plus performants et plus rapides, mais la carte langage est cette fois indispensable, afin d'augmenter la capacité de mémoire vive du système.

Le langage PILOT

D'un point de vue purement informatique, PILOT est un très mauvais langage, qui développe de vilaines habitudes chez le programmeur. Mais il a acquis une très grande popularité car, ne possédant qu'une dizaine de commandes principales, il est très facile à apprendre pour le non-informaticien, et il est spécialement étudié pour les applications à l'enseignement. Jugez-en vous-même par l'étude comparative de ses instructions avec BASIC ou Pascal (voir plus bas).

Les principales commandes du langage sont Remark, Type, Accept, Match, Jump, Use, End, Link, Dimension, Compute... Apple PILOT

et Apple SuperPILOT possèdent de plus des instructions graphiques, sonores et de gestion de fichiers, dont la possibilité d'exécuter des dessins et des airs de musique créés par les éditeurs et stockés sur disque.

Une instruction PILOT comprend une étiquette éventuelle, l'initiale ou les deux premières lettres de la commande, un "modificateur" éventuel qui change légèrement le sens de la commande (eXact ou Single pour Accept, Jump pour Match...),

un "conditionneur" éventuel qui établit un test en rapport avec la commande (Yes, No, Error... ou une expression comme (N > 13), le symbole ":" et enfin l'objet de l'instruction dont la syntaxe dépend de cette dernière.

Le petit programme ci-dessous écrit en SuperPILOT vous permettra de comprendre sa syntaxe. Afin de vous donner les mêmes chances qu'à Champollion, je vous fournis aussi la traduction en BASIC et en Pascal.

Programme en SuperPILOT Programme en BASIC Applesoft

```
R: C'EST UN EXEMPLE                    10 REM C'EST UN EXEMPLE
T: ENTREZ VOTRE NOM                    20 PRINT "ENTREZ VOTRE NOM"
A: $N$                                    30 INPUT N$
T: BONJOUR, $N$                         40 PRINT "BONJOUR, ";N$
M: OLIVIER!HERZ                         40 BON=(N$="OLIVIER" OR N$="HERZ")
JY: DEBUT                                50 IF BON THEN 100
C: A=1982+1                              60 A=1982+1
T: POM'S VOUS ADRESSE SES              70 PRINT "POM'S VOUS ADRESSE SES
   VOEUX POUR #A                        VOEUX POUR ";A
*DEBUT                                    100 GOSUB 1000
U: JEU
```

Programme en Pascal

```
VAR NOM:STRING;
    ANNEE:INTEGER;
    BON:BOOLEAN;
BEGIN
  (*C'EST UN EXEMPLE*);
  WRITELN ('ENTREZ VOTRE NOM');
  READLN (NOM);
  WRITELN (' BONJOUR, ',NOM);
  BON:=(NOM='OLIVIER') OR (NOM='HERZ');
  IF NOT BON THEN
  BEGIN
    ANNEE:=1982+1;
    WRITELN ('POM'S VOUS ADRESSE SES VOEUX POUR ',ANNEE);
  END;
  JEU;
END;
```

Fonctionnement du système PILOT

Pour créer des leçons, l'enseignant a besoin de deux drives, le premier contenant la disquette du système PILOT et le second celle où les leçons seront écrites. Le système boote sur un menu qui offre la possibilité d'utiliser l'un des quatre éditeurs, d'initialiser ou de recopier une disquette de leçons. Notons que, dans le cas de la copie, il est plus rapide d'utiliser un programme de copie tel que COPYA, car les disquettes, étant compatibles avec le Pascal, sont formatées en 16 secteurs. Toutefois, la disquette système est protégée contre la copie, mais un second exemplaire (BACKUP) est fourni avec elle. Une excellente disquette de leçons de démonstration est jointe également.

Une fois la leçon écrite, l'élève peut l'étudier avec un seul drive, car la disquette de leçons boote directement sur un programme HELLO qui affiche les titres des leçons ou qui appelle la leçon (ce peut être la leçon elle-même). Signalons enfin la possibilité intéressante pour le professeur de tester la leçon en la faisant marcher, avec des possibilités de débogage (debugging) étendues car PILOT est comme BASIC un interpréteur interactif.

Un mot à propos de la documentation: comme pour tout produit Apple, elle est très bien fournie et très claire, sous la forme d'un manuel du langage (250 pages) et d'un manuel des éditeurs (140 pages), tous deux en anglais. A quand la version française ?

Les éditeurs

L'éditeur de texte reprend en partie l'éditeur du Pascal Apple, avec quelques légères modifications: il y a 40 colonnes au lieu de 80, inversion des rôles de CTRL-C et ESC et enfin on peut obtenir les minuscules et majuscules, ces dernières étant indiquées en mode INVERSE, grâce aux touches CTRL-A et CTRL-Z.

Le générateur de caractères est identique à celui de DOS ToolKit, écrit en Applesoft. Il permet de définir une police de caractères, à la manière du programme CRECAR de Guy Mathieu publié dans ce numéro de Pom's.

L'éditeur de dessins possède de très puissantes commandes: il permet de tracer des ellipses et des rectangles, de remplir n'importe quelle surface... Les dessins peuvent être utilisés depuis un programme PILOT, soit en étant redessinés à la manière dont ils

ont été créés, soit en étant chargés directement, procédé beaucoup plus rapide, mais aussi plus gourmand en place sur le disque. La seule ombre au "tableau" de cet éditeur est qu'à chaque fois que l'on efface la dernière commande entrée, le dessin doit se redessiner entièrement, ce qui peut être fort long (cet inconvénient disparaît avec le SuperPILOT).

L'éditeur de sons quant à lui permet d'écrire des petits morceaux musicaux monophoniques, quelques effets spéciaux étant permis. Comme tout éditeur de sons assez simple pour Apple, il est très malcommode d'emploi.

Différences entre PILOT et SuperPILOT

En ce qui concerne le langage, on trouve dans SuperPILOT une plus grande variété dans le contrôle du texte à l'écran: simple ou double taille, couleurs variées pour le texte et le fond, caractères épais ou minces, combinaisons logiques entre les caractères et l'écran, possibilité d'animer des dessins en utilisant des polices de caractères spéciales, etc. On peut aussi noter la possibilité d'enregistrer automatiquement sur disque les progrès réalisés par les élèves, un mouvement relatif -genre tortue- dans les commandes graphiques, et quelques autres améliorations de moindre importance (dont la possibilité de commander des périphériques audio-visuels tels que les

Messages de la rédaction

Hard Copy Seikoshia: le programme de hard copy présenté par Olivier Herz dans le numéro 7 de Pom's a été écrit pour la GP80M. Il ne fonctionne pas pour la GP100. Un lecteur peut-il nous envoyer l'adaptation pour la GP100, afin que nous puissions la publier? En effet, nous n'avons pas ce modèle sous la main.

Gestion de marques: nous avons complètement oublié, quand nous vous avons proposé l'article de Gérard Michel sur la gestion de marques, que celui-ci utilisait des marques. Il est en effet indispensable pour le faire fonctionner d'avoir la disquette d'accompagnement du numéro 7 où l'on peut les trouver. C'est de la vente forcée involontaire! Nous avons hélas constaté ce problème trop tard et n'avons pas assez de place pour vous donner les marques. Il vous reste deux solutions en ce moment: acheter la disquette du numéro 7 ou modifier le programme Gesmask pour qu'il fonctionne sans masque... Pardon!

vidéo-disques). Et surtout, les programmes tournent maintenant jusqu'à 10 fois plus vite, ce qui était un progrès rendu indispensable par la lenteur du PILOT original.

L'éditeur graphique quant à lui permet d'insérer ou détruire un ordre graphique n'importe où dans le déroulement du dessin, et non plus seulement à la fin. On peut aussi redessiner un dessin à des fins de vérification, remettre en place les commandes qui avaient été effacées, changer l'origine du dessin ou utiliser comme fond un autre dessin. De plus, cet éditeur est maintenant plusieurs fois plus rapide dans le tracé d'un dessin.

Les fichiers fournis par ces éditeurs sont pratiquement compatibles d'un système à l'autre; il y a peu de modifications à apporter pour faire tourner un programme PILOT en SuperPILOT.

Signalons encore qu'on peut avec SuperPILOT initialiser une disquette de leçons non bootable afin de disposer de plus de place; il faut alors deux drives. On peut en outre ne recopier d'une disquette à l'autre que les leçons voulues et non obligatoirement la disquette toute entière.

La présence de la carte langage augmente la capacité mémoire, ce qui était bien utile, étant donnée la place prise par le système.

Notons enfin qu'avec tout cela les deux manuels ont grossi d'environ un quart et que la disquette de leçons de démonstration est un peu modifiée; signalons aussi la présence de deux disquettes de leçons Super Co-PILOT 1 et 2 qui constituent un excellent cours de SuperPILOT. Le petit exemple ci-dessus doit vous permettre d'apprécier l'apport de ce langage à l'EAO.

Conclusion

PILOT, et surtout SuperPILOT, constituent une excellente application à l'Apple du langage PILOT, grâce aux possibilités audio-visuelles extrêmement performantes et très faciles à mettre en oeuvre, dont le seul gros défaut, atténué dans le SuperPILOT, est la lenteur.

Les programmes ne résident pas en mémoire, mais sont lus sur le disque par petits morceaux, ce qui les rend extensibles à l'envi.

Nous conseillons fortement SuperPILOT à tous ceux qui désirent réaliser simplement et rapidement de longs programmes d'EAO ou des programmes, interactifs ou non, illustrant les capacités graphiques et musicales de l'Apple.

Accélérez vos programmes en BASIC

Hervé Thiriez

Afin d'accélérer l'exécution des programmes en BASIC, vous pouvez appliquer les recettes suivantes.

1 - Les variables

Utiliser des variables au lieu de constantes.

2 - Structurer le programme

Mettre les instructions d'autant plus haut dans le programme qu'elles servent souvent. Ainsi, la structure "optimale" d'un programme en BASIC est la suivante :

1. SUBROUTINES

D'autant plus "haut" qu'elles servent fréquemment

2. MENU PRINCIPAL

Principaux choix de l'utilisateur

3. CORPS DU PROGRAMME

4. INITIALISATION

Initialisation, lecture des tables de référence

Bien entendu, le programme débute par un GOTO vers la phase d'initialisation, laquelle termine par un GOTO vers le menu principal.

3 - Dimensionnement dynamique

Dimensionner les tableaux juste à la taille requise, par exemple avec des instructions telles que DIM A(L), ou DIM A(L+ 50) s'il y a par exemple L éléments dans la table et que l'on n'envisage pas d'en créer plus de 50 dans une session de travail. Ainsi, prenant moins de place en mémoire, votre programme tournera plus vite.

4 - Isoler les REMs

Prévoir de mettre les REMs sur des instructions séparées non adressées par des GOTOs ou GOSUBs. Il sera alors possible de les enlever totalement de la version exécutable du programme.

Si l'on tient à mettre des REMs dans des lignes, il faut les mettre en fin de ligne, car tout ce qui suit le REM est pris comme remarque (y compris des " : " de séparation).



5 - Vive les GOSUBs !

Utiliser les GOSUBs chaque fois qu'un gros segment de programme devrait être reproduit en deux exemplaires ou plus.

6 - Le ménage, vite fait...

Tirer profit du nettoyage rapide de la mémoire. Voir à ce titre "Faites le

ménage dans la mémoire" (Pom's 2) et "Contrôlez le nettoyage mémoire" (Pom's 3).

7 - Compaction du programme

Une fois le programme mis au point et testé, obtenez une version exécutable compactée (voir *Courier des Lecteurs* de ce numéro, lettre du Dr. Rozoy). Si vous possédez un compilateur, cela vous permettra évidemment d'accélérer sensiblement l'exécution du programme.

8 - Overlay dynamique

Si votre programme est vraiment trop long malgré tout cela, mettez en place une structure d'overlay dynamique (Pom's 1).

9 - Tableaux numériques

Il est assez facile d'écrire un programme en assembleur pour effectuer le STORE et le RECALL rapides de tableaux numériques sur disquette.

Attention à la renumérotation !

Si vous utilisez RENUMBER pour renuméroter vos programmes et compacter les programmes en numérotant par exemple avec un pas de 1, vérifiez tout d'abord que votre RENUMBER fonctionne bien.

Essayez de renuméroter le programme à une instruction : 20 A=20*20

Si vous obtenez : 10 A=20*10, votre RENUMBER a besoin de la réparation que nous expliquons dans le Pom's 2.

Nous ne pouvons pas ignorer l'arrivée sur le marché français d'un matériel compatible Apple, d'autant qu'il apporte un certain nombre d'éléments nouveaux.

De prime abord, nous découvrons un matériel à la présentation professionnelle, sans rapport avec celle de notre sympathique Apple II qui, par comparaison, donne l'impression d'être la 2CV de la micro-informati-

Le Basis 108 à l'essai

Hervé Thiriez

que. Le clavier du Basis 108 comporte 100 touches (52 sur l'Apple II) dont 9 touches de déplacement, un bloc de calcul et 15 touches de fonction. Bien entendu, les minuscules sont aussi disponibles. Le clavier est séparé du bloc principal comportant deux lecteurs de disquet-

tes, ce qui est la norme des matériels professionnels.

Caractéristiques techniques

Pour commencer, nous trouvons dans la carte mère deux micropro-

cesseurs, un 6502 et un Z80. D'origine, la RAM est de 128K, le système comportant en outre un moniteur ROM de 2K et cinq supports pour ROM ou EPROM de 2K, pour un total de 10K de ROM utilisateur.

Au niveau du graphique, beaucoup de possibilités sympathiques sont offertes :

- texte en 40*24 ou 80*24, mode HGR mixte : 280*160 avec 4 lignes de texte (40 ou 80 colonnes)
- mode HGR complet : 280*192 avec 6 couleurs

Il existe trois sorties vidéo distinctes : N/B, composite couleur et RGB.

Au niveau des caractères, il est très agréable de pouvoir, avec le Basis 108, basculer de 40 à 80 colonnes, et disposer de quatre polices de caractères à l'écran, changeables à volonté.

En standard, on a une sortie parallèle 8 bits, une interface E/S RS 232C, 6 prises compatibles Apple II pour connexion de périphériques, une prise vidéo RGB et composée (PAL ou NSTC), ...

Enfin, l'alimentation est plus puissante que sur l'Apple, ce qui devrait permettre d'éviter les problèmes que l'on a quand on charge trop l'Apple : il y a huit slots mais, selon le type de carte que l'on y met, on peut se retrouver incapable de booter avec trois ou quatre slots occupés. Nous avons déjà dû dans certains cas augmenter la puissance de nos Apples pour nous en sortir.

Démarrage

Pour des raisons de copyright, le Basis 108 n'est pas livré avec le Basic, le Pascal ou le CP/M 2.2. Par contre, il est aisé, en suivant les instructions fournies avec le matériel, de constituer les disquettes permettant de booter en Basic Integer, en Applesoft ou en Pascal, le tout en 40 ou 80 colonnes. Pour récupérer les Basics Integer ou Applesoft, il suffit de disposer de la disquette Basics (livrée avec tout contrôleur). Le boot Pascal est créé à partir de la disquette Apple 1 livrée avec le Pascal. Les instructions données par BMI pour réaliser ces disquettes de boot sont très claires.

Aspects économiques

Le Basis 108 tout nu est vendu 16.480 F HT. En configuration de base (hors moniteur), il revient à

21.341,40 F HT : Basis 108, 1 drive avec contrôleur et 1 disquette CP/M version 3.0. Le drive sans contrôleur est vendu 3.280 F HT.

Le budget est par conséquent sensiblement plus élevé que celui requis par l'acquisition d'un Apple //e. Il est vrai qu'on dispose d'un équipement compatible CP/M doté d'une présentation et d'une finition plus professionnelles.

Fabriqué en Allemagne, il est vendu depuis peu en France, ce qui ne permet pas encore de tirer de conclusion sur son pouvoir de pénétration du marché, ni sur la qualité du service après-vente.

Compatibilité Apple II

Nous avons essayé de nombreux programmes sur ce matériel, principalement avec les systèmes d'exploitation DOS 3.3 et Pascal. Il n'y a évidemment pas de problème pour faire tourner sur le Basis 108 les produits CP/M courants tels que Wordstar ou Calcstar.

Nous avons commencé par avoir exactement les mêmes difficultés que lors de nos débuts avec l'Apple //e : on se trompe de touches tout le temps, les réflexes de frappe rapide sur Apple II Plus ou //e ne convenant pas au changement de clavier. Cela ne doit pas être gênant une fois que l'on s'est accoutumé à ce nouveau clavier.

Avec Applewriter I, nous avons exactement le même problème que sur l'Apple //e : ce logiciel recodant les caractères d'écran, les espaces sont tous remplacés par une demi-barre transversale. C'est désagréable, mais n'empêche pas de travailler. Par contre, Applewriter //e ne marche pas, ce qui est normal puisque le Basis 108 est censé être compatible avec l'Apple II Plus et non avec l'Apple //e. D'ailleurs, Applewriter II fonctionne à merveille et reconnaît les minuscules.

Pour relancer le programme COPYA sans avoir à le recharger, faire RUN 80.

Ne pas lancer P.L.E. (Program Line Editor) quand TRACE est actif; faire d'abord NOTRACE. En effet, dans le cas contraire, lors de l'appel de PLE, le nom de celui-ci sur la disquette devient dénaturé avec des caractères spéciaux.

Les tentatives d'initialisation de disquette après exécution d'Applewriter I échouent : il faut rebooter avec une disquette normale avant de lancer une initialisation.

Le Pascal marche parfaitement en 40 colonnes; nous n'avons pas eu l'occasion lors de notre essai de le tester en 80 colonnes, mais cela ne pose aucun problème selon BMI avec le programme de boot adéquat.

Nous n'avons pas réussi à faire tourner Multiplan, les programmes de boot Apple II Plus et //e de Multiplan n'arrivant pas à se faire reconnaître. Notre bon vieux Visicalc a, par contre, tourné de façon complètement satisfaisante, mais seulement en 40 colonnes. Avec quelques surprises cependant : il m'a fallu longtemps pour découvrir que l'arobase, indispensable pour la définition des fonctions, s'obtenait avec la touche HOME.

Nous avons essayé de faire tourner GraForth. Le programme boote convenablement et tout se passe apparemment bien, sauf que toutes les minuscules (créées en haute résolution) sont absentes, ce qui est étrange car les majuscules aussi sont obtenues par logiciel sur l'écran HGR.

Conclusion

Pour celui qui a seulement besoin des services que peut rendre un Apple II Plus ou //e, il est certain que le Basis 108 n'est pas compétitif; son prix est trop élevé en comparaison.

Le Basis 108 est intéressant pour les utilisateurs de CP/M, pour ceux qui connaissent avec l'Apple des problèmes d'insuffisance de l'alimentation, pour ceux qui désirent utiliser un matériel plus professionnel (esthétique, fiabilité, clavier, possibilités). C'est aussi intéressant pour tous ceux qui ont des problèmes de vision et apprécieront la possibilité de basculer entre 40 et 80 colonnes, ou de choisir parmi quatre polices à l'écran. A condition de bien vouloir payer le prix de la différence.

Depuis notre banc d'essai de l'Apple //e, nous avons découvert qu'il n'était pas parfaitement compatible avec l'Apple II Plus. Certains jeux (surtout ceux de Programma Software) ne fonctionnent pas, Applewriter I marche imparfaitement, ... Sur l'échelle de la compatibilité Apple II Plus, le //e vaut 9/10 et le Basis 108 environ 8,5/10.

Nous conseillons par conséquent aux futurs acquéreurs de Basis 108 disposant déjà d'une Applethèque de faire des essais avant l'achat, afin de vérifier si leurs programmes préférés fonctionnent bien sur ce matériel.

De toute façon, nous pensons que les utilisateurs d'Apple ont à gagner à la présence de matériels compatibles. Une saine concurrence ne peut avoir pour eux que des effets positifs.

Création de polices de caractères

Guy Mathieu

HAIFA (Pom's 5 - erratum en Pom's 6), ainsi que le DOS ToolKit et d'autres logiciels, utilise des polices de caractères qui fonctionnent sur écran graphique haute résolution, enregistrées en RAM et traitées par POKES sur la page graphique.

Les polices de caractères sont elles-mêmes créées par des POKES dans des zones mémoire réservées, selon un principe expliqué dans l'article sur HAIFA.

Il est plus agréable et plus pratique, au lieu de calculer les POKES en question, de créer graphiquement, en grande dimension, les caractères choisis, surtout si l'on désire créer des caractères graphiques, des symboles, ... C'est le but atteint par le programme CRECAR, qui permet en outre de créer des caractères directement par voie numérique, ainsi que de les relire.

L'instruction 5 fournit au programme l'adresse de début de la table de caractères (il serait évidemment possible de l'entrer par INPUT au lancement du programme). L'adresse décimale correspond à la version 48K de HAIFA.

Le programme s'exécute ensuite de la façon suivante :

Demande d'un nom de police

Si un nom de police est fourni, cette police est chargée en mémoire et l'utilisateur est averti du risque de sa destruction (de la police, bien entendu ...). Si l'on a fait RETURN sans fournir de nom de police, cela signifie que l'on crée une nouvelle police.

Affichage d'un menu

Suit l'affichage d'un menu donnant le choix entre les options :

- 1 - CREER
- 2 - LIRE
- 0 - ARRETER

Nous analysons ci-dessous ce qui se passe lorsque chacune de ces options a été sélectionnée.

Option CREER

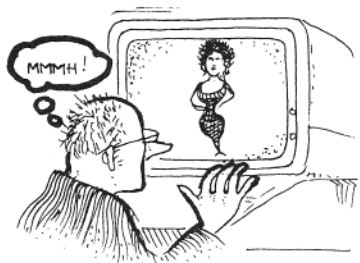
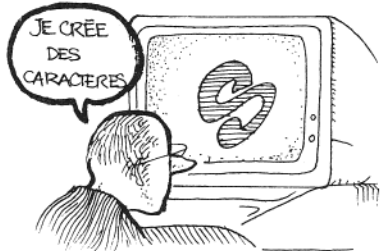
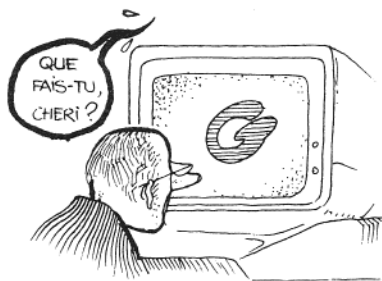
Un second menu nous donne le choix entre les possibilités :

- 1 - NUMERIQUE (donc par POKES, comme avec HAIFA)
- 2 - GRAPHIQUE

Création numérique

En cas de création numérique, la séquence ci-dessous est suivie :

- Quel caractère ?
Répondre par un caractère du cla-



vier. Si ce caractère est précédé de ESC, il sera considéré comme "minuscule", avec la définition donnée dans l'article sur HAIFA.

- Position No... (défilement de 1 à 8)
Fournir la valeur du POKE correspondant, un nombre de 0 à 255.
- Après la dernière entrée, tracé du résultat sur l'écran

- Autre caractère ?
Si la réponse est positive (RETURN), nous retournons à la question "Quel caractère?". Dans le cas contraire (ESC), retour au menu.

Création graphique

En cas de création graphique, l'ordre des opérations est le suivant :

- Dessin d'une grille 7*8
- Un petit carré (créé par SHAPE) peut être déplacé sur la grille par les flèches droite et gauche. La bascule entre déplacement horizontal et vertical s'effectue en appuyant, comme pour Visicalc, sur la barre d'espacement.
- ESC met en place un point du caractère. Un second ESC sur un point déjà tracé efface celui-ci.
- Quand le dessin du caractère est terminé, RETURN. Il y a émission d'un BEEP. En cas de RETURN par erreur, on peut revenir au mode déplacement. Si tout va bien, un second RETURN confirme la fin du dessin.

Option LIRE

- Position verticale (du premier caractère à lire) ? Répondre par une valeur de 0 à 19.
- Position horizontale (idem).
- Caractère ?
La phrase s'inscrit à partir de la position désignée sur l'écran. Attention : quand on va à la ligne, il se passe des choses bizarres...
- RETURN renvoie au menu principal.

Option ARRETER

- Voulez-vous sauver...
Réponse négative (ESC) : "Au revoir".
Si oui (RETURN) : donner le nom choisi. Si aucun nom n'est donné (RETURN à vide), celui de la police lue est utilisé.
- Enregistrement de la police sur le disque sous le nom choisi, puis message d'adieu.

Si l'on sort par mégarde du programme, on peut y revenir par RUN, les polices étant stockées par des POKES.

0 REM

*** CREATION DE CARACTERES ***
*** GRAPHIQUES - G MATHIEU ***

```

1 POKE - 16142,0
2 LOMEM: 17000
3 HIMEM: 36863
4 DS = CHR$(4): DIM E2(7,8)
5 DD = 36864
6 TY = 7
7 MX = 255
8 POKE - 16142,0
9 GOTO 10000
10 REM CREATION
15 HOME : PRINT "1 - CREATION NUMERIQUE
"
16 PRINT : PRINT "2 - CREATION GRAPHIQUE"
17 PRINT : INPUT " VOTRE CHOIX ? "
;CR
18 IF CR = 2 THEN GOSUB 2000
19 IF CR = 1 THEN TEXT : HOME
20 PRINT "QUEL CARACTERE ? " : A = 0
21 GET Z$: PRINT Z$;
22 IF ASC (Z$) = 27 THEN A = 32: GOTO
21
23 PRINT : IF ASC (Z$) < 32 AND ASC (
Z$) < > 13 THEN 19
25 I = ASC (Z$) - 32 + A
30 FOR J = 0 TO TY
35 IF CR = 2 THEN X = N(J): GOTO 60
40 PRINT "POSITION ",J + 1;" : ";
50 INPUT X

```

```

55 IF X > MX OR X < 0 THEN PRINT "VALE
UR INCORRECTE": GOTO 40
60 AD = DD + I * (TY + 1) + J
70 POKE AD,X
80 NEXT
82 HGR
85 LS = Z$:VT = 10:HT = 10: GOSUB 100
90 VTAB 23: PRINT "AUTRE CARACTERE ? (R
ET/ESC) " : GET Z$:Z = ASC (Z$)
91 HOME
92 IF Z = 13 THEN 18
95 TEXT : HOME : RETURN
100 L = LEN (L$): FOR J = 1 TO L
110 V1 = INT (VT / 8):V2 = VT - 8 * V1
115 VV = 8192 + V1 * 40 + V2 * 128
120 VH = VV + HT + J - 1
122 LL$ = MID$(L$,J,1)
125 X = ASC (LL$) - 32 + A
130 FOR I = 0 TO TY
135 A1 = VH + I * 1024
140 A2 = DD + X * (TY + 1) + I
145 POKE A1, PEEK (A2)
150 NEXT I
155 NEXT J
165 RETURN
999 END
1000 REM LECTURE
1001 HOME : VTAB 21
1005 HGR
1010 INPUT "POSITION VERTICALE ? ":V
T: IF VT < 0 OR VT > 19 THEN TEX
T : GOTO 1001
1020 INPUT "POSITION HORIZONTALE ? ":H
T: IF HT < 0 OR HT > 39 THEN TEX
T : GOTO 1001
1025 PRINT "CARACTERE ? (RETURN POUR AR
RETER) "
1030 GET L$: IF ASC (L$) = 27 THEN A =
32: GOTO 1030
1035 IF ASC (L$) = 13 THEN 1050
1037 IF ASC (L$) < 32 THEN VTAB PEEK
(37): GOTO 1025
1040 GOSUB 100
1042 A = 0
1045 HT = HT + 1: GOTO 1030
1050 TEXT : HOME : RETURN
2000 HGR : HCOLOR= 3
2010 FOR I = 100 TO 160 STEP 10: HPLLOT
I,50 TO I,120: NEXT
2020 FOR I = 50 TO 120 STEP 10: HPLLOT 1
00,I TO 160,I: NEXT
2100 DATA 1,0,4,0,27,36,45,45,54,54,
63,63,36,0
2110 FOR I = 1 TO 14: READ V: POKE 3800
0 + I - 1,V: NEXT I
2120 POKE 232,112: POKE 233,148
2130 GG = 1: SCALE= 1: ROT= 0: POKE - 1
6142,0
2140 XA = 100:XX = 100:YA = 50:YY = 50
2150 FOR I = 0 TO 7:N(I) = 0: FOR J = 0
TO 8:E2(I,J) = 0: NEXT : NEXT
2160 RESTORE
2200 REM A:ECRIT. HR
2210 ROT= 0: REM :SHAPE
2220 XDRAW GG AT XA,YA
2230 GET Z$:Z = ASC (Z$)
2240 IF Z = 32 THEN ZD% = 1 - ZD%
2250 IF Z = 8 THEN XX = XX - 10 * ZD%:Y
Y = YY - 10 * (1 - ZD%)
2260 IF Z = 21 THEN XX = XX + 10 * ZD%:
YY = YY + 10 * (1 - ZD%)

```

E . A . O .

Enseignement Assisté par Ordinateur pour Ordinateurs

MOPPE D'ANPICO - SILEX - ITT 2020 - APPLE II - APPLE II +
COURS EN FRANCAIS

de B A S I C	<ul style="list-style-type: none"> . APPLESOFT - APPLE II (+ Carte APPLESOFT) - APPLE II + - APPLE III - MOPPE D'ANPICO (+ APPLESOFT en ROM) 	
	<ul style="list-style-type: none"> . PALSOPT - ITT 2020 (+ Carte PALSOPT) . SILRON - SILEX (de Léanord) 	

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

1/ COURS 1 (BASIC) - En Français

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

A LA FIN DE CE COURS, VOUS SAUREZ PROGRAMMER.

2/ COURS 2 (BASIC +) - En Français

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

3/ CONTRATS-LOCATION du COURS BASIC pour:

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

PRIX (T.V.A. comprise)

<ul style="list-style-type: none"> . COURS 2 (BASIC +) 375 FF. <p>Prix donnés à titre indicatif, pouvant être modifiés sans préavis.</p>	<p>ANDRE F. PINOT 8 Allée BUFFON 91000 EVRY-COURCOURONNES</p>
---	---

R E V E N D E U R S ,
C O N T A C T E Z - N O U S .

```

2265 IF Z < > 13 THEN 2275
2270 GOSUB 20000
2271 IF FIN = 1 THEN 2340
2272 IF FIN < > 1 THEN 2230
2275 IF Z = 27 THEN 2400
2280 IF Z < > 8 AND Z < > 21 AND Z <
    > 32 AND Z < > 13 AND Z < > 27
    THEN PRINT " ": GOTO 2230
2290 IF XX < 100 THEN XX = 100
2300 IF XX > 160 THEN XX = 160
2310 IF YY < 50 THEN YY = 50
2320 IF YY > 120 THEN YY = 120
2330 XDRAW GG AT XA, YA
2340 XDRAW GG AT XX, YY
2350 XA = XX: YA = YY
2360 IF FIN = 1 THEN 2500
2370 GOTO 2230
2400 II = (YY - 50) / 10: JJ = (XX - 100)
    / 10: IF E2(II, JJ) = 1 THEN E2(I
    I, JJ) = 0: N(II) = N(II) - 2 ^ JJ:
    SCALE= 2: ROT= 0: GOTO 2420
2405 SCALE= 2: ROT= 0: XDRAW GG AT XX, Y
    Y
2410 GET Z$: Z = ASC (Z$)
2420 IF Z = 27 THEN XDRAW GG AT XX, YY:
    SCALE= 1: GOTO 2230
2425 SCALE= 1
2450 N(II) = N(II) + 2 ^ JJ: E2(II, JJ) =
    1
2480 GOTO 2240
2500 VTAB 21
2510 FIN = 0
2520 RETURN
9999 END
10000 HOME : PRINT "CREATION/LECTURE":
    PRINT "DE POLICES DE CARACTERES":
    PRINT "EN HAUTE RESOLUTION"
10030 PRINT : PRINT "NOM DE LA POLICE ?
    "
10031 INPUT "( 'RETURN' SI AUCUNE )": NP$
10035 IF NP$ = "" THEN 10100
10040 PRINT D$ "BLOAD": NP$
10050 VTAB 10: PRINT "ATTENTION, VOUS R
    ISQUEZ DE DETRUIRE DES CARACTERES
    DE LA POLICE UTILISEE"
10100 PRINT : PRINT : PRINT "VOULEZ-VOU
    S"
10110 PRINT : PRINT "1 - CREER UN CARAC
    TERE"
10130 PRINT : PRINT "2 - LIRE UN OU DES
    CARACTERE(S)"
10150 PRINT : PRINT " (O POUR ARRETE
    R)"
10180 PRINT : INPUT " VOTRE CHOIX ?
    ": CH
10190 IF CH = 0 THEN 12000
10310 IF CH = 1 THEN GOSUB 10
10320 IF CH = 2 THEN GOSUB 1000
10400 GOTO 10100
12000 HOME : PRINT "VOULEZ-VOUS SAUVER
    ": NP$
12010 PRINT "'RETURN' SI OUI - 'ESCAPE'
    SI NON "
12020 GET Z$: Z = ASC (Z$)
12025 PRINT : PRINT
12030 IF Z = 13 THEN 12045
12035 IF Z = 27 THEN 12100
12040 GOTO 12020
12045 IF NP$ = "" THEN 12060
12050 PRINT "'RETURN' POUR GARDER LE NO
    M, OU BIEN

```

```

12060 PRINT "DONNEZ LE NOM CHOISI"
12070 INPUT " VOTRE CHOIX ? ": NP$
12075 IF NP$ = "" THEN PRINT " ": GO
    TO 12060
12080 PRINT D$ "BSAVE": NP$: ", A": DD: ", L$3
    00"
12100 HOME : VTAB 12: HTAB 12: PRINT "A
    U REVOIR": END
20000 PRINT " "
20010 GET Z$: Z = ASC (Z$)
20020 IF Z = 13 THEN FIN = 1
20030 RETURN

```

Message aux créateurs de logiciel

Nous commençons à commercialiser des programmes en plus de la revue Pom's, la rémunération des auteurs s'effectuant selon le principe des droits d'auteur.

Bien entendu, les programmes destinés à ce mode de distribution doivent être des produits finis, autant par la qualité des programmes que par celle de la documentation et par l'environnement visuel et sonore.

Nous sommes prêts à analyser vos propositions et à vous aider à transformer des idées en produits logiciels. Envoyez-nous vos contributions; nous vous aiderons à diffuser ces programmes.

EXEMPLE DE POLICE DE CARACTERES

'SEMI-GRAPHIQUES'

```

- r T r          - r T r
  + + +          + + +
  L L L |        L L L |
/ \ ^ \ \       / \ ^ \ \
< X >           < X >
  \ v /         \ v /

```

EXEMPLE DE REALISATION D'IMPRIME

Numéro	Désignation	Montant

QUELQUES FIGURES VARIEES

⌘ ⌘ ⌘ ⌘ ⌘

EXEMPLES DE CARACTERES GRECS

α β γ ε ψ δ η ι κ λ
μ ν ο π ρ σ τ υ φ χ ζ

Allo, Questel ?

Pom's

Nous avons encore eu peu de temps pour parler de communication dans Pom's. Il s'agit pourtant là d'un sujet intéressant, mais nous avons tellement de choses passionnantes à vous faire partager ! Pom's a rendu visite à la société Euréquip dans laquelle travaille Guy Mathieu, que nos lecteurs assidus connaissent déjà bien. Nous vous parlerons encore de communication dans les numéros à venir; cet article de présentation joue le rôle d'éclaireur ... Nous retranscrivons ci-dessous les questions posées par Pom's et les réponses de Richard Haddad, collaborateur de Guy Mathieu.

A quoi sert Questel ?

Questel est un service permettant à ses utilisateurs de se relier à de nombreuses bases de données. Nous l'utilisons à de multiples occasions, par exemple pour retrouver toutes les références à certains vocables ou à certains mots-clés dans des publications, dans des conférences. On pourra ainsi rechercher tout ce qui existe dans une base de données et fait référence à la fois aux termes "ergonomie" et "micro-informatique".

Nous l'utilisons aussi pour nous renseigner sur tel ou tel sujet précis, à l'aide d'une des nombreuses bases de données gérées par Questel. Il existe par exemple trois bases de données de l'INPI permettant de se renseigner sur des problèmes de brevets et de brevetabilité. Bien entendu, l'intérêt de telle ou telle base de données dépend essentiellement du sérieux avec lequel elle est mise à jour, et cela varie d'une base à l'autre. Ainsi, la base de données Pascal (rien à voir avec le langage) contient des renseignements pluridisciplinaires (sciences physiques et de l'ingénieur, chimie pure et appliquée, sciences de la Vie et sciences de la Terre) et reçoit 500.000 nouvelles références par an.

Pourquoi Questel ?

Rien ne nous oblige à utiliser exclusivement une base de données précise. Une fois que nous pouvons relier notre Apple à l'extérieur, cela nous autorise à communiquer avec tout serveur avec lequel nous aurions ouvert un compte. Questel est un serveur (il fournit un service), Prestel en est un autre, et ils ne sont pas les seuls.

Ceci dit, il faut investir un certain temps pour apprendre à utiliser le système du serveur. En outre, pour tirer le meilleur parti d'une base de données, il faut en être familier, sinon l'on perd du temps pour trouver les réponses aux questions que l'on se pose, et de l'argent puisque la facturation dépend du temps de connexion.

Lorsque nous avons décidé d'utiliser des services d'information interactifs, nous avons analysé ce que proposaient les différents serveurs. Nous sommes arrivés à la conclusion que le système Questel était celui qui nous serait le plus utile, compte tenu de la grande variété de ses bases de données et de la qualité de leurs mises à jour.

Les serveurs n'offrent pas nécessairement l'accès à plusieurs bases de données. Ils peuvent aussi vous proposer, comme le fait Calvados, de partager certaines ressources informatiques: vous obtenez ainsi au bout de votre Apple une puissance de traitement informatique qu'il ne pourrait vous apporter seul.

Quelles sont les bases de données accessibles ?

Nous avons une brochure décrivant 26 bases de données proposées par le système Questel. Nous ne pouvons les citer toutes; notons par exemple TELEDOC pour les télécommunications, BSI sur l'informatique, LOGOS qui a mémorisé les discours des personnalités politiques françaises en même temps que des données relatives à l'activité politique, économique et sociale française. LOGOS vous donnera par exemple les références de tous les discours des six derniers mois dans lesquels le terme "lutte des classes" a été utilisé.

La base MEETING AGENDA vous renseigne sur les annonces de congrès, conférences et expositions. DEFOTEL garde à jour des informations boursières et financières pour 1500 sociétés françaises et étrangères. CARNETNET s'occupe exclusivement de tout ce qui touche de près ou de loin au cancer. Comme vous le voyez, la variété est grande.

Comment communiquer ?

Il suffit de relier votre Apple au réseau avec un modem téléphonique

ou acoustique, ou avec la nouvelle carte modem intégré disponible depuis peu sur Apple II Plus et //e. Nous nous servons d'un modem acoustique, que nous possédions déjà quand nous avons débuté avec Questel. La carte modem intégré n'était pas disponible quand nous avons commencé à "communiquer".

Combien cela coûte-t-il ?

Ne prenons pas en compte le coût du modem, qui est un coût fixe et tient au fait de communiquer, plutôt qu'à celui d'avoir recours aux services d'un serveur ou d'un autre. Selon la base de données utilisée, le coût par heure de connexion varie entre 180 F HT et 700 F HT, la très grande majorité des tarifs se situant entre 300 F HT et 500 F HT de l'heure. Il faut ajouter à cela le coût de la citation imprimée (de 0,60 F HT à 1 F HT en général).

Une citation sera par exemple la référence d'un article ou d'une conférence où les mots-clés que nous recherchons ont été utilisés.

Peut-on faire un essai ?

Absolument; il suffit de suivre notre procédure habituelle :

- brancher la carte de communication dans le slot 2 ;
- introduire le programme DATA CAPTURE dans le drive 1 ;
- introduire la disquette FICHIER dans le second drive ;
- mettre l'Apple sous tension.

Il se trouve à présent en mode terminal (TTY). Un premier menu apparaît à l'écran.

- faire "ESC T A" pour travailler avec deux lecteurs
- faire "ESC T D" pour travailler en full duplex
- mettre le modem sous tension
- mettre le modem en FULL DUPLEX
- composer le numéro du réseau Transpac
- à l'écoute de la porteuse (tonalité indiquant le lien avec l'ordinateur au bout de la ligne), enfoncer le combiné dans le modem
- taper H (et RETURN)
- taper 106000019 ou 106000027 (et RETURN) pour Questel 1
- entrer le numéro de compte et le mot de passe.

Ce n'est pas plus compliqué que cela. Il ne reste plus qu'à utiliser Questel.

Calcul à 12 chiffres

Michel Crimont

Introduction

L'utilisation des types INTEGER ou REAL en Pascal ne permet pas de disposer de plus de 6 chiffres significatifs, ce qui est insuffisant, ne serait-ce que pour les applications comptables. Le type LONGINTEGER peut comporter jusqu'à 36 chiffres, mais ne permet pas le calcul décimal. Les procédures et fonctions définies dans cet article permettent d'utiliser 12 chiffres significatifs avec des exposants de + à - 30 pour les 4 opérations et les comparaisons. On définit à cet effet les constantes :
MINEXPO = - 30 et MAXEXPO = + 30
SIGNIFICATIF = 12 : maximum de chiffres significatifs

Le type REEL sera alors défini par :
REEL = RECORD

```
EXPO      :MINEXPO..
           MAXEXPO;
MANTISSE  :INTEGER!
           SIGNIFICATIF!;
```

END;
TRESLONG = INTEGER!32! (contrôle des débordements)

Les variables MAX et MAX10 sont utilisées pour représenter la plus grande mantisse et son quotient par 10. Les autres variables ne sont utilisées que par le programme principal servant aux essais, mais la place naturelle de ces procédures et fonctions est dans une UNIT en librairie appelée quand le besoin s'en fait sentir.

Remarquons que les modifications de MINEXPO, MAXEXPO et SIGNIFICATIF peuvent fournir des procédures et fonctions pour un calcul à 18, 24, ... chiffres significatifs, et à exposant différent de ± 30 . La seule précaution à prendre sera de contrôler que la variable "DEBORDE" peut effectivement contenir le produit MAX*MAX.

Premières procédures

PCAR, PCHaine, PREEL, DIAGERREUR ont déjà été décrites dans de précédents articles de Pom's.

La procédure INITMATH fabrique MAX et MAX10.

La procédure NORMALISE aligne le nombre à droite en supprimant les zéros s'il y en a et en incrémentant l'exposant.

La procédure SIGNE renvoie + 1 ou - 1 selon le signe de TL.

DIV10 donne l'arrondi du nombre.

CONTROLLER essaie, si nécessaire, de "faire tenir" le nombre dans la définition en fonction de MAX, MAXEXPO et MINEXPO et renvoie VRAI s'il y parvient.

Procédure EGALISEXPO

EGALISEXPO est utilisée par d'autres procédures du système ; elle permet d'égaliser les exposants de deux nombres afin de faciliter les comparaisons, additions ou soustractions.

Si, à l'entrée de la procédure, les exposants sont déjà égaux, on réalise un EXIT. Si l'exposant du premier nombre est plus grand, on le décroît progressivement en multipliant

la mantisse par 10, ce qui fait alors perdre des chiffres significatifs.

Fonctions de comparaison

Les fonctions de comparaison EGAL, INFÉRIEUR et SUPÉRIEUR appellent EGALISEXPO, comparent les mantisses et renvoient vrai ou faux. Elles sont toutes trois appelées par COMPARE qui reçoit en paramètre M1 et M2 (les deux réels à comparer) et CA, qui est un caractère de comparaison "=", "<" ou ">". Il est bien entendu possible d'appeler directement chaque fonction ou encore, en définissant un type FLAG = (SUP, EG, INF), d'écrire COMPARE :



la mantisse chaque fois par 10, ceci s'arrêtant soit quand une mantisse devient supérieure ou égale à MAX10, soit quand un exposant devient inférieur ou égal à MINEXPO. Ce procédé assure que l'ajustement se fait sans perte de chiffres significatifs.

Si à l'entrée de la procédure l'exposant du premier nombre est plus grand que celui du second, on appelle de façon récursive EGALISEXPO en inversant les deux nombres.

Si l'on sort de la première boucle WHILE sans égalité des exposants, on procède de façon inverse en incrémentant l'exposant du deuxième nombre et en divisant chaque fois sa

```
FONCTION COMPARE(M1,M2 :
REEL) :FLAG;
```

```
Begin
IF EGAL(M1,M2) THEN
  COMPARE := EG
ELSE IF SUPÉRIEUR(M1,M2)
  THEN COMPARE := SUP
  ELSE COMPARE := INF
End;
```

Procédures arithmétiques

Les procédures arithmétiques ont pour rôle, entre autres, de contrôler que l'on ne dépasse jamais les bornes fixées pour la représentation de la mantisse et de l'exposant du résultat. Dans le cas contraire, le drapeau ERREUR est mis à 1. Toutes

les procédures éliminent par avance les cas triviaux $M1=0$ ou $M2=0$.

1. ADDITION

Gestion des cas $M1=0$ et $M2=0$.
Ensuite, égalisation des exposants.
Définition de la variable DEBORDE qui contient la somme des mantisses.
Si CONTROLER est vrai, on crée RESULT et on le normalise, sinon : erreur de débordement.

2. SOUSTRACTION

On transforme $M2$ en $-M2$ et on passe à ADDITION.

3. MULTIPLIE

Elimine les cas triviaux, sinon DEBORDE est égal au produit des mantisses et I à la somme des exposants.
Si CONTROLER est vrai, on fait RESULT et on le normalise, sinon ERREUR est mis à 1.

4. DIVISE

Gestion des cas $M1=0$ et $M2=0$.
Sinon, afin de garder le plus de chiffres significatifs possible, la mantisse du dividende est multipliée par 10 autant que faire se peut sans dépasser MAX et sans que son exposant devienne inférieur à $MINEXPO+M2*EXPO$.
Ensuite, I prend la valeur de la différence des exposants, puis la validité de DEBORDE et de I est contrôlée. Si tout va bien, RESULT est défini et normalisé.

Toutes ces procédures sont appelées par la procédure CALCUL. Elles peuvent être appelées directement, l'appel de calcul se faisant par exemple par :

CALCUL(R1,'+',R2,SOMME)

ou par :

DIVISE(R1,R2,QUOTIENT)

Procédure FORMATAGE

Cette dernière procédure est un PRINTUSING défini de la manière suivante : la position des différents chiffres est indiquée par un "#" et le point décimal, s'il y en a un, par le symbole ".". Si l'affichage se fait à l'aide d'un exposant, la position de ce dernier sera marquée par des "^".

Le format est décrit dans une chaîne de caractères. Tous les caractères alphabétiques de cette chaîne seront conservés tels quels dans l'édition, y compris les espaces éventuels. Si le premier caractère du format est un "*", ce signe remplacera tous les blancs du format jusqu'au premier chiffre. Ainsi, l'utilisation du format "Prix=-# #.# #-Fr." fera apparaître à l'écran "PRIX=- -12.75-Fr." celle de "Valeur : -# # #-Exposant ^^^^".

donnera "Valeur : -12 -Exposant + 12"

et le format '*-# # # #-' aboutira à "*****12-"

Si un nombre ne peut être représenté dans le format donné, il est remplacé par des "?".

La procédure FORMATAGE a pour paramètres le réel $M1$, le format FM et la chaîne résultante SR. Le corps de la procédure détecte tout d'abord les erreurs, l'absence de format ou de chiffres dans le format.

Ensuite, on regarde si le premier caractère est un "*". Si oui, le booléen COMPLET est mis à vrai et le premier caractère du format "*" est supprimé; ensuite, LF prend pour valeur la longueur totale du format et l'on fabrique la chaîne vide résultat de même longueur en forçant le caractère SR!O! à la longueur du format, puis en remplissant cette chaîne d'espaces grâce à la procédure FILL-CHAR.

PEXPO prend alors pour valeur la position de l'exposant dans le format et PDEC celle du point décimal dans le format. Si les deux existent, c'est la représentation avec exposant qui est choisie.

Les procédures EXPOSANT et DECIMALE sont appelées selon le cas et servent à fabriquer SNUM, chaîne de caractères image du nombre. S'il n'y a pas eu d'erreur et si NUM a une longueur inférieure ou égale à celle du format, on recopie grâce à MOVERIGHT SNUM dans le format en prenant la position du chiffre de gauche comme repère. Si SNUM est trop grand, le format est totalement rempli par des "?".

Ensuite, tous les caractères autres que les espaces et les signes de syntaxe du format sont recopiés dans SR. Enfin, si la variable COMPLET est vraie, tout le début du format jusqu'au premier chiffre est rempli avec des "*".

La procédure EXPOSANT est utilisée par FORMATAGE pour placer l'exposant dans SR; on repère d'abord le dernier chiffre de droite dans le format par un SCAN de droite à gauche, et sa valeur est placée dans PDEC. Ensuite, on fabrique la chaîne SNUM représentant le nombre grâce à la procédure STR et l'on calcule le nombre de chiffres NC permis par le format (position du dernier moins position du premier).

Si ce nombre est plus petit que celui autorisé, on supprime des chiffres à droite par :
SNUM := COPY(SNUM,1,NC)
puis on corrige l'exposant en conséquence. On construit ensuite l'exposant par STR. On ajoute un zéro à

gauche si nécessaire; on place le signe devant, puis le E et enfin l'exposant, transféré dans SR par un MOVELEFT.

La procédure DECIMALE est appelée si le format ne comporte pas d'exposant. On repère d'abord deux cas : ou PDEC est différent de zéro et il y a des décimales à afficher, ou il est égal à zéro et il n'y a pas de décimale.

S'il y a des décimales et si le point décimal est le dernier signe (un point non suivi de "#"), le nombre de décimales NDEC est mis à zéro.

Sinon, NDEC est calculé comme distance entre le dernier "#" (à droite) et le point décimal. On appelle alors la procédure AJUSTE qui dans toute la mesure du possible va ajuster le nombre fourni pour le "livrer" avec le nombre de décimales voulues.

Après quoi, PDEC est ramené à l'extrême droite. On crée SNUM comme précédemment. On ajoute le nombre de zéros voulu à gauche du nombre si nécessaire et l'on place le point décimal à la place calculée. Si le nombre n'a pas de décimale, le processus est le même, mais simplifié du positionnement du point décimal.

Le corps du programme n'a d'utilité que pour permettre de contrôler les procédures et fonctions, fournir un nombre, un opérateur, un deuxième nombre et réafficher l'opération sous forme normalisée. Le CASE OF sert à départager ce qui doit être envoyé à CALCUL de ce qui est envoyé à COMPARE.

Pour le calcul, si ERREUR reste à zéro, on écrit le résultat normalisé, puis on demande le format (donné sous forme de chaîne de caractères); cette demande est répétée jusqu'à l'entrée d'un simple RETURN.

Pour la comparaison, affichage de vrai ou faux selon le résultat.

Puis fin de la boucle REPEAT sur "Voulez-vous continuer?"

A noter qu'au lieu d'utiliser directement les caractères "+", "-", "*", etc., il peut être plus simple de définir un type :

OPERATEUR = (PLUS,MOINS,
FOIS,QUOT,EG,
SUP,INF)

cela sera plus élégant, plus "Pascal", mais peut-être moins lisible. C'est pour cette raison que j'utilise directement les caractères dans ce programme.

Par ailleurs, COMPARE pourrait très facilement s'enrichir de "<=", ">=" ou "<>" avec la création des fonctions LE, GE et NE. FORTRAN n'est pas si loin...

```
PROGRAM MATH12;
```

```
CONST MINEXPO = -30; MAXEXPO = 30; SIGNIFICATIF = 12;
TYPE REEL = RECORD
    EXPO : MINEXPO..MAXEXPO;
    MANTISSE: INTEGER[SIGNIFICATIF]
END;
TRESLONG = INTEGER[32];
CHOIDECA = SET OF CHAR;
```

```
VAR R1,R2,
    RESULTAT : REEL;
    MAX,MAX10 : INTEGER[SIGNIFICATIF];
    DEBORDE : TRESLONG;
    SRESULT : STRING;
    ERREUR : INTEGER;
    OP : CHAR;
    FORMAT : STRING;
```

```
function PCAR(BONSET:CHOIDECA):CHAR;
```

```
var CH :CHAR;
    BON :BOOLEAN;
```

```
begin
    repeat
        READ(KEYBOARD,CH);
        if EOLN(KEYBOARD) then CH:=CHR(13);
        BON:=CH in BONSET;
        if not BON then WRITE(CHR(7))
            else if CH in [' '..CHR(130)] then WRITE(CH);
    until BON;
    PCAR:=CH
end;
```

```
procedure PCHAINE(LONGMAX:INTEGER; BONSET:CHOIDECA; var S:STRING);
```

```
var S1 :STRING[1];
    CONT :STRING;
    I :INTEGER;
```

```
begin
    S1:=' '; CONT:='';
    repeat
        if LENGTH(CONT)=0 then S1[1]:=PCAR(BONSET+[CHR(13)])
            else if LENGTH(CONT)=LONGMAX then S1[1]:=PCAR([CHR(13),CHR(8)])
            else S1[1]:=PCAR(BONSET+[CHR(13),CHR(8)]);
        if S1[1] in BONSET then CONT:=CONCAT(CONT,S1)
            else if S1[1]=CHR(8) then
            begin
                WRITE(CHR(8),' ',CHR(8));
                DELETE (CONT,LENGTH(CONT),1)
            end;
    until S1[1]=CHR(13);
    if LENGTH(CONT)<>0 then S:=CONT else WRITE(S)
end;
```

```
PROCEDURE INITMATH;
```

```
VAR BOUCLE :INTEGER;
```

```
BEGIN
    MAX:=0;
    FOR BOUCLE:=1 TO SIGNIFICATIF DO MAX:=MAX*10+9;
    MAX10:=MAX DIV 10
END;
```

```
PROCEDURE DIAGERREUR(ER:INTEGER);
```

```
BEGIN
    CASE ER OF
        1:WRITE('D)bordement');
        2:WRITE('Trop petit');
        3:WRITE('Division par zero')
    END
END;
```



```

PROCEDURE NORMALISE(VAR RL:REEL);
BEGIN
  WITH RL DO
    BEGIN
      IF MANTISSE=0 THEN EXPO:=0
      ELSE BEGIN
        WHILE ((MANTISSE-(MANTISSE DIV 10)*10)=0) AND (EXPO<MAXEXPO) DO
          BEGIN
            MANTISSE:=MANTISSE DIV 10; EXPO:=EXPO+1
          END
        END
      END
    END
  END;

FUNCTION SIGNE(TL:TRESLONG):INTEGER;
BEGIN
  IF TL<0 THEN SIGNE:=-1 ELSE SIGNE:=1
END;

PROCEDURE DIV10(VAR M:TRESLONG);
BEGIN
  IF M>0 THEN M:=(M+5) DIV 10 ELSE M:=(M-5) DIV 10
END;

FUNCTION CONTROLLER(VAR TL:TRESLONG;VAR NP:INTEGER):BOOLEAN;
BEGIN
  WHILE ((SIGNE(TL)*TL>MAX) AND (NP<MAXEXPO)) OR
    ((SIGNE(TL)*TL>0) AND (NP<MINEXPO)) DO
    BEGIN
      DIV10(TL); NP:=NP+1
    END;
  WHILE ((SIGNE(TL)*TL<MAX)) AND (NP>MAXEXPO) DO
    BEGIN
      TL:=TL*10; NP:=NP-1
    END;
  CONTROLLER:=(NP>=MINEXPO) AND (NP<=MAXEXPO) AND (SIGNE(TL)*TL<=MAX)
END;

PROCEDURE PREEL(VAR RL:REEL);
VAR SRL           :STRING;
    DEC,XP        :BOOLEAN;
    I,NP          :INTEGER;
    MOINS,BON     :BOOLEAN;
BEGIN
  REPEAT
    RL.EXPO:=0; RL.MANTISSE:=0; NP:=0; SRL:=''; DEBORDE:=0;
    DEC:=FALSE; MOINS:=FALSE; XP:=FALSE; BON:=TRUE;
    PCHaine(SIGNIFICATIF+4,['+', '-','0'..'9', '.', 'E'], SRL);
    IF SRL='' THEN EXIT(PREEL);
    IF SRL[1]='-' THEN MOINS:=TRUE;
    I:=1;
    WHILE (I<=LENGTH(SRL)) AND NOT XP DO
      BEGIN
        IF SRL[I]='.' THEN DEC:=TRUE
        ELSE IF SRL[I]='E' THEN XP:=TRUE
        ELSE IF SRL[I] IN ['0'..'9'] THEN
          BEGIN
            DEBORDE:=DEBORDE*10+(ORD(SRL[I])-ORD('0'));
            IF DEC THEN RL.EXPO:=RL.EXPO-1
          END;
        I:=I+1
      END;
    IF MOINS THEN DEBORDE:=-DEBORDE; MOINS:=FALSE;
    WHILE I<=LENGTH(SRL) DO
      BEGIN
        IF SRL[I]='-' THEN MOINS:=TRUE
        ELSE IF SRL[I] IN ['0'..'9'] THEN NP:=NP*10+(ORD(SRL[I])-ORD('0'));
      END;
  UNTIL DEBORDE=0 OR NP>=MAXEXPO OR MOINS;
  RL.EXPO:=RL.EXPO+DEBORDE;
  RL.MANTISSE:=RL.MANTISSE DIV NP;
  RL.EXPO:=RL.EXPO-INT(LENGTH(SRL));
  NORMALISE( RL );
  IF BON THEN RETURN TRUE ELSE RETURN FALSE;
END;

```

```

    I:=I+1
END;
IF MOINS THEN NP:=-NP;
NP:=NP+RL.EXPO;
IF CONTROLER(DEBORDE,NP) THEN
BEGIN
    RL.MANTISSE:=DEBORDE;  RL.EXPO:=NP;  NORMALISE(RL);
    END ELSE BEGIN WRITELN('**** ERREUR');  BON:=FALSE END
UNTIL BON
END;
(***
(* FONCTIONS DE COMPARAISON *)
(***)
PROCEDURE EGALISEXPO(VAR M1,M2:REEL);
BEGIN
    IF M1.EXPO=M2.EXPO THEN EXIT(EGALISEXPO);
    IF M1.EXPO>M2.EXPO THEN
    WHILE (M1.EXPO>M2.EXPO) AND (M1.EXPO>MINEXPO)
        AND (SIGNE(M1.MANTISSE)*M1.MANTISSE<MAX10) DO
    BEGIN
        M1.EXPO:=M1.EXPO-1;  M1.MANTISSE:=M1.MANTISSE*10
    END ELSE EGALISEXPO(M2,M1);
    WHILE (M1.EXPO>M2.EXPO) AND (M2.EXPO<MAXEXPO) AND (M2.MANTISSE<>0) DO
    BEGIN
        M2.EXPO:=M2.EXPO+1;  M2.MANTISSE:=M2.MANTISSE DIV 10
    END;
END;

FUNCTION EGAL(M1,M2:REEL):BOOLEAN;
BEGIN
    EGALISEXPO(M1,M2);  EGAL:=M1.MANTISSE=M2.MANTISSE
END;

FUNCTION SUPERIEUR(M1,M2:REEL):BOOLEAN;
BEGIN
    EGALISEXPO(M1,M2);  SUPERIEUR:=M1.MANTISSE>M2.MANTISSE
END;

FUNCTION INFERIEUR(M1,M2:REEL):BOOLEAN;
BEGIN
    EGALISEXPO(M1,M2);  INFERIEUR:=M1.MANTISSE<M2.MANTISSE
END;

FUNCTION COMPARE(M1:REEL;CA:CHAR;M2:REEL):BOOLEAN;
BEGIN
    CASE CA OF
        '=':COMPARE:=EGAL(M1,M2);
        '<':COMPARE:=INFERIEUR(M1,M2);
        '>':COMPARE:=SUPERIEUR(M1,M2);
    END
END;

(* PROCEDURES DE CALCUL *)
(***)
PROCEDURE ADDITION(M1,M2:REEL;VAR RESULT:REEL);
BEGIN
    IF M1.MANTISSE=0 THEN RESULT:=M2
    ELSE IF M2.MANTISSE=0 THEN RESULT:=M1
        ELSE BEGIN
            EGALISEXPO(M1,M2);  DEBORDE:=M1.MANTISSE+M2.MANTISSE;
            IF CONTROLER(DEBORDE,M1.EXPO) THEN
            BEGIN
                RESULT.EXPO:=M1.EXPO;  RESULT.MANTISSE:=DEBORDE;
                NORMALISE(RESULT)
            END ELSE ERREUR:=1;
        END
    END
END;

```

```

PROCEDURE SOUSTRACTION(M1,M2:REEL;VAR RESULT:REEL);
BEGIN
  M2.MANTISSE:=-M2.MANTISSE;  ADDITION(M1,M2,RESULT)
END;

PROCEDURE MULTIPLIE(M1,M2:REEL;VAR RESULT:REEL);
VAR I      :INTEGER;
BEGIN
  IF (M1.MANTISSE=0) OR (M2.MANTISSE=0) THEN
    BEGIN
      RESULT.MANTISSE:=0;  RESULT.EXPO:=0
    END ELSE
    BEGIN
      DEBORDE :=M1.MANTISSE*M2.MANTISSE;  I:=M1.EXPO+M2.EXPO;
      IF CONTROLER(DEBORDE,I) THEN
        BEGIN
          RESULT.EXPO:=I;  RESULT.MANTISSE:=DEBORDE;  NORMALISE(RESULT)
        END ELSE ERREUR:=1
      END
    END
  END;

PROCEDURE DIVISE(M1,M2:REEL;VAR RESULT:REEL);
VAR I      :INTEGER;
BEGIN
  I:=M1.EXPO;
  IF M2.MANTISSE=0 THEN ERREUR:=3
  ELSE IF M1.MANTISSE=0 THEN
    BEGIN
      RESULT.MANTISSE:=0;  RESULT.EXPO      :=0
    END ELSE
    BEGIN
      WHILE (SIGNE(M1.MANTISSE)*M1.MANTISSE<MAX10) AND
            (I>MINEXPO+M2.EXPO) DO
        BEGIN
          M1.MANTISSE:=M1.MANTISSE*10;  I:=I-1;
        END;
        I:=I-M2.EXPO;
        DEBORDE:=M1.MANTISSE DIV M2.MANTISSE;
        IF CONTROLER(DEBORDE,I) THEN
          BEGIN
            RESULT.MANTISSE:=DEBORDE;  RESULT.EXPO:=I;  NORMALISE(RESULT)
          END ELSE ERREUR:=2;
        END
      END
    END
  END;

PROCEDURE CALCUL(M1:REEL;CA:CHAR;M2:REEL;VAR RESULT:REEL);
BEGIN
  ERREUR:=0;RESULT.EXPO:=0;RESULT.MANTISSE:=0;
  CASE CA OF
    '+' :ADDITION(M1,M2,RESULT);
    '-' :SOUSTRACTION(M1,M2,RESULT);
    '*' :MULTIPLIE(M1,M2,RESULT);
    '/' :DIVISE(M1,M2,RESULT);
  END;
END;

(?!*)
(* FORMATAGE DU RESULTAT *)
(***)
PROCEDURE FORMATAGE(M1:REEL;FM:STRING;VAR SR:STRING);
VAR LF,PEXPO,I,NC,PDEC,NDEC      :INTEGER;
  SNUM,SEXPO      :STRING;
  COMPLET        :BOOLEAN;

PROCEDURE AJUSTE;
BEGIN
  WITH M1 DO
    BEGIN

```

```

IF EXPO=-NDEC THEN EXIT(AJUSTE);
WHILE (EXPO>-NDEC) AND (MANTISSE<MAX10) AND (EXPO>MINEXPO) DO
BEGIN
  MANTISSE:=MANTISSE * 10;  EXPO:=EXPO-1
END;
WHILE (EXPO<-NDEC) AND (MANTISSE>0) AND (EXPO<MAXEXPO) DO
BEGIN
  MANTISSE:=MANTISSE DIV 10;  EXPO:=EXPO+1
END;
IF EXPO<>-NDEC THEN ERREUR:=1
END;
END;

```

```

PROCEDURE EXPOSANT;
BEGIN
  PDEC:=LF+SCAN(-LF,='#',FM[LF])+1;  NC:=PDEC-POS('#',FM);
  STR(M1.MANTISSE,SNUM);
  IF NC<LENGTH(SNUM) THEN
    IF (MAXEXPO>=M1.EXPO+LENGTH(SNUM)-NC) THEN
      BEGIN
        M1.EXPO:=M1.EXPO+LENGTH(SNUM)-NC;  SNUM:=COPY(SNUM,1,NC);
        END ELSE BEGIN ERREUR:=1; EXIT(EXPOSANT) END;
  STR(ABS(M1.EXPO),SEXPO);
  WHILE LENGTH(SEXPO)<2 DO SEXPO:=CONCAT('0',SEXPO);
  IF M1.EXPO<0 THEN SEXPO:=CONCAT('-',SEXPO)
    ELSE SEXPO:=CONCAT('+',SEXPO);
  SEXPO:=CONCAT('E',SEXPO);  MOVELEFT(SEXPO[1],SR[PEXPO],4);
END;

```

```

PROCEDURE DECIMAL;
BEGIN
  IF PDEC<>0 THEN
    BEGIN
      IF PDEC=LF THEN NDEC:=0 ELSE NDEC:=SCAN(LF-PDEC,<,'#',FM[PDEC+1]);
      AJUSTE;  PDEC:=PDEC+NDEC+1;  STR(M1.MANTISSE,SNUM);
      WHILE(LENGTH(SNUM)<NDEC+1) DO SNUM:=CONCAT('0',SNUM);
      INSERT('.',SNUM,LENGTH(SNUM)-NDEC+1);
    END ELSE
    BEGIN
      NDEC:=0;  PDEC:=LF+SCAN(-LF,='#',FM[LF])+1;  AJUSTE;
      STR(M1.MANTISSE,SNUM)
    END
  END;

```

```

BEGIN (* FORMATE *)
  ERREUR:=0;  COMPLET:=FALSE;  SR:='';
  IF (FM='') OR (POS('#',FM)=0) THEN EXIT(FORMATE);
  IF FM[1]='*' THEN
    BEGIN
      COMPLET:=TRUE;  FM:=COPY(FM,2,LENGTH(FM)-1)
    END;
  LF:=LENGTH(FM);
  (*$R-*)
  SR[0]:=CHR(LF);
  (*$R+*)
  FILLCHAR(SR[1],LF,' ');  PEXPO:=POS('^^^^',FM);  PDEC:=POS('.',FM);
  IF PEXPO<>0 THEN EXPOSANT ELSE DECIMAL;
  IF (PDEC-POS('#',FM)>=LENGTH(SNUM)) AND (ERREUR=0)
    THEN MOVERIGHT(SNUM[1],SR[PDEC-LENGTH(SNUM)],LENGTH(SNUM))
    ELSE FILLCHAR(SR[1],LF,'?');
  FOR I:=1 TO LF DO IF NOT(FM[I] IN [' ','#','.','^']) THEN SR[I]:=FM[I];
  I:=1;
  IF COMPLET THEN
    WHILE (NOT (SR[I] IN ['0'..'9','?'])) AND (I<=LF) DO
      BEGIN SR[I]:='*';I:=I+1 END;
END;

```

```

BEGIN (* PROGRAMME PRINCIPAL POUR ESSAI *)

```

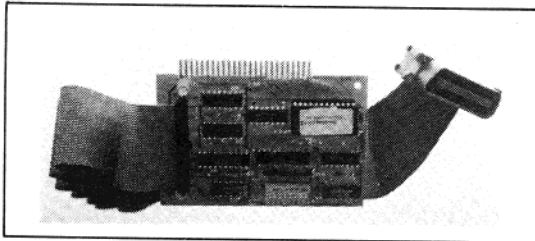
```

INITMATH;
REPEAT
PAGE(OUTPUT);
WRITE('Programme d''essai -calcul a 12 chiffres');WRITELN;WRITELN;
WRITE('N1 : ');PREEL(R1);
WRITE(' Op : ');OP:=PCAR(['+', '-', '*', '/', '=', '<', '>']);
WRITE(' N2 : ');PREEL(R2);WRITELN;
WRITE(R1.MANTISSE, 'E', R1.EXPO, OP, R2.MANTISSE, 'E', R2.EXPO, '=');
CASE OP OF
'+', '-', '*', '/':BEGIN
CALCUL(R1,OP,R2,RESULTAT); FORMAT:='?';
IF ERREUR=0 THEN
BEGIN
WRITELN(RESULTAT.MANTISSE, 'E', RESULTAT.EXPO);
WHILE FORMAT<>' ' DO
BEGIN
FORMAT:=' '; WRITE('Format : ');
PCHaine(20,[' '.CHR(126)],FORMAT);
IF FORMAT<>' ' THEN
BEGIN
WRITE(' -->');
FORMATAGE(RESULTAT,FORMAT,SRESULT);WRITELN(SRESULT)
END
END
END ELSE DIAGERREUR(ERREUR)
END;
'=', '<', '>' :IF COMPARE(R1,OP,R2) THEN WRITE('vrai') ELSE WRITE('faux');
END;
WRITELN;WRITE('Continuer (O/N) ? ');
UNTIL PCAR(['O', 'N', 'o', 'n']) IN ['N', 'n']
END.

```

DUMPLING

CARTE INTERFACE GRAPHIQUE HAUTE
RÉSOLUTION AVEC OU SANS BUFFER



DUMPLING-GX est une carte d'interface parallèle graphique qui permet d'établir une communication intelligente entre un APPLE II et la plupart des imprimantes. La sélection d'imprimante se fait par DIP Switch. EPSON - C-ITOH - NEC - OKIDATA - CENTRONICS - TALLY - APPLE. De simples commandes permettent de gérer la mise en page et les fonctions graphiques, entre autres :

- Longueur de ligne et de page · Marges gauche et droite · Copie d'écran texte · Simulation de graphiques en continu · Copie d'écran graphique page 1 et/ou 2 · Inversion (noir/blanc) · Rotation 90° · Double format · Double frappe

DUMPLING-64 offre toutes les possibilités de DUMPLING-GX avec en plus un buffer/spooler de 64 K. Principales caractéristiques :

- Reconnaissance automatique de la taille du buffer · Multiples copies d'écran consécutives · Compression des espaces · Pause immédiate ou différée · Possibilité de commande à distance · Compatible CP/M ou Pascal.

DUMPLING-GX = 1650 FHT DUMPLING-32 = 2580 FHT
DUMPLING-16 = 2390 FHT DUMPLING-64 = 3390 FHT



sofitec 207 rue Gallieni 92100 BOULOGNE
tél. (1) 605.88.78

IS PipeLine™

UN NOUVEAU CONCEPT

PIPELINE est une mémoire-tampon intelligente qui introduit un nouveau concept révolutionnaire - RANDOM ACCESS PRINTING - qui permet :

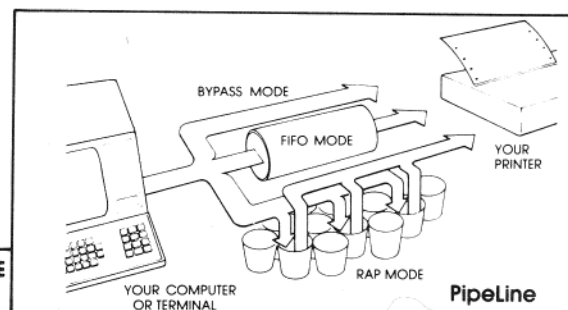
- de composer, réorganiser, dupliquer les données à imprimer ;
- de combiner texte, tableaux, graphiques dans une même impression ;
- de générer automatiquement des états personnalisés à partir de lettres, mémos, rapports ;
- de recopier le document un nombre quelconque de fois.

Le mode RAP permet ainsi de faire travailler ensemble plusieurs programmes pour donner une impression combinée.

Le mode FIFO imprime dans l'ordre des entrées (mode conventionnel).

Le mode BYPASS permet la suspension d'une impression longue pour éditer une impression courte.

IS PIPELINE est facilement extensible de 8 K à 128 K. Il s'intercale entre tout ordinateur et imprimante avec interface parallèle type Centronics.



Deux ans de Pom's : sommaire thématique

Alexandre Duback

Beaucoup de nouveaux lecteurs nous posent des questions dont les réponses se trouvent dans les numéros de Pom's déjà parus. Nous avons donc décidé de reprendre ici le sommaire, classé par thèmes, des huit premiers numéros de Pom's. Il est à noter que les quatre premiers numéros ont été réédités dans le recueil de Pom's numéro 1, que l'on peut commander directement aux Editions MEV.

Matériels et cartes

	Numéro
Inverseur DOS 3.2-DOS 3.3	1
Survol de l'Apple ///	2
Les mémoires de masse	4
La carte M/DOS 6502 à l'essai	4
Banc-test de la carte Legend 128K DE	5
La souris de Lisa	7
L'Apple //e à l'essai	7
Le Basis 108 à l'essai	8

Logiciels commerciaux

Visicalc et Applesoft	1
P.L.E. : le Program Line Editor	1
C.R.A.E. : Co-Resident Applesoft 1 Editor	1
Les utilitaires de documentation :	
Dakin 5 - Apple Doc - DOS Tool Kit	2
Applewriter - Easywriter - Magic Window	
Les éditeurs de texte	2
CCA DMS - PFS - Visifile - DB Master	
Bases de données sur Apple	5
CX Multigestion à l'essai	5
The Last One à l'essai	6
C.O.R.P. à l'essai	6
Visicalc et traitement de texte	7
Le cours de BASIC Applesoft André Finot	7
Multiplan à l'essai	7
PILOT et SuperPILOT à l'essai	8
Présentation du H-BASIC	8

Graphiques

Graphiques : de l'ITT 2020 à l'Apple	1
Les adresses du graphique	1
Applications de graphique H.R.	2
Contrôlez le nettoyage mémoire	3
Routine de présentation graphique	3
Création de tables de formes	4
Création graphique en Pascal	5
Logiciel graphique en Pascal	6
Graphique, quand tu nous tiens...	7
Graphiques et logique	7
Hard Copy Seikosha GP80	7
Un programme de fondu enchaîné	7
Création de caractères graphiques	8

Jeux et loisirs

Un programme aide-mémoire	1
Changez votre poignée de jeu	1
La leçon de calcul	2
Le jeu de la vie	2
Réponse au concours de Pom's	4
Robotwar	4

Tortue Ampersand	6
Le loto, c'est facile...	6
Cryptographie à clef publique	7
Les quatre ponts	7

Programmes utilitaires

Des instructions en une lettre	1
Déplacement du programme en assembleur	1
Analyse du contenu des slots	1 **
Réparez votre APPEND	1,2 **
Réparez votre RENUMBER	2 **
Faites le ménage dans la mémoire	2
Sprechen Sie DOS ?	2
S.H.LAM : une routine bien pratique	2 **
Un exemple de HELLO	3
Copie d'écran texte	3
Personnalisez vos disquettes	3
Un programme de TRACE sélective	4
Un catalogue général en Pascal	2,3,4
Chargez vite vos fichiers binaires	4
Un PRINT USING d'intérêt général	4
Le clavier magique	5
Transfert d'Applesoft vers EXEC	6
Un programme de HELLO complet	6
Un analyseur de syntaxe	6
Dump Pascal	7
Gestion de masques en BASIC	7
FID, MUFFIN et DEMUFFIN	7
Boot P.L.E. + C.R.A.E.	7
Francisez le DOS	8
Calculs en format gestion	8
Recherche de codes binaires	8
Le Pascal à 12 chiffres	8

Pour faire son apprentissage

Programmer en Pascal	1 *
Formatez vos programmes	2 **
Apprentissage de l'assembleur	3,4
Les fichiers EXEC	3
Notions de base (chargement de binaire)	3
Notions de base : les fichiers	4
Notions de base : INPUT généralisé	5
Ergonomie des programmes	5
Des programmes relogeables	7
Création de fichiers EXEC	7
Mini-base de données	8
Notions de base : gestion de fichiers	8
Accélérez vos programmes en BASIC	8
Conseils aux débutants	8

Pour devenir expert sur Apple

Overlay dynamique	1
IncurSION dans les mystères du DOS	2
3 secondes pour trier	2
Conversion Pascal/BASIC/Pascal	3
Les codes ASCII épluchés	4
Les arcanes du moniteur Apple ///	7
Le moniteur étendu	8

Et tant d'autres articles...

Des boucles à s'arracher les cheveux	1 **
Communication grâce à l'Apple	3

Communiquez grâce au format DIF	4
Un Apple à la clinique	4 *
La 7me W.C.C.F.	4 *
La programmation facilitée	5
HAIFA : un amper-interpréteur complet	5
Tableaux de taille déclarée en Pascal	6
Un générateur	7
Un programme de test universel	7
Effacement de directory en Pascal	7
Allo, Questel ?	8
Micro-informations : les nouveautés	8
Saisie de réels en Pascal	8

* Article non repris dans le Recueil 1 de Pom's
 ** Parus dans la rubrique Trucs et Astuces

Remarques

De nombreux articles pouvaient être classés dans plusieurs catégories : ainsi HAIFA est à la fois un outil de création graphique, un merveilleux utilitaire, l'occasion de devenir un expert sur Apple (si l'on analyse les 150 secteurs en source Lisa 2.5 !). Nous nous sommes efforcés, dans les cas de ce genre, de placer l'article dans la catégorie à laquelle il pouvait le plus s'identifier.

D'autre part, il ne faut pas oublier que, même si la compréhension d'un programme peut vous sembler impossible, cela ne vous empêche pas de l'utiliser. On utilise couramment des jeux, Visicalc, ou du traitement de texte, tout en étant pour la plupart d'entre nous incapables de comprendre le programme qui en est la base.

Conseils aux débutants

Alexandre Duback

Vous venez d'acquérir un Apple //e ou un Apple /// flamboyant neuf et vous vous trouvez confronté au premier problème : comment faut-il disposer le matériel ?

Installation

En effet, sur la plupart des publicités et brochures, l'on voit le moniteur posé sur les deux lecteurs de disquettes, alors que le manuel du DOS recommande explicitement d'éloigner les lecteurs d'un récepteur de télévision.

En fait, il n'y a pas péril en la demeure. La meilleure preuve est que le lecteur incorporé de l'Apple /// se trouve, par la force des choses, proche du moniteur !

Les seules fois où nous avons pu constater un problème, le moniteur avait une base en plastique. Cela se manifestait par des I/O ERRORs qui ont disparu quand on a déplacé les lecteurs de disquettes, pour les mettre à droite de l'Apple.

Précautions d'utilisation

La principale règle à respecter est la suivante : ne jamais mettre ou enlever une carte dans un appareil allumé, c'est la meilleure façon de griller la carte.

Le bouton d'allumage de l'Apple a toujours été son point faible, surtout en ce qui concerne l'Apple II Plus. Il est plus prudent (et plus simple) de brancher toutes les prises (Apple, moniteur, imprimante, accessoires éventuels) sur un râtelier de prises, l'idéal étant d'en avoir un muni d'un interrupteur. On laisse alors tous les appareils "allumés" en permanence, sauf l'imprimante qu'il est inutile de faire chauffer pour rien.

Ces précautions étant prises, on peut faire pratiquement ce que l'on veut. Il faut être très fort pour faire "griller" un Apple par programme : nous savons le faire, mais ne vous dirons pas comment ...

Les disquettes

A manier avec amour et délicatesse; il est indispensable à ce titre de lire la rubrique "Trucs et Astuces" à la page 15 du Pom's 6.

Muni d'une bonne paire de ciseaux, ou d'une pince à tiercé, vous pouvez diminuer par 2 votre budget disquettes en pratiquant une encoche opposée à l'encoche latérale

actuelle. Le verso de la disquette devient alors utilisable; il n'est pas garanti par le fabricant, mais ne pose pratiquement pas de problème.

Bien entendu, ne jamais forcer lors de l'introduction ou de l'extraction d'une disquette. Eviter la proximité de tout ce qui comporte des électro-aimants, le support magnétique étant par définition sensible à leur "influence". En particulier, ne pas poser le téléphone sur une disquette !

Que faut-il lire ?

Nous conseillons les ouvrages suivants :

Pratique de l'Apple II, par Henri Lilen (Editions Radio : facile à comprendre, très bien pour les débutants.

Visicalc sur Apple, par Hervé Thiriez (Editions du PSI) : indispensable pour une utilisation efficace de Visicalc.

BASIC Applesoft - Manuel de référence alphabétique, par Paul Merry : il fait bien ce qu'indique le titre. Par contre, mal encollé, il vous reste dans les mains.

Clefs pour l'Apple II, par Nicole Bréaud-Pouliquen (Editions du PSI) : voir la rubrique bibliographique de ce numéro.

La pratique de l'Apple II (volume III), par Nicole Bréaud-Pouliquen et Daniel-Jean David (Editions du PSI) : pour ceux qui désirent apprendre l'assembleur du 6502.

Ceci dit, une mine inépuisable de programmes, idées et renseignements se trouve à portée de votre main dans Pom's. Pourquoi vous en priveriez-vous ? Si une seule page lue dans un numéro de l'année vous évite de perdre une journée, vous avez largement amorti votre abonnement avec disquettes. En outre, de nombreux programmes fournis avec la revue (HAIFA, un programme de HELLO complet, l'analyseur de syntaxe, ...) supportent aisément la comparaison avec des programmes commerciaux vendus de 300 à 900 francs.

On nous demande souvent pourquoi il n'y a pas d'ouvrages dans notre langue sur l'Apple ///. Il y a deux principales raisons à cela. La première est que, pendant longtemps, les ventes d'Apple ///, assez faibles par rapport à celles de l'Apple II, ne justifiaient pas un tel effort. La seconde raison, toujours valable en bonne partie, est que la grande majorité des possesseurs de l'Apple /// sont des utilisateurs professionnels : ceux-ci lisent en général peu sur le sujet, se contentant souvent de faire tourner la machine à longueur de journée avec des programmes standards.

Le H-BASIC : un BASIC Pascalien

Olivier Herz

Pourquoi un nouveau langage ?

En pratiquant un peu le BASIC, on s'aperçoit très vite des limites du langage et peut-être un peu moins vite des mauvaises habitudes qu'il fait prendre: les GOTOs rendent les programmes difficilement lisibles, l'absence de variables locales et de vrais sous-programmes empêche de traiter des problèmes récurrents et rend difficile la structuration d'un programme.

Pour pallier ces limitations, on peut passer au Pascal, mais ce langage coûte cher, se montre plus complexe et plus difficile à déboguer, ne possède pas de graphisme basse-résolution et est très lourd dès qu'il s'agit d'utiliser des routines en assembleur. On peut aussi faire du Lisp ou du Logo, mais leur spécificité en fait des langages un peu à part. On peut enfin faire du FORTH (nous reparlerons certainement de ce langage plein d'avenir).

Nous avons préféré créer un langage hybride possédant les structures de bloc du Pascal et "collant" de près à l'Applesoft. Deux possibilités s'offraient pour cela: l'ampersand-interpréteur (c'est-à-dire ajouter des commandes à l'Applesoft par le biais de la commande ampersand &), qui posait des problèmes de lisibilité du listing, et le compilateur (c'est-à-dire un programme traduisant un programme source écrit en H-BASIC en un programme objet écrit en Applesoft), solution que nous avons retenue ici.

Caractéristiques du langage

Un programme en H-BASIC est formé de blocs emboîtés appelés procédures, qu'on peut appeler avec ou sans paramètres, par l'instruction @

Voici un exemple de programme écrit en H-BASIC :

```
procédure PRINCIPALE
var N, P, A$

procédure MOINS
in A: out B
begin
  B=-A
end

begin
input N
@ MOINS (N,P)
print P: get A$
end
```

Les variables sont celles de l'Applesoft, les tableaux jouant un rôle particulier: ils sont représentés par un dièse devant leur nom dans la partie déclaration de variables et passage de paramètres, et on peut définir une opération directe sur les tableaux si leurs dimensions coïncident: #A=#B

Les exemples ci-dessous montrent les principales instructions permises entre le "begin" et le "end" d'un bloc :

```
for I=1 to 10
  print I
endfor
```

```
while I<2000
  I=2*I
endwhile
```

```
until I<.01
  I=I*.3
enduntil
```

```
if A=1: B=2
elseif A=2: B=4
else B=0
endif
```

De plus, la plupart des instructions Applesoft sont permises (HOME, TEXT, &, etc.).

Un exemple parlant : les tours de Hanoï

Nous joignons un listing type du problème des tours de Hanoï, écrit en Applesoft, en Pascal, en H-BASIC et

en Applesoft après compilation du code H-BASIC.

Avantages et inconvénients du H-BASIC...

Sur l'Applesoft	Sur le Pascal
Mieux structuré Variables locales Affectation de tableaux Vrais sous-prog.	Moins complexe Debugage facile &-interpréteur Instruction DATA Graphisme basse résolution
Compilation assez longue Plus encombrant en mémoire	Lenteur du BASIC Peu de types de données Peu de noms de variables Pas de fonctions

Comment écrire en H-BASIC ?

Il faut créer un fichier TEXT contenant le programme H-BASIC. Deux méthodes sont possibles: utiliser un système de traitement de texte comme AppleWriter II ou bien

```
procedure HANOI
var #N$(2), N*
```

Hanoï H-BASIC

```
procedure DEPLACE: in A, B
begin
  print N$(A); " --> "; N$(B)
end
```

```
procedure TRANSFERT: in !TAILLE, !DEBUT, !FIN
var !AUTRE=3-DEBUT-FIN
begin
  if TAILLE=1: @ DEPLACE(DEBUT,FIN)
  else
    @ TRANSFERT(!TAILLE-1,!DEBUT,!AUTRE)
    @ DEPLACE(DEBUT,FIN)
    @ TRANSFERT(!TAILLE-1,!AUTRE,!FIN)
  endif
end

begin
  until N*>0
    text: home
    input "NOMBRE DE DISQUES? ";N*
    enduntil: N$(0)="GAUCHE"
    N$(1)="MILIEU": N$(2)="DROITE"
    @ TRANSFERT(N*,0,2)
end
```

l'écrire dans les REMs d'un pseudo-programme Applesoft et créer le fichier TEXT selon une procédure simple expliquée dans la documentation du H-BASIC.

Puis on lance le compilateur qui crée à partir de ce fichier un fichier TEXT contenant le programme Applesoft. Il suffira alors de faire EXEC "nom du fichier" et l'on obtiendra le programme Applesoft compilé qu'il faudra sauver par SAVE.

Que contient le progiciel H-BASIC ?

On y trouve bien sûr le compilateur, écrit en H-BASIC et sa version exécutable compilée en Applesoft, ainsi que quelques routines assembleur utiles au compilateur.

On trouve aussi tout un ensemble de routines nécessaires aux programmes une fois compilés, routines que l'on peut bien entendu utiliser directement dans un programme Applesoft : empilement et dépilement de variables et tableaux, affectation de tableaux, RESTORE NNN, GOSUBs quasi-illimités, tout cela par le biais d'une pile qui permet la récursivité en BASIC.

On trouve enfin un certain nombre de programmes de démonstration écrits en H-BASIC et compilés en Applesoft, comme le jeu "Le compte est bon", exemple récursif de l'intérêt du H-BASIC.

Conclusion

Nous conseillons bien entendu le H-BASIC à tous ceux qui sont à la recherche d'un langage facile et plus agréable que le BASIC. Nous insistons de plus sur le caractère didacti-

```
program HANOI;
var NOM: ARRAY[0..2] of INTEGER;
    N : INTEGER;
```

```
procedure DEPLACE (A,B:INTEGER);
begin
  writeln(NOM(A),' --> ',NOM(B))
end;
```

```
procedure TRANSFERT (TAILLE,DEBUT,FIN:INTEGER);
var AUTRE: INTEGER;
```

```
begin
  AUTRE:=3-DEBUT-FIN;
  if TAILLE=1 then DEPLACE(DEBUT,FIN) else
  begin
    TRANSFERT(TAILLE-1,DEBUT,AUTRE);
    DEPLACE(DEBUT,FIN);
    TRANSFERT(TAILLE-1,AUTRE,FIN)
  end
end;
```

```
begin
  repeat
    page(output);
    write('NOMBRE DE DISQUES? ');
    readln(N)
  until N>0; NOM(0):='GAUCHE';
  NOM(1):='MILIEU'; NOM(2):='DROITE';
  TRANSFERT(N,0,2)
end.
```

que de ce package, l'épluchage d'un compilateur écrit "dans lui-même" étant toujours un exercice passionnant, d'autant plus que l'utilisateur peut ainsi facilement modifier le langage à sa guise, sans parler des routines en assembleur qui rendent l'Applesoft récursif.

NB : l'existence de ces routines empêche une compilation du pro-

Hanoï Pascal

gramme Applesoft (TASC, Expediter,...).

Comme il aurait fallu consacrer plus de la moitié de la revue au mode d'emploi et au listing du H-BASIC, il nous a fallu décider d'en faire un progiciel à part. Vous ne trouverez donc pas le H-BASIC sur la disquette d'accompagnement. Celui-ci est vendu par correspondance au prix de 150 francs TTC, manuel d'utilisation compris.

Hanoï Applesoft

```
10 TEXT : HOME : INPUT "NOMBRE DE DISQU
    ES? ";N%: PRINT : IF N% < 1 THEN
    10
20 DIM DEBUT(N%),FIN(N%),AUTRE(N%),N$(2
):N$(0) = "GAUCHE":N$(1) = "MILIE
U":N$(2) = "DROITE":DEBUT(N%) = 0
:FIN(N%) = 2:AUTRE(N%) = 1:GOSUB
30: END
30 IF N% = 1 THEN A = DEBUT(N%):B = FIN
(N%):GOSUB 100: RETURN
40 N% = N% - 1:DEBUT(N%) = DEBUT(N% + 1)
:FIN(N%) = AUTRE(N% + 1):AUTRE(N%
) = FIN(N% + 1):GOSUB 30:A = DEB
UT(N% + 1):B = FIN(N% + 1)
50 GOSUB 100:DEBUT(N%) = AUTRE(N% + 1):
FIN(N%) = FIN(N% + 1):AUTRE(N%) =
DEBUT(N% + 1):GOSUB 30:N% = N%
+ 1: RETURN
100 PRINT N$(A);" --> ";N$(B): RETURN
```

Hanoï H-BASIC compilé

```
100 PRINT CHR$(4)"BLOAD PILE,A32000":
    HIMEM: 32000: DIM ZZ$(19),ZZ(19)
    ,YY(9),YY$(9): & GOSUB 110: END
109 REM
-- DEBUT DE LA PROCEDURE HANOI
110 GOTO 180
119 REM
-- PROCEDURE DEPLACE
120 PRINT N$(A);" --> ";N$(B): & RETUR
N
129 REM
-- PROCEDURE TRANSFERT
130 & > AUTRE:AUTRE = 3 - DEBUT - FIN:
    IF NOT (TAILLE = 1) THEN 150
140 A = DEBUT:B = FIN: & GOSUB 120: GOT
    O 170
150 ZZ(0) = TAILLE - 1:ZZ(1) = DEBUT:ZZ(
```

```

2) = AUTRE: & > TAILLE,DEBUT,FIN
:TAILLE = ZZ(0):DEBUT = ZZ(1):FIN
= ZZ(2): & GOSUB 130: & < FIN,
DEBUT,TAILLE:A = DEBUT:B = FIN: &
GOSUB 120:ZZ(0) = TAILLE - 1:ZZ
(1) = AUTRE:ZZ(2) = FIN: & > TAI
LLE,DEBUT,FIN
160 TAILLE = ZZ(0):DEBUT = ZZ(1):FIN = Z
Z(2): & GOSUB 130: & < FIN,DEBU
T,TAILLE
170 & < AUTRE: & RETURN

```

```

179 REM
-- CORPS DE LA PROCEDURE HANOI
180 & > #N$: DIM N$(2)
190 TEXT : HOME : INPUT "NOMBRE DE DISQ
UES? ";N$: IF NOT (N% > 0) THEN
190
200 N$(0) = "GAUCHE":N$(1) = "MILIEU":N$
(2) = "DROITE": & > TAILLE,DEBUT
,FIN:TAILLE = N%:DEBUT = 0:FIN =
2: & GOSUB 130: & < FIN,DEBUT,T
AILLE: & < #N$: & RETURN

```

Trucs et astuces

Les joies du PR#

PR# n reprogramme le saut de la sortie de caractères vers le slot n (0 pour l'écran) de même. IN# n reprogramme l'entrée de caractères (0 pour le clavier). Tout cela se fait à l'aide des routines situées de \$Cn00 à \$CnFF de la PROM du slot.

Or, si le slot est un contrôleur de disquettes, l'appel de la routine fait rebooter : ainsi, PR# 6 ou IN# 6 entraînent le boot.

Mais le boot n'a pas lieu tout de suite si l'on fait patienter avant la lecture ou l'écriture d'un caractère. Par exemple, faire en Applesoft :

```
:PR# 6:FOR I=1 to 10000:NEXT
```

Voir dans le Pom's, page 26, pourquoi nous débutons par "..."

On peut aussi lancer le programme :

```
10 PRINT CHR$(4)"PR# 6": IF PEEK (-16384)<128
THEN 10
```

Amusant, n'est-ce-pas ?

Apple // Apple ///

**leurs interfaces,
leurs périphériques,
leurs logiciels**

Démonstrations Jeudi

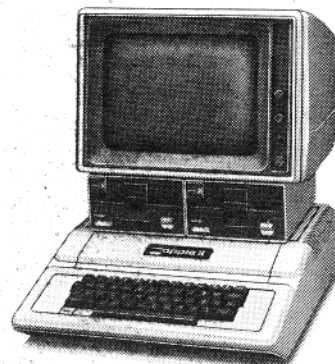


m.b.d.c.

172, rue Solférino - 59800 LILLE TÉL. (20) 57.91.87

concessionnaire agréé APPLE - S.A.V. sur place

ouvert du mardi au samedi de 9h30 à 12h et de 14h30 à 19h



Les Logiciels Saari ont obtenu la Pomme d'Or et le label Bureau Véritas. Vous savez maintenant à qui vous fier.



Tous les éditeurs de logiciels sont contents de leurs logiciels. Nous aussi. C'est humain, c'est normal. Mais la différence c'est qu'il n'y a pas que Saari pour dire du bien des logiciels Saari.

Parmi plusieurs dizaines de logiciels la Paie Gipsi a obtenu la récompense suprême décernée par Apple, la Pomme d'Or du Logiciel.

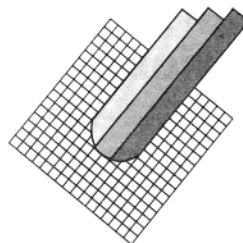
Afin d'apporter un maximum de sécurité à ses clients, Saari a confié au Bureau Véritas le soin de tester ses programmes pour vérifier leurs qualités. Le label Bureau Véritas attestant de ces contrôles est apposé sur les programmes Saari.

La Presse Informatique a beaucoup parlé des Logiciels Saari, louant leurs qualités: facilités d'utilisation ou d'apprentissage, richesse fonctionnelle, sécurité d'emploi ou finition.

Les Logiciels Saari sont également conformes aux derniers textes légaux parus.

Mais il y a aussi plusieurs centaines d'utilisateurs heureux pour dire du bien de nos logiciels. Des utilisateurs rassurés en particulier par le Saari Sécurité Service, un ensemble de services exclusifs à Saari.

Saari, aujourd'hui, ce sont trois logiciels, une Comptabilité, une Paie et une Gestion de Dossiers. Trois logiciels à qui vous êtes sûrs de pouvoir vous fier.



saari®
« L'Esprit Français »

45, rue Galilée, 75116 Paris
Tél. (1) 723 78 56 - Téléx 611 423

AGENCE AUX CLIENTS



Bon pour une documentation gratuite.

Nom _____

Société _____

Adresse _____

Tél. _____

POM'S

Moniteur étendu pour Apple II Plus et IIe

Thierry Le Tallec

NDLR : la très grande longueur de cet excellent moniteur étendu nous a amenés à couper une partie du listing. Nous avons ainsi coupé du "source" imprimé les routines de TRACE (732-749), STEP (750-846) et MINI-ASSEMBLER (918-1089), qui correspondent à des thèmes déjà traités dans Pom's. En outre, pour faire de la place, les routines de conversion (501-638) et de recherches de codes (394-500) ont elles aussi été court-circuitées. Le programme source complet se trouve sur la disquette d'accompagnement, ainsi qu'un programme DUMP en Basic permettant à ceux qui ne possèdent pas le Big Mac de faire la liste complète du source sur leur imprimante. La récapitulation à la fin de cet article est bien entendu complète; bon courage à ceux qui tiennent à l'entrer au clavier (la disquette d'accompagnement est disponible pour 50 FF TTC).

Le moniteur d'origine de l'APPLE II occupe 2K octets, de \$F800 à \$FFFF. Dans cet espace plutôt réduit, ses concepteurs ont tout de même réussi à loger un désassembleur, en plus de toutes les commandes indispensables. Lors de l'introduction de la ROM AUTOSTART, certaines fonctions furent rajoutées (ESC-I, J, K, M; CTRL-S; Auto-Boot; ...) au détriment des fonctions STEP et TRACE.

L'un des premiers objectifs de ce moniteur étendu fut de recréer ces fonctions; vous allez voir qu'il va finalement bien au-delà !

Parmi les fonctions les plus remarquables, il faut noter :

- le pas-à-pas (STEP)
- le pistage (TRACE)
- le mini-assembleur
- le point d'arrêt
- la recherche d'octets
- les minuscules au clavier
- les conversions HEXA → DECIMAL et DECIMAL → HEXA
- l'addition et la soustraction sur 16 bits.

1. Appel du moniteur étendu (&)

Ce moniteur est écrit en Big Mac. Son assemblage nécessite la version 64K du Big Mac; sinon, il faut couper les remarques pour regagner de la place et assembler en version 48K. Il occupe les emplacements mémoire

\$8D00-\$95FF. Pour l'appeler, il suffit de taper :

8D00G depuis le moniteur,
CALL 36096 (ou CALL -29440)
depuis le BASIC.

Une autre solution consiste à simplement faire "BRUN MONITEUR" à partir du BASIC.

Vous pouvez savoir que vous êtes sous son contrôle grâce au signe (PROMPT) et au nouveau son de la cloche. Le nouveau signe est un astérisque affiché en inverse.

Si vous êtes en BASIC, tapez maintenant "&" puis "RETURN". Le moniteur vous répond : "<< AMPERSAND OK >>". Par la suite, pour appeler le moniteur depuis le BASIC, il vous suffira de taper à nouveau "&" et "RETURN". De plus, on retourne au moniteur si on tombe sur un BRK (par exemple CALL -35 ou FFDDG) car le moniteur reprogramme le BRK (119-122).

2. Les commandes

D'une manière générale, la syntaxe des commandes a été calquée sur celle du moniteur d'origine (sauf RETURN seul qui n'affiche plus les octets de la mémoire) : les adresses ou arguments se placent devant, immédiatement suivis par une lettre de commande, sans espace. Plusieurs commandes peuvent s'écrire sur une même ligne. Les commandes "normales" du moniteur fonctionnent toujours de la même façon, mais affichent parfois des indications complémentaires. C'est le cas par exemple de la commande ":" qui, après le RETURN suivant, affiche l'adresse et le contenu du prochain emplacement mémoire dès que l'on enfonce la barre d'espace :

```
2000 :A9 00 A2 FF [RETURN]
[ESPACE]
2004(AA) :
```

Cela est souvent utile lorsqu'il faut taper toute une liste d'octets; on est sûr de ne pas sauter une adresse.

Marche arrière(^)

Pour corriger une erreur, il est possible de revenir à l'adresse précédente en tapant "^" juste derrière le prompt, au début de la ligne suivante. L'adresse précédente s'affiche

alors, suivie de son contenu entre parenthèses. Après le ":", tapez alors la donnée, cette fois-ci sans erreur ...

Stockage de texte (;)

Il est parfois utile de mettre des caractères ASCII directement en mémoire, comme on entre des octets par la commande ":". Pour cela, on tape l'adresse, un point-virgule, et le mot ou la phrase que l'on veut stocker en mémoire :

```
2004;CECI EST UN EXEMPLE
```

Tous les caractères, y compris les espaces seront rangés à partir de l'adresse spécifiée, jusqu'au dernier caractère qui précède le RETURN. Si l'APPLE est en mode INVERSE à ce moment, les caractères sont stockés en inverse (commandes "I" pour Inverse et "N" pour Normal).

Examen mémoire Hexa-ASCII (H)

Cette commande permet de visualiser le contenu d'une portion de la mémoire en hexadécimal et en ASCII. Tapez la première adresse à examiner, puis "H". L'ordinateur affiche alors 22 lignes comportant 8 octets et leur représentation ASCII :

```
2000H
```

```
2000- A9 00 A2 FF C3 C5 C3 C9 )@" CECI
2008- A0 C5 D3 D4 A0 D5 CE A0 EST UN
2010- C5 D8 C5 CD D0 CC CE 26 EXEMPLE&
2018- 53 38 A5 52 E5 54 AA A5 S8*rft*%
2020- 53 90 06 .....
```

Le bit de poids fort est ignoré; les caractères de contrôle sont affichés en clignotant. Un "H" non précédé d'une adresse affiche les 22 lignes suivant celles du dernier H normal. Les adresses de \$C000 à \$COFF sont automatiquement sautées, car elles ne correspondent pas à de la mémoire et leur "lecture" accidentelle est dangereuse pour les programmes qui se trouveraient en RAM à cet instant. Appuyer sur la barre d'espace affiche les 22 lignes suivantes. N'importe quelle autre touche revient en mode commande.

NDLR: il n'est pas tout-à-fait vrai que le bit de poids fort soit ignoré. Faire pour le voir "FOR I=0 TO 255: POKE 8192+I, NEXT I", puis 2000H sous le moniteur.

Recherche d'octets (/)

Cette commande permet de rechercher un octet ou une série d'octets, entre deux adresses. Donnez la première adresse à partir de laquelle vous désirez commencer à chercher, un point, la dernière adresse à examiner, une barre de fraction (/), l'octet ou la série d'octets que vous cherchez, et RETURN.

F800.FFFF/20 ED FD

Le moniteur va explorer toutes les adresses depuis \$F800 jusqu'à \$FFFF et afficher toutes les adresses où il trouve la suite d'octets \$20, \$ED et \$FD. Par exemple :

F903,F91B,F923,F94C,FAE6,FAEC,FAF1,FD47,FD64,FD6C,FDB8,FDD6,FE46,FE4B,FE55,FF2F,FF34,FF37

Notez que, là encore, le moniteur saute automatiquement les adresses de \$C000 à \$COFF. Notez aussi qu'il range votre liste d'octets dans le buffer clavier (en \$200); ne soyez donc pas surpris qu'il l'y retrouve.

Recherche de caractères (")

Variation de la précédente, cette commande suit à peu près la même syntaxe :

F800.FFFF"APPLE

FB09,

Il est important de noter que le moniteur retrouvera un mot qu'il soit en NORMAL, en INVERSE, en FLASHING ou même en minuscules. Si vous n'êtes pas sûr de l'orthographe d'un mot, vous pouvez remplacer une, ou plusieurs lettres par "=".

800.9600"ROU=E

Dans ce cas, le moniteur retrouverait indifféremment ROUEE, ROUGE, ROULE, etc... Le signe "=" est ce qu'on appelle en anglais un caractère "wild card".

Ne soyez pas surpris non plus que le moniteur retrouve la liste de caractères dans la mémoire associée à l'écran (\$400-\$7FF).

Conversion en décimal (\$)

Tapez le nombre à convertir, un dollar, et RETURN. L'équivalent déci-

mal est alors affiché, suivi de sa représentation en négatif s'il y a lieu :

C000\$= 49152 (-16384)

Conversion en hexadécimal (=)

Tapez le nombre à convertir, le signe "=" et RETURN. L'équivalent hexa est affiché, suivi de la conversion complémentaire si cela est possible :

16384=\$4000 (-16384=\$C000)

Addition et soustraction (+, -)

Ces opérations fonctionnent de la même manière que sous le moniteur d'origine, mais peuvent traiter des nombres de 16 bits (4 chiffres hexa) :

1AF7+ C3B = 2732

C080-BFAF= 00D1

La pause, ou point d'arrêt (P,O,X)

Il est fréquemment utile de connaître l'état des registres à certains endroits du déroulement d'un programme, lors de sa mise au point par exemple. La méthode usuelle consiste à placer un "break" (00) à l'endroit désiré, et à rétablir la valeur d'origine après son exécution. Ce moniteur propose une alternative moins primaire ; tapez l'adresse où vous désirez voir le programme s'interrompre, "P", et RETURN

2004P

L'astérisque (le prompt) se met à clignoter, signalant la modification de votre programme. Lors du lancement de celui-ci, tout se déroule normalement jusqu'à l'adresse du point d'arrêt. A cet instant, le contenu des registres est affiché, le point d'arrêt est supprimé, le programme est restauré et l'astérisque arrête de clignoter :

2004 A=00 X=FF Y=7

P=NV.BDIZC S=E7

Remarquez l'affichage détaillé du registre "P". Chaque drapeau est repéré par son initiale standard s'il est à 1, ou par un point s'il est à 0. Cet affichage est obtenu également lors de la commande "CTRL-E".

Une seule pause peut être mise en place à la fois. Si vous oubliez l'adresse où vous avez placé un point d'arrêt, tapez "O" et RETURN pour retrouver cette adresse. Si vous changez d'avis et désirez supprimer un point d'arrêt sans exécuter votre programme, tapez "X" puis RETURN.

Le moniteur mémorise l'adresse de la pause et son ancien contenu à partir de \$2FB; il est donc primordial de ne pas toucher à ces adresses tant que la pause est en vigueur. Notez aussi que le point d'arrêt doit être en RAM.

Le STEP, ou pas-à-pas (S)

Comme son nom l'indique, cette commande permet d'exécuter un programme instruction par instruction. L'adresse est tapée, suivie de "S" et RETURN :

2000S

2000- A9 00 LDA #00
A=00 X=32 Y=07
P=...B..Z. S=E7

Ensuite, il suffit d'appuyer sur la barre d'espace pour exécuter l'instruction suivante, ou sur n'importe quelle autre touche pour sortir de ce mode. Si aucune adresse n'est spécifiée avant le "S", l'adresse prise par défaut est celle rangée en \$3A,\$3B (PCL,PCH) lors d'un STEP antérieur ou d'un point d'arrêt par exemple. Ceci permet de partir en pas-à-pas immédiatement après une pause.

La TRACE, ou pistage (T)

Cette commande n'est autre qu'un STEP perpétuel, et obéit donc aux mêmes règles que celui-ci :

2000T

2000- A9 00 LDA #00
A=00 X=32 Y=07
P=...B..Z. S=E7

2002- A2 FF LDX #FF
A=00 X=FF Y=07
P=N..B.... S=E7

2004- C3

La vitesse de défilement est réglable par le paddle 0. Une pression sur le bouton fige l'écran afin de pouvoir le lire tranquillement; relâcher le bouton pour continuer. Appuyez sur n'importe quelle touche du clavier pour en sortir. Cela peut ne pas marcher si les poignées ne sont pas connectées.

Utilisant les mêmes adresses en page zéro que les anciens STEP et TRACE, ces deux commandes souffrent des mêmes contraintes : impossible de pister un programme qui utiliserait les dites adresses.

Le mini-assembleur (!)

Voici une vieille connaissance ! Disponible d'origine avec l'INTEGER BASIC, cette facilité est de nouveau à votre disposition. Il vous suffit pour

cela de taper un point d'exclamation (!) et RETURN. Son utilisation est décrite en détail dans le manuel de référence APPLE. Tapez "*" en début de ligne et RETURN pour revenir sous le contrôle du moniteur étendu.

Copie en remontant, ou mode inverse (>)

Il arrive assez souvent que l'on ait à déplacer un bloc mémoire de quelques octets vers le haut, la commande MOVE (M) est alors impossible à utiliser car les premiers octets sont recopiés "par-dessus" les suivants. Exemple :

2005<2000.2100

Dans cet exemple, les octets de \$2000 à \$2004 seront bien recopiés en \$2005-\$2009, mais la suite \$200A-\$2105 contiendra également cette même série de 5 octets répétés sans arrêt. Pour remédier à ce problème, utilisez la commande :

2000.2100>2005

Ne tapez pas "M", sous peine de "plantage" catastrophique dans certains cas; tapez tout de suite RE-

TURN. Notez le choix des signes "<" et ">" qui rappellent le sens dans lequel se fait le déplacement.

La saisie des minuscules (K)

Tapez tout de suite "K" et RETURN. Le curseur change pour vous signaler que vous avez quitté le fonctionnement habituel de votre clavier. A partir de ce moment, CTRL-S commute le mode "minuscules" et CTRL-A commute le mode "majuscules". Notez que le DOS est provisoirement déconnecté. Notez également que la modification de la touche SHIFT est reconnue.

Sortie du moniteur (Q)

Pour quitter le moniteur étendu et revenir au BASIC, tapez "Q" et RETURN. Le DOS est alors connecté (= 3DOG); la saisie des minuscules n'est pas affectée; vous pouvez donc entrer des minuscules sous BASIC et sous DOS.

Attention: RESET ou PR#n déconnectent la commande "K" (plus de minuscules au clavier) !

Re-vectoriser le RESET (%)

Si vous désirez pouvoir retomber dans le moniteur étendu par appui sur RESET, et que vous avez la ROM AUTOSTART, tapez "%". Le moniteur étendu répond << RESET OK >>, c'est tout. Pour renvoyer le RESET au BASIC, dans le cas d'un 48K avec le DOS, tapez :

3F2 :BF 9D 38

Tapez ensuite RETURN, c'est terminé. Remarque: la ROM Autostart déconnecte systématiquement l'espace \$C800-\$CFFF).

NDLR : il y a un léger "bug" dans l'entrée du moniteur. Faire deux flèches à gauche puis taper sur les touches "L" et "RETURN" avec ce moniteur et celui de l'Applesoft donne des résultats différents. En fait, le buffer d'entrée est décalé d'un caractère par rapport à l'Autostart (faire NNN200L dans les deux cas pour voir); changer cet état de choses s'avère compliqué à la vue du source : nous ne l'avons pas fait.

à Tignes

STAGE

A PARTIR DU 2 JUILLET 83

MICRO-INFORMATIQUE

Nous avons créé une formule de stage d'une semaine qui permet une formation à la carte dans un programme d'enseignements informatiques de 84 heures. Ce stage s'adresse à ceux qui n'ont jamais programmé, aux débutants en informatique, et peut être pris en charge au titre de la formation.

Nous avons choisi Tignes pour la beauté du site et pour ses nombreuses activités sportives et de détente. Nous avons ainsi organisé pour vous un ensemble de prestations de qualité à partir de 2900 F.

Veillez me faire parvenir, sans engagement de ma part, une documentation détaillée sur le stage de micro-informatique CEMIA

Nom: _____

Prénom: _____

Adresse: _____

TOUR DE L'EUROPE 11 - B.P. 3206 - 68064 MULHOUSE CEDEX. TEL. (89) 46.42.57 - TELEX 881 720 code 78

publicité yves rühmann mulhouse

TIGNES

Veillez retourner ce coupon à.

Big Mac 64 K

```

1 *****
2 *
3 *   EXTENSION DU MONITEUR *
4 *   DE L'APPLE II +      *
5 *
6 *   Thierry Le Tallec   *
7 *
8 *****
9
10 * / 5 Mai 1983
11
12   ORG $8D00 ; (CALL -29440)
13
14 CH   = $24
15 BASL = $28
16 RTNL = $2C
17 RTNH = $2D
18 FORMAT = $2E
19 LENGTH = $2F
20 MODE = $31
21 INVFLG = $32
22 PROMPT = $33
23 YSAV = $34
24 L = $35
25 CSWL = $36
26 CSWH = $37
27 KSWL = $38
28 KSWH = $39
29 PCL = $3A
30 PCH = $3B
31 XQT = $3C
32 DPRND = $3C
33 A1L = $3C
34 A1H = $3D
35 A2L = $3E
36 A2H = $3F
37 A3L = $40
38 A3H = $41
39 A4L = $42
40 A4H = $43
41 FMT = $44
42 ACC = $45
43 XREG = $46
44 STATUS = $48
45 TEMP = $4A
46 DIGIT = $4E
47 RNDL = $4E
48 RNDH = $4F
49 STARTFLG = $4F
50 INBUFF = $200
51 QUIT = $3D0
52 SOFTEV = $3F2
53 AMPERSND = $3F5
54 SHFTLOCK = $77E
55 MINFLAG = $7FE
56 KBD = $C000
57 KBDSTRB = $C010
58 SPKR = $C030
59 PUSHBO = $C061
60 SHIFT = $C063

```

Pour garder les accents dans les commentaires, nous avons mis l'imprimante en mode français. Ainsi, "£" signifie "#".

```

61 INSDS1 = $F882 ;détruit A,X,Y
62 INSDS2 = $F88E
63 INSTDSP = $F8D0 ;détruit A,X,Y
64 PRNTYX = $F940 ;détruit A
65 PRNTAX = $F941 ;détruit A
66 PRBLNK = $F948 ;détruit A,X
67 PRBL2 = $F94A
68 PCADJ = $F953
69 PCADJ2 = $F954
70 PCADJ3 = $F956
71 CHAR1 = $F9B4
72 CHAR2 = $F9BA
73 MNEML = $F9C0
74 MNEMR = $FA00
75 RTBL = $FB19
76 PREAD = $FB1E ;détruit A,Y
77 INIT = $FB2F
78 UP = $FC1A
79 CLREOP = $FC42 ;détruit A,Y
80 WAIT = $FCAB ;annule A
81 RDKEY = $FDOC
82 RDCHAR = $FD35
83 GETLNZ = $FD67
84 NXTCHAR = $FD75
85 CROUT = $FD8E ;détruit A
86 PRA1 = $FD92 ;détruit A,X,Y
87 PRBYTE = $FDDA ;détruit A
88 COUT = $FDED ;détruit A
89 COUT1 = $FDF0
90 BL1 = $FE00
91 BLANK = $FE04
92 SETMODE = $FE18
93 LT = $FE20 ;détruit A,X
94 MOVE = $FE2C
95 VFY = $FE36
96 LIST = $FE5E
97 A1PC = $FE75
98 A1PCLP = $FE78
99 SETINV = $FE80
100 SETNORM = $FE84
101 SETKBD = $FE89
102 INPRT = $FE8D
103 SETVID = $FE93
104 OUTPRT = $FE97
105 XBASIC = $FEBO
106 BASCONT = $FEB3
107 GO = $FEB6
108 USR = $FECA
109 WRITE = $FECD
110 READ = $FEFD
111 RESTOR = $FF3F ;détruit A,X,Y,P
112 SAVE = $FF4A ;détruit A
113 GETNUM = $FFA7 ;détruit A,X,Y
114 ZMODE = $FFC7
115 TR ON ;tronque le listing (col. 2)
116
117 *** DEBUT DU MONITEUR ETENDU ****
118
119 START LDA $XBRK ;revectorise le break
120 STA $3F0

```

```

121 LDA &XBRK
122 STA $3F1
123 JSR SETVID ;PR&O
124 JSR SETKBD ;IN&O
125 JSR SETNORM ;NORMAL
126 JSR INIT ;TEXT
127 MON CLD
128 JSR BELL
129 MONZ BIT PROMPT ;prompt = "$" ou ")" ?
130 BMI SETPRMPT ;oui => prompt = '*' inverse.
131 LDA &$6A ;non => teste si pause
132 BVS STPRMPT1 ; flashe '*' si oui.
133 SETPRMPT LDA &'*' ;inverse '*' sinon.
134 STPRMPT1 STA PROMPT
135 MONZ1 JSR CROUT
136 LDA PROMPT
137 JSR COUT ;affiche le prompt
138 JSR RDCHAR ;et attend une touche.
139 CMP &"^"
140 BNE MONZ3
141 LDA ASL
142 BNE MONZ2
143 DEC ASH
144 MONZ2 DEC ASL
145 DEC AIL
146 LDA &" "
147 MONZ3 CMP &" " ;touche espace ?
148 BNE MONZ4 ;non => attend une ligne
149 LDX ASL ;oui => affiche la prochaine
150 LDA ASH ; adresse
151 JSR PRNTAX
152 DEY ; (raz Y)
153 LDA &"("
154 JSR COUT
155 LDA (ASL),Y
156 JSR PRBYTE ; et son contenu
157 LDA &"")"
158 JSR COUT ; entre parenthèses.
159 LDA &"::" ;mode stockage d'octets
160 MONZ4 LDX &1 ;simule la fin de GETLNZ
161 JSR NXTCHAR+3 ;attend la suite de la commande
162 JSR ZMODE
163 INY ;(il y a déjà eu une touche)
164 NXTITM JSR GETNUM ;lit une adresse
165 STY YSAV
166 LDY &SUBTBL-CHRTBL ;34 commandes !
167 CHSRCH DEY
168 BMI MON
169 CMP CHRTBL,Y
170 BNE CHSRCH
171 JSR TOSUB
172 LDY YSAV
173 JMP NXTITM
174 CRMON CMP &"+" ;addition?
175 BNE CRMON1
176 JMP ADDITN
177 CRMON1 CMP &"-" ;soustraction?
178 BNE CRMON2
179 JMP SOUSTR
180 CRMON2 CMP &"::" ;stockage en mémoire?

```

```

181 BNE CRMON3
182 JMP STORE
183 CRMON3 JSR BL1
184 CRMON4 PLA ;ajuste la pile
185 PLA
186 BNE MONZ1
187 TOSUB TYA
188 ASL ;Y=Y*2
189 TAY
190 LDA SUBTBL,Y
191 PHA ;sauve la partie haute
192 LDA SUBTBL+1,Y ;et la partie basse
193 PHA ;de l'adresse (ordre).
194 LDA MODE
195 JMP ZMODE
196
197 ***** TABLES DES ORDRES ET DE LEUR ADRESSE ****
198
199 * (CHR = commande XOR $30 + $09)
200
201 CHRTBL DFB $82:48+9 ;(ctrl-B)asic
202 DFB $83:48+9 ;(ctrl-C)
203 DFB $85:48+9 ;(ctrl-E)xamine les registres
204 DFB $88:48+9 ;(ctrl-K)
205 DFB $8D:48+9 ;(RTN)
206 DFB $90:48+9 ;(ctrl-P)
207 DFB $99:48+9 ;(ctrl-Y)
208 DFB " ":48+9 ;(espace)
209 DFB "!:48+9 ;(!) mini-assembleur
210 DFB "":48+9 ;(") recherche en ASCII
211 DFB "$":48+9 ;($) convertit en décimal
212 DFB "%":48+9 ;(%) vectorise le reset
213 DFB "&":48+9 ;(&) vectorise l'ampersand
214 DFB "+":48+9 ;(+)
215 DFB "-":48+9 ;(-)
216 DFB ".":48+9 ;(.)
217 DFB "/":48+9 ;(/) recherche d'octets
218 DFB ":":48+9 ;(:)
219 DFB ";":48+9 ;(;) stockage de texte
220 DFB "<":48+9 ;(<)
221 DFB "=":48+9 ;(=) convertit en hexa.
222 DFB ")":48+9 ;(>) déplace vers le haut
223 DFB "G":48+9 ;(G)o
224 DFB "H":48+9 ;(H)exa/ASCII
225 DFB "I":48+9 ;(I)nverse
226 DFB "K":48+9 ;(K)eyin minuscules
227 DFB "L":48+9 ;(L)ist
228 DFB "M":48+9 ;(M)ove
229 DFB "N":48+9 ;(N)ormal
230 DFB "O":48+9 ;(O)bserve pause
231 DFB "P":48+9 ;(P)ause
232 DFB "Q":48+9 ;(Q)uite -> $3D0
233 DFB "R":48+9 ;(R)ead
234 DFB "S":48+9 ;(S)tep
235 DFB "T":48+9 ;(T)race
236 DFB "V":48+9 ;(V)erify
237 DFB "W":48+9 ;(W)rite
238 DFB "X":48+9 ;(X) supprime la pause
239 SUBTBL DDB XBASIC-1
240 DDB BASCONT-1

```

```

241 DDB REGDSP-1
242 DDB INPRT-1
243 DDB CRMON-1
244 DDB OUTPRT-1
245 DDB USR-1
246 DDB BLANK-1
247 DDB MINIASM-1
248 DDB SRCHASC-1
249 DDB PRDEC-1
250 DDB RSTVECT-1
251 DDB AMPVECT-1
252 DDB SETMODE-1
253 DDB SETMODE-1
254 DDB SETMODE-1
255 DDB SEARCH-1
256 DDB SETMODE-1
257 DDB STORETXT-1
258 DDB LT-1
259 DDB PRHEX-1
260 DDB MOVEUP-1
261 DDB GO-1
262 DDB HEXASC-1
263 DDB SETINV-1
264 DDB LWINKEY-1
265 DDB LIST-1
266 DDB MOVE-1
267 DDB SETNORM-1
268 DDB OBSERV-1
269 DDB PAUSE-1
270 DDB QUIT-1
271 DDB READ-1
272 DDB STEP-1
273 DDB TRACE-1
274 DDB VFY-1
275 DDB WRITE-1
276 DDB SUPPAUSE-1
277
278 **** RE-VECTORISE L'AMPERSAND ***
279
280 AMPVECT LDA #4C
281 STA AMPERSND
282 LDA #START
283 STA AMPERSND+1
284 LDA #)START
285 STA AMPERSND+2
286 JSR UP ;remonte le curseur
287 LDY #0
288 PRMESSG1 LDA MESSG1,Y ;affiche (( OK ))+bip!
289 JSR COUT1
290 INY
291 CPY #MESSG2-MESSG1
292 BNE PRMESSG1
293 JMP BELL
294
295 **** RE-VECTORISE LE RESET *****
296
297 RSTVECT LDA #START ;à partir de maintenant,
298 STA SOFTEV ;reset renverra au moniteur.
299 LDA #)START

```

```

300 STA SOFTEV+1
301 EDR #45
302 STA SOFTEV+2
303 JSR UP ;remonte le curseur
304 LDY #0
305 PRMESSG2 LDA MESSG2,Y ;affiche (( RESET OK ))+bip!
306 JSR COUT
307 INY
308 CPY #MESSG3-MESSG2
309 BNE PRMESSG2
310 JMP BELL
311
312 ***** STOCKAGE D'OCTETS *****
313
314 STORE DEX ;RTN précédé d'un octet ?
315 BNE STOR1
316 LDA A2L ;oui => va le ranger
317 STA (A3L),Y
318 JMP STOR2
319 STOR1 LDA A3L
320 CMP A1L
321 BEQ STOR3 ;oui => conserve l'adresse
322 STOR2 INC A3L
323 BNE STOR3
324 INC A3H
325 STOR3 INC A1L
326 JMP CRMON4
327
328 ***** STOCKAGE DE TEXTE *****
329
330 STORETXT LDY #0 ;initialise pointeur écriture
331 STORTXT1 LDX YSAV ;récupère pointeur lecture
332 LDA INBUFF,X ;lit caractère suivant
333 CMP #8D ;retour chariot ?
334 BEQ STORTXT2 ;oui => fin
335 AND INVFLG
336 STA (A2L),Y ;non => range le caractère
337 INC YSAV ;caractère suivant
338 INY
339 BNE STORTXT1 ; (saute toujours)
340 STORTXT2 RTS
341
342 ***** HEXA/ASCII DISPLAY *****
343
344 HEXASC LDX #22
345 STX XREG
346 HE0 LDY #0
347 JSR PRA1 ;affiche l'adresse
348 HE1 JSR PRSPACE ; un espace,
349 LDA (A1L),Y
350 JSR PRBYTE ; un octet,
351 INY
352 CPY #8
353 BNE HE1 ;recommence huit fois
354 INC CH ;deux espaces
355 INC CH
356 LDY #0
357 HE2 LDA (A1L),Y
358 CMP #20 ;inverse ?

```

```

359 BGE HE3
360 ORA 2%CO ;oui => normal
361 HE3 CMP 2%60 ;clignotant ?
362 BGE HE4
363 ORA 2%80 ;oui => normal
364 HE4 CMP 2%A0 ;caractère de controle ?
365 BGE HE5
366 EDR 2%CO ;oui => clignotant
367 HE5 JSR COUT ;affiche le caractère
368 INY
369 CPY 28
370 BNE HE2 ;huit fois
371 LDA A1L
372 ADC 27 ;avance de huit (carry=1)
373 STA A1L
374 BCC HE7
375 HE6 INC A1H
376 HE7 LDA A1H
377 CMP 2%CO
378 BEQ HE6 ;saute les adresses %COxx
379 CMP 2%CF ;%CF8 ( A1 ( %CFF ?
380 BNE HE8
381 LDA A1L
382 CMP 2%F8
383 BLT HE8
384 LDA 20 ;oui => A1=%D000
385 STA A1L
386 INC A1H
387 HE8 DEC XREG
388 BNE HE0
389 JSR RDKEY
390 CMP 2" "
391 BEQ HEXASC
392 RTS
638
639 ***** PAUSE *****
640
641 PAUSE LDA (A1L),Y ;sauve 3 octets
642 STA INBUFF+253,Y ;dans le buffer
643 LDA JSRPAU,Y ;les remplace
644 STA (A1L),Y ;par "JSR XPAUSE"
645 LDA A1L-1,Y
646 STA INBUFF+250,Y ;sauve aussi l'adresse
647 INY
648 CPY 23
649 BNE PAUSE
650 LDA 2%6A ; ('*' clignotante)
651 STA PROMPT
652 RTS
653
654 JSRPAU JSR XPAUSE
655
656 XPAUSE JSR SAVE ;sauve les registres
657 PLA
658 SEC
659 SBC 2
660 STA PCL ;même le compteur pgm.
661 PLA
662 SBC 20

```

```

663 STA PCH
664 JSR SUPPSEO ;remet les 3 octets
665 JSR CROUT
666 LDX PCL
667 LDY PCH
668 JSR PRNTYX ;affiche l'adresse
669 JSR RGDSP1 ;affiche les registres
670 JMP MON ;et retour au moniteur.
671
672 **** SUPPRESSION DE LA PAUSE ****
673
674 SUPPAUSE LDA PROMPT
675 CMP 2%6A ;y-at-il une pause ?
676 BNE SUPPSE2 ;non => retour
677 SUPPSEO LDA INBUFF+251
678 STA RNDL
679 LDA INBUFF+252
680 STA RNDH
681 LDY 2 ;remet les trois octets
682 SUPPSE1 LDA INBUFF+253,Y
683 STA (RNDL),Y
684 DEY
685 BPL SUPPSE1
686 LDA 2'*'
687 STA PROMPT
688 SUPPSE2 RTS
689
690 ***** AFFICHAGE DE L'ADRESSE ****
691
692 OBSERV JSR CROUT ;nouvelle ligne
693 LDX INBUFF+251
694 LDY INBUFF+252
695 JMP PRNTYX
696
697 **** AFFICHAGE DES REGISTRES ****
698
699 RGDSP JSR CROUT ;nouvelle ligne
700 RGDSP1 LDA 27
701 STA CH
702 LDA 2ACC
703 STA A3L
704 LDA 20
705 STA A3H
706 LDX 2%FB
707 RDSP1 JSR PRSPACE ;affiche un espace,
708 LDA RTBL-%FB,X
709 JSR COUT ; " le nom du registre
710 JSR PREGUAL ;affiche "=",
711 LDA ACC+5,X
712 CPX 2%FE ;exception pour P
713 BNE RDSP2
714 LDY 28
715 NXTFLG ASL ;un flag dans la carry
716 PHA ;sauve le reste
717 BCC NOTSET
718 LDA FLGNAM-1,Y
719 BNE DSPFLG
720 NOTSET LDA 2"."
721 DSPFLG JSR COUT ;affiche le flag

```

```

722     PLA
723     DEY
724     BNE  NXTFLG
725     BEQ  RDSP3
726 RDSP2 JSR  PRBYTE
727 RDSP3 INX
728     BMI  RDSP1
729     JMP  CROUT      ;revient par 'return'
730 FLGNAM ASC  "CZIDB.VN"
847
848 ***** ADDITION *****
849
850 ADDITN CLC
851     LDA  A1L
852     ADC  A2L
853     STA  A1L
854     LDA  A1H
855     ADC  A2H
856     STA  A1H
857 FINADD JSR  UPCURSOR ;remonte le curseur
858     LDX  A1L
859     LDY  A1H
860     JSR  PRNTYX
861     JMP  MONZ
862
863 ***** SOUSTRACTION *****
864
865 SOUSTR SEC
866     LDA  A1L
867     SBC  A2L
868     STA  A1L
869     LDA  A1H
870     SBC  A2H
871     STA  A1H
872     CLC
873     BCC  FINADD
874
875 ***** NOUVEAU SON DE CLOCHE *****
876
877 BELL   LDX  2$FF
878 BELL1  LDA  2$58
879 BELL2  LDY  2$18
880 BELL3  DEY
881     BNE  BELL3
882     BIT  SPKR
883     TAY
884 BELL4  DEY
885     BNE  BELL4
886     SBC  21
887     BEQ  BELL1
888     BIT  SPKR
889     DEX
890     BNE  BELL2
891     RTS      ;revient AVEC X=Y=0
892
893 **** ALIGNEMENT DES RESULTATS ***
894
895 UPCURSOR LDY  YSAV
896     STY  CH      ;positionne le curseur

```

```

897     JSR  UP      ;à la ligne supérieure
898
899 PREQUAL LDA  2$="      ;affiche un "="
900 GOCOUT  JMP  COUT   ;et revient
901
902 PRSPACE LDA  2$ "
903     BNE  GOCOUT
904
905 PRCOMMA LDA  2$,"
906     BNE  GOCOUT
907
908 PRNTDOL LDA  2$"$"
909     BNE  GOCOUT
910
911 CLOSBRAC LDA  2$5"
912     BNE  GOCOUT
913
914 OPENBRAC LDA  2$""
915     JSR  COUT
916     LDA  2$"-
917     BNE  GOCOUT
1090
1091 ***** SAISIE DES MINUSCULES *****
1092
1093 LWINKEY LDA  2$NEWROUT ;modifie les entrées/sorties
1094     STA  C$WL
1095     LDA  2$NEWROUT
1096     STA  C$WH
1097     LDA  2$NEWKEYIN
1098     STA  K$WL
1099     LDA  2$NEWKEYIN
1100     STA  K$WH
1101     LSR  MINFLAG ;majuscules par défaut
1102     RTS
1103
1104 NEWKEYIN PHA      ;sauve l'ancien caractère
1105 NK0     LDA  (BASL),Y ;regarde ce qui est affiché
1106     AND  2$DF
1107     CMP  2$DF      ;c'est le curseur ?
1108     BEQ  NK1
1109     LDA  2$ "      ;non => met-le !
1110     BIT  SHFTLOCK
1111     BPL  NK2
1112     LDA  2$FF      ;curseur spécial en minuscules
1113     BMI  NK2      ;(saute toujours)
1114 NK1     PLA      ;récupère l'ancien caractère
1115     PHA
1116 NK2     STA  (BASL),Y ;affiche le caractère
1117 NK3     INC  RNDL      ;incrémente le compteur
1118     BNE  NK4      ;saute si pas débordé
1119     INC  RNDH
1120     LDA  RNDH
1121     AND  2$3F
1122     CMP  2$01      ;délai écoulé ?
1123     BCC  NK0      ;oui => échange curseur/caractère
1124 NK4     BIT  KBD      ;touche enfoncée ?
1125     BPL  NK3      ;non => avance tempo
1126     PLA
1127     STA  (BASL),Y ;restaure l'affichage

```

1128	LDA	KBD	;lit le clavier	1180	STA	INBUFF,X	;range dans le buffer clavier
1129	BIT	KBDSTRB	;acquitte le clavier	1181	NORMOUT	JMP	COUT1
1130	CMP	£93	;ctrl-S ?	1182			
1131	BNE	CTRLA		1183	****	COPIE EN REMONTANT	*****
1132	STA	SHFTLOCK	;oui => passe en minuscules	1184			
1133	NEXTKEY	LDA	(BASL),Y	1185	MOVEUP	STA	MODE ;ne touche plus A1
1134	JMP	NEWKEYIN		1186	JSR	LT	;recopie A2 dans A4
1135	CTRLA	CMP	£81 ;ctrl-A ?	1187	LDY	YSAV	;récupère pointeur lecture
1136	BNE	NLOCK		1188	JSR	GETNUM	;lit la destination
1137	LSR	SHFTLOCK	;oui => passe en majuscules	1189	DEY		
1138	BPL	NEXTKEY		1190	STY	YSAV	;sauve pointeur lecture
1139	NLOCK	BIT	SHFTLOCK ;on est en minuscules ?	1191	SEC		
1140	BPL	NOSHIFT	;non => saute	1192	LDA	A4L	;calcule la longueur
1141	STY	RNDL	;oui => sauve Y	1193	SBC	A1L	;de la zone à déplacer
1142	LDY	£KTBL2-KTBL1		1194	STA	A3L	
1143	TRYCODE	DEY		1195	LDA	A4H	
1144	BMI	NOCHANGE		1196	SBC	A1H	
1145	CMP	KTBL1,Y	;trouvée dans la table ?	1197	STA	A3H	
1146	BNE	TRYCODE	;non => boucle Y fois	1198	CLC		
1147	LDA	KTBL2,Y	;oui => lit correspondance	1199	LDA	A3L	;en déduit la dernière
1148	NOCHANGE	LDY	RNDL ;récupère Y	1200	ADC	A2L	;adresse écrasée
1149	CMP	£8D	;retour chariot ?	1201	STA	A3L	
1150	BEQ	RETURN		1202	LDA	A3H	; (on commence par la fin)
1151	CMP	£98	;ctrl-X ?	1203	ADC	A2H	
1152	BEQ	RETURN		1204	STA	A3H	
1153	CMP	£9B	;escape ?	1205	LDY	£0	
1154	BEQ	RETURN	;oui => passe en majusc.	1206	MOVEUP1	LDA	(A4L),Y ;copie un octet
1155	BIT	SHIFT	;touche shift appuyée ?	1207	STA	(A3L),Y	
1156	BPL	NOSHIFT	;oui => ne convertit pas	1208	DEC	A4L	;décrémente pointeur source
1157	CMP	£"a"		1209	LDA	A4L	
1158	BLT	NOSHIFT	;ne convertit que les lettres	1210	CMP	£FF	
1159	ORA	£20	;convertit en minuscule	1211	BNE	++4	
1160	NOSHIFT	PHA	;sauvegarde le bon code	1212	DEC	A4H	
1161	CMP	£95	;c'est la flèche '-' ?	1213	DEC	A3L	; et pointeur destination
1162	BNE	NOARROW		1214	LDA	A3L	
1163	LDA	(BASL),Y	;oui => lit l'écran	1215	CMP	£FF	
1164	NOARROW	CMP	£E0 ;c'est une minuscule ?	1216	BNE	++4	
1165	BLT	ENDKEY		1217	DEC	A3H	
1166	STA	MINFLAG	;oui => positionne le drapeau	1218	LDA	A4L	;test de fin de déplacement
1167	ENDKEY	PLA	;revient avec le vrai code	1219	CMP	A1L	
1168	RTS		;et retour	1220	LDA	A4H	
1169				1221	SBC	A1H	
1170	RETURN	LSR	SHFTLOCK ;passe en majuscules	1222	BCS	MOVEUP1	;boucle si A4)= A1
1171	RTS		;et revient.	1223	RTS		
1172				1224			
1173	KTBL1	DFB	"s", "^", "a"	1225	*****	MESSAGES	*****
1174	KTBL2	DFB	"m", "n", "p"	1226			
1175				1227	MESSG1	INV	" AMPERSAND OK "
1176	NEWROUT	BIT	MINFLAG ;c'est une minuscule ?	1228	HEX	8D	
1177	BPL	NORMOUT	;non => affiche telle quelle	1229	MESSG2	INV	" RESET OK "
1178	LDA	MINFLAG	;oui => affiche la minuscule	1230	HEX	8D	
1179	LSR	MINFLAG	;raz du drapeau	1231	MESSG3	HEX	FF

Pom's s'adresse à tous ses lecteurs

Quel que soit votre niveau, vous pouvez avoir des informations, des idées ou des programmes à transmettre aux autres. Nous vous prions de nous les faire parvenir afin que nous puissions faire partager ce savoir que vous avez acquis. L'idéal est de nous envoyer une disquette avec vos programmes et un article en traitement de texte (Applewriter I ou II, ou tout fichier TEXT) sur disquette et si possible aussi imprimé, au cas où nous aurions du mal à le récupérer. N'oubliez pas d'indiquer votre adresse et votre numéro de téléphone !

MONITEUR : Récapitulation

*8D00.951E

8D00- A9 E2 8D F0 03 A9 91 8D
8D08- F1 03 20 93 FE 20 89 FE
8D10- 20 84 FE 20 2F FB D8 20
8D18- 79 92 24 33 30 04 A9 6A
8D20- 70 02 A9 2A 85 33 20 8E
8D28- FD A5 33 20 ED FD 20 35
8D30- FD C9 DE D0 0C A5 40 D0
8D38- 02 C6 41 C6 40 C6 3C A9
8D40- A0 C9 A0 D0 19 A6 40 A5
8D48- 41 20 41 F9 88 A9 AB 20
8D50- ED FD B1 40 20 DA FD A9
8D58- A9 20 ED FD A9 BA A2 01
8D60- 20 78 FD 20 C7 FF C8 20
8D68- A7 FF 84 34 A0 26 88 30
8D70- A5 D9 AA 8D D0 F8 20 9A
8D78- 8D A4 34 4C 67 8D C9 AB
8D80- D0 03 4C 4F 92 C9 AD D0
8D88- 03 4C 69 92 C9 BA D0 03
8D90- 4C 60 8E 20 00 FE 68 68
8D98- D0 8C 98 0A AB B9 D0 8D
8DA0- 48 B9 D1 8D 48 A5 31 4C
8DA8- C7 FF BB BC BE C4 C6 A9
8DB0- B2 99 9A 9B 9D 9E 9F A4
8DB8- A6 A7 AB 93 94 95 96 97
8DC0- 00 01 02 04 05 06 07 08
8DC8- E9 EA EB EC ED EF F0 F1
8DD0- FE AF FE B2 91 2B FE 8C
8DD8- 8D 7D FE 96 FE C9 FE 03
8DE0- 93 4A 8F 46 8F C0 8E 3D
8DE8- 8E 1B FE 17 FE 17 FE 17
8DF0- 8E F3 FE 17 8E 7A FE 1F
8DF8- 90 2E 94 B7 FE B5 8E 8F
8E00- FE 7F 94 02 FE 5D FE 2B
8E08- FE 83 91 1F 90 C2 03 CF
8E10- FE FC 91 9B 91 75 FE 35
8E18- FE CC 91 00 A9 4C 8D F5
8E20- 03 A9 00 8D F6 03 A9 8D
8E28- 8D F7 03 20 1A FC A0 00
8E30- B9 04 95 20 F0 FD C8 C0
8E38- 0F D0 F5 4C 79 92 A9 00
8E40- 8D F2 03 A9 8D 8D F3 03
8E48- 49 A5 8D F4 03 20 1A FC
8E50- A0 00 B9 13 95 20 ED FD
8E58- C8 C0 0B D0 F5 4C 79 92

8E60- CA D0 07 A5 3E 91 40 4C
8E68- 70 8E A5 40 C5 3C F0 06
8E70- E6 40 D0 02 E6 41 E6 3C
8E78- 4C 96 8D A0 00 A6 34 8D
8E80- 00 02 C9 8D F0 09 25 32
8E88- 91 3E E6 34 C8 D0 EE 60
8E90- A2 16 86 46 A0 00 20 92
8E98- FD 20 A0 92 B1 3C 20 DA
8EA0- FD C8 C0 08 D0 F3 E6 24
8EA8- E6 24 A0 00 B1 3C C9 20
8EB0- B0 02 09 C0 C9 60 B0 02
8EB8- 09 80 C9 A0 B0 02 49 C0
8EC0- 20 ED FD C8 C0 08 D0 E4
8EC8- A5 3C 69 07 85 3C 90 02
8ED0- E6 3D A5 3D C9 C0 F0 F8
8ED8- C9 CF D0 0C A5 3C C9 F8
8EE0- 90 06 A9 00 85 3C E6 3D
8EE8- C6 46 D0 A8 20 0C FD C9
8EF0- A0 F0 9D 60 85 31 84 4A
8EF8- 20 20 FE A4 34 20 A7 FF
8F00- 84 34 A4 4A 48 A5 3E 99
8F08- 00 02 E6 4A 68 C9 99 F0
8F10- EA 48 A5 42 85 3E A5 43
8F18- 85 3F 20 8E FD A0 00 B1
8F20- 3C D9 00 02 D0 0F C8 C4
8F28- 4A D0 F4 A6 3C A4 3D 20
8F30- 40 F9 20 A4 92 20 A0 8F
8F38- 90 E3 68 C9 AF F0 02 C6
8F40- 34 20 C7 FF 4C 8E FD 84
8F48- 4A A4 34 B9 00 02 C9 8D
8F50- D0 02 A9 C0 E6 34 A4 4A
8F58- 48 29 3F 99 00 02 E6 4A
8F60- 68 C9 A2 F0 04 C9 A0 B0
8F68- E0 20 8E FD C6 4A A0 00
8F70- B9 00 02 F0 0F B1 3C C9
8F78- E0 90 02 29 1F 29 3F D9
8F80- 00 02 D0 0F C8 C4 4A D0
8F88- E7 A6 3C A4 3D 20 40 F9
8F90- 20 A4 92 20 A0 8F 90 D6
8F98- C6 34 20 C7 FF 4C 8E FD
8FA0- A5 3C C5 3E A5 3D E5 3F
8FAB- 08 E6 3C D0 02 E6 3D A5
8FB0- 3D C9 C0 F0 F8 C9 CF D0
8FB8- 06 A5 3C C9 FF F0 EA 28
8FC0- 60 20 94 92 20 42 FC 20
8FC8- 20 FE 20 ED 8F A9 7F C5
8FD0- 43 B0 19 20 48 F9 20 B0
8FD8- 92 A9 00 38 E5 42 85 3E
8FE0- A9 00 E5 43 85 3F 20 ED
8FE8- 8F 20 AC 92 60 A2 04 86
8FF0- 4F A9 B0 85 4E A5 3E DD

8FF8- 25 90 A5 3F FD 2A 90 90
9000- 0D 85 3F A5 3E FD 25 90
9008- 85 3E E6 4E D0 E7 A5 4E
9010- EB CA F0 0A C9 B0 F0 02
9018- 85 4F 24 4F 10 03 20 ED
9020- FD CA 10 CD 60 01 0A 64
9028- E8 10 00 00 00 03 27 A9
9030- 00 85 3C 85 3D A4 34 88
9038- 10 0F B9 00 02 38 E9 B0
9040- 30 0A C9 0A B0 06 99 00
9048- 02 88 10 EE C8 B4 3E 84
9050- 4A A4 3E B9 00 02 30 1F
9058- 20 BE 90 65 3C AA A9 00
9060- 65 3D A8 20 BE 90 20 BE
9068- 90 8A 65 3C 85 3C 98 65
9070- 3D 85 3D E6 3E 90 DA C6
9078- 34 20 94 92 20 42 FC E6
9080- 34 20 A8 92 A6 3C A4 3D
9088- 20 40 F9 A9 7F C5 3D 90
9090- 2C 20 48 F9 20 B0 92 A4
9098- 4A B9 00 02 30 08 09 B0
90A0- 20 ED FD C8 D0 F3 20 ED
90A8- FD 20 A8 92 A9 00 38 E5
90B0- 3C AA A9 00 E5 3D A8 20
90B8- 40 F9 20 AC 92 60 06 3C
90C0- 26 3D 60 B1 3C 99 FD 02
90C8- B9 DD 90 91 3C B9 38 00
90D0- 99 FA 02 C8 C0 03 D0 EB
90D8- A9 6A 85 33 60 20 E0 90
90E0- 20 4A FF 68 38 E9 02 85
90E8- 3A 68 E9 00 85 3B 20 07
90F0- 91 20 8E FD A6 3A A4 3B
90F8- 20 40 F9 20 2F 91 4C 16
9100- 8D A5 33 C9 6A D0 18 AD
9108- FB 02 85 4E AD FC 02 85
9110- 4F A0 02 B9 FD 02 91 4E
9118- 88 10 F8 A9 2A 85 33 60
9120- 20 8E FD AE FB 02 AC FC
9128- 02 4C 40 F9 20 8E FD A9
9130- 07 85 24 A9 45 85 40 A9
9138- 00 85 41 A2 FB 20 A0 92
9140- BD 1E FA 20 ED FD 20 9B
9148- 92 B5 4A E0 FE D0 16 A0
9150- 08 0A 48 90 05 B9 6D 91
9158- D0 02 A9 AE 20 ED FD 68
9160- 88 D0 EE F0 03 20 DA FD
9168- E8 30 D2 4C 8E FD C3 DA
9170- C9 C4 C2 AE D6 CE A9 01
9178- 85 31 20 9C 91 A2 00 20
9180- 1E FB 98 49 FF C9 08 90
9188- 03 20 A8 FC AD 61 C0 30

9190-	FB AD 00 C0 10 E0 8D 10	92C0-	A6 3E D0 01 88 CA 8A 18	93F0-	BB F0 04 C9 8D D0 80 4C
9198-	C0 4C 16 8D 20 75 FE 20	92C8-	E5 3A 85 3E 10 01 C8 98	93F8-	15 93 B9 00 02 C8 C9 A0
91A0-	D0 FB 68 85 2C 68 85 2D	92D0-	E5 3B D0 6B A4 2F B9 3D	9400-	F0 F8 60 A9 A7 85 36 A9
91A8-	A2 08 BD 46 92 95 3C CA	92D8-	00 91 3A 88 10 F8 20 1A	9408-	94 85 37 A9 17 85 38 A9
91B0-	D0 FB A1 3A F0 2C A4 2F	92E0-	FC 20 1A FC 20 D0 F8 20	9410-	94 85 39 4E FE 07 60 48
91B8-	C9 20 F0 43 C9 60 F0 2F	92E8-	53 F9 84 3B 85 3A 4C 4E	9418-	B1 28 29 DF C9 DF F0 0B
91C0-	C9 4C F0 46 C9 6C F0 43	92F0-	93 20 9A 8D A4 34 20 A7	9420-	A9 DF 2C 7E 07 10 06 A9
91C8-	C9 40 F0 1F 29 1F 49 14	92F8-	FF 84 34 A0 1F 88 30 4B	9428-	FF 30 02 68 48 91 28 E6
91D0-	C9 04 F0 02 B1 3A 99 3C	9300-	D9 AA 8D D0 F8 C9 C6 D0	9430-	4E D0 0A E6 4F A5 4F 29
91D8-	00 88 10 F8 20 3F FF 4C	9308-	E8 A5 31 A0 00 C6 34 20	9438-	3F C9 01 90 DB 2C 00 C0
91E0-	3C 00 20 82 FB 20 2F 91	9310-	00 FE 4C 4E 93 A5 3D 20	9440-	10 ED 68 91 28 AD 00 C0
91E8-	4C 16 8D 18 68 85 48 68	9318-	8E F8 AA BD 00 FA C5 42	9448-	2C 10 C0 C9 93 D0 08 8D
91F0-	85 3A 68 85 3B A5 2F 20	9320-	D0 13 BD C0 F9 C5 43 D0	9450-	7E 07 B1 28 4C 17 94 C9
91F8-	56 F9 84 3B 18 90 14 18	9328-	0C A5 44 A4 2E C0 9D F0	9458-	81 D0 05 4E 7E 07 10 F2
9200-	20 54 F9 AA 98 48 8A 48	9330-	88 C5 2E F0 9F C6 3D D0	9460-	2C 7E 07 10 28 84 4E A0
9208-	A0 02 18 B1 3A AA 88 B1	9338-	DC E6 44 C6 35 F0 D6 A4	9468-	03 88 30 08 D9 A1 94 D0
9210-	3A 86 3B 85 3A B0 F3 A5	9340-	34 98 AA 20 4A F9 A9 DE	9470-	F8 B9 A4 94 A4 4E C9 8D
9218-	2D 48 A5 2C 48 20 2C 91	9348-	20 ED FD 20 79 92 24 33	9478-	F0 23 C9 98 F0 1F C9 9B
9220-	A5 31 D0 0E AD 00 C0 10	9350-	A9 21 50 02 09 40 85 33	9480-	F0 1B 2C 63 C0 10 06 C9
9228-	FB 2C 10 C0 C9 A0 D0 02	9358-	20 67 FD 20 C7 FF AD 00	9488-	C0 90 02 09 20 48 C9 95
9230-	C6 34 60 18 A0 01 B1 3A	9360-	02 C9 AA D0 03 4C 16 8D	9490-	D0 02 B1 28 C9 E0 90 03
9238-	20 56 F9 85 3A 98 38 B0	9368-	C9 A0 F0 13 C8 C9 A4 F0	9498-	8D FE 07 68 60 4E 7E 07
9240-	B2 20 4A FF 38 B0 AE EA	9370-	85 88 20 A7 FF C9 93 D0	94A0-	60 DD DE C0 CD CE D0 2C
9248-	EA 4C 41 92 4C 33 92 18	9378-	C8 8A F0 C5 20 78 FE A9	94A8-	FE 07 10 09 AD FE 07 4E
9250-	A5 3C 65 3E 85 3C A5 3D	9380-	03 85 3D 20 FA 93 0A E9	94B0-	FE 07 9D 00 02 4C F0 FD
9258-	65 3F 85 3D 20 94 92 A6	9388-	BE C9 C2 90 B4 0A 0A A2	94B8-	85 31 20 20 FE A4 34 20
9260-	3C A4 3D 20 40 F9 4C 1A	9390-	04 0A 26 42 26 43 CA 10	94C0-	A7 FF 88 84 34 38 A5 42
9268-	8D 38 A5 3C E5 3E 85 3C	9398-	F8 C6 3D F0 F4 10 E4 A2	94C8-	E5 3C 85 40 A5 43 E5 3D
9270-	A5 3D E5 3F 85 3D 18 90	93A0-	05 20 FA 93 84 34 D0 B4	94D0-	85 41 18 A5 40 65 3E 85
9278-	E3 A2 FF A9 58 A0 1B 88	93A8-	F9 D0 13 20 FA 93 D0 BA	94D8-	40 A5 41 65 3F 85 41 A0
9280-	D0 FD 2C 30 C0 AB 88 D0	93B0-	F9 F0 0D BD BA F9 F0 07	94E0-	00 B1 42 91 40 C6 42 A5
9288-	FD E9 01 F0 EE 2C 30 C0	93B8-	C9 A4 F0 03 A4 34 18 88	94E8-	42 C9 FF D0 02 C6 43 C6
9290-	CA D0 EA 60 A4 34 84 24	93C0-	26 44 E0 03 D0 0D 20 A7	94F0-	40 A5 40 C9 FF D0 02 C6
9298-	20 1A FC A9 BD 4C ED FD	93C8-	FF A5 3F F0 01 E8 86 35	94F8-	41 A5 42 C5 3C A5 43 E5
92A0-	A9 A0 D0 F9 A9 AC D0 F5	93D0-	A2 03 88 86 3D CA 10 C9	9500-	3D B0 DE 60 20 01 0D 10
92A8-	A9 A4 D0 F1 A9 DD D0 ED	93D8-	A5 44 0A 0A 05 35 C9 20	9508-	05 12 13 01 0E 04 20 0F
92B0-	A9 DB 20 ED FD A9 AD D0	93E0-	B0 06 A6 35 F0 02 09 80	9510-	0B 20 8D 20 12 05 13 05
92B8-	E4 E9 81 4A D0 14 A4 3F	93E8-	85 44 84 34 B9 00 02 C9	9518-	14 20 0F 0B 20 8D FF

Abonnez-vous à Pom's

*La vie de votre revue dépend de son nombre d'abonnés. Plus vous êtes nombreux, plus nous pouvons vous rendre service, par l'augmentation correspondante de nos moyens comme par le plus grand nombre de contributions que nous recevrons forcément de vous. Nous avons volontairement choisi une politique de disquettes à bon marché ; ainsi, lorsque vous vous abonnez avec disquettes, chaque disquette vous revient à moins de 45 francs TTC, port compris ! Un abonnement à quatre numéros coûte aujourd'hui 295 francs. Cela ne sera-t-il pas largement amorti si **une seule** des 200 pages de texte que vous trouverez en quatre numéros vous résoud ce problème sur lequel vous séchez (on a vite fait de perdre deux ou trois jours sur un problème) ? Dans de nombreux numéros, nous vous avons offert des programmes comparables à ceux qui sont vendus de 250 à 500 francs dans les boutiques. Profitez-en !*

1095F HT
Prix pour Apple II ou Apple II E

**Vous avez un Apple ou
un ordinateur personnel IBM.
Beaucoup de logiciels
sont utiles.
Celui-ci est indispensable.**

PFS Gestion de Fichiers :

- Permet en un minimum de manipulations et de temps de créer votre Fichier (clients, fournisseurs, stock, bibliothèque, etc.).
- Très nombreuses possibilités de recherche : nombreux critères de sélection applicables à plusieurs zones du Fichier en même temps.
- Nombreuses possibilités d'édition (fiches, étiquettes, mailings, etc.).
- Autres fonctions : modification du dessin du

Fichier, copie totale ou partielle du Fichier, ajout de nouvelles fiches, suppression, recherche et mise à jour, recherche et édition.

Best-seller aux États-Unis, plus de 150.000 programmes vendus.

PFS Gestion de Fichiers peut être complété par PFS Report (programme de tri et d'édition de listes et calculs) et PFS Graph (programme d'édition sous forme de graphiques, histogrammes, parts de gâteau, etc.).



Bon de commande

POM'S

A retourner à Médipost 45212 Montargis Cedex - Tél. (38) 85 54 73

Je commande PFS Fichier
PFS Report
PFS Graph

PRIX PAR
PROGRAMME

Apple II ou II E	APPLE III ou IBM PC
1095F + TVA = 1298,67F	1475F + TVA = 1749,35F

Fonctionnant pour Apple II , Apple II E , Apple III et IBM PC (sauf PFS Graph) soit un total de _____ F

Règlement à la commande : Chèque bancaire Chèque postal à l'ordre de Médipost

Nom _____ Société _____

Adresse _____ Tél. _____

Date

Signature

Une facture acquittée sera jointe à votre livraison.

Notions de base : gestion de fichiers

Gérard Michel

Cet article est complémentaire de celui publié dans le Pom's 4. Il présente une autre façon de gérer des fichiers selon le principe de la "table de référence" (ou table d'index) et des fichiers de données à Accès Direct.

Ainsi que vous pourrez le constater en examinant le programme, ou en l'utilisant, les procédures de "menu", de saisie et de contrôle des données sont réduites à leur plus simple expression : nous ne nous intéresserons ici qu'aux aspects spécifiques de la gestion des fichiers.

Principes généraux

Les données sont stockées dans un fichier à accès direct (FIC). Chaque enregistrement de ce fichier correspond à un "article" et chaque article est repéré par une clé unique et strictement personnelle (nom d'un client, numéro de code d'un produit...). Dans notre exemple, la clé se compose de 6 caractères alphanumériques quelconques.

Pour retrouver un enregistrement à partir de sa clé, il faut constituer une table d'index qui donne, en face de chaque clé, l'adresse de l'enregistrement correspondant dans FIC (augmentée de 1000 dans notre exemple). Ainsi, un élément de la table est une variable V\$, avec :
LEFT\$(V\$,6) valeur de la clé et
VAL(MID\$(V\$,7,4))-1000 adresse de l'enregistrement.

Les éléments de cette table doivent être classés dans l'ordre alphanumérique. Pour en retrouver un dans la liste, on utilise une procédure de recherche dite dichotomique.

La question qui se pose maintenant est de savoir comment sera stockée cette table et où s'effectueront les opérations de recherche.

Dans le Pom's 4, nous avons abordé la solution consistant à stocker la table dans un fichier séquentiel. Ce fichier est lu intégralement en mémoire centrale au début des traitements et recopié à la fin. Toutes les opérations de recherche s'effectuent en mémoire.

Cette solution permet des recherches et une mise à jour de la table très rapides, en cas d'ajout ou de suppression. Par contre, elle impose de lire la table intégralement chaque fois qu'on lance les traitements et surtout,

si les clés sont longues et nombreuses, elle pose des problèmes de saturation de la mémoire centrale.

L'alternative envisagée ici consiste à stocker la table d'index dans un fichier à accès direct (TAB). Ceci permet d'effectuer les recherches sur disquette (toujours selon les principes de la recherche dichotomique) et de libérer une capacité mémoire non négligeable.

Evidemment, la recherche d'une clé donnée est légèrement plus lente, mais c'est surtout pour la mise à jour de la table que le temps de traitement risque de subir une dégradation sensible. Pour ajouter une clé dans la liste, par exemple, il faut "reculer"

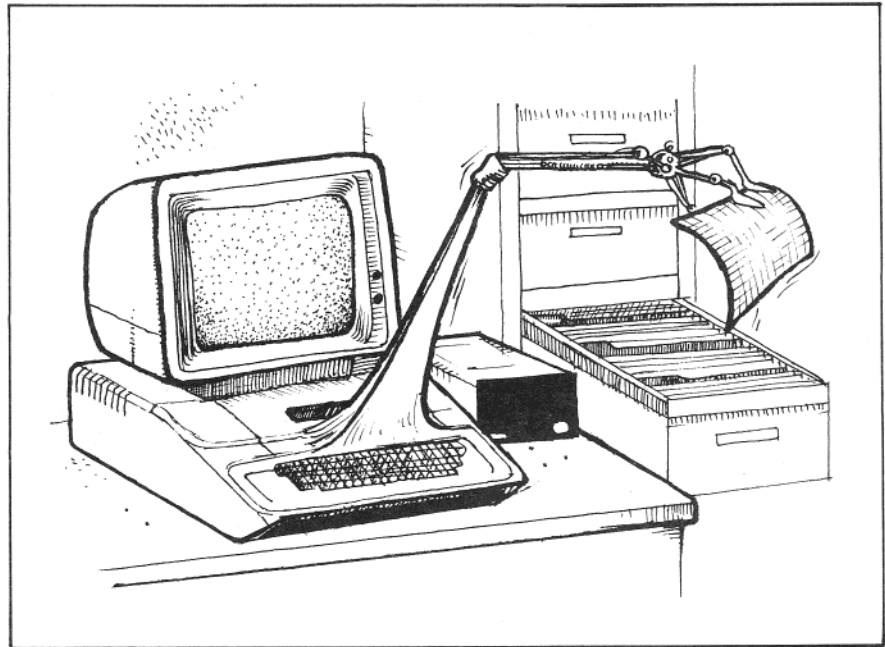
Pour réduire la portée de ce problème, nous utilisons ici une mini-table de référence en mémoire (pour les créations notamment) et un décalage des mises à jour sur disquette par rapport aux traitements courants.

Variables Principales

Les variables principales sont listées dans l'ordre de leur apparition dans le programme.

K1, K2, A\$: indices et variable courants

NC: nombre d'éléments dans la table "tampon" en mémoire centrale.



toutes celles qui lui sont supérieures; si elles sont nombreuses, et compte tenu des performances du DOS, on pourrait parfois attendre bien plus d'une minute avant de passer à l'article suivant.

Une autre solution consiste à avoir une table d'index qui indique, pour chaque article, la position du précédent et celle du suivant dans l'ordre alphabétique. La recherche d'un article donné est alors plus longue, mais la mise à jour plus rapide. En effet, l'insertion d'un article, par exemple, ne nécessite la modification que de l'index des articles entre lesquels le nouvel article doit s'insérer.

S\$: clé, complétée par des blancs si nécessaire.

NT: nombre d'éléments dans la table sur disquette.

N\$: tableau des clés en mémoire.

PL%: liste des adresses libres dans le fichier FIC (libérées par une annulation). Cette liste est stockée dans un fichier séquentiel.

TD: nombre de clés annulées sur disquette au cours d'une phase de traitement.

NL: nombre d'adresses libres dans FIC.

NZ: nombre d'éléments "vivants" dans la table sur disquette.

ZK: indique la position virtuelle

d'une nouvelle clé dans la table sur disquette.

A : utilisé pour la re-crédation de clés annulées.

Analyse du programme

Dans tous les cas, la clé entrée au clavier est tout d'abord recherchée sur la disquette. Si elle existe, on signale une éventuelle annulation antérieure (ligne 350). Si elle n'existe pas, on la cherche alors en mémoire, après avoir noté sa position virtuelle sur disquette (ligne 360).

1) Création d'un article (homonymes interdits). La limite de 50 imposée au nombre de créations sans mise à jour sur disquette dépend en fait, bien sûr, du niveau de saturation de la mémoire (ligne 390).

Le principe consiste à faire la mise à jour uniquement sur la table en mémoire (à la restriction ci-dessus près) pendant le traitement lui-même. Les créations peuvent ainsi s'enchaîner sans délai d'attente sensible entre chacune d'elles (lignes 410 et 420).

S'il s'agit de la re-crédation d'une clé préalablement annulée, celle-ci est simplement réécrite intégralement (clé plus adresse de l'article) à sa place dans la table sur disquette (ligne 405). Aucune autre mise à jour n'est nécessaire pour ce type de création, ni sur disquette, ni en mémoire.

Les données sont ensuite enregistrées à l'adresse R dans FIC.

2) Modification et consultation d'un article. Ces deux opérations ne posent guère de problèmes, dans la mesure où l'on sait déjà comment retrouver la clé et l'article qui lui correspondent.

3) Annulation. La limite de 50 imposée au nombre d'adresses libres gardées en mémoire appelle la même remarque que pour les créations.

L'annulation d'une clé qui se trouve dans la table tampon (clé créée au cours de la même exécution du programme) est prise directement en compte (ligne 520). Cette clé ne sera donc pas recopiée sur disquette en fin de traitement.

Si la clé se trouve dans la table permanente, on remplace simplement l'adresse de l'article correspondant par A, sans procéder au décalage de la table (ligne 510).

4) Fin de traitement

Nous abordons maintenant le problème principal, du moins en ce qui concerne le temps de traitement, à savoir la mise à jour de la table sur disquette.

S'il s'agit de la toute première phase de création, cependant, il suffit de recopier directement la table tampon dans TAB (ligne 212).

Dans le cas contraire, il faut insérer les nouvelles clés dans la liste d'index, en respectant l'ordre alphanumérique. Nous vous proposons, à titre d'exemple, deux solutions possibles :

- La première consiste à décaler les enregistrements du fichier TAB lorsque cela est nécessaire, exactement comme on le faisait en mémoire centrale (lignes 215 et 220 du programme de base).

- La seconde utilise un fichier intermédiaire (TB2), constitué progressivement par la copie séquentielle des éléments de TAB entre lesquels viennent s'intercaler, le cas échéant, les clés de la table tampon.

Pour une même dimension du fichier TAB, le choix entre ces deux solutions dépend surtout du nombre de clés à insérer; la première est plus rapide si ce nombre est faible (sauf si elles viennent majoritairement en tête de liste), et inversement pour la seconde. Il est d'ailleurs possible d'utiliser les deux dans le même programme et de choisir en fonction d'un test sur la valeur de NC.

En ce qui concerne les annulations, le traitement assure "l'écrasement" des clés marquées d'un A (ligne 280). Cette opération peut être réalisée à l'issue des annulations elles-mêmes, ou de façon indépendante quand aucun travail plus urgent ne l'interdit.

Programme FICHIERS

```
1 ONERR GOTO 9000
5 GOTO 100
38 REM
39 REM *****
   * RECHERCHE DICHO. *
   * EN MEMOIRE *
   *****
40 K1 = 1:K2 = NC: IF NC = 0 THEN K = 1:
   RETURN
41 K = INT ((K1 + K2) / 2):A$ = LEFT$(
   (N$(K),6): IF S$ = A$ THEN K1 =
   - 1: RETURN
42 IF S$ < A$ AND K2 > K1 + 1 THEN K2 =
   K: GOTO 41
43 IF S$ < A$ THEN RETURN
44 IF K2 > K1 + 1 THEN K1 = K: GOTO 41
45 K = K + 1: IF S$ > LEFT$( N$(K2),6)
   THEN K = K2 + 1
46 IF S$ = LEFT$( N$(K2),6) THEN K1 =
   - 1:K = K2: RETURN
47 RETURN
48 REM
49 REM *****
   * RECHERCHE DICHO. *
   * SUR DISQUETTE *
   *****
50 K1 = 1:K2 = NT: IF NT = 0 THEN K = 1:
   RETURN
```

```
51 K = INT ((K1 + K2) / 2): PRINT RT$:K
   : INPUT N$:A$ = LEFT$( N$,6): IF
   S$ = A$ THEN K1 = - 1: PRINT RU
   $: RETURN
52 IF S$ < A$ AND K2 > K1 + 1 THEN K2 =
   K: GOTO 51
53 IF S$ < A$ THEN PRINT RU$: RETURN
54 IF K2 > K1 + 1 THEN K1 = K: GOTO 51
55 K = K + 1: PRINT RT$:K2: INPUT N$: IF
   S$ > LEFT$( N$,6) THEN K = K2 +
   1
56 IF S$ = LEFT$( N$,6) THEN K1 = - 1
   :K = K2: PRINT RU$: RETURN
57 PRINT RU$: RETURN
98 REM
99 REM *****
   * DEBUT PROGRAMME *
   *****
100 BL$ = "          ":D$ = CHR$(4):D1$ =
   CHR$(13) + D$:OT$ = D1$ + "OPE
   NTAB,L11":RT$ = D$ + "READTAB,R":
   WT$ = D$ + "WRITETAB,R":OF$ = D1$
   + "OPENFIC,L100":RF$ = D$ + "REA
   DFIC,R":WF$ = D$ + "WRITEFIC,R"
105 OL$ = D1$ + "OPENPL":RL$ = D$ + "REA
   DPL":WL$ = D$ + "WRITEPL":CL$ = D
   $ + "CLOSE":RU$ = D$ + "PREO"
110 DIM N$(50),PL*(50)
118 REM
```

```

119 REM *****
    * OUVERTURE DES *
    * FICHIERS *
    *****

120 Z% = 1:NC = 0:TD = 0:Z = 0: PRINT OT
    $: PRINT RT$:0: INPUT NZ,NT
130 Z% = 0: IF Z = 5 THEN NZ = 0:NT = 0
140 Z% = 2:Z = 0: PRINT OL$: PRINT RL$:
    INPUT NL
150 Z% = 0: IF Z = 5 THEN NL = 0: GOTO 1
    70
160 FOR I = 1 TO NL: INPUT PL%(I): NEXT
170 PRINT D$"CLOSEPL": PRINT OF$: PRINT
    RU$:Z = 0
180 PRINT : PRINT : PRINT "1-CRE. 2-MO
    D. 3-CONS. 4-ANN. 5-MAJ ANN.
    6-FIN";: GET ZZ$: PRINT : PRINT :
    ZZ% = VAL (ZZ$): IF ZZ% < 1 OR Z
    Z% > 6 THEN 180
190 IF ZZ% < 6 THEN 340
198 REM
199 REM *****
    * FIN TRAITEMENT *
    *****

200 IF NC = 0 THEN 230
205 GOSUB 210: PRINT RU$: GOTO 230
208 REM
209 REM *****
    * MISE A JOUR DES *
    * CREATIONS *
    *****

210 IF NT > 0 THEN 215
212 PRINT D$"CLOSETAB": PRINT OT$: FOR
    I = 1 TO NC:Z$ = LEFT$ (N$(I),10
    ): PRINT WT$:I: PRINT Z$:NZ = NZ
    + 1:NT = NT + 1: NEXT : RETURN
215 FOR I = 1 TO NC:K = VAL ( MID$ (N$
    (I),11)) + I - 1: IF K < = NT TH
    EN FOR L = NT TO K STEP - 1: PR
    INT RT$:L: INPUT Z$: PRINT WT$:L
    + 1: PRINT Z$: NEXT
220 NZ = NZ + 1:Z$ = LEFT$ (N$(I),10):N
    T = NT + 1: PRINT WT$:K: PRINT Z$
    : NEXT I: RETURN
230 IF TD = 0 THEN 310
240 INPUT "OK POUR M.A.J. ANNULATIONS "
    ;Z$: IF Z$ < > "O" AND Z$ < > "
    N" THEN 240
250 IF Z$ = "N" THEN NZ = NZ - TD: GOTO
    310
258 REM
259 REM *****
    * MISE A JOUR DES *
    * ANNULATIONS *
    *****

260 R = 1: IF NT = 0 OR NZ = NT + TD THE
    N 310
270 PRINT RT$:R: INPUT Z$: IF MID$ (Z$
    ,7) < > "A" THEN 290
280 NT = NT - 1: IF R < = NT THEN FOR
    L = R TO NT: PRINT RT$:L + 1: INP
    UT Z$: PRINT WT$:L: PRINT Z$: NEX
    T
290 R = R + 1: IF R < = NT THEN 270
300 NZ = NT

```

```

310 PRINT WT$:0: PRINT NZ: PRINT NT: PR
    INT CL$: PRINT OL$: PRINT WL$: PR
    INT NL: FOR I = 1 TO NL: PRINT PL
    %(I): NEXT : PRINT CL$: END
340 IF ZZ% = 5 THEN 260
348 REM
349 REM *****
    * RECHERCHE CLE *
    *****

350 ZK = 0:A = 0: INPUT "VALEUR DE LA CL
    E ";S$:S$ = LEFT$ (S$ + BL$,6):
    GOSUB 50: IF K1 = - 1 AND MID$
    (N$,7) = "A" THEN PRINT "ANNULE"
    :A = K: ON ZZ% GOTO 400,180,180,1
    80
360 IF K1 > 0 THEN ZK = K: GOSUB 40
370 ON ZZ% GOTO 380,430,430,430
378 REM
379 REM *****
    * CREATION *
    *****

380 IF K1 = - 1 THEN PRINT "EXISTE DE
    JA": GOTO 180
390 IF NC = 50 THEN PRINT "MAJ CREATIO
    NS...PATIENCE": GOSUB 210:NC = 0:
    PRINT RU$
400 FOR I = 1 TO 5: INPUT "VARIABLE/DON
    NEE ";ZY$(I): NEXT :R = NZ + NC +
    1: IF NL > 0 THEN R = PL%(NL):NL
    = NL - 1
405 IF A > 0 THEN N$ = S$ + STR$ (1000
    + R): PRINT WT$:A: PRINT N$: GOT
    O 425
410 IF K < = NC THEN FOR L = NC TO K
    STEP - 1:N$(L + 1) = N$(L): NEXT
420 NC = NC + 1:N$(K) = S$ + STR$ (1000
    + R) + STR$ (ZK)
425 PRINT WF$:R: PRINT S$: FOR I = 1 TO
    5: PRINT ZY$(I): NEXT : PRINT RU
    $: GOTO 180
430 IF K1 > 0 THEN PRINT "N'EXISTE PAS
    ": GOTO 180
438 REM
439 REM *****
    * LECTURE DONNEES *
    *****

440 Z$ = N$: IF ZK > 0 THEN Z$ = N$(K)
450 R = VAL ( MID$ (Z$,7,4)) - 1000: PR
    INT RF$:R: INPUT S$: FOR I = 1 TO
    5: INPUT ZY$(I): NEXT : PRINT RU
    $: FOR I = 1 TO 5: PRINT "DONNEE
    "ZY$(I): NEXT : ON ZZ% GOTO 380,4
    60,470,480
458 REM
459 REM *****
    * MODIFICATION *
    *****

460 FOR I = 1 TO 5: INPUT "NOUVELLE DON
    NEE ";ZY$(I): NEXT : GOTO 425
468 REM
469 REM *****
    * CONSULTATION *
    *****

470 PRINT : PRINT "FRAPPEZ UNE TOUCHE";
    : GET Z$: GOTO 180

```

MILOG

VOICI LA CLEF
POUR VOTRE
INFORMATIQUE



MILOG s'adapte à vos besoins
Matériels: choix, décision

APPLE IIe
APPLE III
GOUPII 3

COMMODORE VIC 20 ... coffret promotion 3 150 F

Promotion Sharp ENSEIGNEMENT

PC 1212 850 F TTC

MZ 80 A 6 800 F TTC

EPSON HX-20 prix: nous consulter

SANCO avec traitement de texte nous consulter

promotion carte Z 80 colonnes pour Apple + nous consulter

IMPRIMANTES

EPSON RX-80 • EPSON FX-80 • CENTRONIC •
SANCO • PRAXIS

LOCATION APPLE IIe 630 F la semaine



De nouveaux programmes performants.
Des solutions pour le dossier médical.

Nombreux logiciels et progiciels professionnels ou de jeux.

DEPOT-VENTE

Vendez votre micro dans notre boutique

De particulier à particulier	Neuf	2 ^e main
ATARI CX 2600S	1 400	1 000
EPSON MX 80	5 500	4 000
OLIVETTI 1450	8 500	6 000
GOUPII 2	—	5 000

FORMATION

Initiation au basic et utilisation de progiciels.

FOURNITURES DIVERSES

Disquettes 3M-Multiset - Rubans - Listings, etc.

CREDIT BAIL

A BIENTOT CHEZ:
MILOG informatique
12, rue de Constantinople
75008 PARIS - Tél.: 293.53.38

```
478 REM
479 REM *****
      *   ANNULATION   *
      *****
```

```
480 IF NL = 50 THEN PRINT "CAPACITE PL
      ACES LIBRES ATTEINTE": GOTO 180
490 PRINT "TAPEZ 'O' POUR ANNULER";: GE
      T Z$: PRINT : IF Z$ < > "O" THEN
      180
```

```
500 NL = NL + 1: PL*(NL) = R: PRINT WFS;R
      : PRINT "ANNULE": IF ZK > 0 THEN
      520
```

```
510 PRINT WTS;K:N$ = LEFT$(N$,6) + "A
      ": PRINT N$: PRINT RU$:TD = TD +
      1: GOTO 180
```

```
520 PRINT RU$:NC = NC - 1: IF K < = NC
      THEN FOR L = K TO NC:N$(L) = N$
      (L + 1): NEXT
```

```
530 GOTO 180
```

```
8998 REM
```

```
8999 REM *****
      *   ROUTINE ONERR   *
      *****
```

```
9000 Z = PEEK (222): IF Z = 5 AND Z% =
      1 THEN 130
```

```
9010 IF Z = 5 AND Z% = 2 THEN 150
```

```
9020 PRINT PEEK (222): PRINT CL$
```

Programme FICHIERS (variante)

```
199 REM *****
      * M.A.J. CREATION *
      * SOLUTION NO 2 *
      * REMPLACE -->220 *
      *****
```

```
200 IF NC = 0 THEN 230
```

```
205 GOSUB 210: PRINT RU$: GOTO 230
```

```
210 IF NZ > 0 THEN 215
```

```
212 PRINT D$"CLOSETAB": PRINT OT$: FOR
      I = 1 TO NC:Z$ = LEFT$(N$(I),10
      ): PRINT WTS;I: PRINT Z$:NZ = NZ
      + 1:NT = NT + 1: NEXT : RETURN
```

```
215 LL = 1:NN = NT:NT = 0:NZ = NZ + NC:
      PRINT D1$"OPENTB2,L11": FOR I = 1
      TO NN: PRINT RT$:I: INPUT N$:S$
      = LEFT$(N$,6):A$ = LEFT$(N$(L
      L),6): IF S$ < = A$ OR LL > NC T
      HEN Z$ = N$: GOTO 222
```

```
220 Z$ = LEFT$(N$(LL),10):LL = LL + 1:
      I = I - 1
```

```
222 NT = NT + 1: PRINT D$"WRITETB2,R";NT
      : PRINT Z$: NEXT I: IF LL > NC TH
      EN 226
```

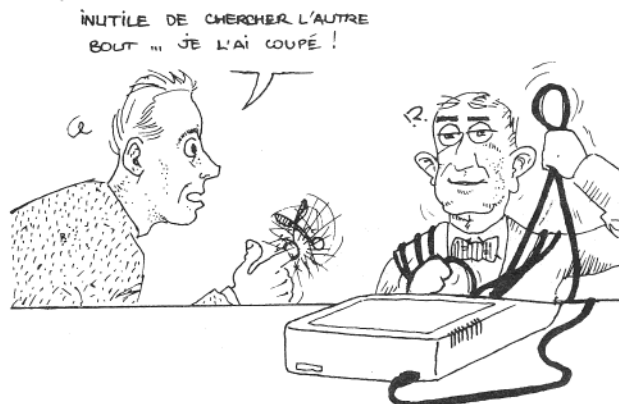
```
224 FOR I = LL TO NC:NT = NT + 1:Z$ =
      LEFT$(N$(I),10): PRINT D$"WRITET
      B2,R";NT: PRINT Z$: NEXT
```

```
226 PRINT CL$: PRINT D$"DELETETAB": PRI
      NT D$"RENAME TB2,TAB": PRINT OT$:
      PRINT OF$: RETURN
```

Francisez le DOS

Philippe Krepper

Si vous êtes lassé de recevoir les messages d'erreur du DOS en anglais, voici un petit programme qui vous permettra de les franciser. Les messages habituels apparaissent en anglais un par un; il vous reste à indiquer à chaque fois le message en français que vous voulez utiliser à sa place. Dès la fin du programme, les nouveaux messages d'erreur sont opérationnels, et mémorisés sur la disquette dans le fichier MESSAGE; leurs positions sont enregistrées dans le fichier TABLE.



Sauvegarde du DOS

Si vous venez d'utiliser ce programme pour modifier les messages du DOS, il vous suffira d'initialiser une disquette comme d'habitude pour que son DOS soit modifié en conséquence, sous réserve que la longueur totale de vos messages soit restée inférieure à 202 caractères. Si celle-ci est supérieure à 202 caractères mais inférieure ou égale à 252, vous ne pourrez pas initialiser une disquette avec ce DOS mais vous pourrez charger les messages en français en "BLOADant" TABLE et MESSAGES.

Chargement des messages

Si vous désirez à un moment donné modifier les messages d'erreur du DOS par ceux que vous avez créés avec ce programme, il suffit de mettre la disquette sur laquelle se trouve la table dans le lecteur et de faire "BLOAD TABLE". A partir de ce moment, vous pouvez vérifier que les nouveaux messages sont actifs.

Vérification des messages

Supposons que vous ayez un programme verrouillé baptisé HELLO, un programme integer INTEX, et que le nom de fichier ZZZ ne soit pas utilisé. Voici les messages habituels que vous devez recevoir, en essayant les exemples non exhaustifs indiqués ci-dessous.

OPEN ZZZ	Not direct command
LOAD ZZZ	File not found
SAVE ZZZ,V49	Volume not found
SAVE ZZZ,S9	Range error
BLOAD HELLO	File type mismatch
INT	Syntax error (si non chargé)
LOAD INTEX	Language not available
SAVE HELLO	File locked
Avec une disquette protégée contre l'écriture :	
SAVE ZZZ	Write protected
Après avoir enlevé la disquette :	
SAVE ZZZ	I/O error
En mettant une disquette pleine :	
SAVE ZZZ	Disk full

NDLR : faites BLOAD TABLE.POM'S et BLOAD MESSAGES.POM'S avec la disquette d'accompagnement de Pom's. Vous aurez un échantillon de ce que l'on peut obtenir en essayant ces exemples.

```
0  REM  *** DOS.ERREUR ***
    *** APPLE 48 K ***
    *** AVEC DOS ***

1  TEXT : HOME : VTAB 5: PRINT " AVEC LE
    PROGRAMME SUIVANT VOUS ALLEZ
    POUVOIR TRADUIRE LES MESSAGES D'E
    RREUR DU S.E.D EN FRANCAIS , MAI
    S L'ENSEMBLE DE VOS MESSAGES NE
    DOIT PAS DEPASSER 252 CARACTER
    ES."

2  PRINT : PRINT " ET SI L'ENSEMBLE DE
    VOS MESSAGES NE DEPASSE PAS 2
    02 CARACTERES ALORS LES NOUVEAU
    X MESSAGES SERONT ENREGISTRES
    DANS UNE ZONE MEMOIRE DU S.E.
    D ET TOUTES LES DISQUETTES
    INITIALISEES"

3  PRINT " A LA FIN DE CE PROGRAMME CON
    TIENDRONT LES MESSAGES FRANCAIS.
    ": PRINT : PRINT " POUR COMMENCER
    APPUYEZ SUR UNE TOUCHE": POKE
    - 16368,0

5  DATA "LANGUAGE NOT AVAILABLE","RANGE
    ERROR","WRITE PROTECTED","END OF
    DATA","FILE NOT FOUND","VOLUME MI
    SMATCH","I/O ERROR","DISK FULL","
    FILE LOCKED","SYNTAX ERROR","NO B
    UFFERS AVAILABLE","FILE TYPE MISM
    ATCH","PROGRAM TOO LARGE"

7  HIMEM: 38143: DIM A$(15),B$(15): FOR
    L = 1 TO 15: IF L = 2 THEN L = 3:
    DATA "NOT DIRECT COMMAND"

10 READ B$(L): NEXT L:E = 3:B = 3: WAIT
    - 16384,128: HOME : POKE - 163
    68,0: FOR L = 1 TO 15: PRINT "MES
    SAGE ERREUR NO:":L: IF L = 2 THE
    N PRINT ",3":L = 3

20 INVERSE : PRINT : PRINT : PRINT B$(L
    ): PRINT : NORMAL : INPUT "NOUVEA
    U MESSAGE ":A$(L):X = LEN (A$(L)
    ): IF X = 0 THEN A$(L) = B$(L):X
    = LEN (A$(L)): PRINT : PRINT "JE
    PRENDS ";B$(L);"": PRINT "COMM
    E NOUVEAU MESSAGE":

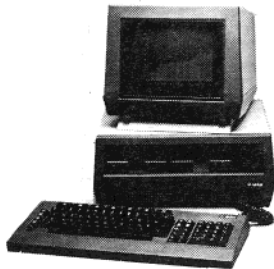
25 B = B + X:E = E + LEN (B$(L)): IF B
    > 252 THEN PRINT : PRINT "VOUS N
```



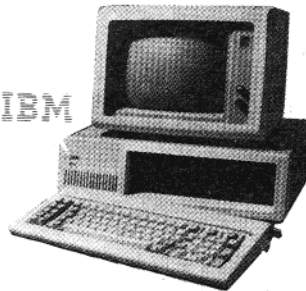
INFORMATIQUE ÉLECTRONIQUE FRANÇAISE

Société Anonyme au Capital de 1099080 F
228, rue Lecourbe, 75015 PARIS - Tél. : 828.06.01 +

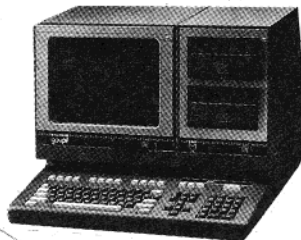
Le SPÉCIALISTE FRANÇAIS de la MICRO-INFORMATIQUE



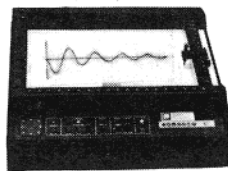
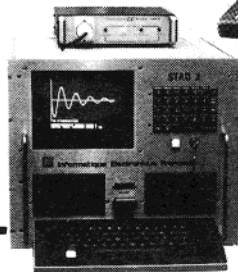
IBM



GOUPIL



I.E.F.



DISTRIBUTION, CONCEPTION, FABRICATION
ASSISTANCE TECHNIQUE

- MICRO-ORDINATEURS
- TERMINAUX INTELLIGENTS
- SYSTÈMES INDUSTRIELS
- SYSTÈMES SCIENTIFIQUES
- CARTES INTERFACES
- CAISSE ENREGISTREUSE INTELLIGENTE
- SYSTÈMES CODE-BARRE
- LECTEURS DE BADGES
- DISQUES
- MÉMOIRES DE MASSE
- SYSTÈME MULTIPOSTES
- RÉSEAUX MULTI-UTILISATEURS
- ALIMENTATIONS DE SÉCURITÉ
- GESTION STOCK
- FACTURATION, etc.
- COMPTABILITÉ GÉNÉRALE
- MAINTENANCE

CENTRE DE DÉMONSTRATION : 193, rue de Javel, 75015 PARIS

```

'AVEZ DROIT QU'A 252 CARACTERES !
"; CHR$(7): FOR T = 1 TO 2500: N
EXT T: RUN
30 PRINT : IF B > E THEN PRINT "VOUS A
VEZ DEJA UTILISE ";B - 3;" CARACT
ERES": PRINT "ALORS QUE LE S.E.D
N'EN UTILISE QUE ";E - 3: GOTO 35
33 PRINT "VOUS N'AVEZ UTILISE QUE ";B -
3;" CARACTERES": PRINT "AU MEME
NIVEAU LE S.E.D EN UTILISE ";; IF
B < E THEN PRINT : PRINT "DEJA
";
34 PRINT E - 3
35 PRINT "-----"
-----": NEXT L:A = 43377:Q
= A: IF B > 205 THEN A = 38144:Q
= A
40 B = 3:C = INT (A / 256):D = A - (C *
256): POKE 42762,D: POKE 42763,C
: POKE A,13: POKE A + 1,7: POKE A
+ 2,141:A = A + 3: FOR L = 1 TO
15: IF L < > 2 THEN FOR M = 1 T
O LEN (A$(L)): POKE A, ASC ( MID
$(A$(L),M,1)):A = A + 1: NEXT M:
POKE A - 1, PEEK (A - 1) + 128:B
= B + LEN (A$(L))
45 POKE 43584 + L,B: NEXT L: HOME : VTA
B 5:X$ = " VOUS POUVEZ CONSERV
ER VOS MESSAGES"
50 IFC = 169 THEN PRINT " VOUS N'AVEZ
PAS DEPASSE 202 CARACTERES VOUS
POUVEZ DONC CONSERVER LES NOUVEA
UX MESSAGES SIMPLEMENT EN INITIAL
ISANT UNE DISQUETTE OU ALORS":X$
= ""
60 PRINT X$;" EN SAUVANT LA TABLE ET
LES MESSAGES ET EN LES FAISANT
CHARGER PAR VOTRE PROGRAMME D
E SALUTATION": IF C < > 169 THEN
PRINT " ET ENFIN EN Y AJOUTANT
LES INSTRUCTIONS SUIVANTES": INV
ERSE : PRINT : HTAB 3: PRINT "HIM
EM:38143:POKE42762,";D;":POKE4276
3,";C
70 NORMAL : PRINT : POKE - 16368,0: PR
INT " VOULEZ VOUS QUE JE SAUVE L
ES MESSAGES ET LA TABLE (O/N)":;
GET R$: IF R$ = "O" OR R$ = "O"
THEN PRINT : PRINT CHR$(4)"BSA
VE TABLE,A$AA3F,L16": PRINT CHR$(
4)"BSAVE MESSAGES,A";Q;"",L";B
80 HOME : PRINT "ANCIENS MESSAGES /";;
INVERSE : PRINT "/ NOUVEAUX ME
SSAGES": NORMAL : IF C = 169 THEN
HIMEM: 38400
90 FOR L = 1 TO 15: IF L = 2 THEN L = 3
100 X = LEN (B$(L)) + LEN (A$(L)): IF
X > 39 THEN 120
110 PRINT B$(L);; INVERSE : HTAB 41 -
LEN (A$(L)): PRINT A$(L);; GOTO 1
40
120 PRINT B$(L);;Y = LEN (A$(L)):Z =
LEN (B$(L)):W = 40 - Z: FOR M = W
TO 1 STEP - 1: IF MID$(A$(L),
M,1) < > " " THEN NEXT M
125 INVERSE
130 W$ = LEFT$(A$(L),M - 1):Y$ = RIGH
T$(A$(L),Y - M): HTAB 41 - LEN
(W$): PRINT W$;: HTAB 41 - I.FN (
Y$): PRINT Y$;
140 NORMAL : NEXT L

```


Saisie de réels en Pascal

Philippe Chemin

La procédure Pascal que je vous propose autorise la saisie de réels, hors exposant, mais avec pavé numérique. Celui-ci est formé par le bloc de touches à partir de "7", "8" et "9" sur un clavier QWERTY. Il est facile de modifier le programme à votre convenance si vous possédez un clavier AZERTY, ou si vous préférez situer le bloc ailleurs.

Touches du clavier	Bloc numérique
<pre> 7 8 9 U I O J K L M, </pre>	<pre> 7 8 9 4 5 6 1 2 3 0. </pre>

NDLR : la procédure PREEL de Michel Crimont (Pascal à 12 chiffres) fait la même chose, mais sans le pavé et de façon moins interactive. Par contre, elle permet l'utilisation d'exposants.

```

PROGRAM ESSAI;
TYPE CHOIDECA=SET OF CHAR;
VAR TEST:REAL;
    HOME,BS,CR,SON,ESC:CHAR;
    POINTEXISTE:BOOLEAN;
    PAVE:BOOLEAN;
PROCEDURE PRINFO;
BEGIN
    BS:=CHR(8);
    HOME:=CHR(12);
    CR:=CHR(13);
    SON:=CHR(7);
    ESC:=CHR(27);
END;(*PRINFO*)
FUNCTION PRENCAR(BONSET:CHOIDECA):CHAR;
VAR CH :CHAR;
    BON : BOOLEAN;
BEGIN
    REPEAT
        READ(KEYBOARD,CH);
        IF EOLN(KEYBOARD) THEN CH:=CR;
        BON:=CH IN BONSET;
        IF NOT BON
            THEN WRITE(SON)
            ELSE IF CH IN ['.',' ','0']
                THEN BEGIN
                    IF PAVE
                        THEN BEGIN
                            CASE CH OF
                                'M':CH:='0';
                                'J':CH:='1';
                                'K':CH:='2';
                                'L':CH:='3';
                                'U':CH:='4';
                                'I':CH:='5';
                                'O':CH:='6';
                                ',':CH:='.';
                            END;
                            IF CH IN ['.',' '] THEN POINTEXISTE:=TRUE;
                        END;
                    END;
                WRITE(CH);
            END;
    UNTIL BON;
    PRENCAR:=CH;
END;(*PRENCAR*)
PROCEDURE REEL(X,Y,LMAX,DECI:INTEGER;VAR NOMBRE:REAL);
VAR S1 :STRING(1);
    I :INTEGER;
    CONT :STRING;
    OKSET :CHOIDECA;
    VAL :INTEGER;
    DENOM :INTEGER;
    VIRG :INTEGER;
    POSITIF:BOOLEAN;
    DEBUT :INTEGER;
PROCEDURE SORTIE;
BEGIN
    GOTOXY(X,Y);
    WRITE(NOMBRE:LMAX:DECI);
    PAVE:=FALSE;
    EXIT(REEL);
END;(*SORTIE*)
BEGIN
    PAVE:=TRUE;
    OKSET:=['0'..'9','U','I','O','J'..'M',ESC];
    POINTEXISTE:=FALSE;
    S1:=' ';
CONT:='';
VIRG:=0;
POSITIF:=TRUE;
GOTOXY(X,Y);
WRITE(' ':LMAX);
GOTOXY(X,Y);
REPEAT
    IF LENGTH(CONT)=0
        THEN S1(1):=PRENCAR(OKSET+[CR,'-'])
        ELSE IF (LENGTH(CONT)=LMAX) OR (VIRG=DECI)
            THEN S1(1):=PRENCAR([CR,BS])
            ELSE IF POINTEXISTE
                THEN S1(1):=PRENCAR(OKSET+[CR,BS])
                ELSE S1(1):=PRENCAR(OKSET+[CR,BS,'.',' ']);
        IF S1(1)=ESC THEN SORTIE;
        IF S1(1) IN OKSET+['.',' -']
            THEN BEGIN
                CONT:=CONCAT(CONT,S1);
                IF S1(1) IN ['-'] THEN POSITIF:=FALSE;
                IF (S1(1) <> '.') AND (POINTEXISTE) THEN VIRG:=VIRG+1;
            END
        ELSE IF S1(1)=BS
            THEN BEGIN
                WRITE(BS;' ',BS);
                IF POINTEXISTE THEN VIRG:=VIRG-1;
                CASE CONT[LENGTH(CONT)] OF
                    '.':BEGIN
                        POINTEXISTE:=FALSE;
                        VIRG:=0;
                    END;
                    '-':POSITIF:=TRUE;
                END;
                DELETE(CONT,LENGTH(CONT),1);
            END;
    UNTIL S1(1)=CR;
    IF LENGTH(CONT)<>0
        THEN BEGIN
            NOMBRE:=0;
            POINTEXISTE:=FALSE;
            DENOM:=10;
            IF POSITIF THEN DEBUT:=1 ELSE DEBUT:=2;
            FOR I:=DEBUT TO LENGTH(CONT) DO BEGIN
                VAL:=ORD(CONT[I]);
                IF CONT[I]='.'
                    THEN POINTEXISTE:=TRUE
                ELSE BEGIN
                    IF POINTEXISTE
                        THEN BEGIN
                            NOMBRE:=NOMBRE+(VAL-ORD('0'))/DENOM;
                            DENOM:=DENOM*10;
                        END
                    ELSE BEGIN
                            NOMBRE:=NOMBRE*10;
                            NOMBRE:=NOMBRE+(VAL-ORD('0'))
                        END;
                END;
            END;
            IF NOT POSITIF THEN NOMBRE:=-NOMBRE;
            SORTIE;
        END;(*REEL*)
    BEGIN
        PRINFO;
        WRITE(HOME);
        REPEAT
            GOTOXY(0,5);
            WRITE('XXXXXXXXXXXXXXXXXXXX');
            TEST:=-1000.05;
            REEL(4,5,8,2,TEST);
            GOTOXY(0,7);
            WRITELN(TEST,' ');
        UNTIL TEST=0;
    END.

```

3 LIVRES ET LEUR DISQUETTE POUR VOTRE APPLE

Pour Apple II, II plus et //e DOS 3.3, version 48 K ou plus



Vous avez acheté le livre "La comptabilité sur Apple II" de Serge et Gérard Lillio, et vous n'avez pas le temps nécessaire pour taper vous-même les programmes sur votre Apple. Dans ce cas, cette disquette vous est tout particulièrement destinée!

L'ensemble des programmes présentés ici constitue la réplique exacte des listings proposés dans l'ouvrage, à l'exception des ordres pour imprimante.

Cette disquette doit être considérée comme une aide au lecteur et non comme un progiciel de comptabilité. Ainsi, les trois exemples de libellés cités en annexe dans le livre et ci-inclus, doivent être envisagés en tant que points de départ pour une utilisation plus personnelle.

La disquette seule : 185,00FF / 1425,00FB

Le livre décrit un ensemble de programmes de comptabilité générale en Basic pour petites entreprises, professions libérales, artisans, commerçants. Avec édition des livres-journal, grands livres, balances, bilans. Avec calcul des ratios. Programme spécial intéressant l'adaptation et la personnalisation du plan comptable. Et... quelques "ficelles" pour Apple II.

Le livre seul : 102,00FF / 785,00FB

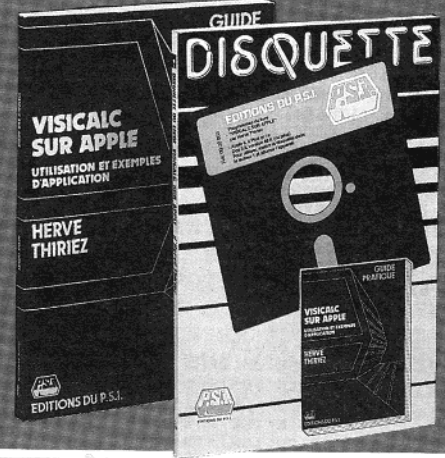
Voici une disquette qui permet aux lecteurs de "Visicalc sur Apple" d'accéder directement à tous les tableaux Visicalc présentés dans le livre. Ils trouveront sur cette disquette la réplique exacte des tableaux présentés dans l'ouvrage. S'y ajoute un programme en Basic permettant l'impression des formules d'un tableau, sur une ou deux colonnes, dans l'ordre de A1 vers le bas à droite du tableau.

Attention : cette disquette ne peut être utilisée sans la disquette maîtresse Visicalc.

La disquette seule : 185,00FF / 1425,00FB

Après une présentation progressive du modèle Visicalc, l'ouvrage présente un ensemble d'exemples d'applications, échéancier de remboursement, feuille d'impôt, gestion de copropriété, paye, facturation..., permettant d'introduire les différentes instructions et astuces d'utilisation.

Le livre seul : 82,00FF / 635,00FB



Vous avez acheté le livre "Mathématiques et statistiques", et vous n'avez pas le temps de taper vous-même les programmes sur votre Apple? Alors cette disquette vous est tout particulièrement destinée!

L'ensemble des programmes présentés ici constitue la réplique exacte des listings proposés dans l'ouvrage.

Cette disquette doit être considérée comme une aide au lecteur et non comme un progiciel de mathématiques.

La disquette seule : 185,00FF / 1425,00FB

Le livre décrit un ensemble de programmes en Basic de mathématiques et de statistiques, niveau supérieur. Chaque problème traité comporte une introduction numérique, un exposé de la technique de programmation utilisée, un organigramme détaillé et un programme complet suivi d'un exemple d'utilisation. Les programmes sont structurés de manière à permettre une insertion dans des programmes plus importants.

Le livre seul : 92,00FF / 710,00FB



Demandez notre catalogue gratuit

MICRO-EXPO Stand N 14 - N 15



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

Envoyer ce bon accompagné de votre règlement à P.S.I. DIFFUSION ou, pour la Belgique et le Luxembourg, à P.S.I. BENELUX

Paiement par chèque joint Paiement en FF par carte bleue VISA (à P.S.I. DIFFUSION uniquement) Signature (obligatoire pour paiement par carte de crédit)

DESIGNATION	NOMBRE	PRIX
TOTAL		

N° _____ Date d'expiration _____

NOM _____ PRENOM _____

rue _____ N° _____

Code post. _____ Ville _____

au Canada
SCE Inc.
3449 rue Saint-Denis
Montréal Québec H 2X3 L 1
Tél. : (514) 843.76.63

en Espagne
P.S.I. IBERICA
Ferraz 11 Madrid 8
Tél. : 247.30.00

en Suisse
CRISPA
16, avenue de Beaumont
1700 Fribourg
Tél. : (037) 24.43.76

Micro-informations : les nouveautés

Jean-François Duvivier

Pom's vous offre à partir de ce numéro une rubrique "micro-informations" régulière. On y trouvera pêle-mêle informations et nouveautés pour la gamme Apple (II, //e et ///). En fin de rubrique sont regroupées les adresses des différentes entreprises citées, ainsi que les références dans le cas d'informations tirées de revues étrangères. Les prix mentionnés sont habituellement donnés en francs français TTC. Ils ne sont qu'indicatifs et peuvent être sujets à variation.

Les lecteurs de disquettes

Les progrès effectués dans le domaine des lecteurs de disquettes prennent plusieurs aspects : formats réduits (disquettes 3.5 pouces que l'on devrait voir bientôt apparaître en compatibles Apple), densités accrues (plus de 1 mégaoctet sur une disquette 5 pouces 1/4) et encombrement réduit. RMI propose des unités de disquettes de marque TEAC, entièrement compatibles contrôleur Apple, mais 30% moins hautes. Prix sans contrôleur : 2950 FF.

Pour demain, avec les disquettes à enregistrement vertical, des disquettes 5 pouces avec des capacités formatées de 5 mégaoctets (2,5 Mo par face). Ces merveilles sont encore au niveau prototype. Patientez donc un peu...

Les imprimantes

Domaine très actif actuellement, les imprimantes où l'on vient d'enregistrer coup sur coup plusieurs nouveautés. Tout d'abord, à tout seigneur tout honneur, les deux imprimantes de marque Apple. La première est de type matriciel (97 points) et ressemble furieusement à la NEC. Rapidité (120 caractères par seconde), souplesse (sept alphabets spéciaux, huit tailles de caractères), redéfinition des caractères, espacement proportionnel et copie d'écran graphique. Ses possibilités ainsi que son prix (7200 FF avec interface) devraient en faire une concurrente sérieuse des Centronics, Microline et Epson.

La deuxième, grâce à une impression à marguerite, permet une frappe de

très haute qualité. Destinée aux applications de traitement de texte ou à la sortie de documents de qualité professionnelle, elle présente des caractéristiques intéressantes : 40 caractères par seconde (cps), largeur jusqu'à 38 cms (198 colonnes avec caractères resserrés), tabulations, jeux de caractères internationaux... Son prix de 19000 FF avec interface la réserve cependant aux applications professionnelles.

De son côté, Epson vient de sortir deux nouveaux modèles. La FX80, tout d'abord, offre de nouvelles améliorations par rapport à la série MX82 : matrice 9x9, 160 cps, mémoire vive de 2K utilisable soit en buffer d'entrée, soit en redéfinition de caractères, possibilité de sauts de ligne arrière... Son prix est d'environ 9900 FF avec interface. Quant à l'autre modèle, la RX80, je ne pourrai vous décrire ici ses caractéristiques car je n'ai pu me procurer la documentation. Seul élément connu, son prix : 6500 FF avec interface.

Les cartes

Une nouvelle sorte de carte est en train d'apparaître. Il s'agit d'une carte comprenant un 6502C (fonctionnant à 4 MHz au lieu de 1 MHz pour le 6502 normal) et 64K de RAM. Un petit programme lui permet de désélectionner le 6502 en place sur la carte mère et d'effectuer toutes les opérations à sa place, à la seule différence qu'il les exécute 3 à 4 fois plus vite. Ce gain de temps ne s'applique bien entendu pas aux opérations avec la disquette qui s'effectuent toujours avec la même lenteur.

Pour exploiter ce gain de temps, il faudra que j'essaie le programme d'échecs Sargon 2.5 au niveau 6 : je ne l'ai utilisé qu'une fois au niveau 4, pour abandonner par "KO" à 3h du matin après 5h d'échanges désespérément lents. Outre l'exécution de gros programmes nécessitant peu d'accès disquette, les applications graphiques semblent pouvoir offrir de nouvelles perspectives dans l'animation avec cette carte.

Conçue par la société américaine Number Nine, cette carte va être importée en France par BIP (Prix français *non encore précisé* ; prix américain : 650 dollars).

Steve Watt est anglais. Il s'est installé à Paris et a fondé Microperiph pour distribuer des produits spécifiquement consacrés à l'Apple. Parmi les bonnes affaires qu'il propose, retenons la carte 80 colonnes U-Term à 1490 FF, un dispositif permettant la répétition automatique des caractères lorsqu'une touche est laissée enfoncée (inutile pour le //e !) pour 110 FF, des disquettes 3M renforcées à 180 FF la boîte de 10. Il a également importé la carte "Accelerator II" qui, grâce à un 6502 fonctionnant à 4 MHz, accélère dans un facteur 3 à 4 la vitesse d'exécution des programmes sur Apple (3950 FF). Si vous passez chez lui, dites que vous venez de notre part, vous serez bien accueilli. Attention, n'hésitez pas à entrer, c'est au premier étage.

Parmi les nouvelles cartes, citons aussi toutes celles qui permettent de conserver les informations une fois l'Apple éteint. MPC Peripherals a conçu une carte 128K à base de mémoires à bulles, qui permet de remplacer réellement un lecteur de disquettes grâce au "Disk Emulator", y compris au niveau de la non-volatilité des informations.

Autre démarche, celle adoptée par Legend qui propose une carte 18K en technologie CMOS, avec batterie incorporée. Pourquoi 18K ? Parce que le DOS prend exactement 18K de mémoire vive et peut se stocker dans cette carte. Ainsi, dès la mise sous tension, le DOS est présent dans l'Apple. Bien entendu, tout autre programme peut être mis dans cette carte, pour obtenir un système "turnkey" exécutant le programme dès la mise sous tension de l'appareil. Toutes les cartes Legend sont importées par BIP. Prix de la carte 18SRC : 1550 FF.

La carte Legend 128K DE fonctionne maintenant sur Apple II et //e ; elle est fournie avec interface Legend, Boot Visicalc (160K utiles), programme de gestion de fichiers ultra-rapide, simulateur de disque, ... et manuel en français. Nouveau prix : 5.600 FF ; la carte 64K est disponible à 3.600 FF.

Vous trouverez aussi chez BIP la carte "80 colonnes" UltraTerm, qui vous donne les choix suivants : 24*180, 32*80, 48*80, 24*96, 32*128, 24*132 et 24*160. Compatibilité avec Visicalc et Applewriter II.

Avec la carte PCPI 88CARD, compatible Apple II et //e, vous ajoutez 64K aux 64K de votre Apple, pour un total de 128K de mémoire centrale. Mais surtout, vous disposez du système d'exploitation MS-DOS de l'IBM OP. La carte avec le MS-DOS et le MBASIC est vendue 595 FF. Le CP/M-86 est disponible en option. Fabricant : Personal Computer Products Inc. (PCPI).

Autre produit PCPI : l'APPLI-CARD offre 64K de plus, un écran de 70 colonnes, et le choix entre des processeurs Z80 à 4MHz (295) ou 6MHz. Avec cela, votre CP/M tourne 2 à 3 fois plus vite qu'avec une carte Z80 habituelle.

Les communications

Le Service Calvados de l'American College in Paris étoffe ses services. Citons notamment la prochaine mise en place d'un service Bourse qui permettra l'accès à une base de données de 20.000 cotations quotidiennes (marchés français et étrangers) et la possibilité de création de bases de données privées. Une documentation complète peut être obtenue sur demande.

La société Communication Interface présente un ensemble Microtelex qui permet d'accéder à partir d'un Apple au réseau télex national et international. Les avantages par rapport aux matériels télex classiques résident dans les possibilités de traitement de texte, de numérotations abrégées, de transmission différée et de gestion des différents types de rejet du réseau. Autre avantage, et non des moindres : le silence ! Le matériel est bien entendu agréé par les PTT. Documentation disponible sur demande.

Acquisition et traitement de données

EUROTRON introduit en France l'ISAAC 91A de Cyborg (USA) : il s'agit là d'un système polyvalent d'acquisition rapide de données et de contrôle de processus. Entrées analogiques (tensions, courants, capteurs et transmetteurs 4/20 mA, thermocouples) ou numériques (via RS232 et IEEE); sorties analogiques ou numériques.

Les jeux

Saluons ici la société Penguin Software qui, face au piratage des logiciels, essaie de trouver des solutions originales. Après avoir l'an dernier ôté toutes les protections de ses logiciels utilitaires graphiques (Special Effects, The Complete Graphics Sys-

tem et Graphics Magician), elle a annoncé à partir du 1er mars un prix unique pour tous ses jeux : 19,95 dollars, ce qui représente entre 200 et 250 FF. Parmi les jeux édités par Penguin Software, notons Pie-Man, Spy's Demise, Transylvania, Thunderbombs et Crime Wave. Souhaitons d'une part que l'expérience soit concluante pour Penguin Software, d'autre part qu'elle fasse des émules parmi les autres sociétés. Dernier souhait, et non des moindres, que tous ces jeux soient très vite distribués par des boutiques françaises à des prix raisonnables.

Les jeux d'aventures ne connaissent plus de limites. L'un des plus célèbres, Time Zone, dispose de 1400 images graphiques différentes, le tout étant présenté sur 6 disquettes recto-verso ! Plus récent, Sherwood Forest charge ses images en 1 seconde !

Les programmes

LOGAWAL lance la version 3.0, entièrement en français, du logiciel WalCore de gestion de fichiers tournant sur CP/M. Ce progiciel peut traiter 13.000 fiches de 500 caractères avec accès instantané par l'index, tris et sélections multi-critères, impression en tableaux, étiquettes et lettres personnalisées, modification de la structure des fichiers existants. Prix : 3.143 FF.

Autres produits de la même société : version 3.0 de Wordstar en français et système de gestion de base de données InfoStar (combinaison de DataStar et ReportStar). Wordstar et InfoStar sont vendus au prix de 25.500 FB HT chacun.

Magicalc vous donne toutes les fonctions du Visicalc plus des améliorations à un prix plus intéressant : 2.000 FF. Possibilités : colonnes de largeur variable, utilisation avec disque dur, colonnes invisibles ou protégées, 70 colonnes par logiciel, reconnaissance automatique de nombreuses extensions mémoire. Magicalc est vendu par BIP.

Executive Briefing System (EBS) est un ensemble de deux programmes distribués par SOFITEC dont le but est de vous permettre la projection d'images graphiques. Le premier programme sert à composer la projection, le second à la présenter. Pour la composition : bordures, choix de caractères, histogrammes, ...

Le programme de gestion de cabinet dentaire Agatha du Dr. Gausson vient d'apparaître en version IV. Avec la carte Legend, vitesse cinq fois supérieure. Capacités possibles : 300, 510 ou 675 fiches de patients en cours de traitement.

La base de données Omnis en Pascal est à présent francisée et commercialisée par la société KA. Omnis est, selon la publicité, plus puissant que DB Master et aussi facile à utiliser que PFS.

Enfin, si vous cherchez à classer vos disquettes dans des enveloppes en plastique rangeables dans des classeurs, FRANIC vous donne le choix entre plusieurs types de pochettes. La feuille plastique avec deux poches pour disquettes 5 pouces est vendue 40 FF la dizaine.

Le marché des micro-ordinateurs

Le domaine des ordinateurs compatibles Apple se développe rapidement. Procès à la clé, Apple tente d'enrayer le phénomène. Il semble même que des saisies et destructions soient effectuées par les douanes américaines lors de l'importation de certains de ces matériels aux USA. Outre le BASIS 108 dont nous vous parlons dans ce numéro et le Golem dont nous vous parlerons, on peut trouver aux Etats-Unis le "Orange+", le "Pineapple", le "AP-II", le "Franklin Ace 1000 et 2000", le "Syscom"... et la liste n'est certainement pas complète !

D'après une étude de marché de la société "Future Computing", le parc des différents micro-ordinateurs s'établissait comme suit en fin 1982 : Commodore VIC 20 : 750.000 unités, Apple II et ZX80/81 : 600.000, Texas TI-99/4A : 575.000. Une étude qui a le mérite de faire le point des différentes forces en présence, excepté l'IBM OP qui démarre très fort, mais dont les chiffres sont, comme toujours, sujets à caution.

Les informations

La revue Softalk (la plus épaisse des revues consacrées à l'Apple : 300 pages mensuelles) publie le résultat d'une étude effectuée auprès de ses lecteurs sur les logiciels les plus populaires parmi ceux commercialisés entre fin 81 et fin 82. Grand gagnant : Choplifter (vous savez, l'hélicoptère qui va sauver les otages !) créé par Broderbund Software. Le second, très détaché également par rapport aux suivants, est Wizardry de Sir-tech. Le troisième n'est autre que Locksmith 4.1, effectivement très populaire chez certains utilisateurs.

Pour les possesseurs d'Apple ///, une publication "The Apple Three Newsletter" est éditée trimestriellement aux USA par Mediaworks. Coût annuel : 15 dollars (prévoir près du double à cause des frais d'envoi).

Ceux de nos lecteurs à qui Pom's ne suffit pas et qui se gavent de revues étrangères, pourront trouver les principales d'entre elles chez SIDEG ou chez SIVEA, qui se révèlent avoir le choix le plus étendu (mais surtout n'y allez pas pour acheter votre Pom's trimestriel : mystère d'entre les mystères, il y est souvent en rupture de stock !).

Adresses

BIP - 22 rue Joseph Dijon - 75018 Paris - Tel (1) 255.44.63
 Communication Interface - 2, place d'Aligre - 75012 Paris - Tel (1) 341.85.85

EUROTRON - 34 avenue Léon Jouhaux - Z.I. - 92167 ANTONY CEDEX.

FRANIC - Rue de la Chapelle - B.P.2 - 49190 Beaulieu s/Layon-Tel (41) 78.36.55

Gausson (Dr. Pierre) - CD Soft - 59bis rue de la Biche - 30000 Nîmes

KA - 212 rue Lecourbe - 75015 Paris - Tel (1) 533.13.50

LOGAWAL - 200 avenue Winston Churchill (Bte 22) - 1180 Bruxelles - Belgique

Mediaworks - POB 2757 - San Francisco CA 94126

Microperiph - 60 rue Ducouedic - 75014 Paris - Tel (1) 321.41.80. Ouvert de 10 à 12h et de 14 à 18h du mardi au samedi (1er étage).

MPC Peripherals Corp. - 9424 Chesapeake Drive - San Diego, CA 92123 - USA. Voir Byte Nov 82, page 287.

Personal Computer Products Inc. - 16776 Bernardo Center Drive - San Diego, Ca. 92128 - USA

Penguin Software - 830 4th Avenue - Geneva IL60134 - USA

RMI - 71 Bd Marceau - 92700 Colombes - Tel 785.68.89

Service Calvados - BP 21-07 - 75327 Paris Cedex 07 - Tel (1) 705.09.04

SIDEG - 170 rue Saint Charles - 75015 Paris - Tel (1) 557.79.12

SIVEA - 31 bd. des Batignolles - 75008 Paris - Tel (1) 522.70.66

SOFITEC - 207 rue Galliéni - 92100 Boulogne - Tel (1) 605.88.78

Trucs et astuces

Listings désassemblés

Quand on désassemble des programmes en langage machine, la commande L du moniteur n'affiche que 20 instructions. Pour en afficher plus, on peut appuyer plusieurs fois sur L, mais c'est fatigant, surtout s'il faut désassembler de longs programmes sur imprimante.

Pour éviter cela depuis le moniteur, il suffit de faire :

[adresse] L34:n
 où n est le nombre de caractères de l'adresse, de 0 (s'il

n'y en a pas) à quatre chiffres. Il ne faut pas mettre d'espace, sauf après le n, où il est obligatoire.

Exemples :
 L34:0
 300L34:3

On trompe ainsi le moniteur : remettre la mémoire 34 à n lui indique qu'il faut exécuter des commandes à partir du n^e caractère de la ligne (ici L), l'espace lui indiquant justement que l'on n'est pas en fin de ligne.

Il faut appuyer sur RESET pour arrêter le listing (ou CTRL-C si PLE est en mémoire).

ISSAC™ TRANSFORME UN APPLE II EN SYSTÈME CALCULATEUR POUR L'INDUSTRIE & LA RECHERCHE SCIENTIFIQUE

GRACE A :

SES ENTRÉES

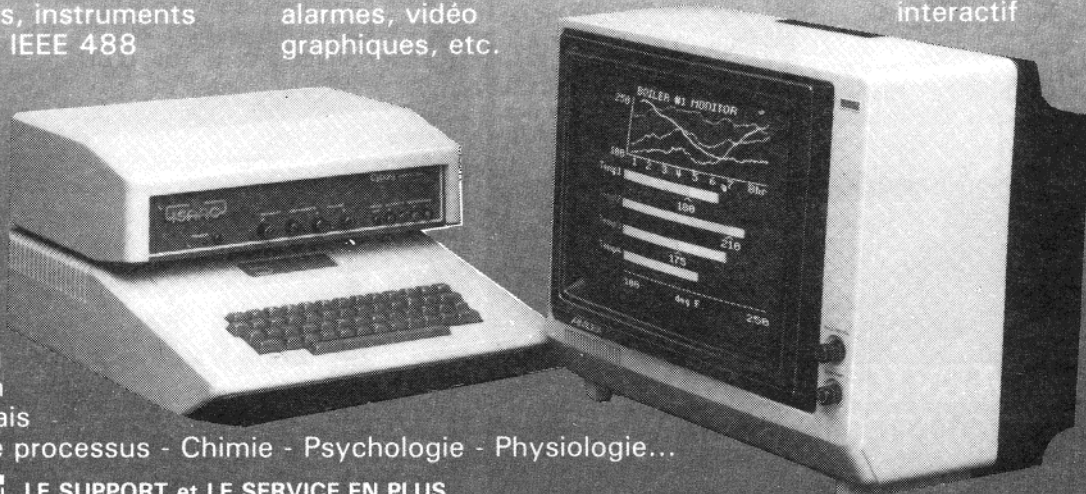
Volts, mA, capteurs, sondes
 transmetteurs, instruments
 RS 232, IEEE 488

SES SORTIES

analogiques, numériques
 alarmes, vidéo
 graphiques, etc.

et... SON LabSoft

langage évolué
 interactif



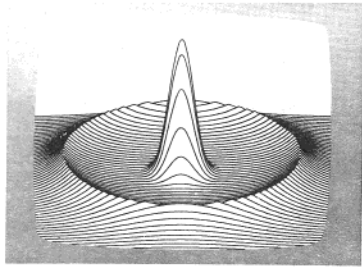
Applications :

- Engineering
- Automation
- Banc d'essais
- Contrôle de processus - Chimie - Psychologie - Physiologie...

EUROTRON
 DIVISION MICROINFORMATIQUE

LE SUPPORT et LE SERVICE EN PLUS

34, Avenue Léon-Jouhaux, ZI 92167 ANTONY Cédex, Tél. 668 10-59 (5 lignes)



VOUS INVESTISSEZ EN MICROINFORMATIQUE, FAITES LE EFFICACEMENT.

Vous allez investir dans un ordinateur personnel.
Vous voulez être guidé dans votre choix et garantir au maximum votre investissement.

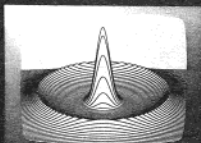
Vous voulez vous servir efficacement de votre ordinateur personnel et souhaitez une assistance soutenue et suivie tant sur votre disposition un système d'assistance efficace qui vous permettra de réussir votre investissement et d'optimiser vos résultats.

MINIGRAPHE MICROINFORMATIQUE, créée à Boulogne-sur-Seine il y a trois ans par Jean-Louis Orsini, met à votre disposition un système d'assistance efficace qui vous permettra de réussir votre investissement et d'optimiser vos résultats.

Téléphonez-lui, il vous donnera toutes les informations nécessaires.



En nous rendant visite, vous pourrez voir fonctionner des logiciels performants sur nos ordinateurs personnels.



MINIGRAPHE MICROINFORMATIQUE

263, Boulevard Jean-Jaurès 92100 Boulogne

608/44/31

DB création

B.I.P. Information

Un système informatique ne se limite pas à sa configuration de base.

Il faut pour l'exploiter un ensemble de logiciels et interfaces spécialisées efficaces.

C'est dans cet esprit que BIP propose ses logiciels et interfaces pour Apple 2.

UTILISATEURS DE ...CALC :

Le temps des pionniers de Visicalc avec 40 colonnes et 18K mémoire est bien dépassé !

Aujourd'hui, grâce à BIP :

- Utilisez les touches nouvelles de l'Apple II^e.
- Jusqu'à 512K mémoire.
- 80 colonnes avec carte, 70/40 sans.
- Largeur des colonnes indépendante, cachées ou protégées.
- Stockez les fichiers sur disquette ou disque dur.
- 16 000 cases utilisables au lieu de 8 000 en Visicalc.
- Profitez de toutes ces possibilités supplémentaires et beaucoup d'autres, **demandez MAGICALC** dès maintenant disponible chez les meilleurs revendeurs.

BON TEMPS AVEC TAMPON d'imprimante

Une interface Tampon d'imprimante a une mémoire additionnelle qui emmagasine très vite l'information à imprimer et la redistribue à l'imprimante à sa vitesse.

Elle vous permet de continuer à travailler sans attendre la fin de l'impression.

BIP a toute une gamme d'interfaces, renseignez-vous pour trouver celle qui correspond à vos besoins.

BIP

C'est aussi toute une gamme d'accessoires pour les jeux graphiques : joystick, paddles, interfaces d'animation stéréo sonore et paroles.

Et toujours :

- Ventilateur RH Electronic.
- Carte Interface Vidéo.
- Carte Synthétiseur Vocal.
- NOUVELLE Carte 128k LEGEND pour Apple II+ , II^e et ///

PROMOTION SPECIALE AU MOIS DE JUIN :
Carte LEGEND 128K + MAGICALC : 5 700 FF. H.T.

BIP sera présent à MICRO-EXPO stand N 26
Documentation sur demande.
En vente chez les meilleurs revendeurs.



B.I.P.

l'informatique personnalisée

Distributeur : BIP 22 rue Joseph Dijon 75018 Paris Tél. 255 44 63

Magicalc est une Marque déposée de Artsci.
Visicalc est une Marque déposée de Visocorp.
Apple est une Marque déposée de Apple Computer Inc.
Legend est une Marque déposée de Legend Ind.

Courrier des lecteurs

Olivier Herz

1 - J'aurais grand besoin d'un programme à compacter les programmes. Où trouve-t-on cela ? On peut gagner 1/4 de la place mémoire occupée par le programme.

2 - S'il vous arrive de retrouver une ligne de programme modifiée, par exemple un "*" que vous avez mis et qui a disparu, ou un token pris pour un autre, et si la modification se reproduit après correction, ne cherchez pas la réponse pendant trois jours (comme cela m'est arrivé). J'ai été jusqu'à ramener ma machine au vendeur pour qu'il l'essaye...

Commencez par vérifier les POKES dans votre programme. En particulier, si vous avez eu le malheur de taper POKE 333,33 au lieu de POKE 33,33 pour un programme de capture ou pour l'édition de lignes, le POKE modifié à contre-temps provoque, de façon difficilement prévisible, ce genre de modification.

Cela vient de m'arriver en entrant en machine le programme "Les 4 ponts", pour lequel il faut remercier Olivier Herz. En ligne 100, j'avais entré POKE 333,0 au lieu du 233,3. Résultat, lors du RUN, erreur en ligne 310 où le E avait disparu de la formule à l'endroit du "10*E". J'ai essayé de mettre le produit entre parenthèses : c'est alors le "*" qui disparaissait !

Docteur J. Rozoy - 26 rue du Petit Bois - 08000 Charleville-Mézières

1 - Il existe deux programmes commerciaux que nous connaissons pour compacter les programmes BASIC. Vous en trouverez un dans Dakin (voir l'analyse des utilitaires de documentation dans Pom's 2), qui offre par ailleurs d'autres services.

Un autre programme commercial, Applesoft Optimizer, compacte encore plus les programmes car, outre l'élimination des REM et le raccordement des lignes raccordable, il opère une renumérotation complète et rebaptise les variables afin de les raccourcir autant que possible.

Bien entendu, dans un cas comme dans l'autre, il est malaisé de modifier un programme compacté. L'idéal est de ne compacter qu'en fin de mise au point, et de garder pour les modifications éventuelles du programme une version non compactée.

La place mémoire que vous gagnez dépend de la façon dont votre pro-

gramme était écrit. Vous gagnerez beaucoup de place (parfois jusqu'à 40%) s'il comporte de nombreuses instructions REM, s'il utilise des noms de variables longs, s'il possède des numéros d'instructions longs et s'il met peu d'instructions par ligne.

2 - Merci ! Nous savions qu'il fallait se méfier des POKES intempestifs, mais nous n'avions pas encore rencontré ce problème.

Je n'ai pas la fonction CHOOSE sur mon Visicalc. Comment doit-on modifier la formule de la case B20 dans l'exemple de feuille d'impôts du livre "Visicalc sur Apple" pour obtenir le bon résultat sans cette fonction ?

M. Morcrette - 02100 St Quentin

Il suffit de déplacer avec [/M] la ligne 27 après les deux lignes suivantes (en position 29), puis de remplacer dans la formule de la case B20 la fonction CHOOSE(B26,A27...M27) par LOOKUP(B26,A28...M28).

Après avoir lu la lettre d'Olivier Herz dans le Courrier des Lecteurs du Pom's 4, j'ai essayé de mettre en place le mini-assembleur. J'obtiens bien le mini-assembleur, mais aussi la première ligne assemblée. Par exemple :

```
!-2000--LDA-#00 me donne :  
2000--A9-00-LDA-#800
```

A partir de là, le système "se plante" et je dois faire RESET pour en sortir. Que faut-il faire ?

G. Gilbert - 97300 Cayenne

Il y a en effet une coquille dans ce Courrier des Lecteurs : il faut lire "8537 :85".

Je tiens à vous livrer une expérience désagréable que j'ai eue avec le compilateur Pascal. J'avais tapé la séquence suivante :

```
VAR NOM : STRING ;  
TB : TAB ;  
I,J : INTEGER ;  
(* Compteur de boucle)  
LBM : LIBMOR ;  
LBV : LIBVER ;
```

(* Procédures courantes *)
etc ...

A la compilation, j'avais l'erreur de syntaxe "UNDECLARED IDENTIFIER" à la première occurrence de LBM et LBV, alors que je croyais les avoir déclarées en tête de programme. J'ai mis plus d'une heure à découvrir qu'il manquait effective-

ment l'astérisque avant la parenthèse fermante en ligne 3, ce que je ne voyais pas à cause de mon écran à 40 colonnes.

Michel Lafon - 72 rue du Maréchal Foch - 42300 Riorges

Merci pour cet exemple : il est bon d'être au courant de problèmes de ce genre ; cela évite que chacun perde du temps pour la même raison. Voici d'ailleurs une autre expérience de "plantage", pour les amateurs d'Applesoft.

Le programme de "Gestion de masques" de Gérard Michel dans le numéro 7 m'a fort intéressé. Mais je dois dire que j'ai eu des résultats variés lors de l'impression des masques :

- avec une Centronics, il y avait un saut de ligne excédentaire à chaque ligne ;
- avec une Silentype et une IDS, aucun problème ;
- avec une Epson équipée d'une carte graphique, j'avais une ligne sur trois ?

Guy Mathieu - 92 Vaucresson

Pour notre part, nous avons testé le programme avec une Silentype, une Centronics 739 et une Oki 92. Le tout sans problème...

1 - Il est peu pratique que les pages HGR de l'Apple soient situées en plein milieu de la RAM et obligent ainsi à situer un programme BASIC soit entièrement en-dessous, soit entièrement au-dessus des pages graphiques H.R. Y a-t-il un truc pour placer un morceau de programme avant les pages HGR et un autre après ?

2 - Ceux qui ont la couleur sur leur Apple ont sans doute remarqué que la définition est moins bonne qu'en N/B. Y a-t-il un moyen soft pour se passer de la couleur et récupérer la vraie définition ? J'ai une carte RVB Sonotec et un téléviseur Thomson. Parfois, il y a des amorçages (étincelles) au niveau de la THT du téléviseur et, par miracle, mon Apple écrit en vert sur fond noir avec la vraie haute résolution ou en noir sur fond vert. RESET annule cet effet et fait revenir la couleur. Je suppose que des tensions incontrôlées modifient l'état de certaines mémoires en remontant les fils de la RVB. Comment provoquer cela par un moyen moins aléatoire ?

3 - Et voilà un tuyau : à cause de la couleur, l'impression de certains graphes sur mon écran n'est pas satisfaisante. Des parties de tracé sont bleues, d'autres vertes. On obtient de très belles courbes bien homogènes en traçant en blanc (HCOLOR= 3) sur un écran constitué d'une alternance de lignes horizontales vertes (HCOLOR= 1) et bleues (HCOLOR= 2).

Joël Delannoy - Quartier des Grées
-04510 Aiglun - Mallemoisson

1 - Faire un "trou" dans un programme

Le programme SPLITTER, publié par Call APPLE et repris dans sa publication "All About Applesoft" permet de créer un "trou" à l'intérieur d'un programme Applesoft, trou qui occupe un nombre donné de pages mémoire, à définir par l'utilisateur.

Voici la récapitulation binaire de ce programme :

Programme SPLITTER

Récapitulation

```

9400- 38 A5 B0 85 41 65 06 85
9408- 43 A5 AF 85 40 85 42 A0
9410- 06 A9 00 91 42 88 D0 F9
9418- B1 40 C6 40 91 42 C6 42
9420- A5 42 C9 FF D0 F2 A5 41
9428- C6 43 C6 41 C5 07 10 E8
9430- A5 07 85 08 C6 08 38 A5
9438- B0 65 06 85 B0 85 6A 85
9440- 6C A5 67 85 40 A5 68 85
9448- 41 A0 01 B1 40 C5 08 F0
9450- 0B AA 88 B1 40 85 40 86
9458- 41 C8 D0 EF 88 B1 40 F0
9460- 02 D0 05 C8 B1 40 D0 E9
9468- C8 C8 C8 C8 B1 40 D0 FB
9470- C8 C8 C8 C8 C8 A9 AB 91
9478- 40 A5 40 85 42 A5 41 85
9480- 43 C8 84 09 A0 01 B1 40
9488- 38 65 06 91 40 88 AA B1
9490- 40 85 40 86 41 C8 A6 09
9498- D0 11 B1 40 D0 EA 4C 00
94A0- 00 10 27 E8 03 64 00 0A
94A8- 00 01 00 C8 B1 40 85 19
94B0- C8 B1 40 85 1A A9 A1 85
94B8- 1B A9 94 85 1C A0 01 A2
94C0- 01 A9 2F 85 1D E6 1D A5
94C8- 19 88 38 F1 1B 85 19 C8
94D0- A5 1A 30 02 A2 00 F1 1B

```

```

94D8- 85 1A 10 E9 8A D0 E6 A5
94E0- 19 88 18 71 1B 85 19 C8
94E8- A5 1A 71 1B 85 1A 84 1E
94F0- A5 1D A4 09 91 42 C8 A9
94F8- 00 91 42 E6 09 A4 1E C8
9500- C8 98 C9 0A 30 8B 86 09
9508- A0 01 4C 9A 94

```

Mode d'emploi :

- le sauvegarder sous le nom SPLITTER
- faire BLOAD SPLITTER
- POKE 6, nombre de pages du trou
- POKE 7, page de début du trou
- CALL 37888

Exemple : pour épargner la page HGR2, il faut faire POKE 6,32 : POKE 7,64 : CALL 37888.

Précautions à prendre : on ne peut BLOAD SPLITTER et faire le CALL depuis le programme Applesoft concerné, mais il faut le faire au clavier ou depuis un fichier EXEC. Il est fortement déconseillé de relancer SPLITTER sur un programme déjà "splitté", ainsi que d'essayer de créer un trou en dehors du programme ou dans un programme trop gros qui déborderait alors sur le DOS. Enfin, sauver puis recharger un programme "splitté" ne conserve pas le trou.

2 - Carte RVB Sonotec

Voici les POKES à faire avec la carte RVB Sonotec :

POKE -16144,0 : texte en blanc et graphiques en couleur. C'est le mode par défaut à l'allumage ou après un RESET.

POKE -16143,0 : texte en vert et graphiques en couleur

POKE -16142,0 : texte en blanc et graphiques en N/B

POKE -16141,0 : texte en vert et graphiques en noir et vert.

3 - Merci pour vos tuyaux sur les couleurs. Pour en savoir un peu plus, vous pouvez vous reporter à mon article sur HAIFA (Pom's numéro 5, pages 41 et 42) : la fonction SCRIN haute résolution.

Dans la "Programmation facilitée" (Pom's 5) de Denis Sureau que je remercie vivement pour son article original et agréable, l'instruction & INPUT A\$, ..., "LABEL" ne lit pas les espacements. Ainsi : \$ INPUT A\$, ..., "QUI EST LA " donne à l'exécution : "QUIESTLA ?".

B. Tambosco - 14 rue des Canadiens
- St Didier des Bois -
27370 Amfreville-la-Campagne

Vous avez découvert un bug qui s'avère encore plus dangereux que vous ne l'avez dit, car un oubli des guillemets finaux dans "LABEL" fait que l'ampers-interpréteur affichera

TOUS les octets du programme Applesoft jusqu'aux premiers guillemets rencontrés, d'où du "garbage" à l'écran.

Le problème vient du fait que le programme assembleur affiche la chaîne caractère par caractère jusqu'au prochain guillemet, chaque caractère étant obtenu par la routine CHRGET (\$B1) qui justement "saute" les espaces.

Je propose une modification utilisant les routines Applesoft STRTXT, STRPRT et CHRGET. STRTXT (\$DE81) fabrique un descripteur de chaîne pour la chaîne pointée actuellement par le pointeur de programme TXTPTR (\$B8,B9) et stocke ce descripteur en FACMO,LO (\$A0,A1). CHRGET (\$B7) lit le caractère du programme pointé par TXTPTR, sans toutefois incrémenter TXTPTR comme CHRGET le fait.

En conclusion, il faut remplacer les lignes 115 à 121 par :

```

115 STRING SEC
116 JSR $DE81
117 JSR $DB3D
118 RTSSTR JSR $B7

```

Si vous n'avez pas d'assembleur, vous pouvez utiliser le patch suivant, pour la version 48K de la programmation facilitée :

```

93A3 : 38 20 81 DE 20 3D DB
93AA : EA EA EA EA EA EA EA
93B2 : 20 B7 00

```

Pour la version 32K, utiliser les adresses 53A3, 53AA et 53B2.

Mon fils Loïc (16 ans et demi) aimerait rencontrer des utilisateurs d'Apple II pour parfaire ses connaissances et progresser ensemble. Y a-t-il un club dans le 20è (sinon le 12è, le 11è ou le 10è) ou des abonnés à Pom's qui seraient intéressés ?

M. Lepechon - 94 rue de Buzenval -
75020 Paris

Nous invitons les lecteurs de Pom's intéressés à répondre directement. Voici en outre une liste de clubs du voisinage :

Computerland Paris-Est - 135 Bd. Voltaire-75011 - Tel 379.21.01 -
Contacter M. Cung
Ecole Professionnelle Supérieure (EPS) - 45 rue des Petites Ecuries -
75010 Paris - Tel 523.35.30 -
Contacter Lionel Godard
Club 702 - 9 rue de Rambouillet-
75012 Paris - Tel 345.33.80 -
Contacter Gilles Probst

Lecteur récent de Pom's, je me permets de vous écrire pour vous dire BRAVO; bravo pour une revue remarquablement lisible pour le programmeur moyen que je suis, bravo pour les programmes originaux et intéressants, bravo pour la qualité

d'impression et le panachage des différents langages.

Une petite remarque cependant; je ne suis pas abonné et ne désire pas l'être. J'achète tous les numéros de Pom's au jour le jour et possède le recueil et tous les autres numéros. Je n'ai pourtant pas pu acheter les disquettes chez mon revendeur habituel. Pourquoi donc ?

J-M Mousseau - 36 rue de Gometz - 91440 Bures s/Yvette

Tout d'abord, merci pour vos encouragements. De nombreux autres lecteurs nous en envoient; nous profitons de l'occasion pour les remercier en même temps que vous. Nous avons voulu dès le début de Pom's vendre les disquettes d'accompagnement à très bas prix. Où trouverez-vous ailleurs des disquettes pleines de programmes et vendues moins de 45 FF TTC (prix sur abonnement) ?

Le seul moyen de vendre à ce genre de prix est de pratiquer la vente directe. A titre indicatif, la disquette d'accompagnement du livre "Visicalc sur Apple" était vendue par nous directement à 75 FF TTC port compris. Elle est à présent vendue dans les boutiques, distribuée par les Editions du PSI, à un peu moins de 200 FF. C'est d'ailleurs le prix du marché pour ce genre de produit. En effet, il faut incorporer le coût de l'emballage, la marge du diffuseur, celle de la boutique, ... Seriez-vous disposé à payer la disquette de Pom's 100 FF pour l'avoir dans les boutiques ?

En réponse à la lettre d'André Babin dans le Courrier des Lecteurs du Pom's 7, il est exact que le chargement rapide utilisé ne copie que des secteurs entiers de 256 octets. Si l'on charge un programme en \$300, cela écrase donc les vecteurs du

DOS en \$3D0 et du RESET en \$3F2. Il n'y a pas de parade autre que de réécrire le MENU. Le programme &BLOAD publié antérieurement, par contre, comporte un test de longueur destiné à contourner cet obstacle.

Quand au patch de la ROM LC, je pense qu'il serait intéressant de modifier le DOS lui-même et de ne pas modifier les buffers pour conserver le maximum de compatibilité avec les autres programmes. De toute façon, le MENU est destiné à ne fonctionner que lors du boot, le programme choisi peut ensuite compter sur une configuration "normale", libre à lui de modifier ce que bon lui semble (vecteurs de RESET, d'entrées/sorties, ...).

Thierry Le Tallec - 69 rue Sauveur Tobelem - 13007 Marseille

Vacances informatiques pour vos enfants

"Les amis du logiciel proposent des stages de micro-informatique pour les enfants de 12 à 14 ans. Les séances de travail alterneront avec les activités de plein air (tennis, ping-pong, natation). Le matériel utilisé est Apple II. Lieu : à 30 kms d'Angers.

LOGAMI - Christian de Menthon - 9 rue Las Cases - 75005 Paris - (1) 551.74.07

Ma Pomme

Le club Ma Pomme continue à organiser des soirées de travail autour d'un thème. Prochaines réunions de ce genre : Fichiers et bases de données sur Apple II, //e et /// (15 juin), Applications musicales et vocales (20 juillet) et Mémoires de masse pour tous Apples (21 septembre). Les séances ont lieu à partir de 19 heures au 110, avenue du Général Leclerc à Paris (XIV).

Courrier des clubs

Ma Pomme - 6 rue Paul Saunière - 75016 Paris - Tel le soir (tard) : (1) 558.05.78

Microtel Club Monaco

Notre Club se réunit un vendredi soir sur deux dans notre local provisoire, au Club des Sports et Loisirs, à l'adresse indiquée ci-dessous.

Nous sommes équipés d'un Goupil II avec TV couleur, mais beaucoup de nos membres possèdent des Apple II, qui constituent un centre d'intérêt privilégié de nos rencontres. Pour tous renseignements, contacter C. Perisi au (93) 50.68.56

Microtel Club Monaco - 24, avenue Prince Pierre - 98000 Monaco

Informatique - architecture

Les prochaines journées de démonstration organisées par notre club se-

ront consacrées aux applications disponibles pour les architectes. Elles auront lieu les 24 juin et 22 juillet. Pour tout renseignement, s'adresser à :

Club informatique-architecture - 110 avenue du Général Leclerc - 75014 Paris-Tel (1) 45.75.07

Informatique "Populaire"

Une publication mensuelle d'une vingtaine de pages est née en décembre dernier. Il s'agit d'Informatique "Populaire", bulletin de liaison des clubs populaires d'informatique, pour le moment implantés dans la région Nord et Pas-de-Calais.

Ecrire à Hélène Weiss - EDICO - 13 rue E. Rostand - 62200 Boulogne s/Mer - Tel (21) 31.42.32

OLIVETTI PRAXIS AVEC INTERFACE APPLE

5400 F h.t.

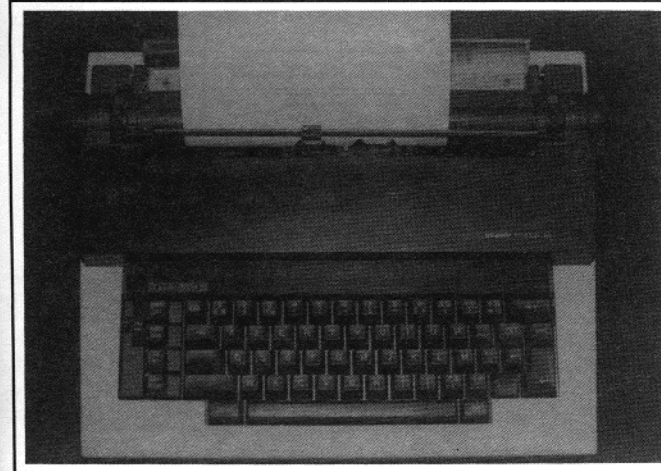
L'imprimante qui est aussi une machine à écrire.

Marguerites interchangeables.




TOTALE FORMATION

114, avenue Charles-de-Gaulle 92200 Neuilly - Tél. 745.62.73.



MICRO-PERIPH

ouvre les portes de votre 

Ouvert du mardi au samedi : 10 h - 12 h et 14 h - 18 h
60, rue Ducoüedic - 75014 PARIS (Tél. : 321.41.80)

NOUVEAU

RAMEX 128 K AVEC V-C EXPAND VERSION 80-2 (5 600 F T.T.C.)

Si vous souhaitez utiliser votre VISICALC avec :

- 136 K de mémoire • chargement d'un tableau complet en 20 secondes
- 80 colonnes sur l'écran (logiciel compatible avec 4 différentes cartes 80 colonnes)
- Largeur de colonnes variable individuellement
- Curseur programmable
- Beaucoup d'autres améliorations très nombreuses

(Ce logiciel qui est également compatible avec les cartes Saturn est disponible séparément (1 150 F T.T.C.). La carte RAMEX 128 K est disponible aussi séparément pour 4 600 F T.T.C., y compris le logiciel SOLIDOS.)

NOUVEAU

ACCELERATOR II (3 950 F T.T.C.)

Cette carte, fabriquée par Saturn Systems, est équipée d'un microprocesseur 6502 C qui tourne à 3,58 MHz et de 64 K de RAM. Cette augmentation de vitesse de 358 % se traduit par une augmentation de rapidité d'exploitation de tous les programmes. La carte est directement compatible avec tous les langages et programmes (Applesoft, Integer, Binaire, Pascal, Fortran 77, Forth ainsi que tous les progiciels tels que Visicalc, DB Master, etc.). A la commande, spécifier version Apple II ou Apple IIe.

NOUVEAU

LECTEUR DE CODE BARRE (2 340 F T.T.C.)

GÉNÉRATEUR DE CODE BARRE (2 340 F T.T.C.)
CARTE RVB (1 500 F T.T.C.) avec système d'amélioration d'anomalies couleur.
CARTE 80 COLONNES (1 900 F T.T.C.) : cette carte fonctionne avec notre carte RVB.

NOUVEAU

PROGRAMMEUR D'EPROMS (1 360 F T.T.C.). La carte se branche dans un slot de l'Apple II et permet de programmer directement les EPROMS (2758, 2716, 2732 et 2764). Un logiciel de gestion du programmeur se trouve sur la carte de mémoire morte. Une carte existe également pour adapter les EPROMS programmés sur l'Apple II (630 F T.T.C.).

LES CARTES « PILOTS »

U-TERM (1 490 F T.T.C.)

Cette carte qui vous donne 80 colonnes sur l'écran vous permet d'avoir deux jeux de caractères à la fois qui sont co-résidents dans la mémoire vive et la mémoire morte de la carte. Par exemple, les caractères français majuscules et minuscules ainsi que les caractères anglo-saxons. Elle a une très grande gamme de compatibilité : APPLESOFT, PASCAL, CPM, APPLEWRITER II, etc. (Photo 2).

U-Z80 (1 150 F T.T.C.) : son point fort est la vitesse : 4 MHz. Compatible avec tous les logiciels CPM ou de Z80 (Photo 1).

U-RAM16 (890 F T.T.C.) : c'est une carte langage entièrement compatible avec INTEGER BASIC, PASCAL, VISICALC, etc. Pourquoi payer plus cher ?
U-S232 (990 F T.T.C.) : une carte de série, vitesse variable entre 75 et 19200 bauds. Cette carte contient son propre logiciel de *handshake*, ce qui permet de faire fonctionner une imprimante à sa vitesse optimale. Compatible avec le logiciel de communication VISITERM (Photo 9).

EXTENSIONS DE MÉMOIRE

U-RAM 32K (1 450 F T.T.C.), U-RAM 64K (2 590 F T.T.C.), U-RAM 128K (4 250 F T.T.C.) (Photos 7 et 4). VERSA RAM émulateur de disque (690 F T.T.C.).

CARTES SCIENTIFIQUES

U-A/D (4 530 F T.T.C.) : carte analogue digitale haute vitesse, 12 bits, 16 gammes (Photo 6).
Carte A/D 8 bits (1 400 F T.T.C.), 1 gamme (470 F T.T.C.)
Carte D/A 8 bits (1 889 F T.T.C.), 1 gamme (450 F T.T.C.)
U-BCD (1 050 F T.T.C.) (Photo 8)
U-DT (1 350 F T.T.C.), 32 lignes entrée-sortie avec temporisateur et *buffer*
U-TIM (1 090 F T.T.C.), temporisateur
U-2 PORT (1 800 F T.T.C.) : 2 portes RS 232 sur la même carte extensible jusqu'à 8 portes (Photo 5)
THE MILL 6809 (NC) : co-processeurs à haute vitesse et de multiprogrammation
Carte 68000 (NC) 16 bits + 32 bits interne (Photo 3).
Nous avons plusieurs autres cartes qui ne sont pas mentionnées ici ; renseignez-vous

DIVERS POUR L'APPLE II

JOYSTICK TRACKBALL (660 F T.T.C.) : une grande boule pour diriger vos navires spatiaux.

CARTE HORLOGE (770 F T.T.C.) : au quartz avec pile et logiciel agenda.
ROBOCOM BITSTICK (3 397 F T.T.C.) : un système très sophistiqué de dessin graphique.

D BASE II (5 200 F T.T.C.) : système de base de données relationnel (Z 80).
JOYSTICK KRAFT (510 F T.T.C.) et JOYSTICK TG PRODUCTS (570 F T.T.C.) : fabriqués aux États-Unis, ils sont d'excellentes qualité et fiabilité.

LECTEURS DE DISQUES : toutes les configurations possibles entre 143 K octets et 2 M octets, entièrement compatibles avec l'Apple II.

CCS carte synchrone (2 120 F T.T.C.) : pour dialoguer avec un IBM.
DB MASTER (2 050 F T.T.C.) : très puissant. Base de données.

DATA FACTORY (3 172 F T.T.C.) : peut-être la plus puissante des bases de données.

ASCII EXPRESS PROFESSIONAL (1 376 F T.T.C.) : permet d'envoyer n'importe quel fichier en DOS par modem.

MUMATH/MUSIMP Z80 (2 542 F T.T.C.) : très puissant logiciel mathématique.

VENTILATEUR (750 F T.T.C.) : se branche à gauche de l'Apple II. Alimente l'Apple II.

ENSEMBLES D'EXPANSION VISICALC

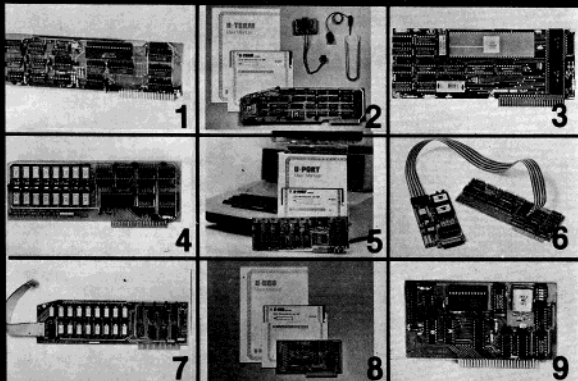
V-C 50K (3 560 F T.T.C.) : 1 carte 80 colonnes, V-C expand logiciel, U-RAM 32K
V-C 82K (4580 F T.T.C.) : 1 carte 80 colonnes, V-C expand logiciel, U-RAM 64K
V-C 145K (6 360 F T.T.C.) : 1 carte 80 colonnes, V-C expand logiciel, U-RAM 128K
V-C consolidator (400 F T.T.C.) : pour consolider plusieurs tableaux de VISICALC
V-C super expand (700 F T.T.C.) : version 40 colonnes avec sauvegarde sur disquette en 20 secondes (le RAMEX 128K est nécessaire)
V-C super expand 80 (900 F T.T.C.) : version 80 colonnes compatible avec les cartes omnivision 80, Videx et U-Term (le RAMEX 128K est nécessaire)

NOUVEAU

EPSON FX 80 (5 800 F T.T.C.) - PROMOTION

Vitesse : 160 c.p.s., 80, 40, 137, 68, 96 ou 48 caractères par ligne, friction/traction, 9 formats graphiques, 9 jeux de caractères, RAM avec logiciel ouvert pour générer votre propre jeu de caractères, matrice de 11 fois 9, espacement proportionnel, mode silencieux.

Si vous ne voyez pas ce que vous cherchez ou si vous désirez de plus amples renseignements, téléphonez à MICRO-PÉRIPH (321.41.80). Si vous ne pouvez vous déplacer, téléphonez-nous pour connaître le revendeur le plus proche de chez vous ou complétez le bon de commande ci-dessous.
TOUTES NOS CARTES SONT GARANTIES 12 MOIS



BON DE COMMANDE

A découper et envoyer à :
MICRO-PÉRIPH, 60, rue Ducoüedic - 75014 Paris - Tél. : 321.41.80
Je désire recevoir rapidement, sous pli recommandé :

QUANTITÉ DÉSIGNATION TOTAL T.T.C.

Total _____
Frais de port recommandé _____ 15,00 F
Ci-joint un chèque/C.C.P. de F : _____

JCR, DES MICRO-ORDINATEURS PROFESSIONNEL ET GRAND PUBLIC.



APPLE II E

Entièrement compatible avec l'Apple II il possède un clavier étendu Azerty/ Qwerty et une mémoire de 64 K oct. d'origine.

Déjà de nombreux logiciels et extensions disponibles sur ce nouvel ordinateur.

APPLE III

L'outil professionnel par excellence. 128 Ko ou 256 Ko. Unité de disque incorporée. Sortie RS 232. Nombreux interfaces disponibles. Adjonction possible d'un disque dur de 5 méga. Profilé. Écran vert haute résolution antireflets. Clavier Azerty - Qwerty.

MONITEURS VIDÉO

Prince 12" N/B.	1 050 F
Prince 12" Vert.	1 400 F
Prince 12" Ambre.	1 600 F
CIAEGI 12" Vert.	1 720 F
Philips 12" Ambre.	1 720 F
Cabel 12" Couleur RVB.	3 500 F

Cartes pour Apple II :

U-RAM 16 K.	890 F
U-TIM carte horloge.	1 090 F
U-TERM carte 80 col.	1 490 F
U-Z80 carte CP/M.	1 150 F
CP/M pour U-Z80.	750 F
Microbuffer EPSON 16 K.	2 450 F
Microbuffer ext. 64 K.	3 300 F
Clavier numérique.	950 F
Carte chat Mauve.	1 400 F
Carte EVE pour Apple II E.	2 900 F
Carte 80 col. TEXT A II E.	990 F
Carte 80 col. + 64 K A II E.	2 300 F
Poignées de contrôle.	185 F
Joystick.	380 F
Carte Superterm 80 col.	2 590 F
Ventilateur Super Fan.	850 F
Carte mémoire 64 K.	2 590 F
Carte mémoire 128 K.	3 950 F

Produits consommables :

Disq. SF/DD 3M les 10.	210 F
Disq. DD/DD 3M les 10.	452 F
K 7 vierges les 10.	70 F
Papier listing 80 col.	290 F
Papier listing 132 col.	350 F

VISICORP

Une série de logiciels d'aide à la décision : Visicalc - Visiplot - Visidex - Visi-file.

PROMOTION

ORIC I

Un ordinateur pour tous de 16 à 48 K oct. RAM.

16 couleurs en graphisme haute résolution.

Synthétiseur de son et interface parallèle incorporés d'origine.

Version 48 K.

2 080 F



— Catalogue gratuit sur demande — Crédit 4-36 mois — Leasing 36-48 mois — détaxe à l'exportation.



BOUTIQUE

58, rue Notre-Dame-de-Lorette
75009 PARIS
Tél. (1) 282.19.80 - Télex : 290350 F

59, rue du Docteur Escat
13006 MARSEILLE
Tél. (91) 37.62.33

Fermeture annuelle du 14 au 23 août 1983

▶▶▶ VICTOR CASIO SINCLAIR SIRIUS ESPON

COMMODORE THOMSON OSBONE SHARP VISICORP SEIKO APPLE

En raison des fluctuations monétaires, ces prix sont susceptibles d'être modifiés sans préavis. Nous consulter pour confirmation.