

tremplin micro

**Copie d'écran texte
sur imprimante**
(Apple IIe, IIc, II GS)

**Le nouveau
ProDOS**

**Routines
Applesoft**

**Initiez-vous
au C**

**Réveillez
votre GS !**

M 1631 - 22 - 33,00 F



3791631033006 00220

N° 22 - Bimestriel - Quatrième année
3 Novembre - 24 Janvier 1988
241 FB - 11 FS - **33 F**

Apple, Apple II GS et ProDOS sont des marques déposées par Apple Computer Inc.
MS-DOS est une marque déposée par Microsoft.

tremplin micro 22

SOMMAIRE

Avec la collaboration de :

Claude AUBRY, Pierre CHAUVIÈRE, Stéphan HADINGER, Madeleine HODÉ, Yvan KOENIG, NESTOR, Clément RENARD, Emile SCHWARZ.

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

BIMESTRIEL

Le numéro : 33 F
Abonnement d'un an : 190 F
(6 numéros)

Tous nos prix sont indiqués TTC.

EDITIONS JIBENA

Direction-Rédaction :

Editions JIBENA

Guy-HACHETTE

La Petite Motte — Senillé —
86100 CHÂTELLERAULT.

Téléphone :
49.93.66.66

PUBLICITÉ :
Raymond JULLIEN
(1) 45.75.41.81

Commission Paritaire :

Les revues qui choisissent d'être réellement au service du Lecteur, en ne l'obligeant pas à glaner, dans plusieurs magazines, les renseignements concernant sa machine, ne bénéficient pas du numéro de Commission Paritaire, et pas d'avantage des tarifs postaux réduits.

TREMPLIN MICRO — Bimestriel — C'est une publication des Editions JIBENA, 4, rue de la Cour-des-Noues, 75020 PARIS — S.A. au capital de 3600000 F — Imprimé par CITÉ-PRESS/PARIS — Service de vente : Presse-Promotion, tél. : 49.93.65.03. — Dépôt légal à la date de parution — Inscription à la Commission Paritaire des Publications et Agences de Presse : en cours — Directeur de la Publication : Guy-Clément COGNÉ — Diffusion N.M.P.P.

La disquette TREMPLIN MICRO contient tous les programmes du numéro, ainsi que les sources trop longs pour être publiés dans les colonnes de la revue.

PERPLEXE... GUY-HACHETTE

l'est... et c'est ce qui explique le retard de parution de T.M. 2

PRODOS continue. La preuve, une version 3.2 existe aux Etats-Unis, Emile Schwarz vous en parle... en long, en large et en travers 3

FONCTIONS PLUS POUR ACCENT

L'accessoire de bureau de Stephan Hadinger n'en finit pas de passionner Yvan Koenig, lequel se permet de lui offrir quelques fonctions supplémentaires 7

FILTRINPUT & SONO

deux routines en Applesoft, fonctionnant aussi bien sous ProDOS qu'avec le vieux DOS 3.3 11

Le nouveau GRIBOUILLE arrive sur l'Apple //GS 19

La petite fille avec une boîte de chocolat...
ou le C par l'exemple, sur Apple //GS... ou autre chose 23

KOENIG a visité Apple Expo 88
et vous en parle 29

PC ORGANISER, le tout en un de votre PC 32

INSTADUMP RECOPIE D'ÉCRAN-TEXTE SUR APPLE //e
et //c facile, grâce à une nouvelle
commande ProDOS 33

Turbo Pascal 4.0 dynamisé (sur PC) 36

ALPHA Lexis vous facilitera la vie 37

Traducteur d'anglais (Collins-On-Line) 39

Langage C : TEXT, TOOL

Les fonctions de l'outil texte
comprendre votre GS avec Emile Schwarz 41

TextScreenDump, mais sur GS 51

Et bien sûr, dans TREMPLIN MICRO, la revue des livres informatique...
un peu partout dans le journal.

PERPLEXE... GUY-HACHETTE

Je suis perplexe. Terriblement. Et je ne suis pas le seul. Je n'aime pas être floué et c'est cependant ce qui m'arrive avec mes deux beaux GS, acquis à prix d'or dès leur mise en vente sur le marché français.

Du GS, nous avons tout espéré. La machine était attendue par une clientèle impatiente. Elle arriva... puis, en quelques mois, parvint à passer totalement inaperçue.

Combien, parmi nos lecteurs, candidats à l'achat de la bête, furent aimablement aiguillés vers le MAC ? Nous ne les comptons plus.

Je me demande encore pourquoi Apple sortit une machine que nul, aux ULIS, ne semblait vraiment désireux de vendre.

L'Apple IIGS servit simplement de repoussoir... en faveur du seul, du beau, du grand MAC.

Très bien. C'est une politique commerciale qui vaut ce qu'elle vaut. J'estime personnellement qu'elle est mauvaise. On ne joue pas ainsi avec le côté sentimental d'une clientèle de fans. Qui aime bien châtie bien... je crains que cette arête ne reste en travers de la gorge de nombreux possesseurs de GS.

*Et que l'on ne vienne pas comparer mon micro personnel à quelque automobile. Ça n'a rien à voir. **Questions** : combien a coûté et rapporté chacun des Apple IIGS vendus en France ? A qui a profité cette étrange opération ?*

Sûrement pas à ceux qui, plusieurs années après leur achat, attendent encore une ROM définitive et un système d'exploitation capable de rivaliser avec le MS.DOS par exemple.

Pas davantage à Apple France. Alors, à qui ? Et pourquoi ?

Paradoxalement, la machine se porte plutôt bien aux Etats-Unis, mais il est vrai que l'on n'est pas obnubilé, outre-Atlantique, par le désir de vendre absolument du Mac.

*Dans ce contexte, que devient **TREMLIN MICRO**, revue de programmation SUR APPLE II ?*

Continuerons-nous, contre vents et marées, d'œuvrer dans l'intérêt des Applemaniaques attachés à leur machine et à leurs habitudes ?

Nous tournerons-nous davantage vers d'autres processeurs, et notamment vers ceux qui équipent la gamme des PC / et compatibles ?

C'est une question que je me pose et que je pose à tous nos lecteurs.

Merci d'y répondre.

GUY-HACHETTE. (Prochain numéro le 25 janvier 1989).

PRODOS



La disquette système version 3.2 US

ProDOS 8, ProDOS 16 et le Loader ont pour numéro de version 1.6. Il est très important de considérer ces trois fichiers (la disquette entière) comme un ensemble. En effet, le système refusera de lancer ProDOS 8 avec un numéro de version inférieur à 1.6.

Launcher version 2.2. Dans *version reporter*, les outils 29 à 32 ont été ajoutés dans la liste. Pour accéder à *version reporter*, il suffit de presser la touche pomme tout en cliquant sur la ligne *Version 2.2* inscrite en bas, à gauche de la fenêtre du Launcher. Dans **A+** daté du mois d'Août 1988, un lecteur US signale l'existence et le mode d'emploi de cet accessoire, alors que *version reporter* figure dans le Launcher depuis le système 2.0.

Le **FINDER** porte le numéro de version 1.1 et demande 375 K de mémoire pour être exécuté.

Le fichier **BASIC.SYSTEM** a été légèrement modifié et porte maintenant le numéro 1.2 ainsi que **COPYRIGHT APPLE, 1983-87**. Quelques octets de code ont été changés, mais ce ne sont que des corrections de détails.

Voyons maintenant les différents sous-catalogues.

SYSTEM.SETUP :

Dans ce directory, nous avons deux types de fichiers :

1. Fichiers patchs des outils :

Tool Setup version 2.2. Ce fichier fait une analyse du numéro de ROM. Trois cas sont à considérer :

- a) Il s'agit de la ROM originale (ROM 00), le fichier patch TS1 sera chargé.
- b) Il s'agit de la ROM de mise à niveau 1987 (ROM 01), le fichier TS2 sera chargé.
- c) Ce n'est ni la ROM 00, ni la ROM 01 : on ne charge aucun fichier.

D'autre part, un patch a été implanté à la fonction **SysBeep**. Il fait clignoter la partie pourtour de l'écran lors d'un passage sur la position zéro du volume au tableau de bord.

Il est important de faire installer la nouvelle ROM ainsi que le nouveau VGC dans votre machine, d'autant que cette opération est effectuée gratuitement par votre concessionnaire.

Note : Vous pouvez effacer le fichier TS1 ou TS2 suivant la ROM de votre Apple IIGS. Si vous possédez la ROM révision 01, vous aurez un gain de 59 blocks sur votre disquette système de travail.

2. Les fichiers Appletalk :

- a) ATINIT. Contient des informations sur la configuration de l'utilisateur.
- b) ATPATCH. Patche les bugs en ROM et implémente les fonctions suivantes : RPM, PAP, EP et ZIP.
- c) SPLOAD. Implémente la fonction ASP qui permet l'ouverture de 8 fichiers simultanément lors d'une session de travail.
- d) PFILOAD. Implémente les fonctions ProDOS Filing Interface.
- e) ATSTART. Démarre *Appletalk*.
- f) ATSETUP. Détermine si *Appletalk* est actif et charge les fichiers ad hoc.
- g) ATROM. Contient le code de la ROM *Appletalk*. Il apparaît que patcher la ROM *Appletalk* est plus gourmand en codes et en temps que de recharger complètement le contenu débuggé de la prochaine ROM *Appletalk*.
- h) ATRESPONDER. Enregistre les noms des utilisateurs du réseau *Appletalk*.

Note : Si vous n'utilisez pas le réseau *Appletalk*, vous pouvez enlever ces huit fichiers de votre disquette système de travail.

TOOLS :

A. L'outil *Note Sequencer* - TOOL026 - a

été ajouté. Il n'était plus dans la disquette système depuis un an et demi (Décembre 1986).

B. Le *ACE (Audio Compression and Expansion toolkit)* - TOOL029 - figure pour la première fois sur une disquette système.

Cet outil est un compresseur/expandeur de fichier son. Il permet de compresser un fichier avant de le sauver et ainsi de gagner de la place sur disquette. Lors du chargement, un appel à cet outil opérera l'expansion des données son. Un gain de temps important lors des opérations disque est également permis du fait de la compression du fichier.

C. Les outils 30 et 31 - TOOL030 et TOOL031 - sont réservés à une utilisation interne.

D. Le *Midi Tool Set* - TOOL032 - permet d'interfacer l'Apple IIGS et les instruments de musique au format MIDI.

E. Les outils suivants sont, à présent, dans une version qui permet leur implantation en ROM : Control Manager, Dialog Manager et Font Manager.

F. Dans le numéro de version d'un outil, il y a quelques octets signifiant que cet outil est dans une version prototype. Tous les outils présents dans la version 3.2 du système ont ces octets à zéro.

Il y a plusieurs nouvelles fonctions dans les outils suivants : *Tool Locator*, *Memory Manager*, *Window Manager*, *Font Manager*, *Print Manager* et *Quick Draw Auxiliary*.

D'autre part, lors du premier accès à un menu déroulant, l'image de ce menu est sauvegardée pour être transférée - et apparaît plus rapidement - lors d'un appel ultérieur sauf en cas de modification dans ce menu. Il en résulte un affichage accéléré à l'écran.

DRIVERS :

A. *IMAGEWRITER LQ*. Non, ne bondissez

pas encore chez votre concessionnaire pour acheter la LQ ! Ce driver permet de trouver la LQ dans un réseau Appletalk ainsi qu'une émulation ImageWriter II. Ce n'est pas un driver qui permet d'utiliser la LQ dans toute sa splendeur...

B. IMAGEWRITER. Oui, bondissez de joie. Jugez vous-même : un texte GSWRITE contenant une image était imprimé en 12,50 minutes avec le système 3.1 ; maintenant, il est imprimé en 3,50 minutes... Ça laisse rêveur...

C. LASERWRITER. Cette version est compatible avec le driver Macintosh.

D. PRINTER, MODEM et APPLE TALK : quelques bugs on été corrigés.

E. APPLE.MIDI. Driver pour le *Midi Tool Set*.

F. CARD6850.MIDI. Driver pour le *Midi Tool Set*.

Note : Si vous n'utilisez pas le réseau Appletalk, l'ImageWriter LQ, l'outil Midi ou une LaserWriter, vous pouvez enlever ces fichiers de votre disquette système de travail.

FONTS :

FONT.LISTS. Un nouveau fichier dans ce directory. Créé lors de la première initialisation du *Font Manager*, il contient les informations sur le nom des fontes dans ce directory, leur taille etc... Lors d'un appel à *FMStartUp*, le *Font Manager* lit le contenu de ce fichier ; s'il n'existe pas, il cherchera les informations dans les différentes fontes, puis les sauvera sous le nom de **FONT.LISTS**. Un gain de temps appréciable est ainsi obtenu.

SYS.UTILS :

Il s'agit - pour la plupart d'entre nous - d'un nouveau directory. En effet, lors de la localisation du système 3.1, ces fichiers n'ont

pas été francisés. Vous avez pu voir à la page 13 du "Manuel abrégé du Finder" qu'un directory de ce nom existait, mais de directory... point !

A. SYSUTIL.SYSTEM. C'est un utilitaire en mode texte et en 8 bits. Il permet notamment de lister le contenu de disques formatés sous *DOS 3.3*, *DOS 3.2*, *CP/M*, en *PASCAL* aussi bien que sous *ProDOS*. Les copies de fichiers texte depuis *DOS 3.3* sont pour la première fois effectuées correctement (ce que fait *COPY II+*, par exemple, depuis fort longtemps). Et, chose importante pour les souris-phobes, nul n'est besoin de la souris pour obtenir un catalogue ou pour renommer un fichier, etc.

B. FASTCOPY.SYSTEM. Cet utilitaire permet la copie volume de disques au format *PASCAL*, *ProDOS*, *DOS 3.3* et *Macintosh*. Il possède une option de vérification ainsi qu'une option de comparaison de disque.

Au chapitre des pertes, il faut classer les fichiers suivants :

- *INIT.UTIL*, formateur de disque dur.
- *ICON.EDITOR*, crée ou modifie les fichiers icônes.
- *SOUND.INIT*, est maintenant inclus dans le fichier *TS2*.
- *PANEL.SETUP*, est un fichier en français pour les Français. Il nous permet d'obtenir presque tous les messages du tableau de bord du GS en français, mais prend 11 blocks sur la disquette et il est incomplet. En effet, un message existant sur la ROM 01 a été oublié. Apple France en a été informé.

La documentation complète est disponible chez :

APDA
(Apple Programmer's and Developer's Association)
290 S.W. 43rd Street
Renton, WA 98055
United States of America

APDA est une association qui demande 35\$ de droit annuel d'entrée pour nous, Européens. Pour ce prix, nous avons droit à un catalogue (APDAlog) qui paraît tous les trois mois. Ce dernier propose tous les produits développeur pour les deux gammes d'Apple, que ce soit des logiciels ou des documentations (les "Toolbox reference" en version définitive de chez Addison-Wesley par exemple)...

L'Apple IIGS System Disk v. 3.2 est livré avec une disquette et 225 pages de documentation réparties comme suit :
 Release Notes (33 pages), A.C.E. Tools ERS (10 pages), MIDI Tools ERS (50 pages), Note Sequencer ERS (41 pages), Note Syn-

thesizer ERS (26 pages), Sound Tools ERS (36 pages) et enfin, Defining your own Windows (29 pages). L'ensemble est vendu 13,50\$.

Pour les personnes intéressées, je signale qu'elles peuvent obtenir, pour la somme de 72 F ttc, le lot de deux disquettes système 3.1 et 3.2. Il leur faudra 100 F ttc de plus pour avoir 33 pages des "release notes" du système 3.2.

La documentation partitionnée est disponible chez :
 Prim'Vert Editions, 36 rue des Etats Généraux
 78000 VERSAILLES — Tél. : (16-1) 39.02.33.44

Dernière minute : Apple U.S.A. a présenté le système 4.0 qui répond désormais au nom de GS/OS. Dès que la version française sera disponible, nous vous la décrirons.

Rappel : Il est très important que les possesseurs de GS fassent la mise à jour gratuite de leur ROM/VGC car GS/OS a besoin de la ROM 01 pour fonctionner.

DES NOUVELLES DU FRONT

Nous allons commencer par les nouvelles :

1. Une nouvelle version du *GS Basic* est actuellement étudiée par son auteur **John O. Arkley**.
2. La version 3.2 du système d'exploitation est diffusée depuis début juillet aux U.S.A. (voir le banc d'essai).
3. Une anomalie de fonctionnement - tout d'abord attribuée au *GS Basic* - vient d'être corrigée. Il s'agissait de l'impossibilité d'envoyer un catalogue de plus de 12 fichiers (ou plus d'un block) dans un fichier texte. Dans la version 3.2 du système d'exploitation, ce bug a été corrigé.
4. **Apple Numerics Manual, Second Edition (SANE)** est paru chez Addison-Wesley. Cette nouvelle édition (en anglais) est mieux présentée. Elle comporte quelques modifications par rapport à la première, entre autres :
 - les chapitres 11 à 18 portent sur *SANE 65C816* pour le *GS*,
 - les chapitres 27 à 31 portent sur *SANE MC68881* pour le *Macintosh*.
5. L'Apple *IIC+* est sorti aux Etats-Unis. Il a un lecteur 3 pouces 1/2 en standard et est équipé d'un 65C02 version 4 Mhz.

Et maintenant les rectifications :

1. Dans le livre "A la découverte du *GS Basic*" :
 Page 10, exemple ligne 100 il faut lire :
 100 DEBUT: GOSUB 30
 Page 254, le code ASCII 137 représente le caractère graphique a

Page 254, le code ASCII 140 représente le caractère graphique a.

L'anomalie numéro 11 page 259 n'en est pas vraiment une et s'explique parfaitement. En effet :

1% = 43 : J% = 1% / 2 : PRINT J%:REM Résultat = 22

1% = 43 : J% = 1% DIV 2 : PRINT J%:REM Résultat = 21

Le résultat (22) est normal car une conversion est effectuée avant de faire la division parce que le résultat de l'opération ne peut pas être un entier. On peut dire, en fait, que le programme effectue les opérations suivantes : 1% = 43 : R = 1% : J% = R / 2.0
 Une fois ces opérations terminées, une conversion du résultat en entier est réalisée, avec un arrondi qui produit le passage à un résultat de 22 au lieu de 21. C.Q.F.D !

La meilleure démonstration en est la ligne suivante :

1% = 43 : J% = 1% / 2 : PRINT 1%/2, INT (1%/2), J%, 1% DIV 2

2. Il vous faut acheter le langage *GS Basic* pour pouvoir l'utiliser. Je dois vous le préciser car quelques lecteurs ont cru que le langage faisait partie de la disquette d'accompagnement du livre.
3. **John O. Arkley** - auteur du *GS Basic* - me précise que la valeur attribuée actuellement aux *TOKENS* n'est pas figée et peut donc varier dans une version future. Ceci n'a aucune incidence sur les programmes actuels ; il faudra simplement sauvegarder sous format TXT le programme avec la version 1.0B4, puis, après avoir booté sur la nouvelle version, EXECuter la sauvegarde.

Fonctions supplémentaires pour ACCENT

de
**Stephan
HADINGER**

ACCENT de Stephan HADINGER a fêté son anniversaire. A cette occasion, j'ai pris la liberté de lui offrir quelques fonctions supplémentaires.

Dans la version publiée, nous ne disposions que d'un mode de fonctionnement autorisant la saisie de TOUS les caractères complémentaires codés de \$80 à \$ff. C'est puissant, mais il n'est pas facile de mémoriser les combinaisons de touches donnant accès à un caractère donné. Dans un premier temps, j'ai réglé le problème en saisissant, table en main, la série des 127 caractères. Je chargeais systématiquement ce texte lors d'une séance de travail MultiScribe. A l'aide du couper-coller, je pouvais insérer tout caractère sans me soucier de son mode d'accès clavier et, ACCENT ne me servait plus. Avantage de cette formule : en sélectionnant tout le document, on peut aisément contrôler si le caractère désiré existe dans la police que l'on a prévu d'employer.

Revenant à ACCENT, je lui ai ajouté trois modes de fonctionnement et des fonctions annexes :

- Taper Pomme Option flèche droite active le mode originel.
- Pomme Option flèche en haut active le mode "circonflexe", tandis que Pomme Option flèche en bas active le mode "tréma". Quand l'un de ces modes est sélectionné, la séquence Pomme Option Relâche (AEIOUaeiou) envoie la lettre surmontée d'un circonflexe ou d'un tréma selon le mode.
- Pomme Option flèche gauche active le mode "réunion". Dans ce cas, la séquence Pomme Option Relâche (AOao) envoie le caractère choisi - union - (Ee). En prime, dans ce mode, Pomme Option Relâche (àçé) donne le caractère sélectionné, mais en majuscule.

- Enfin, la séquence Pomme Option (1234567) avec un chiffre du pavé numérique sélectionne un langage conformément au codage APPLE :

1 = anglais 2 = français 3 = danois
4 = espagnol 5 = italien 6 = allemand
7 = suédois

Lorsqu'un langage est choisi, taper Pomme Option (èà*çš`èùè`); cela donne accès au caractère correspondant de cette langue (cf. table page 8). Pour obtenir notre bon vieux é, il suffit donc de taper Pomme Option é.

Vous trouverez sur la disquette T.M. le fichier MORE.ACCENT (texte MultiScribe) qui résume les possibilités d'ACCENT modifié. Si vous essayez de l'imprimer, ne soyez pas surpris de ne pas obtenir les mêmes résultats que moi. Les fontes livrées par APPLE et divers autres fournisseurs sont défectueuses. Elles ne respectent pas les règles de définition établies par APPLE. Avec HELVETICA 12 (24 sur papier), il y a génération d'espaces parasites. Avec TIMES 12 (24 sur papier), au contraire, on a des reculs intempestifs. Cela affecte un caractère ' et le caractère TM.

Heureusement, un éditeur de fontes, dont je teste la version préliminaire, permet, entre autres, de corriger ces anomalies.

Il apparaît que nombre de problèmes qui furent hâtivement baptisés "bugs de GS.WRITE" étaient, en fait, dus aux anomalies des fontes livrées par APPLE (et d'autres, hélas). Je tenais à rétablir la vérité sur cette attribution erronée de responsabilités. L'équipe VersionSoft a abandonné le GS : souhaitons que ce n'en soit pas la cause.

(suite page 9).

Précisons que ce n'est pas parce que nous allons bientôt disposer d'un ou plusieurs éditeur(s) de fontes qu'il faut se laisser aller à jeter les standards aux orties. Redessiner le caractère \$7b pour qu'il s'affiche sous forme de é tente certains ; hélas, s'amuser à cela aurait plus d'inconvénients que d'avantages. D'abord, le caractère accolade serait perdu ; ensuite, la portabilité, qui devrait nous être chère ; disparaîtrait.

Il est vrai que, d'ores et déjà, les fontes disponibles prennent des libertés avec les "normes" :

- Un caractère XXX24 n'est pas toujours deux fois plus large qu'un caractère normal XXX12.
- L'espace collant \$CA devrait avoir le même encombrement qu'un espace normal.
- En dehors des fontes de symboles, les caractères \$00 à \$D8 devraient, lorsqu'ils sont définis, respecter la liste APPLE...

Il existe un accessoire de bureau fort bien conçu et baptisé KEYUTIL qui permet, en gros, de prendre en compte en mode SHGR la configuration "langage affiché" sélectionnée au tableau de bord. Lorsque cet accessoire est actif, taper é affiche un é, tandis que Option é nous donne la bonne vieille accolade. En prime, on peut, à tout instant, effectuer une copie de l'écran SHGR sur papier (ou dans un

fichier \$C1). Cet accessoire affiche la mention "Copyright APPLE 1988". Le marché allemand étant moins développé que le marché français en ce qui concerne la gamme Apple II, on peut se demander pourquoi KEYUTIL est disponible en Allemagne (tous messages traduits) et ne l'est pas en France. Serions-nous, une fois encore, des sous-hommes ?

Dans la fiche jointe, j'ai employé MONACO 12 et COURIER 12 pour souligner la différence de qualité entre les impressions obtenues lorsque la fonte choisie existe dans la taille 2xCHOIX et lorsqu'elle n'existe pas.

Quand on travaille en mode "best", c'est, en effet, la fonte 2xCHOIX qui est employée, lorsqu'elle existe, donnant un résultat bien supérieur à ce que l'on obtient dans le cas contraire (mais attention aux jeux de fontes mal conçus !).

Avant de vous quitter, j'aimerais rappeler à APPLE France qu'il serait urgent, sur GS comme sur MAC, de définir le code qu'il convient d'attribuer au caractère FRANC (F barré 2 fois). La liste actuelle étant saturée, il faudrait sacrifier un caractère. Ça pourrait, me semble-t-il, être le second signe MOINS codé \$D0 ou la seconde VIRGULE en \$E2. Une décision rapide accompagnée d'une mise à jour des fontes LaserWriter seraient les bienvenues.

Tous les fichiers concernant cet article figurent sur la disquette Tremplin Micro. **Y. K.**



Le livre des imprimantes

Le fonctionnement incorrect de l'imprimante, la disparition des accents à l'impression, la recherche de certains caractères spéciaux sont des problèmes de plus en plus courants.

Une multitude de paramètres doivent être pris en compte pour résoudre ce type de problèmes : le choix du bon driver, de la position des interrupteurs, du câble approprié, du type de liaison, etc.

LE LIVRE DES IMPRIMANTES analyse, dépouille, réorganise la liaison entre l'imprimante et le micro-ordinateur. Il aide à résoudre tous les problèmes d'accentuation qu'une mauvaise configuration du matériel serait susceptible de causer ou bien d'accéder à certains caractères accentués sur des imprimantes matricielles qui n'en disposent pas. L'utilisateur pourra réaliser des impressions graphiques sur imprimante matricielle, créer son propre jeu de caractères grâce à un éditeur de fontes très simple d'utilisation, etc.

EDITIONS MICRO APPLICATION, 58, rue du Faubourg Poissonnière 75010 PARIS —
Tél. (16-1) 47.70.32.44 380 pages / Disquette intégrée 299 F

● **COMMENT AMÉLIORER
LES PERFORMANCES
DU MATÉRIEL
ET DES LOGICIELS
DES IBM PC
ET COMPATIBLES ?**

Auteurs : Alain MARIATTE
Marcel COTTINI
Michel MARTIN
Michel BUJARDET

Weka France, fondée en 1980, étend ses principales activités dans l'édition d'ouvrages d'information spécialisée, aussi bien dans le domaine professionnel que dans celui des loisirs. Cette société pratique une politique d'édition très particulière, ce qui la différencie des autres éditeurs. Chaque sujet technique est développé sous la forme d'un ouvrage de base autour duquel sont conçus des compléments à un rythme de un par trimestre. Parce que chaque jour de nouveaux développements et de nouveaux produits rendent caduques les informations d'hier, votre ouvrage a besoin en permanence d'être complété.

L'ouvrage "Comment améliorer les performances du matériel et des logiciels des IBM PC et compatibles ?" se présente comme une sorte de "bible" destinée à venir en aide au lecteur à tout moment de ses activités informatiques sur PC. C'est, en fait, un manuel de référence, dont l'organisation claire et hiérarchique doit faciliter l'accès au renseignement désiré. L'ouvrage comporte des exposés, des mises au point à propos de ce qu'il faut absolument savoir sur la gamme d'ordinateurs la plus vendue dans le monde. L'ouverture aux systèmes PS/2 de ce manuel centré sur l'univers PC/XT/AT est particulièrement élaborée pour satisfaire l'amateur désireux d'apprendre ce qu'il faut pour être efficace avec son PC, mais aussi intéressé par l'évolution de la technologie. Grâce aux compléments et mises à jour publiés régulièrement, le possesseur de cet ouvrage de base a l'assurance de protéger et valoriser son investissement initial : ce livre sera toujours d'actualité, contrairement à bien d'autres.

Les auteurs, regroupés dans une équipe parfaitement cohérente, ont mis sur pied et ordonné une somme de

connaissances sur les ordinateurs IBM. Ils se sont efforcés de traiter, le plus clairement possible, tous les sujets de la micro-informatique grand public. L'ouvrage de base traite des chapitres suivants :

- un historique sur les PC et PS/2,
- l'architecture matérielle et logicielle du PC,
- les systèmes d'exploitation,
- les langages de programmation,
- les techniques de base de la programmation,
- les extensions du PC,
- les solutions types,
- les utilisations spéciales du PC,
- sous oublier le chapitre des utilitaires, trucs et astuces.

Besoin d'exemples concrets ? Cet ouvrage présente clairement l'architecture interne des ordinateurs IBM. Tous les composants majeurs sont passés en revue, avec de courts exemples de programmation permettant de les utiliser au mieux, ou d'en tirer des ressources insoupçonnées. Les hobbyistes chevronnés ne sont pas oubliés. Pour eux, l'ouvrage de base et ses compléments proposent des réalisations performantes (interfaces, programmeur d'EPROM, etc.).

Etant donné que le PC est un ordinateur constitué d'un système d'exploitation (MS/DOS, UNIX, OS/2, etc.) et de logiciels, des aperçus exhaustifs et pratiques sur l'emploi des différents SED sont diffusés très largement tout au long de l'ouvrage. Pour les programmeurs, des synthèses portant sur les grands langages de programmation ont été inclus, avec des exemples concrets d'application. Au menu, un Basica minutieusement décortiqué, une synthèse à propos de Pascal et une représentation de l'assembleur des 8/16/32 bits Intel. "C" et les autres suivront dans la foulée.

Pour les hobbyistes exercés, le chapitre traitant des "ruses des programmeurs avertis" (partie 7, chapitre 4) est tout à fait recommandé. Les prochains compléments y apporteront les "trucs" de programmation de la carte d'affichage EGA, des drivers de souris, et bien d'autres astuces encore. L'ouvrage de base développe déjà le

mode opératoire d'accès à la mémoire non-volatile de Setup sur un AT.

Une disquette d'accompagnement regroupe tous les utilitaires proposés tout au long de l'ouvrage. Vous pourrez, par exemple, modifier les couleurs affichées à l'écran depuis le DOS, ce que votre progiciel favori ne sait peut-être pas faire. Si vous cherchiez vainement un logiciel de communication, capable d'envoyer et de recevoir des messages et des télex par le biais des grands serveurs, ou encore susceptible d'enregistrer des pages vidéo-text, pour emploi ou visualisation ultérieurs, ne cherchez plus, vous trouverez ce dont vous avez besoin dans l'ouvrage de base.

ÉDITIONS WEKA — Collection Loisirs
12, Cour Saint Eloi
75012 PARIS

Ouvrage de base de 722 pages, complété tous les 3 mois à raison de 150 pages en moyenne.

Prix normal de l'ouvrage de base : 450 Francs.
Pour profiter de l'offre de souscription, 375 Francs au lieu de 450 Francs, contactez Madame Brigitte Morvant aux éditions Weka.

● **TRUCS ET ASTUCES
dBASE III Plus**

Cet ouvrage comprend un rapide descriptif des commandes de la base de données, plusieurs programmes et routines d'aide tels que la rédefinition des touches de fonction, la protection des données par mot de passe, un programme complet de publipostage, etc.

Un chapitre complet est dédié à la méthode de programmation du logiciel : comment planifier son travail, établir un cahier des charges, structurer et organiser son programme, éviter les erreurs courantes, créer sa propre documentation...

350 pages / Disquette intégrée 299 F
ÉDITIONS MICRO APPLICATION
58, rue du Faubourg Poissonnière,
75010 PARIS
Tél. : (16-1) 47.70.32.44.

FILTRINPUT & SONO

- **FILTRINPUT** est une routine écrite en BASIC, qui, incorporée dans un programme, filtre et contrôle les saisies, refusant les caractères non prévus. De plus, le champ de saisie est matérialisé et positionné sur l'écran tout au long de celle-ci par une ligne de pointillés à toutes les positions de caractères supprimés ou non encore saisis.
- **SONO** est une routine en langage machine utilisée en association avec **FILTRINPUT**, mais qui peut également être mise en œuvre indépendamment de celle-ci. Elle assure la "sonorisation" des opérations qui se déroulent tout au long du programme auquel elle est associée, à l'aide de signaux sonores typiques répertoriés.

SONO

Cette routine, écrite en LANGAGE MACHINE, occupe 79 octets. Parfaitement transportable, on peut l'installer où l'on veut, mais on constatera en examinant le début du listage qu'il est facile de l'implanter au sommet de la mémoire disponible, juste sous l'*HIMEM* existant : celui-ci est alors descendu de 79 octets.

La routine utilise 3 adresses habituellement disponibles en page zéro : \$06, \$07 et \$08. Par ailleurs, elle fait appel à la routine système dite "SPKR" située en \$C030 (susciter cette adresse provoque un "clic" du haut-parleur).

On accède à la routine **SONO**, en six endroits répartis de 8 en 8 octets à partir de son début. Ces six entrées correspondent à six signaux sonores caractéristiques, dont la dénomination explicite peut également être admise comme *nom de variable* en BASIC AppleSoft, ce qui rend son utilisation très claire (par exemple : *CALL ALARME*, ou bien *CALL COUIC...*).

Ces six signaux sonores sont les suivants :

PRET

Destiné plus particulièrement à appeler l'opérateur pour lui signaler qu'il a quelque chose à faire.

CLAVIER

Fait le bruit conventionnel d'une touche de machine à écrire. Peut servir à rendre plus vivante une saisie.

COUIC

Reproduit à peu près le bruit suggéré par son nom.

ALARME

Emet un bruit de klaxon d'atelier : destiné à attirer l'attention sur un événement important.

ERREUR

Utilisé pour prévenir l'opérateur qu'il vient de commettre une faute dans son intervention.

TERMINE

Joue un arpegge pouvant servir de conclusion à l'exécution d'un programme.

Le fonctionnement de la routine repose sur les valeurs de 3 paramètres : une *FRÉQUENCE* et un *CYCLE* interférant l'un sur l'autre, le tout répété un nombre de fois défini par le facteur de *RÉPÉTITION*.

Le listage de l'assemblage de la routine permettra de générer de nouveaux sons à volonté. Je suggère aux connaisseurs d'essayer les trios de valeurs suivants à affecter, dans l'ordre, aux trois paramètres ci-dessus :

95, 90, 10

9, 92, 1

10, 255, 1

65, 0, 1

129, 56, 1

129, 120, 1

109, 0, 1

142, 255, 1

Celui-ci est l'un de mes préférés :

0, 0, 12

FILTRINPUT

Cette routine, en BASIC APPLESOFT (lignes 10 à 69), intègre elle-même la routine **SONO** (lignes 47 à 66, celles-ci pouvant être remplacées par un chargement direct grâce à un *BLOAD SONO*).

Son utilisation est très simple : pour entreprendre une saisie, il faut disposer, d'une part d'une *chaîne-format FMT\$* contenant les valeurs des paramètres nécessaires à la saisie et, d'autre part, d'une forme initiale de *SA\$*, la *chaîne de saisie* proprement dite, qui sera éventuellement modifiée par addition, par suppression, ou par changement de tout ou partie de ses éléments.

La chaîne-format *FMT\$* est construite selon le schéma suivant (les parenthèses ne sont là que pour détacher les éléments facultatifs : elles ne doivent pas figurer dans la chaîne réelle) :

FMT\$ = "vt,ht,ch (,sp.sd (il.i2...in))"

- vt** : valeur d'un *VTAB* qui positionne le champ de saisie.
- ht** : valeur d'un *HTAB* qui positionne le début du champ.
- ch** : longueur du champ de saisie.
- sp.sd** : premier (*sp*) et dernier (*sd*) d'une série continue de caractères admis (exemple : *AZ* pour l'ensemble des lettres majuscules, ou *09* pour l'ensemble des chiffres).
- il.in** : liste des caractères isolés acceptés (10 caractères maxi).

Les caractères ne figurant pas dans la chaîne-format seront impitoyablement refoulés par la routine, avec émission d'un signal sonore d'erreur. A noter que si les paramètres numériques doivent être séparés par des virgules, par contre, les caractères alphanumériques licites sont contigus (les points figurant dans les définitions plus haut ne sont là que pour la clarté des explications).

Exemples de chaînes-formats :

FMT\$ = "6,25,15,AZ" :

Champ de 15 caractères dont le premier se trouve à la 6^e ligne, 25^e colonne, les seuls caractères admis étant des lettres majuscules.

FMT\$ = "8,10,12,az " :

Champ de 12 caractères, ligne 8, colonne 10, lettres minuscules et espace.

FMT\$ = "12,23,5,09ABCDEFS" :

Champ de 5 caractères, ligne 12, colonne 23, chiffres de 0 à 9, lettres de A à F, et caractère "\$" (concerne vraisemblablement la saisie d'un nombre hexadécimal).

FMT\$ = "14,21,19, i" :

Champ de 19 caractères, ligne 14, colonne 21, tous les caractères sont acceptés (depuis l'espace, code ASCII 32, jusqu'au tréma, code ASCII 126).

La *chaîne de saisie SA\$* peut être nulle (*SA\$ = ""*), ou quelconque (elle peut même comporter des caractères qui seraient interdits lors de la saisie), mais sa longueur ne doit évidemment pas excéder celle du champ déterminée dans la *chaîne-format FMT\$*.

Les deux chaînes *SA\$* et *FMT\$* étant définies, la routine est appelée par l'instruction : *GOSUB 13*.

En retour, on retrouve la chaîne de saisie *SA\$* mise à jour par la saisie, et la variable *ASKI* qui donne le *code ASCII* de la commande de validation. Au programme d'en faire, éventuellement, bon usage pour gérer l'écran de saisie.

FILTRINPUT est organisé sur le plan suivant :

10 : saut à la ligne 37, afin de donner les tous premiers numéros de lignes aux sous-programmes dont l'appel très fréquent nécessite des temps d'accès courts.

13 à 18 : extraction des valeurs de la chaîne-format *FMT\$*, à savoir :

ZV = ligne de l'écran (1-24) où doit se trouver le champ.

ZH = colonne (1-40) où se situe le 1^{er} caractère à saisir.

ZC = longueur du champ de saisie.

Z1\$ = premier de la série continue des caractères acceptés.

Z2\$ = dernier de la série continue des caractères acceptés.

ZI\$() = liste des caractères isolés acceptés (s'il y en a), au nombre de ZI caractères, de ZI\$(1) à ZI\$(ZI). S'il n'y en a pas, ZI est nul.

18 : POP désarme le GOSUB 17 de la ligne 15 en inhibant le RETURN de la ligne 17 en fin d'analyse de la chaîne FMT\$.

20 : début de l'analyse de la saisie. Pendant cette opération, la routine va gérer simultanément 2 chaînes de travail : ZS\$ et ZA\$. ZS\$ est une copie de la chaîne SA\$ initiale, qui va la supplanter en subissant les modifications apportées par la saisie. ZA\$ est la chaîne d'affichage, maintenue identique à sa sœur ZS\$, mais complétée d'une ligne de pointillés jusqu'à la longueur du champ de saisie. Noter qu'après avoir été recopiée dans ZS\$ au début du processus, la chaîne de saisie SA\$ va demeurer intacte jusqu'à la frappe de la commande de validation. Si cette dernière est ESC (ligne 22), la chaîne SA\$ sera retournée inchangée, avec signal sonore adéquat. Sinon, elle deviendra la copie de la chaîne de travail ZS\$ avant d'être retournée : copie complète si c'est une flèche verticale qui valide (ligne 23) ; tronquée à hauteur de la position ZK du curseur, si c'est la touche RETURN qui a été pressée (ligne 24). Dans les deux cas, le retour est accompagné d'un bruit de touche.

21 : affichage de la chaîne ZA\$ (saisie en l'état actuel) et positionnement du curseur (ZK est sa position relative dans le champ de saisie).

22 : saisie d'un caractère, inscription de son code ASCII dans la variable ASK1, et début du contrôle : le caractère est-il ESC ? (voir plus haut). C'est la variable ASK1 qui nous permettra de savoir quelle aura été la commande de validation employée pour conclure la saisie, et qui autorisera, par exemple, une gestion efficace d'un écran de saisie (voir l'application dans FILTRINPUT.DEMO).

23 et 24 : suite des tests sur caractères de validation (voir plus haut). Si on échappe à ces lignes, c'est que le caractère frappé n'est pas une commande de validation mais, soit une commande d'édition (lignes 25 à 27), soit un caractère de saisie proprement dite (lignes 28 à 31).

25 : si la commande est la touche DEL (code

ASCII 127) ou CTRL-D (code ASCII 4), nous allons supprimer de la saisie déjà faite le caractère qui suit la position ZK du curseur. Au préalable, annulation (par ZS=Ø) d'un précédent mode insertion éventuel (voir ligne 26), puis mise à jour par suppression de la position ZK+1 dans les chaînes de travail ZS\$ et ZA\$. Dans cette dernière chaîne, la suppression d'un caractère est compensée par l'ajout d'un point en fin de chaîne : ainsi reste sauvegardée la matérialisation de la longueur de champ. Un bruitage adéquat accompagne le retour à la saisie.

26 : si la commande est la touche "←|→" (ou son équivalent : CTRL-I, tous deux de code ASCII 9), nous allons mettre à 1 l'indicateur ZS pour signaler à la suite de la routine (ligne 31) que, jusqu'à ce que l'indicateur ZS soit à nouveau rendu nul, les caractères suivants n'écraseront pas les caractères déjà saisis, mais seront intercalés à la position ZK du curseur. Un bruit de touche est émis avant le retour à la saisie.

27 : si la commande est une flèche horizontale, une fois l'indicateur d'insertion ZS remis à zéro, deux autres indicateurs sont suscités : ZA (= -1 si flèche gauche, +1 si flèche droite) et ZL (= 1 si le mouvement doit faire sortir du champ le curseur, Ø si le curseur reste dans le champ).

Si ZL=Ø, le mouvement du curseur a lieu : CALL -1008 pour reculer quand ZA=1, CALL -1036 pour avancer quand ZA=+1, puis bruit de touche et retour à la saisie. Si ZL=1, on ne bouge pas et un bip d'erreur est envoyé par CALL ERREUR, et retour à la saisie.

28 : désormais, le caractère est un caractère de saisie. On vérifie d'abord qu'il reste de la place pour l'accueillir : si la position relative du curseur ZK a déjà atteint la longueur de champ ZC, il y a envoi d'un bip d'erreur et retour à la saisie.

29 et 30 : on vérifie si le caractère frappé appartient au domaine des caractères admis. Un indicateur d'erreur ZR est d'abord armé à 1 ; après quoi, on compare le caractère frappé à la liste des caractères isolés admis (ligne 29) puis à la série continue admise (ligne 30) : s'il y a une égalité, ZR est annulé. Sinon, un bip d'erreur est envoyé et retour à la saisie.

31 : le caractère frappé étant désormais *licite*, les deux chaînes de travail ZA\$ et ZS\$ sont mises à jour en conséquence, la position de caractère ZK est incrémentée d'une unité et, on retourne à la saisie après envoi d'un bruit de touche.

33 : sous-programme d'initialisation des chaînes de travail ZA\$ et ZS\$ à partir du contenu original de la chaîne de saisie SA\$.

35 : sous-programme construisant, à la 24^e ligne de l'écran, un bandeau explicatif rappelant (sous une forme évidemment très concise) les différentes commandes d'édition et de validation dont dispose l'opérateur au cours d'une saisie. La lettre "n" du mot "fin" étant en 40^e colonne de la 24^e ligne de l'écran est affichée sur celui-ci par l'instruction *POKE 2039,238*, plutôt que par *PRINT* qui provoquerait un "scroll" intempestif. L'instruction *VTAB 21* qui suit, permet de désamorcer le *point-virgule* placé après l'impression précédant cet affichage du "n" final, en expédiant le curseur à un endroit tranquille de l'écran permettant d'envoyer sans risque le *PRINT* libérateur !

37 et 45 : calcul de l'*HIMEM* précédent, duquel on retranche 79 octets (longueur de la routine *SONO*), la nouvelle valeur étant réattribuée à *HIMEM* et affectée à la variable *DÉBUT* qui va servir d'origine pour le calcul des adressages des six appels par *CALL*.

38 à 43 : calcul des 6 adressages ci-dessus, de 8 en 8 octets à partir de *DÉBUT*, et affectation des 6 valeurs à 6 variables (*PRET, CLAVIER, COUIC, ALARME, ERREUR, TERMINE*), permettant ainsi des appels explicites à la routine *SONO*.

47 à 66 : implantation de la routine *SONO* en mémoire, à partir de l'adresse portée par *DÉBUT*. Cette implantation peut évidemment être remplacée par un *BLOAD* à la même adresse ; la réalisation en sera confiée à *DOS*.

FILTRINPUT.DEMO

C'est une application destinée à illustrer l'utilisation des routines *FILTRINPUT & SONO* : réalisation, par exemple, d'un écran de saisie complet dont la gestion est pilotée à partir du *code ASCII* retourné par la variable *ASKI*.

10 DEBUT = PEEK (115) + PEEK (116) * 256 - 256	9A6E
11 :	003A
12 PRET = DEBUT : PRINT CHR\$ (17) : HOME	7AE4
13 CLAVIER = DEBUT + 8	F84A
14 COUIC = DEBUT + 16	81E6
15 ALARME = DEBUT + 24	D824
16 ERREUR = DEBUT + 32	2946
17 TERMINE = DEBUT + 40	9C84
18 :	003A
19 HIMEM = DEBUT	5AB4
20 :	003A
21 :	003A
100 HOME : POKE 49337,47 : HOME : INVERSE : FOR I = 1 TO 120 : PRINT "	
": : NEXT : VTAB 2 : HTAB 13 : PRINT "S ""O ""N ""O" : NORMAL	BF8C
110 :	003A
120 DATA PRET,CLAVIER,COUIC,ALARME,ERREUR,TERMINE	E2AE
130 :	003A
140 VTAB 7 : HTAB 11 : PRINT "INDICATIFS SONORES : " : HTAB 11 : PRINT "--	
-----" : FOR I = 0 TO 5 : READ BIP\$: VTAB 11 + I * 2 :	
HTAB 20 : PRINT BIP\$: NEXT	E298
150 :	003A
160 PRINT CHR\$ (4) ; "BLOAD SONO,A" ; DEBUT	E982
170 :	003A

HELLO

SONO Cette routine était prévue pour fonctionner sous DOS 3.3 mais marche sous ProDOS. Si vous utilisez le DOS 3.3, vous pouvez remplacer le -256 de la ligne 10 par -79.

```

180 RESTORE : FOR I = 0 TO 5: READ BIP$: VTAB 11 + I * 2: FLASH : HTA
    B 14: PRINT " CALL ";BIP$;" ": NORMAL : FOR J = 1 TO 700: NEXT :
    CALL DEBUT + I * 8: FOR J = 1 TO 2000: NEXT : VTAB 11 + I * 2: HT
    AB 14: PRINT " " ";BIP$;" ": NEXT
190 :
200 :
210 HOME : POKE 49337,47: INVERSE : FOR I = 1 TO 120: PRINT " ";: NEX
    T : VTAB 2: HTAB 8: PRINT "C "A "T "A "L "O "G "U "E": NORMAL : V
    TAB 7
220 PRINT CHR$(4);"CAT"
230 VTAB 22: PRINT "PRESSEZ UNE TOUCHE ";; GET R$: PRINT : PRINT CHR
    $(4)"RUN /T22/STARTUP"

```

D5BE
003A
003A
A478
E77D
EC19

FILTRINPUT avec DÉMO

```

10 GOTO 37: REM (pour laisser les sous-programmes en tete afin d'ac
    celerer l'execution.)

```

4E15

SOUS-PROGRAMMES

SAISIE FILTRÉE

```

13 ZK = 1:ZV = VAL (FMT$): GOSUB 16:ZH = VAL ( MID$ (FMT$,ZK)): GO
    SUB 16:ZC = VAL ( MID$ (FMT$,ZK))
14 Z1$ = CHR$(0):Z2$ = CHR$(255):ZI = 0: GOSUB 16:Z1$ = MID$ (F
    MT$,ZK,1):ZK = ZK + 1:Z2$ = MID$ (FMT$,ZK,1)
15 GOSUB 17:ZI = ZI + 1:Z1$(ZI) = MID$ (FMT$,ZK,1): GOTO 15
16 ZK = ZK + 1: ON ZK > LEN (FMT$) GOTO 18: ON MID$ (FMT$,ZK - 1,1
    ) < > ", " GOTO 16: RETURN
17 ZK = ZK + 1: IF ZK < = LEN (FMT$) THEN RETURN
18 POP
19 :
20 GOSUB 33:ZK = 0:ZS = 0:ZX = 0: CALL PRET: VTAB ZV
21 HTAB ZH: PRINT ZA$;; HTAB ZH + ZK
22 GET Z$:ASKI = ASC (Z$): IF ASKI = 27 THEN GOSUB 33: HTAB ZH: PR
    INT ZA$: CALL COUIC: RETURN
23 IF ASKI = 10 OR ASKI = 11 THEN SA$ = ZS$: HTAB ZH: PRINT ZA$: CAL
    L CLAVIER: RETURN
24 IF ASKI = 13 THEN SA$ = MID$ (ZS$,1,ZK): GOSUB 33: HTAB ZH: PRIN
    T ZA$: CALL CLAVIER: RETURN
25 IF ASKI = 127 OR ASKI = 4 THEN ZS = 0:ZA$ = MID$ (ZA$,1,ZK) + M
    ID$ (ZA$,ZK + 2) + ".":ZS$ = MID$ (ZS$,1,ZK) + MID$ (ZS$,ZK + 2
    ): CALL COUIC: GOTO 21
26 IF ASKI = 9 THEN ZS = 1: CALL CLAVIER: GOTO 22
27 IF ASKI = 8 OR ASKI = 21 THEN ZS = 0:ZA = (ASKI = 21) - (ASKI = 8
    ):ZL = (ZK < > (ZC - 1) * (ZA + 1) / 2): CALL ERREUR * NOT ZL -
    (1022 + 14 * ZA) * ZL:ZK = ZK + ZA * ZL: CALL CLAVIER: GOTO 21
28 IF ZK = ZC THEN CALL ERREUR: GOTO 21
29 ZR = 1: IF ZI THEN FOR I = 1 TO ZI:ZR = ZR * (Z$ < > Z1$(I)): N
    EXT
30 ZR = ZR * (Z$ < Z1$ OR Z$ > Z2$): IF ZR THEN CALL ERREUR: GOTO 2
    2
31 ZN = ZK + 2 - ZS:ZA$ = LEFT$ ( MID$ (ZA$,1,ZK) + Z$ + MID$ (ZA$
    ,ZN) + ZP$,ZC):ZS$ = MID$ (ZS$,1,ZK) + Z$ + MID$ (ZS$,ZN):ZK =
    ZK + 1: CALL CLAVIER: GOTO 21
32 :
33 ZA$ = LEFT$ (SA$ + ZP$,ZC):ZS$ = SA$: RETURN

```

5F30
0B4B
42E5
E431
12F4
43A1
003A
7055
B405
728E
9D27
6FBB
2397
5365
08D9
302C
8EE2
D30D
9E56
003A
255D

BANDEAU INSTRUCTIONS

```
35 VTAB 24:J = 1: FOR I = 1 TO 5: INVERSE : PRINT MID$ ("FLV->IDELR
    ETESC",I * 3 - 2,3);: NORMAL :Z = I * (I < 5) + 3 - (I = 3): PRIN
    T MID$ ("=ok =ins =sup =tronc =fi",J,Z);:J = J + 2: NEXT : POKE 2
    039,238: VTAB 21: PRINT : RETURN
```

5070

INITIALISATIONS

```
37 DEBUT = PEEK (115) + PEEK (116) * 256 - 79: REM ctM      La rout
    ine SONO s'installera d'elle-meme sous l'ancien HIMEM, celui-ci s
    e reajustant en consequence.ctM
38 PRET = DEBUT
39 CLAVIER = DEBUT + 8
40 COUIC = DEBUT + 16
41 ALARME = DEBUT + 24
42 ERREUR = DEBUT + 32
43 TERMINE = DEBUT + 40
44 :
45 HIMEM: DEBUT
```

BA41
DF7F
F84A
81E6
D824
2946
9C84
003A
1417

ATTENTION ! sous ProDOS, si vous chargez SONO, remplacez -79 de la ligne 37 par -256.

LES LIGNES CI-DESSOUS POURRAIENT ÊTRE REMPLACÉES PAR PRINT CHR\$(4) ; "BLOAD SONO,A" ; DÉBUT

```
47 DATA 5, "5, 20: REM parametres pour PRET
48 DATA 10, 40, "1: REM parametres pour CLAVIER
49 DATA 0,120, "1: REM parametres pour COUIC
50 DATA 10, 40, 30: REM parametres pour ALARME
51 DATA 190, "0, "1: REM parametres pour ERREUR
52 DATA 31, "0, "1: REM parametres pour TERMINE
53 :
54 FOR I = 0 TO 5:J = DEBUT + I * 8: READ J1,J2,J3
55 POKE J,162: POKE J + 1,J1
56 POKE J + 2,160: POKE J + 3,J2
57 POKE J + 4,169: POKE J + 5,J3
58 IF I < 5 THEN POKE J + 6,208: POKE J + 7,38 - I * 8
59 NEXT
60 :
61 DATA 134, "6,132, "7,133, "8,164, "7,152
62 DATA 170,202,208,253, 44, 48,192, 69, "6
63 DATA 170,202,208,253, 44, 48,192
64 DATA 136,208,236,198, "8,208,230, 96
65 :
66 FOR I = DEBUT + 46 TO DEBUT + 78: READ J: POKE I,J: NEXT
67 :
68 :
69 ZP$ = "":ZB$ = "": FOR Z = 1 TO 40:ZP$ = ZP$ + ".":ZB$ = ZB$ + "
    ": NEXT :Q$ = CHR$ (34)
```

F007
E831
B20F
F843
F656
A620
003A
6F28
37A5
DEA0
4BAE
256E
0582
003A
4DB7
96F8
409B
2D5E
003A
8225
003A
003A
E89B

PROGRAMME DÉMO

```
110 DIM SA$(5),FMT$(5),ECRAN$(5),ECRAN%(5,1): REM ctMctM      S
    A$( ) = chaines de saisies successivesctM      FMT$( ) =
    formats successifsctM      ECRAN$( ) = libellesctM
    ECRAN%( ) = ordonnees suivante,precedente.ctMctM
120 DATA "PIPE A PAPA","6,25,15,AZ ", "MAJUSCULES et ESPACE",2,5
```

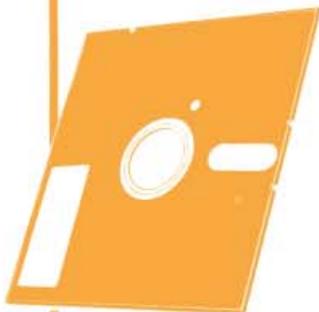
545D
DAFF

```

130 DATA "pom pom","8,25,15,az ","MINUSCULES et ESPACE",3,1           CE31
140 DATA "(7.8+9)*3/4-5","10,26,14,09+*./.=()", "ARITHMETIQUE DECIMALE  A147
    ",4,2
150 DATA "$FADA","12,23,5,09ABCDEF$", "NOMBRE HEXADECIMAL",5,3       6D32
160 DATA "(I,;.:I)","14,21,19, """, "TOUS CARACTERES",1,4           10B2
170 :                                                                     003A
180 FOR I = 1 TO 5: READ SA$(I),FMT$(I),ECRAN$(I): FOR J = 0 TO 1: RE    OE40
    AD ECRAN$(I,J): NEXT J: SA$(5) = LEFT$(SA$(5),5) + Q$ + Q$        003A
    + Q$ + MID$(SA$(5),6): REM La 5eme option autorise la saisie de    003A
    n'importe quoi (y compris "").
190 :
200 :
210 TEXT : HOME :TITRE$ = " ""E C R A N ""D E ""S A I S I E """:      B3A9
    INVERSE : PRINT ZB$;TITRE$;ZB$: NORMAL
220 FOR I = 1 TO 5: VTAB 4 + I * 2: HTAB 2: PRINT ECRAN$(I);" :": NEX  9690
    T                                                                     DEB5
230 VTAB 16: FOR I = 1 TO 40: PRINT "_": NEXT I: PRINT
240 VTAB 18: PRINT "Chaine retournee ": VTAB 20: PRINT "Code ASCII c   21D3
    aract.commande retour :
250 VTAB 21: FOR I = 1 TO 40: PRINT "_": NEXT I: PRINT                49B1
260 GOSUB 35: REM (bandeau instructions)                                6C18
270 :                                                                     003A
280 :                                                                     003A
290 CALL ALARME:LIGNE = 1                                               08E8
300 :                                                                     003A
310 SA$ = SA$(LIGNE):FMT$ = FMT$(LIGNE): GOSUB 13:SA$(LIGNE) = SA$: R   9C68
    EM ctMctM                    SA$ = chaine de saisiectM            FMT$ =
    format de saisie et filtre (caracteres autorises).ctMctM
320 VTAB 18: HTAB 20: CALL - 868: PRINT Q$;SA$;Q$                     ED7E
330 VTAB 20: HTAB 37: CALL - 868: PRINT ASKI                           428F
340 :                                                                     003A
350 IF ASKI = 27 THEN VTAB 23: CALL - 958: CALL TERMINE: GOTO 400:     9F61
    REM touche ESC
360 IF ASKI = 10 OR ASKI = 13 THEN LIGNE = ECRAN$(LIGNE,0): GOTO 310:   3656
    REM touche FLECHE BAS ou touche RETURN
370 IF ASKI = 11 THEN LIGNE = ECRAN$(LIGNE,1): GOTO 310: REM touche F  C02E
    LECHE HAUT
380 :                                                                     003A
390 :                                                                     003A
400 PRINT CHR$(4)"RUN/T22/STARTUP"                                       9EAB

```

SONO (lire l'article)



/T22/SON/SONO.A\$95B1.L\$004F

```

95B1: A2 05 A0 05 A9 14 D0 26 A2 0A A0 28 A9 01 D0           ECED
95C0: 1E A2 00 A0 78 A9 01 D0 16 A2 0A A0 28 A9 1E D0       BB73
95D0: 0E A2 BE A0 00 A9 01 D0 06 A2 1F A0 00 A9 01 86       191F
95E0: 06 84 07 85 08 A4 07 98 AA CA D0 FD 2C 30 C0 45       E303
95F0: 06 AA CA D0 FD 2C 30 C0 88 D0 EC C6 08 D0 E6 60       4F8B

```

Les sources des programmes figurent sur la disquette T.M. n°22.

● **TECHNIQUES DE PROGRAMMATION EN C LA GESTION DE FICHIERS (François-Xavier Eloundou)**

Cet ouvrage s'adresse aux programmeurs utilisant le langage C. L'auteur, Ingénieur Système, a voulu faire de ce livre le complément indispensable aux ouvrages de référence et aux manuels d'initiation. C'est pourquoi il explique en détail les concepts et propose un certain nombre d'outils nécessaires à tout utilisateur de C développant des applications faisant intervenir les fichiers de données.

L'ouvrage est organisé en quatre parties :

- La première présente les outils de base du langage C utilisés dans la gestion de fichiers. Elle traite de la nature et des attributs des données utilisées ainsi que des fonctions classiques de la bibliothèque de gestion de fichiers.
- La deuxième traite des fichiers séquentiels et de la construction d'un ensemble complet de fonctions assurant une gestion efficace de l'accès séquentiel.
- La troisième analyse tous les aspects de l'utilisation de fichiers directs.
- La quatrième explique en détail le séquentiel indexé B-Tree ainsi que son extension B+Tree et sa mise en œuvre : dBASE.

Tous les concepts sont illustrés par de nombreux exemples permettant à l'utilisateur de mettre en pratique ses connaissances au fur et à mesure du développement du livre.

Livre broché, 512 pages — 288 F

SYBEX, 6-8 impasse du Curé, 75018 PARIS
Tél. : (16-1) 42.03.95.95.

● **ALGORITHMIQUE ET PROGRAMMATION SCIENTIFIQUE EN TURBO PASCAL (A. Warusfel et T. Lachand-Robert)**

Cet ouvrage est le premier de trois volumes traitant de la programmation

en Turbo Pascal. Les deux autres volumes à paraître seront consacrés aux applications en mathématique et en physique.

Les auteurs, familiers du contexte universitaire, ont développé ces ouvrages avec le souci de répondre aux besoins des enseignants et des étudiants dans le cadre de l'enseignement informatique du premier cycle universitaire. Leur démarche a été de réaliser un outil de travail conforme aux normes universitaires auxquelles les élèves du premier cycle ont l'habitude de s'adresser.

Ce premier volume couvre l'approche de l'informatique à travers le sous-ensemble officiel de Turbo Pascal. Il s'agit d'une présentation raisonnée des principales caractéristiques de l'utilisation scientifique du micro-ordinateur.

Parmi les chapitres traités, deux sont à vocation universelle : le chapitre 2 étudie les structures de contrôle et le chapitre 7, les méthodes de programmation rationnelle. D'autres sont davantage liés à Turbo Pascal, tels ceux consacrés aux procédures et aux types. Un dernier chapitre esquisse une définition des termes abstraits liés à toute activité de programmation.

Les auteurs proposent aussi des références théoriques et pratiques nécessaires à la programmation en Turbo Pascal, et en particulier un guide syntaxique complet ainsi que la liste des procédures et fonctions standard usuelles.

Cinq annexes complètent l'ouvrage :

- Quelques maximes de programmation.
- Bibliographie.
- Textes officiels.
- Codes ASCII.
- Mots réservés de Turbo Pascal.

SYBEX. Livre broché, 352 pages — 178 F

● **LES MÉMOMAC**

Suite au succès de nos aide-mémoire (plus de 90 000 exemplaires vendus à ce jour), et du standard Macintosh dans les entreprises, une nouvelle série est disponible depuis le 9 septembre dernier, les *MémoMac*.

Une nouvelle collection, un nouvel état d'esprit : il ne s'agit pas ici de décrire

le B-A BA d'un logiciel, ni d'entreprendre une pédagogie évoluée pour en faire comprendre les mécanismes.

MémoMac s'adresse à tous les utilisateurs de Macintosh qui, faute de temps, n'en tirent pas tout le bénéfice possible. Chaque MémoMac est consacré à un logiciel spécifique (Word, Excel, PageMaker, 4^e Dimension, etc...) ou à un sujet plus général tel que le Macintosh, les communications sur Macintosh.

Ces "Mémos" reprennent les fonctions du logiciel, les explicitent, et en démontrent l'intérêt et la finalité, donnant à l'utilisateur l'opportunité d'améliorer l'ordinaire de son travail. Ils constituent une référence immédiate et pratique pour le lecteur qui trouvera les réponses à ses questions exposées dans l'ordre logique d'utilisation et illustrées d'exemples.

La structure des MémoMac Cedic/Nathan permet à l'utilisateur confirmé de se remémorer l'utilisation des principales commandes et au débutant de découvrir rapidement le logiciel.

Comme rien n'est laissé au hasard aux Editions Cedic/Nathan, le format de nos MémoMac (13,5x25 cm) a également été particulièrement étudié, d'une part pour nous démarquer des aide-mémoire Cedic/Nathan destinés au monde PC et, d'autre part, pour ne pas avoir à trop réduire les nombreuses mais néanmoins nécessaires copies d'écran du Macintosh.

NOUVEAUTÉS DISPONIBLES :

- MémoMac Excel base de données, graphiques, macros par Nathalie Barbary.
- MémoMac Excel tableur par N. Barbary.
- MémoMac Excel macros par N. Barbary.
- MémoMac Word par Jean-Christophe Courte.

A paraître : Tech 2D par Xavier Toumarie — CricketDraw par Pascale Cerf — 4D-initiation par Patricia et Gilles Batteux — 4D-programmation par P. et G. Batteux — PageMaker par Olivier Castelli — HyperCard par Pierre Brandeis.

Distribué par PCV diffusion Prix : 55 F
9, rue Méchain — 75680 PARIS Cedex 14

Le nouveau GRIBOUILLE

VU par son auteur

Eh bien nous y sommes ! Les "collégiens" aidant, nous sortons une nouvelle version de GRIBOUILLE, spécialement destinée à l'APPLE IIGS.

Que vous apporte ce nouveau Gribouille ?

Il a conservé, souvent en les améliorant, les fonctions de la version précédente. Deux exceptions cependant : le calcul est supprimé (il m'aurait fallu passer trop de temps pour reprendre mes procédures). Supprimée aussi l'impression des graphiques : sur le GS, leur définition est beaucoup plus fine que sur le *Ile* ou le *Ilc*, et ils passent mal sur l'Image Writer.

A cela près, vous retrouverez :

- La coupure des mots, avec possibilité, maintenant, de marquer ceux que Gribouille doit laisser entiers.
- Le glossaire, directement accessible par un clic de souris.
- Tous les petits mécanismes de correction : écriture en remplacement, interversion de deux caractères, conversion Majuscule/minuscule, sauts par Pomme-chiffres, déplacement de quelques mots par les flèches. S'y ajoutent, la possibilité de forcer les majuscules en début de phrase, même si vous oubliez le shift, et surtout un jeu de flèches beaucoup plus commode, doublé par la panoplie ascenseur-souris. Vous avez, maintenant, un saut direct aux débuts des paragraphes et, une exploration par écrans entiers.
- La recherche-remplacement, toujours aussi rapide, avec jokers et sauts de

lignes, s'est affinée (mots ou syllabes, choix de l'emplacement du curseur dans la chaîne trouvée).

- Sont également restées : la frappe au kilomètre (conservée pour la rapidité d'écriture et la souplesse de correction), la visualisation de la page formatée, l'introduction des codes de l'imprimante, et toutes les commandes de mise en page. Aucun problème pour reprendre sur Gribouille-GS les fichiers écrits avec Gribouille *Ile* ou *Ilc*.
- Ont été singulièrement magnifiés : le tiroir et le dispositif général de la page.

Pourquoi passer sur le GS ?

Mais alors, si tout est resté, à quoi bon passer sur GS ? That is the question !

Apple-GS nous procure deux importants avantages : l'écran graphique et un processeur 16 bits, c'est-à-dire l'accès à une grande capacité de mémoire.

Aussi avons-nous écrit une version qui **tourne vite** et qui est **adaptée à l'équipement actuel de nos clients**, c'est-à-dire un GS avec un méga de mémoire et une imprimante Image Writer *I* ou *II*.

Notre driver d'imprimante est "orienté caractères". L'autre option aurait été le "bit image".

BIT IMAGE OU DRIVER ORIENTÉ CARACTÈRES ?

En bit image, quand on écrit **A**, on envoie à l'imprimante un inventaire de tous les points noirs ou blancs qui composent le **A**. Cet inventaire ne peut guère occuper **moins de 8 octets**, mais généralement davantage. Il permet, par contre, d'employer une grande variété de polices, dans des tailles diverses.

Avec le driver orienté caractères, **un seul octet suffit** : celui qui donne la valeur ASCII du **A**. Cet octet a pu être précédé par des codes qui, pour toute une suite de caractères, commandent :

- La taille (élite ou pica, en condensé, normal ou expansé).
- La police (standard, proportionnel ou caractères créés par l'utilisateur).
- Le gras, le soulignement.
- Eventuellement, la hauteur de ligne.

Le bit image est lent, et la sortie série, avec ses 9600 bauds de vitesse maximum, est un frein qu'il est impossible de desserrer.

Corollaire de cette décision : pas non plus de "Wysiwyg" (What you see is what you get, l'écran reflétant exactement l'image de la page imprimée). Qu'il s'agisse de caractères ou de bit image, les outils actuels du GS (Quick Draw et Font Manager) ne peuvent pas réaliser le Wysiwyg pour les caractères condensés ou expansés. Il ne nous est donc pas possible de donner à l'écran l'image de la page puisque Gribouille peut imprimer (et justifier) non seulement en proportionnel, mais aussi en condensé ou en expansé. Nous aurions en tout cas écarté le Wysiwyg à cause de sa lenteur.

GRAPHIQUE malgré tout !

Pas de bit image et pas de Wysiwyg, mais nous ne renonçons pas pour autant à l'écran graphique, qui apporte à la version GS de Gribouille des fonctions utiles :

- En premier lieu, il permet l'affichage des caractères personnalisés créés par l'utilisateur.

Comme l'ancien, le nouveau Gribouille comporte un générateur de caractères enregistrables et téléchargeables sur l'Image Writer I ou II.

Ils sont, maintenant, dessinés à la souris et ils sont affichables. Vous pouvez voir à l'écran, pendant la frappe, vos racines carrées, sigmas, italiques et vos alphabets grec, cyrillique ou arabe. Vous pouvez aussi les débrancher pour voir les caractères standards. (A propos de caractères arabes, j'ai dû programmer l'écriture de droite à gauche, que certains linguistes me réclamaient avec véhémence).

- En second lieu, vous avez le grand jeu des souris, menus déroulants et du multifenêtrage. Amateurs de clavier, rassurez-vous : lâcher les touches, attraper la souris, trouver de la place sur un coin de table pour rouler la bille, c'est la vie rêvée, certes, mais nous avons été discrets. La quasi totalité des commandes sont doublées par des pomme-lettres, dont la liste sera généreusement affichée au moindre appel.

Et vous verrez que le multifenêtrage n'est pas un apport négligeable. Vous pouvez fort bien travailler dans une fenêtre unique qui occupe tout l'écran. Mais vous constaterez que, pour certains travaux (tris, création à la volée de glossaires, traductions, remaniements de plans et autres pétrissages de paragraphes) un

bon multifenêtrage rend service. Je sais que le nôtre est bon : trois fenêtres de 65000 octets chacune, qu'on peut réduire, agrandir, glisser à volonté...

Et la capacité ?

Elle nous permet, sur le GS, d'ouvrir au texte et au glossaire 195000 octets répartis en trois fenêtres.

Il faut ajouter l'amélioration du tiroir. Ce n'est plus un tiroir, mais DIX. Le "tiroir" est une sorte de couper-coller particulier à Gribouille. Nuancé (il permet d'enregistrer sur disquettes des parties de textes), souple et surtout beaucoup plus puissant que le couper-coller classique : il donne un accès immédiat aux dix passages marqués en dernier lieu par l'utilisateur. L'inventaire de ces passages et la récupération de l'un d'eux est simple et rapide.

- La numérotation peut être placée en haut ou en bas de la page, décalée et justifiée au gré de l'utilisateur. Il en est de même pour la date et l'heure.
- On peut aussi décaler et justifier librement les titres et changer à tout instant le texte qui s'y inscrit. Cela permet d'avoir toujours une parfaite concordance entre le titre et le sujet développé.
- La page peut enfin être disposée en recto verso. Combinée à la possibilité d'imprimer d'abord les pages impaires, de retourner les feuilles et d'imprimer les pages paires, cette nouvelle fonction

donne des documents qui ont une marge large du côté de l'onglet. On peut ainsi les relier ou les perforer sans endommager le texte.

Il a été question, dans ces colonnes, de Gribouille and Co. A-t-il vu le jour ?

Pas vraiment. Il y a bien un papier à lettre de ce nom. Il n'a pas été beaucoup utilisé. J'aurais bien voulu, mais je suis écrasée de travail, et je n'ai pas pu.

Mais l'esprit Gribouille and Co, lui, oui, il existe pleinement et vigoureusement. J'ai reçu des centaines de lettres, et j'ai répondu à chacune. Dans ces lettres, il y avait un peu de tout. La dame qui "vire" les fichiers du programme pour faire de la place aux siens, et qui hélas, ne boote plus ! Ceux qui veulent greffer Gribouille sur leur programme en basic, qui a juste besoin d'une "partie traitement de texte". Ceux qui veulent m'envoyer leurs disquettes favorites, pour que je m'en pénètre, que je les marie à Gribouille, et que je réalise ainsi l'idéal, un Gribouille-Multi-Scribe, ou un Gribouille avec WPL. Surtout tous ceux qui, comme moi, veulent avoir sous les doigts un outil commode pour écrire. Beaucoup ont aimé Gribouille, et m'ont aidée à percevoir les améliorations, à imaginer les fonctions nouvelles qu'on pouvait lui apporter sans le surcharger. La version GS est un peu leur œuvre. J'espère qu'ils s'y reconnaîtront et qu'ils l'aimeront.

Madeleine HODÉ.

5, rue Humblot, 75015 PARIS, tél. : (16-1) 40.59.49.77
Gribouille-GS — Prix TTC 990 francs (880 pour les anciens clients).

PROGRAMMATION DU 80386 (par Martine et Jean-Michel TRIO) Application à l'utilisation du mode protégé sous MS-DOS

Cet ouvrage étudie le nouveau "moteur" des micro-ordinateurs de la dernière génération, le 80386, et plus particulièrement sa programmation. En effet, ce microprocesseur offre une large variété de possibilités, pour la plupart encore peu utilisées. Par exemple : repousser la limite d'un segment de 64 KO à 4 Giga octets ou utiliser la capacité multi-tâches. Cette étude de la

programmation du 80386 a été conçue selon deux principes :

- Autonomie des thèmes et étude progressive de la programmation. Afin de faciliter la compréhension du lecteur, chaque thème peut être appréhendé de manière indépendante. Cependant, l'étude des différents thèmes de l'ouvrage et leur mise en application sous forme d'exemples, sont abordées

de façon progressive au fil des pages.

- Choix du cadre MS-DOS. Ce cadre a été retenu pour cet ouvrage parce qu'il constitue le standard du marché et que sa simplicité en fait une excellente base de développement.

Le livre comporte trois parties :

- Fonctionnement du 80386 en mode réel.
- Les thèmes du mode protégé.
- L'interface avec le système MS-DOS.

EYROLLES, éditions-librairie 432 pages, 350 F
61, boulevard Saint-Germain — 75240 PARIS Cedex 05
Tél. : (16-1) 46.34.21.99.

● **NATHAN ET P.S.I
LANCENT UNE
NOUVELLE COLLECTION
LES INFOGUIDES**

Une nouvelle collection pour les étudiants et pour tous ceux qui veulent découvrir :

- Un langage (Turbo C, Turbo Pascal...).
- Un système (MS-DOS, Unix...).
- Un logiciel (dBASE, Multiplan...).

L'objectif de cette nouvelle collection est de mettre entre les mains des étudiants en informatique des outils pédagogiques en résonance avec leurs programmes de cours. Ils y trouveront, écrits par des enseignants de l'université, des manuels complets, didactiques et précis, sur les grands langages de programmation abordés dans les cursus informatiques.

L'utilisateur est donc nettement ciblé : il ne s'agit pas du débutant qui vient d'acheter un micro ni du cadre moyen dont la direction a péremptoirement posé un PC sur son bureau.

La collection Infoguide est dirigée par Jean-Michel Gaudin, Inspecteur départemental de l'Education nationale.

Prix : 68 F
Distribué par PCV diffusion,
9, rue Méchain — 75680 PARIS Cedex 14

NOUVEAUTÉS DISPONIBLES :

- INFOGUIDE MS-DOS par Augustin Garcia-Ampudia.
- INFOGUIDE TURBO PASCAL par Bénédicte Hudault
- INFOGUIDE TURBO C par Jean-Michel Gaudin.

● **EXCEL GRAPHIQUE
VERSION 1.5 SUR MAC**

Auteur : Jean-Christophe Courte est consultant dans le domaine de la communication. Il travaille beaucoup sur Macintosh et connaît à fond tous les logiciels sur cette machine.

Description : De nombreuses personnes font du graphique mais sans vraiment savoir l'utiliser comme outil d'analyse et de communication ou encore sans savoir choisir le graphique

adapté à un besoin précis de représentation.

Excel graphique version 1.5 sur Mac essaie de combler cette lacune tout en proposant une initiation complète au grapheur de la dernière version Excel ainsi qu'à la programmation simple de macros dédiées au graphique. L'auteur soumet également un programme exhaustif permettant de gérer des formats graphiques.

Ce livre sera un outil idéal pour tous les cadres et consultants d'entreprise ayant à communiquer des données d'une manière plus percutante et efficace qu'un simple tableau de chiffres.

232 pages — 195 F
Distribué par PCV diffusion.

● **CLEFS POUR
MICROSOFT C 5.0
SOUS MS-DOS ET OS/2**

Auteur : François Piette est un ingénieur polyvalent spécialisé en informatique et électronique. Il est Directeur du B.E.I. (Bureau d'Etudes Informatiques) et responsable du service informatique du C.R.I.F. (Centre de Recherche des Industries de Fabrication Métallique). Il est l'auteur de "Clefs pour C" ancienne édition, de "Plus loin avec MS/PC-DOS" et du "Livre du MS-PC/DOS sur PC et AT" chez BCM.

Thème : Le C de Microsoft est le langage de développement officiel sous OS/2 et le langage utilisé par la majorité des programmeurs professionnels. Il est à noter que le C est également le langage utilisé par tous les développeurs sous Unix. Le Microsoft C est vendu 5000 F.

Description : Ce livre est le premier à traiter de C 5.0. C est désormais le langage standard de développement sous Unix et MS-DOS.

6 parties articulent cet ouvrage :

- Règles et programmation.
- Opérateurs et données.
- Fichiers et allocation mémoire.
- Environnement de compilation.
- Librairies de fonctions standard OS/2.
- Librairies de fonctions standard MS-DOS.

272 pages — 190 F — Distribué par PCV diffusion.
ÉDITIONS P.S.I., 6-10 bl Jourdan, 75014 PARIS
Tél. : (16-1) 45.65.06.06.

● **LE LIVRE DE L'OS/2**

Pour une prise en main efficace de ce nouvel environnement, LE LIVRE DE L'OS/2 permet d'acquérir une connaissance approfondie de la philosophie du système, tant dans l'utilisation des commandes que dans l'organisation propre à OS/2 (sa programmation, ses capacités multi-tâches...).

Même s'il n'est pas familier de l'univers micro-informatique, le lecteur assimilera aisément les thèmes traités ; de nombreux exemples accompagnent la description des commandes, le texte étant lui-même d'un abord aisé. Enfin, pour une navigation efficace parmi les nombreuses fonctions du système, un glossaire complet de la syntaxe et du rôle des commandes OS/2 épaulera l'utilisateur durant chaque étape de son apprentissage.

409 pages — 199 F
ÉDITIONS MICRO APPLICATION
58, rue du Faubourg Poissonnière, 75010 PARIS
Tél. : (16-1) 47.70.32.44.

● **SQL Bases Relationnelles
(par Christian Bonnin)**

Longtemps considérées comme un mythe dans le monde informatique, les BASES DE DONNÉES RELATIONNELLES constituent aujourd'hui le fondement des applications informatiques modernes.

L'ouvrage de Christian Bonnin expose toutes les possibilités du langage SQL afin de permettre à l'utilisateur de bâtir, mettre à jour et gérer les Bases de Données. En effet, SQL/DS (Structured Query Language/Data System) offre un système de gestion des BASES DE DONNÉES RELATIONNELLES opérationnel et performant grâce aux langages d'interrogation généralisés conviviaux et efficaces QMF (Query Management Facility) et QBE (Query By Example).

L'utilisateur voit ainsi ses applications considérablement facilitées. Cet ouvrage de base s'appuie sur les langages de programmation les plus utilisés en informatique de gestion : COBOL et PL/1.

Livre d'apprentissage du langage SQL de traitement des BASES DE DONNÉES RELATIONNELLES s'adresse à tous les programmeurs d'applications

EYROLLES, éditions-librairie 136 pages — 160 F
61, boulevard Saint-Germain, 75240 PARIS Cedex 05
Tél. : (16-1) 46.34.21.99.

La petite fille avec une boîte de chocolat

... où Claude Aubry se surpasse !

Mon engouement pour les choses de l'esprit remonte à ma rencontre, à l'âge de trois ans, avec une boîte de chocolat... dont je tairai la marque.

Sur cette boîte jaune et rouge figurait au centre, une petite fille blonde tenant à la main une boîte de chocolat jaune et rouge où l'on voyait une petite fille blonde tenant elle-même une boîte de chocolat. A l'aide des lunettes de ma grand mère j'avais, alors, découvert que ces univers imbriqués ne s'arrêtaient pas là et que la dernière petite fille discernable à l'œil nu avait dans ce qui ressemblait à sa main quelque chose comme une boîte.

Ma mère, troublée par mon regard fixe et mon immobilité traumatique à chaque petit déjeuner décida, alors, d'acheter une sorte de cacao à la banane...

Trop tôt apparue dans mon existence, la récursivité disparaissait avant que j'en aie élucidé le mystère.

LA RECURSIVITE

Dans l'un des premiers articles de cette saga des temps modernes qui me rend riche et célèbre en même temps qu'elle vous rend intelligent et beau, je me souviens d'avoir défini des «niveaux algorithmiques» : SEQUENCE, TEST, ITERATION.

J'avais alors sauté volontairement et discrètement un quatrième niveau qui fait les délices de tous les masochistes que nous sommes : la RECURSIVITE.

Nous allons aujourd'hui nous rouler avec volupté dans cette fange, et réaliser des programmes que personne ne pourra comprendre, même pas nous...

On dit d'un objet qu'il est RECURSIF lorsqu'il est lui-même utilisé dans sa composition ou sa définition.

En dehors des petites filles sur les boîtes de choco-

lat, la récursivité apparaît sous forme de définition récurrente en mathématique où un élément S_n d'une suite est défini par $S_n = f(S_{n-1}, S_{n-2}, \dots)$ et en informatique où certains langages permettent à une procédure de s'appeler elle-même.

Sa puissance réside dans la possibilité de décrire un ensemble infini d'objets au moyen d'une phrase finie.

L'outil indispensable à la gymnastique récursive est la procédure (sous-programme) utilisant une structure de pile pour gérer ses variables locales et ses adresses de retour.

FORTRAN et COBOL ne connaissent pas ; BASIC ignore sauf dans ses versions les plus pascalisées ; PASCAL ALGOL et C permettent une utilisation raisonnable de la récursivité. Il est impensable de programmer LOGO ou LISP autrement qu'en mode récursif.

Un premier programme récursif

```
void recurs(i)
int i;
{
  if (i > 0) recurs(i-1);
  printf("%d\n",i);
}

main()
{
  recurs(3);
}
```

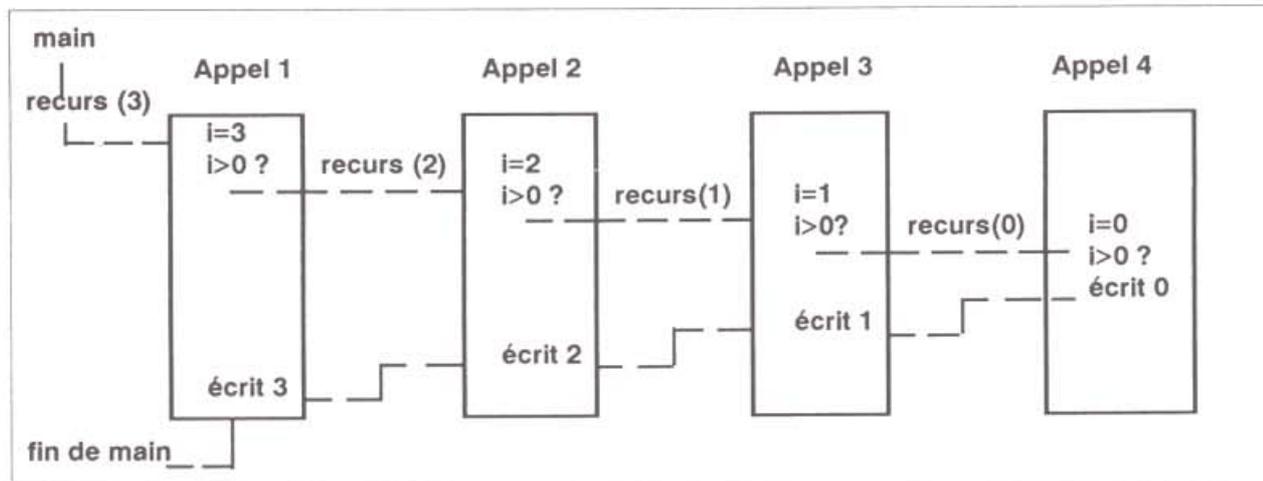
Lorsque nous exécutons ce programme, nous voyons s'afficher en colonne, et devant nos yeux émerveillés, les chiffres 0 1 2 et 3. Que s'est-il passé?

Le programme principal appelle la procédure «re-curs» en transmettant à la variable locale i la valeur 3. Un test est effectué et, si la valeur de i est différente de 0 (ce qui est évidemment le cas), la procédure recurs est appelée depuis elle-même avec comme nouveau paramètre la valeur 2.

Comprenez bien, UN NOUVEL ENSEMBLE DE VARIABLES LOCALES (ici une seule) EST CREE A CHAQUE ACTIVATION DE LA PROCEDURE. Bien qu'elles portent le MEME NOM, ces variables sont DISTINCTES et seul le jeu de variables correspondant à un environnement activé est visible.

Nouveau test et nouvel appel de recurs avec, cette fois, la valeur 1 et comme la condition $i > 0$ est toujours satisfaite, appel de recurs avec la valeur 0. Cette fois la condition n'est pas satisfaite ; recurs se termine donc et affiche la valeur de i (0). L'adresse de retour est alors dépilée et l'on revient à la procédure précédente, celle où l'environnement de recurs était $i=1$. On affiche i et ainsi de suite jusqu'au dernier environnement qui fournit l'adresse de retour de la procédure main, le programme se termine.

Le diagramme suivant sera peut-être plus clair que mes fumeuses explications :



Dans notre exemple, la procédure recurs contient une référence explicite à elle-même. On dit qu'elle est **directement** récursive.

Si elle faisait appel à une procédure machin faisant elle-même appel à recurs, on parlerait alors d'une procédure **indirectement** récursive.

Terminaisons

De même que les structures répétitives, les structures récursives présentent le risque de calculs infinis. Il est indispensable de vérifier que les conditions de terminaison sont convenablement remplies. Comme dans notre exemple, une clause conditionnelle doit être introduite de telle sorte que l'exécution puisse se faire sans récursivité.

La technique courante consiste à

- 1/ définir une variable x telle que $x \leq 0$ implique l'arrêt de la récursivité
- 2/ prouver que x décroît à chaque appel

Dans la pratique, il faut également montrer que non seulement le nombre d'appels est fini, mais qu'il est également faible. Songez qu'à chaque appel, il faut sauvegarder les variables et l'état courant du calcul.

Une fonction utile

Dans sa merveilleuse simplicité, le programme précédent présentait l'avantage absolu de ne rien faire. Nous allons maintenant aller plus loin dans l'utile et créer la fonction **power** à peu près équivalente à la fonction **pow**, disponible dans toutes les bonnes stdlib mais pas dans celle de l'Apple.

syntaxe $x^n \rightarrow$ `int power(int x,int n);`

La fonction d'élevation à la puissance est définie par

$$x^n = x \times x^{n-1} \text{ avec } x^0 = 1$$

donc x à la puissance n est égal au produit de x par x à la puissance $n-1$
avec x puissance $0 = 1$

Peut-on trouver définition plus récursive...?

```
/* définition récursive de la fonction power */  
  
int power(x,n)  
int x,n;  
{  
    if (n==0) return 1;  
    else return ( x * power(x,n-1));  
}  
  
main()  
{  
    printf("%d",power(2,2));  
}
```

Vous pouvez constater que l'algorithme suit de très près la définition récursive de l'élevation à la puissance. Vous pouvez vous amuser à écrire vous-même une fonction factorielle où factorielle n ($n!$) est le produit des n premiers nombres entiers : la factorielle de 5 est $5! = 5 \times 4 \times 3 \times 2 \times 1$ définie par

$$n! = n \times (n-1)! \text{ avec } 0! = 1$$

Quand ne pas utiliser la récursivité

Les exemples que nous avons rencontrés jusqu'à maintenant ne sont pas typiquement récursifs et l'on peut leur trouver aisément des solutions itératives. En fait, en raison du coût élevé en mémoire et en temps de calcul de la récursivité, chaque fois qu'une solution itérative est envisageable sans complication excessive, elle doit être choisie.

A titre d'exercice, cherchez la forme itérative de la fonction **power** et factorielle.

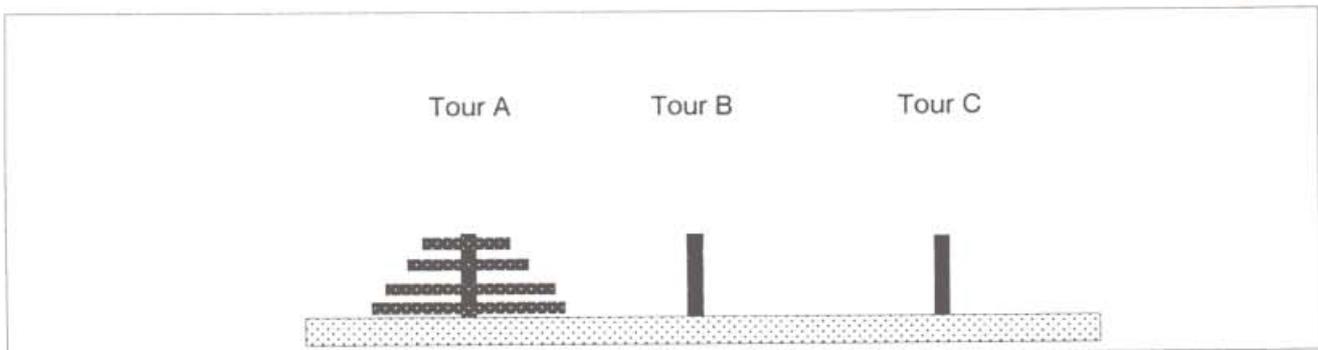
Quand faut il utiliser la récursivité ?

Que ma réflexion ci-dessus ne vous dissuade pas d'utiliser la récursivité : il y a beaucoup de bonnes applications de celle-ci. Les algorithmes qui sont récursifs par nature doivent être exprimés sous forme de procédures récursives ; de même, les procédures manipulant des données récursives, telles les arbres, seront de préférence codées en mode récursif.

Enfin, choisissez la solution récursive quand elle est plus simple que la solution itérative (cherchez par exemple la solution du problème posé ci-dessous en basic...)

Un problème classique

Un vrai casse-tête itératif est fourni par le problème bien connu des «TOURS DE HANOI». Il s'agit d'un jeu où l'on dispose de trois piquets susceptibles de recevoir des disques empilés. La position de départ peut être représentée de la façon suivante :



```
void hanoi(n,tour_départ,tour_tempo,tour_arrivée)
int n;
char tour_départ,tour_tempo,tour_arrivée;

{
if(n>0){
/* Transfert de n-1 disques de A à B en passant par C*/
hanoi(n-1,tour_départ,tour_arrivée,tour_tempo);

printf("Déplacer le disque %d de %c à %c\n",n,tour_départ,tour_arrivée);

/* Transfert de n-1 disques de B à C en passant par A */
hanoi(n-1,tour_tempo,tour_départ,tour_arrivée);
}

}
main()
{
int n;
printf("Combien de disques :");
scanf("%d",&n);
hanoi(n,'A','B','C');
}
```

Exemple de sortie écran :

- Combien de disques : 3
- Déplacer le disque 1 de A à C
- Déplacer le disque 2 de A à B
- Déplacer le disque 1 de C à B
- Déplacer le disque 3 de A à C
- Déplacer le disque 1 de B à A
- Déplacer le disque 2 de B à C
- Déplacer le disque 1 de A à C

Etonnant non?

le jeu consiste à faire passer les disques de la tour numéro A à la tour numéro C en utilisant la tour B de telle sorte qu'à aucun moment un disque ne soit empilé sur un autre de plus petite dimension.

La solution réursive est très simple :

- (1) Si on dispose d'un seul disque, la solution est immédiate : on transfère de A à C.
- (2) Pour résoudre le problème posé avec n disques, on peut supposer que $n-1$ disques ont été transférés sur la tour B et qu'il reste le plus grand disque sur la tour A.
- (3) Il est alors possible de transférer directement ce disque sur la tour C.
- (4) Il ne reste plus qu'à transférer les $n-1$ disques de la tour numéro B à la tour C en utilisant la tour A comme stockage temporaire.
- (5) Le même raisonnement est possible avec une tour comportant $n-1$ disques au départ.
- (6) On peut ainsi revenir au cas où un seul disque est à transférer, ce que nous savons faire (1)

/ LES LIVRES /

• **CALCUL NUMÉRIQUE
SUR PC, PS ET COMP.**
**Applications en GW-Basic
et Turbo-Pascal**
(Claude Nowakowski)

L'auteur, Ingénieur CNAM, docteur-Ingénieur (LIMSI-Orsay), découvre l'informatique lors d'un stage en 1962, à la SETI (filiale de la Cie des Compteurs). Il est particulièrement intéressé par le calcul scientifique : la simulation, la CAO... et passionné par la micro-informatique.

Thème : Aujourd'hui, beaucoup de logiciels sont disponibles pour des applications à caractère technique ou scientifique. Cependant, dans de nombreux cas, il est encore nécessaire de développer soi-même un programme de calcul numérique pour résoudre des problèmes précis.

Pour un étudiant, la programmation d'algorithmes de calcul numérique rend plus concret l'apprentissage des mathématiques et, bien souvent, le facilite. L'utilisation de langages relativement simples comme Basic et Pascal autorise une programmation assez immédiate.

Description : Ce livre apprend à résoudre un grand nombre de problèmes de calcul numérique en mettant en œuvre des théories mathématiques variées et en y associant un algo-

rithme spécifique. Les thèmes abordés par l'auteur sont : le calcul matriciel, l'interpolation, l'approximation de fonctions, les équations non linéaires, le calcul polynomial, le calcul intégral et la résolution d'équations différentielles.

Dans chaque cas, l'auteur expose : les principes généraux, un énoncé d'exercices, le type de solution algorithmique, le résumé du programme dans un pseudo-code, le programme en Basic et en Pascal. L'auteur précise également la liste des modules à prendre pour qui utilise Turbo-Pascal et sa bibliothèque mathématique. Pour tirer profit de ce livre, il faut être initié à Basic et à Pascal et avoir un bon niveau en mathématiques (supérieures ou spéciales).

Renseignements au public :

EDITIONS P.S.I. 320 pages — 235 F
6-10, boulevard Jourdan, 75014 PARIS
Tél. : (16-1) 45.65.06.06.

• **LE MODE
RELATIONNEL BINAIRE**
Méthode I.A. (NIAM)
(H. Habrias)

Voici le premier ouvrage en langue française consacré au modèle relationnel binaire utilisé en conception de systèmes d'information et basé sur le langage naturel.

Le modèle élaboré par le Professeur NIJSEN, est mondialement

connu sous le nom de la méthode qui l'utilise : I.A. (Information Analysis) ou N.I.A.M. (Nijssen Information Analysis Method).

Bien que peu diffusée en France, la méthode NIAM est déjà employée dans quelques grands Groupes Français.

Conçu comme un cours de base sur la modélisation conceptuelle et comme un manuel pour utilisateurs du modèle binaire, tous les termes mathématiques et spécifiques sont expliqués et le modèle relationnel n-aire est longuement étudié. De nombreux exercices et des études de cas facilitent la compréhension des lecteurs.

Enfin, *Le Mode Relationnel Binaire* regroupe les différentes méthodes de conception et compare les différents formalismes binaires et n-aires.

Henri HABRIAS possède un D.E.S. de sciences économiques. Diplômé de l'I.A.E., il est titulaire d'un doctorat en "informatique des organisations" complété par une licence de psychologie et de certificats de linguistique générale et d'histoire des sciences.

Maître de Conférence à l'I.U.T. de Nantes, il anime le groupe de travail des Utilisateurs Européens de l'approche Entité-Association. Il est rédacteur de la "Lettre de l'Adéli".

EYROLLES 320 pages — 380 F
Editions-Librairie
61, boulevard Saint-Germain, 75240 PARIS
CEDEX 05 Tél. : (16-1) 46.34.21.99.

● **ASSEMBLEUR 8086 à 80386
ANALYSE
ET PROGRAMMATION
STRUCTURÉE AVEC
EXERCICES RÉSOLUS**
(Gilles Louise)

Thème : On parle souvent de méthodes de programmation pour des langages évolués comme Pascal ou C. On montre qu'il est souhaitable de ne pas produire du code n'importe comment mais qu'il faut au contraire concevoir des programmes modulaires, structurés, faciles à lire et à modifier. Ces conseils s'appliquent aussi à la programmation en assembleur. Les méthodes d'analyse et de programmation trouvent parfaitement leur justification avec ce type de langage. Ceci est d'autant plus vrai que l'intérêt de la programmation en assembleur est de fournir du code très rapide à exécuter, donc aussi fiable que possible.

Description : Ce livre s'adresse à ceux qui ont déjà fait leurs premiers pas en assembleur 8086. Son but est de montrer comment réduire les erreurs de programmation et vous donner, de page en page, les moyens d'accroître la productivité de votre travail ou celle d'une équipe de programmeurs. L'ouvrage vous livre d'abord d'utiles rappels sur les registres internes et le jeu d'instructions du 8086, choisi pour son intérêt pédagogique. Vous trouvez ensuite une présentation claire et détaillée des éléments de base d'une programmation structurée. Pour concrétiser votre apprentissage, l'auteur a prévu pour vous plusieurs cas d'analyse et de programmation. Vous débuterez par des algorithmes simples comme à la recherche d'un mot dans une table. Vous continuerez à l'aide d'algorithmes numériques et de tri et vous terminerez en essayant de résoudre des problèmes de récursivité et d'analyse combinatoire.

Voici un livre qui convient parfaitement aux étudiants en BTS d'informatique, Maîtrise, Ecoles d'Ingénieurs.

272 pages — 195 F

Renseignements au public :

Editions P.S.I.

6-10, boulevard Jourdan — 75014 PARIS
Tél. : (16-1) 45.65.06.06.

● **PAGEMAKER EFFICACE
SUR MACINTOSH**
(Philippe Mathieu)

L'auteur est professeur d'informatique dans une école d'ingénieurs en agriculture. Il utilise régulièrement PageMaker.

Description : En deux ou trois jours, on apprend à se servir de PageMaker sur Macintosh. Après plusieurs mois d'usage, il n'est pas rare qu'un utilisateur ne connaisse et n'exploite véritablement que la moitié des possibilités de ce logiciel de mise en page et pas toujours de manière optimale. Cet ouvrage vous dévoilera tous les aspects méconnus de PageMaker. Il vous permettra de découvrir les mille et une "astuces" qui facilitent et accélèrent son utilisation.

Ce livre est découpé en douze chapitres traitant chacun d'un thème différent. Le lecteur peut ainsi approfondir ses connaissances au fur et à mesure de ses besoins, en consultant à chaque fois les rubriques qui l'intéressent. Avec PageMaker sur Macintosh, vous apprendrez ainsi à configurer une page, vous servir de l'éditeur de textes, placer un texte dans un cadre, exporter un texte, modifier et assembler des graphiques, placer un graphique dans un cadre, ajuster votre texte, modifier l'ordre des pages de votre publication...

PageMaker n'est souvent utilisé qu'à 30% de ses capacités et on peut découvrir peu à peu toutes les facettes méconnues de ce logiciel. En donnant des astuces d'utilisation, ce livre fait gagner un temps précieux.

Renseignements au public :

Editions P.S.I. 176 pages — 185 F

● **TURBO BASIC PAS À PAS**
(A. Garcia et J.-M. Gaudin)

Auteurs : Augustin Garcia a écrit "Clefs pour Basic 2", le livre de "Word 3 sur PC" et a traduit "Flight Simulator copilot" et "Voler avec Flight Simulator".

Jean-Michel Gaudin est inspecteur à l'Education Nationale. Il est venu tout seul à l'informatique et a fondé deux associations d'initiation à l'informatique. Il a personnellement testé la méthode d'apprentissage dans "Turbo Basic pas à pas". Il a écrit "Turbo Pascal pas à pas" version 3 et 4 ainsi que "Turbo C pas à pas".

Thème : Turbo-Basic est un Basic créé par Borland qui, comme tous les produits de cette firme, présente de nombreuses caractéristiques attractives : prix, rapidité, simplicité d'utilisation, compatibilité avec GW-Basic, utilisation de programmation structurée. Ce langage est donc amené à rencontrer un grand succès.

Description : La philosophie de cet ouvrage, comme de tous ceux qui portent le titre "pas à pas", est d'amener l'utilisateur débutant à une connaissance des principales instructions et règles de programmation d'un langage, en le guidant complètement dans l'élaboration d'un programme. Le premier programme proposé comporte 6 lignes et présente 4 instructions. Il est peu à peu étoffé au fur et à mesure que l'on étudie de nouvelles instructions pour arriver à une taille de 3 pages. Deux programmes sont ainsi développés dans ce livre et mettent en évidence les principales instructions et les aspects les plus importants de Turbo Basic.

Cet ouvrage, proposé par les Editions P.S.I., est le seul s'adressant aux débutants en Turbo Basic.

Renseignements au public :

Editions P.S.I. 128 pages — 105 F

● **VENTURA PAR LA PRATIQUE**
(Matthew Holtz)

Ce livre s'adresse à la fois, à l'utilisateur confirmé et au néophyte de la PAO. Ce dernier y trouvera le descriptif des mécanismes de Ventura tandis que l'utilisateur averti pourra y examiner les nombreux styles et formats disponibles. Il y découvrira aussi les explications relatives à l'utilisation de Ventura avec d'autres logiciels. A l'aide de nombreux exemples concrets, l'ouvrage étudie tous les menus et dialogues. Tous les exemples sont présentés de manière progressive pour suivre l'apprentissage de l'utilisateur.

Voici, entre autres, les principaux sujets abordés :

La création d'un bulletin, l'utilisation des fonctions ciseaux et colle électroniques, la construction de blocs pour le formatage... etc.

Les nombreux tableaux et figures sont accompagnés d'explications détaillées afin d'en faciliter la compréhension.

620 pages — 328 F

SYBEX, 6-8 Impasse du Curé, 75018 PARIS
Tél. : (16-1) 42.03.95.95

APPLE EXPO 88

APPLE, le GS et NOUS

Après avoir passé quatre jours à Apple Expo, il me faut essayer de vous faire part de ce que j'y ai vu et entendu. *Tremplin Micro* étant une revue dédiée à la gamme Apple II, vous ne vous attendez pas à ce que je vous parle des multiples Macintosh que j'ai pu voir alimentés de programmes plus performants les uns que les autres. C'est entendu, le Mac est une très belle machine, qui attire de très bons programmeurs et leur permet d'afficher de très bons résultats économiques.

Mais ce que je cherchais, ce sont des informations sur l'avenir de nos machines.

Les propriétaires de II+, IIe ou IIc sont fixés depuis plusieurs mois. Leurs appareils, qui n'ont plus de vie commerciale, ne semblent pas avoir perdu pour autant l'affection de leurs propriétaires. Ce n'est certainement pas M. Brésard, le directeur de Bréjoux, qui me contredira. Il était extrêmement difficile de l'approcher en raison de l'affluence enregistrée autour de son stand pourtant installé hors de la zone privilégiée que constituait le rez-de-chaussée de la Grande Halle.

Côté GS...

Côté GS, la situation m'a paru beaucoup moins claire. Les journalistes du Figaro et tous ceux, nombreux, qui étaient présents à la conférence de J. Sculley ont cru comprendre qu'Apple décidait d'interrompre la carrière commerciale du GS en Europe (c'était le mercredi 28). Le jeudi 29, je rencontre, en compagnie de l'ami Emile Schwarz, le directeur du marketing d'Apple France et nous entendons alors des propos peu encourageants quant à l'avenir commercial de la machine. Il nous a, naturellement, été promis que les utilisateurs ne seraient pas abandonnés par la marque à laquelle ils ont fait confiance. En fin d'après-midi, c'est le nouveau P.D.G., Gian

Carlo Zanni, qui accepte de me consacrer une heure, ce dont je dois le remercier.

Là encore, je plaide la cause de notre chère machine, rappelant que nous sommes nombreux à penser que le relatif échec de l'introduction du GS sur le marché français est dû, au moins en partie, au peu d'enthousiasme manifesté aussi bien aux Ulis que dans le réseau. Je rappelle que l'attitude des utilisateurs des machines Apple II de tous types est très singulière et entre difficilement dans des critères mercantiles.

Vous semblez être attachés à votre engin par un lien quasi-amoureux difficile à appréhender à l'aide d'un tableur, aussi puissant soit-il.

Durant cet entretien, il m'a semblé une fois encore, que la décision de J. Sculley était irrévocable. Les choix restant ouverts concernent la définition de ce qu'il sera possible de faire en matière de support du parc existant.

Et le GS.OS ?

Le nouveau système d'exploitation que nous attendons tous, GS.OS, sera francisé et distribué en Janvier 89. Il faut préciser à ce sujet, que la "nouvelle" Rom est INDISPENSABLE et que ceux d'entre vous qui n'ont pas encore pu ou voulu faire la mise à jour sont cordialement mais fermement invités à la réaliser.

Si votre concessionnaire se montre "réticent", écrivez sans hésiter chez APPLE aux Ulis (à l'attention de M. Pivert par exemple). Apple a diffusé suffisamment d'informations dans le réseau pour que celui-ci ne puisse faire mine d'ignorer les conditions de cette mise à niveau. D'autre part, les heureux possesseurs d'un ou plusieurs périphériques SCSI devront remplacer la Rom de leur interface susnommée par la nouvelle version connue sous l'appellation Rom C. Celle-ci est, en effet, indispensable au fonc-

tionnement normal de GS.OS. Ne comptez pas sur ce compte-rendu pour une présentation détaillée de ce dernier. Je n'ai eu que des contacts brefs et superficiels. Je sais que l'on introduit un nouveau mode d'organisation des disquettes (pas de panique, c'est quasiment transparent pour l'utilisateur) et que les relations avec le monde extérieur passent désormais par des programmes que nos amis américains appellent des *Drivers*.

Il semble possible d'en inférer que nous disposerons un jour de "drivers" pour format Mac ou MS.DOS (eh oui ! j'ai réussi à éviter d'écrire trois lettres qui me donnent des boutons !). J'ai remarqué la présence, au sein même de GS.OS, d'une option permettant à tout le monde d'effectuer, en toute connaissance de cause, un redémarrage à froid reconfigurant éventuellement un possible Ram Disque. Adieu donc à mon bon vieux Freezer !

Ça suffit pour le moment en ce qui concerne GS.OS.

Finesses de langage

Vendredi, j'ai une nouvelle fois rencontré mes deux interlocuteurs réunis à cette occasion. Nous venions de lire dans *Décision Informatique* un démenti des propos précédemment attribués à J. Sculley. Nous sommes apparemment en présence de finesse de langage qui m'échappent. Peu enclin à me laisser aller à un optimisme débridé, j'ai insisté lourdement, pour qu'Apple permette au moins aux acquéreurs du GS actuel de pouvoir mettre à jour leur machine lorsque la nouvelle carte mère, tant attendue, apparaîtra.

Là encore, on est dans le flou. Certains parlent de quelques semaines, J.-L. Gassée parle de 18 mois. Il n'est pas évident que ce soit incompatible. La carte décrite un peu partout (un clin d'œil en passant à nos amis de *Tilt*) est une évolution importante, intéressante mais ne remettant pas en cause l'image même du GS. Plus de rapidité, plus de définition, peut-être même plus de Rom, c'est bien, mais ce n'est pas la révolution.

Je tiens à être clair : si cette carte voit le jour, il nous (vous) la faudra car il s'agit d'une évolution

logique. En ce qui concerne la nouvelle machine évoquée par J.-L. Gassée, il ne faut rien brusquer. Certains, qui se disent bien informés, annoncent que la compatibilité actuellement presque totale avec l'ancienne gamme serait malmenée, pour ne pas dire abandonnée.

Dans ce cas, je m'avoue perplexe. Quelques-uns vont même jusqu'à envisager une carte mère équipée d'un 68000, ce qui, à première vue, ressemblerait bigrement à un petit Mac. C'est peut-être dans la logique des choix quasi-mystiques d'Apple, mais il ne me semble pas évident que ce soit une réponse à l'attente des utilisateurs. Il est bon parfois de créer des produits donnant naissance à des besoins nouveaux, mais lorsque d'anciens existent, pourquoi refuser de les satisfaire alors qu'on en a les moyens ?

Même aux Etats-Unis

Même aux Etats-Unis, la grogne semble monter chez de nombreux auteurs (cf. *Open Apple* par exemple) qui, constatant que la machine est fort bien accueillie en tant que super *Ile*, supportent mal que l'on cherche à les pousser vers le "look" Mac qui ne les passionne guère. Les avatars du Basic System semblent confirmer que les développements sur l'écran texte ne sont pas en odeur de sainteté à Cupertino.

Comme vous le voyez, ce n'est pas l'euphorie de 87. Et pourtant, il me semble, que tout pourrait redémarrer aisément. Supposons un instant que ceux d'entre vous qui souhaitent une amélioration des performances de leur vieux *Ile* ou *Ilc* fassent des comptes détaillés. Transwarp + RamFactor est un doublé possible mais, n'en déplaise à Bréjoux (et à Nestor), une unité centrale GS coûte presque la même somme, et c'est tout de même autre chose ! J'entends déjà la critique relative à l'absence de programmes français. Il serait difficile de la nier mais, je ne suis pas sûr que ce soit un vrai problème. D'abord, nous ne sommes pas obligés de travailler avec des programmes de la nouvelle génération, le succès des extensions destinées à Appleworks le montre bien (un grand bravo à POM's pour la francisation des TimeOut). En outre, même avec les programmes modernes, il y a des solutions. J'utilise Multiscribe GS et GraphicWriter, deux produits bien de chez eux.

Grâce à un accessoire de bureau créé par Apple en Allemagne (haro sur nos amis des Ulis !), j'ai accès, sans aucune gymnastique, aux accentués chers à notre langue. Un petit coup de Sourceror puis de Merlin sur la bête et la voilà capable de nous donner tous les âœœ dont nous n'osions plus rêver, tellement nous avons intégré les limites de notre vieille pomme.

Que voulez-vous de plus ? Une compta imprimant les nombres à la française ? bon, vous n'avez pas tort ! mais soyez honnête un instant, même les comptas bien de chez nous se permettent parfois de recourir au point décimal anglo-saxon ! Pire encore, la Chambre des Métiers des Alpes-Maritimes n'est pas capable de mettre les accents sur les é è lorsqu'elle établit un extrait de registre des métiers ! C'est une honte, mais ça existe !

Il paraît qu'il faut que j'abrège, mais je tiens à vous rappeler que, l'an passé, après Apple Expo, mon édito faisait référence à un spectacle consacré par Mort Schumann à Jacques Brel. Cela n'a hélas pas empêché Brel de mourir aux Marquises. Mais, si 10 ans après sa mort commerciale, le GS est aussi vivant que le Grand Jacques aujourd'hui, ils auront l'air fin chez Apple !

Tremplin Micro et le GS

Ce n'est pas tout. Il n'y a pas que le GS dans la vie. Vous avez du remarquer un petit détail : *Tremplin Micro n°22* s'est fait désirer. Non, Schwarz n'était pas en grève et SIVEA Cannes ne lui a pas encore interdit l'accès à leur LaserWriter. Si grève il y a, c'est plutôt de votre côté qu'il faut chercher. Lors de mon séjour à Paris, tout le monde semblait inquiet. Il y a presque de quoi. Vous nous demandez de défendre le GS, mais vous ne faites pas l'effort de vous abonner. Or, sachez bien qu'un abonné vaut deux clients kiosque (là, je joue au comptable car côté courrier des lecteurs, ça n'en fait qu'un). Depuis que nous avons ouvert nos pages au GS, les aficionados des *Ile, Ilc* nous délaissent, mais les GSistes ne se bousculent pas. Notre stock de cartes stéréo comportait 300 unités. Comme il s'agissait d'un cadeau, je ne pouvais être trop gourmand mais, finalement, il n'y a pas eu trop de commandes. Secouez-vous donc un peu, si vous tenez à pouvoir nous lire ;

un petit effort s'il vous plaît ! On ne va pas vous proposer une souscription façon *LIBÉ* ou des actions façon *L'ÉVÈNEMENT*, mais nous avons des bulletins d'abonnement. On n'a tout de même pas acheté des quinquaux d'agrafes pour des prunes !

Nous songeons à une nouvelle formule avec photos pornos digitalisées ! Comment, qu'ouïs-je ? Ah bon, j'avais pourtant cru comprendre...

Désolé, le cahier spécial envisagé pour les abonnés se bornera à contenir des programmes GS que nous ne mettrons plus dans la revue de base qui se consacrera davantage aux racines d'Apple comme le dit si joliment J.-L. Gassée.

Bien entendu, toutes vos suggestions seront les bienvenues (en fait, presque toutes car M. Hachette ne veut pas entendre parler des MACs et je suis allergique à MS.DOS).

Et si vous êtes bien sages, il y aura une surprise pour Noël (ou le jour de l'an).

Dernière chose, si vous avez eu des difficultés à obtenir la mise à niveau de votre GS, n'hésitez pas, écrivez dès aujourd'hui aux Ulis en demandant qu'on veuille bien vous faire parvenir (en Janvier 89) le GS.OS francisé. Indiquez l'adresse de votre fournisseur habituel. Ainsi, ce dernier ne pourra pas feindre l'ignorance comme cela fut trop souvent le cas lors de l'échange de Rom.

Vous savez tous que, si j'exerce actuellement le métier de céramiste, je suis aussi Ingénieur des Ponts et Chaussées. Si Apple nous fait faux bond, je les laisse à leurs MACs et je retourne à mes grues ! **Yvan KOENIG — 10 Octobre 1988**

Dernière minute :

Dans le texte présentant mes extensions à l'accessoire ACCENT de Stephan Hadinger, je citais l'accessoire KEYUTIL diffusé par APPLE Allemagne. Je l'ai désassemblé, traduit et y ai ajouté les fonctions supplémentaires d'ACCENT. Nous tenons ainsi un accessoire très commode et performant. Deux remarques :

1. Parfaitement compatible avec le système 3.1 et GS.OS, l'accessoire a des problèmes avec le système 3.2. Il suffit de remplacer les fichiers IMAGEWRITER et PRINTER de ce système par ceux de GS.OS.
2. L'origine Apple de cet accessoire m'oblige à solliciter l'autorisation de la firme avant de le diffuser. Si elle m'est accordée, vous trouverez KEY.FR sur la disquette T.M. n°23.

Yvan KOENIG — 14 Octobre 1988

PC ORGANISER

Le tout en un de votre micro

Cherchez-vous un logiciel intégré aussi simple que possible ? Pensez par exemple au **PC ORGANISER** de MICRO APPLICATION... et vous disposerez en permanence d'un agenda, d'un traitement de texte élémentaire, d'un carnet d'adresses, d'une gestion de fiches... et d'une calculatrice. Tout en un !

Vous pourrez ou bien considérer **PC ORGANISER** comme un programme résident et il se fera alors aussi discret que possible... ou le charger lors de chaque utilisation, ce qui reste très rapide, à partir d'un disque dur.

Evidemment, la solution programme résident est de loin la plus pratique puisque **PC ORGANISER** peut alors être appelé à partir d'un autre logiciel... sauf incompatibilité, bien sûr !

On n'en finirait pas d'énumérer les avantages présentés par un tel gestionnaire :

- Compositeur téléphonique automatique.
- Horloge, calculatrice, table des codes ASCII/HEXA (pratique quand on programme).
- Traitement de texte résolument simplifié, mais néanmoins pratique (vous n'écrirez tout de même pas votre prochain bouquin en passant par lui !).
- Un système de fiches par index, avec recherche performante par index et par mot.

N'oublions pas l'alarme qui vous rappelle un rendez-vous, le bloc-notes... ou encore le fichier d'adresses.

PC ORGANISER s'offre le luxe de vous présenter des modules d'impression de lettre (commande, publi-postage).

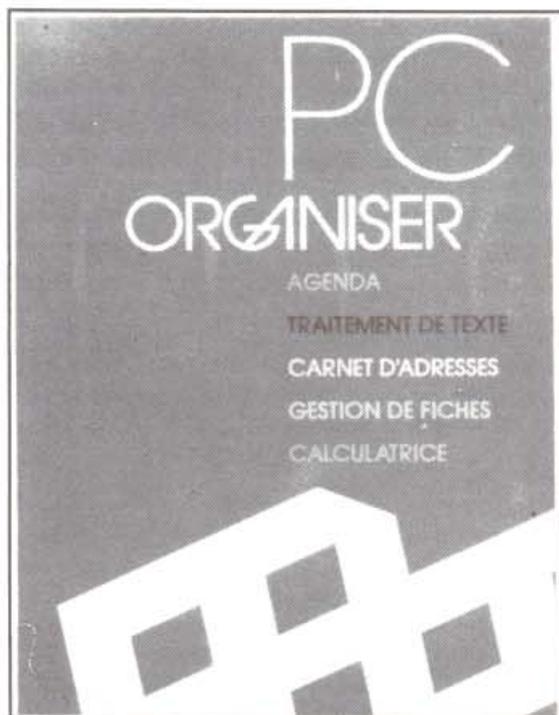
Notez enfin qu'il autorise le couple Copier/Coller à partir / ou vers d'autres programmes.

C'est amusant et terriblement pratique. Autre avantage, ce n'est pas ruineux (690 F) et pas davantage protégé. Un bon petit logiciel intégré.

Clément RENARD

MICRO APPLICATION

58, rue du Faubourg Poissonnière 75010 PARIS.



Des nouvelles du front

Dans Tremplin Micro n°20 :

- Page 31, ligne 10, il fallait lire :
10 E\$ = CHR\$ (27) :a = 19:P\$ = "/T20/FONTES/FAPL"
- Page 33, ligne 62120, il fallait lire :
62120 IF PEEK(1529) 128

Emile SCHWARZ

InstaDump

Recopie d'écran-texte sur Apple IIe et IIc

La gamme Apple II, même si elle ne comporte plus — actuellement — qu'un seul modèle, est encore bien vivante. Vos lettres nous le rappelleraient s'il en était besoin. J'ai saisi l'occasion de la publication d'un accessoire de bureau GS permettant la recopie d'écran texte, pour sortir de mes tiroirs une commande externe ProDOS effectuant le même travail sur les IIe et IIc, dans des conditions de confort comparables.

Il vous faudra, indépendamment de l'unité centrale, s'il s'agit d'un IIe, une carte interface souris (et, si je ne m'abuse, des roms mises à niveau).

J'utilise des informations déjà publiées dans *Tremplin Micro* et je ne m'y attarderai donc pas outre mesure.

L'interface souris possède, entre autres, la faculté de pouvoir générer une interruption tous les 1/60^e de seconde. Vous profiterez de cette caractéristique pour scruter régulièrement le clavier. Si celui-ci reçoit la séquence PommeFermée + H, la machine effectuera une recopie papier de l'écran en service.

Pour ce faire, il vous faut :

- Implanter la commande INSTON/INSTOF. Vous utiliserez une fois encore, ce bon vieux RELOPRO.S (*T.M.* n°20 page 45...). Rappelons qu'il y avait une erreur dans la ligne 313. Il fallait lire : ADC R__OFFSET+1.
- Activer la routine de scrutation par PRINT CHR\$(4) "INSTON". La syntaxe de la commande est analysée par BASIC.SYSTEM et par la routine PARSE du source figurant sur la disquette. Ensuite, le contenu des connecteurs (slots) est étudié afin de rechercher l'interface souris requise. N'oubliez pas qu'elle est baladeuse par nature sur le IIe, mais, qu'Apple s'est amusé à la déplacer du slot4 vers le slot7 sur les dernières versions du IIc. Le module de recherche vous a été proposé à deux re-

prises dans *Tremplin Micro* n°5 (sous la plume de C. Aubry) et dans *T.M.* n°14 (sous celle de F. Gallet). Après l'avoir localisée, vous initialiserez l'interface souris en mode interruption par rafraîchissement écran.

Désormais, tous les soixantièmes de seconde (ou presque car, pendant les accès disques, les interruptions sont interdites), vous exécuterez la routine INTERUP qui se chargera d'appeler la routine DO__DUMP si PommeFermée + (H) a été tapée.

Cette routine DO__DUMP, simple dans son principe, se complique du fait de la multiplicité des types d'écrans-texte générés sur nos machines.

Dans l'ordre chronologique, nous avons eu : l'écran 40 colonnes avec mode INVERSE et FLASH, mais SANS minuscules inversées ; ensuite, Apple a introduit le FirmwareVidéo du slot3 et son Mode Nouveau (Alternate Mode) qui peut être utilisé en 40 ou 80 colonnes, mais ne possède plus FLASH ; il permet cependant d'afficher des minuscules inversées. Enfin, avec le IIc, puis avec les nouvelles roms du IIe, sont arrivées les Icônes (MouseText).

DO__DUMP identifie le mode en service et transcrit le contenu de l'écran en conséquence.

Il restait un problème délicat lié aux différents modèles d'imprimantes connectées à nos Apple II. Le programme publié correspond à la configuration majoritaire, à savoir imprimante série IMW ou IMW2.

Les IMW2 disposent d'origine de la faculté de

reproduire directement les icônes mais, comme ce n'est pas le cas des IMW1, j'ai introduit un module qui les reproduit par le biais des caractères téléchargés. Pour les caractères INVERSE(S), je n'ai pas retenu les caractères téléchargés (C. Aubry l'avait fait dans le bureau de MINIE) parce qu'à mon goût, cela impliquait une utilisation trop importante de l'espace mémoire.

Il faut noter que la formule Icônes téléchargées présente un inconvénient non négligeable. Toute fonte de caractères préalablement chargée sera détruite. Les possesseurs d'IMW2 auront donc tout intérêt à modifier DO_DUMP pour utiliser les icônes internes de leur imprimante. Ils pourront consulter le source de l'accessoire de bureau GS pour y récupérer les informations qui pourraient leur faire défaut (comme quoi les programmes GS peuvent être utiles aux utilisateurs des vieilles machines).

Lorsque vous n'aurez plus besoin de la fonction recopie d'écran (pour utiliser APPLEWORKS par exemple), n'oubliez pas de déconnecter la routine d'interruption par :

PRINT CHR\$(4)" INSTOF"
ou, plus cavalièrement, par Ctrl RESET.

L'installation, un tantinet fruste de la routine d'interruption, la rendrait *aléatoire* sur un GS qui manipule des interruptions à tour de bras, mais il n'y a aucune raison d'utiliser INSTON/INSTOF sur un GS puisque l'on a désormais le CDA TextScreenDump.

Rappelons une énième fois que le source de mes commandes externes est composé de trois fichiers MERLIN à savoir ici :
K.INSTH.S, INSTH.S et RELOPRO.S.

Comme toujours, je conseille d'installer la commande par un programme de démarrage appelant éventuellement PRO.FP (cf. T.M. n°20) pour éviter des chargements à répétition du programme *actif*.

Exemple de démarrage :

```
10 D$=chr$(4) : print D$"-PRO.FP"
20 print D$"-INSTH"
30 print D$"-une autre commande ou plus si nécessaire"
90 print D$"-MonVraiProgramme"
```

Vous trouverez sur la disquette les sources K.INSTH.S, INSTH.S, ainsi que la démo en Basic. Pour RELOPRO.S et PRO.FP, reportez-vous aux références citées.

Pour les amateurs de saisie, voici le listage de la démo et des routines en Hexa de INSTH.

Yvan KOENIG — 11 Octobre 1988

START . UP

```
10 D$ = CHR$ (4)           83A4
20 PRINT D$"-PRO.FP"      9248
30 PRINT D$"-INSTH"      7819
40 PRINT D$"-DEMO.INSTH"  A56C
```

DEMO . INSTH

```
10 D$ = CHR$ (4): PRINT D$"INSTON"  991F
15 FOR YK = 1 TO 20                7449
20 TEXT : HOME : PRINT D$"PRÉ3": PRINT 1CE6
25 GOSUB 620                        1D48
60 FOR I = 64 TO 95: INVERSE : PRINT CHR
  $ (27) CHR$ (I) CHR$ (24):: NORMAL : P
  RINT " ";: NEXT : PRINT           D114
70 GOSUB 670                        3B4D
110 PRINT CHR$ (17)                 AA5A
120 GOSUB 620                        1D48
160 FOR I = 64 TO 95: INVERSE : PRINT CHR
```

```
$ (27) CHR$ (I) CHR$ (24):: NORMAL : P
RINT " ";: NEXT : PRINT           D114
170 GOSUB 670                       3B4D
210 PRINT CHR$ (21)                  A455
220 GOSUB 620                        1D48
260 FLASH : GOSUB 500: NORMAL : PRINT A7E9
270 GOSUB 670                        3B4D
280 NEXT YK                           AB26
380 HOME : VTAB 10: PRINT "N'oubliez pas d
  e déconnecter la commande": VTAB 12: H
  TAB 9: PRINT "par INSTOF ou Ctrl-Reset
  ": VTAB 14: HTAB 6: PRINT "si vous n'e
  n avez plus besoin."             CA78
400 END                               8188
490 REM "Routines
500 PRINT "à A B C D E F G H I J K L M N O
  P Q R S T U V W X Y Z " ç s ' _"  D0CE
510 RETURN                            6381
610 :                                  883A
620 GOSUB 500                          1845
630 PRINT : PRINT                     15AE
```

640 INVERSE : GOSUB 500: NORMAL
 650 PRINT : PRINT
 660 RETURN
 665 :
 670 VTAB 18: PRINT "Pomme Fermée- H = Reco

F8F4 pie écran": VTAB 24: PRINT "-----
 15AE Ligne basse -----"; VTAB 28
 63B1 680 FOR KY = 1 TO 1500: NEXT KY
 003A 690 HOME
 700 RETURN

E357
 3F8D
 2F97
 63B1

SDUMP/INSTH,A#6000,L#0558

6080:	A9 04 20 F5 BE 90 65 A0 16 B9 95 60 20 ED FD 88 10 F7 4C 00 BE 8D 87 C5 C3 C1 CC D0 A0 C5 C4 A8	D33E
60A0:	D3 C1 D0 A0 C1 A0 D9 A7 CE A0 CC C9 04 15 04 2A 85 14 00 2A 16 15 04 2A 15 15 00 2A 04 15 04 2A	6276
60C0:	15 14 00 28 04 15 04 2A 15 15 00 2A 15 15 00 2A 15 15 04 2A 05 15 00 2A 05 15 04 2A	9E98
60E0:	15 14 00 28 05 15 04 2A 15 14 00 28 85 43 AE 00 BE 8E 85 62 8D 00 BE AE 07 BE 8E 04 62 A9 00 85	AF13
6100:	3C 85 04 A9 62 85 3D 85 05 A9 4A 85 3E A9 64 85 3F A9 D7 85 86 A9 65 85 87 D8 A9 00 85 42 8D 07	7524
6120:	BE 38 E5 04 85 02 A5 43 E5 05 85 03 A0 00 B1 3C C9 20 D0 17 C8 B1 3C D0 12 C8 B1 3C C9 BF D0 88	37CC
6140:	A0 00 A2 02 20 C3 61 A9 02 D0 16 A0 00 B1 3C 48 29 03 A8 68 4A 4A AA BD AC 60 88 30 84 4A 4A D0	9A51
6160:	F9 29 03 AA A0 00 E0 02 90 05 C8 20 9C 61 88 20 C3 61 90 88 90 12 A9 4E 85 3E A9 64 85 3F 20 9C	FBC8
6180:	61 A2 01 20 C3 61 90 F6 90 00 A5 86 85 3E A5 07 85 3F 20 2C FE 20 CE 61 60 6C FD 00 A5 86 D1 3C	EE61
61A0:	C8 A5 07 F1 3C 90 1A 88 A5 04 D1 3C C8 A5 05 F1 3C 80 8E 88 B1 3C 65 02 91 3C C8 B1 3C 65 03 91	843D
61C0:	3C 88 60 B1 3C 91 42 20 B4 FC CA 10 F6 60 EA A5 05 18 65 03 AA 18 69 84 85 43 8A 48 4A 4A 4A	2F47
61E0:	8A 29 07 AA A9 00 38 6A CA 10 FC 19 58 BF 99 58 BF 68 AA E8 E4 43 90 E2 60 00 00 00 00 00 00	EA58
6200:	4C 1E 62 4C 9E BE 4C 58 FF D0 FB A0 00 89 C5 65 20 ED FD C8 C0 11 90 F5 20 B7 00 F0 E9 EA A2 05	9CCE
6220:	8E 52 BE AD 05 02 29 DF 49 CE F0 85 C9 88 38 D0 D2 8D 5D 62 BD FF 01 29 DF 5D C4 65 38 D0 C4 CA	8B3E
6240:	D0 F2 8E 53 BE 8E 54 BE 8E 55 BE A9 58 8D 50 BE AD 06 63 8D 51 BE 18 60 78 20 C0 62 A9 00 F0 23	80DE
6260:	AD D7 65 F8 44 AE 80 62 BD 38 07 29 F7 20 AC 62 AD D7 65 8D FF 03 AD D6 65 8D FE 03 A9 00 8D D7	4722
6280:	65 F0 26 AD D7 65 D0 8C AD FE 03 AE FF 03 8D D6 65 8E D7 65 A9 FB AE 06 63 8D FE 03 8E FF 03 A0	6FA9
62A0:	19 20 AE 62 A9 89 20 AC 62 58 18 60 A0 12 BE 00 C4 8E BD 62 A0 40 AE 80 62 8E BE 62 20 00 00 60	41A8
62C0:	A2 07 A9 00 85 86 A9 C8 85 07 C6 07 CA 30 1E A0 8C B1 06 C9 20 D0 F3 A0 FB B1 06 C9 D6 D0 EB A5	2F1F
62E0:	07 8D B0 62 0A 0A 0A 8A 8D B5 62 D0 D2 A0 EB 89 C5 64 20 ED FD C8 D0 F7 4C D0 03 D8 78 48 8A 48	3FA3
6300:	98 48 A0 13 20 AE 62 2C 62 C0 10 0A 20 1D 63 C9 C8 D0 83 20 29 63 68 A8 68 AA 68 58 40 2C 10 C0	52F9
6320:	AD 00 C0 10 FB 2C 10 C0 60 A9 01 8D 4C 63 89 C0 8D 55 63 8D 93 63 8A 0A 0A 69 8A 8D 99 63 A5	5694
6340:	24 48 A5 25 48 A5 36 48 A5 37 48 A9 01 20 95 FE 20 8E FD AD 11 C1 C9 85 D0 08 AC 4C 63 A9 89 99	9218
6360:	F8 05 A5 00 48 A5 01 48 AD 48 64 85 80 AD 4C 64 85 81 A0 00 B1 00 20 ED FD A5 00 CD 4D 64 A5 01	77C8
6380:	ED 4E 64 E6 00 D0 02 E6 01 90 E7 68 85 01 68 85 00 AD 11 C1 C9 85 D0 03 8C 9A C0 84 25 2C B3 FB	F399
63A0:	38 40 2C 1F C0 10 24 20 3A 64 8D 55 C0 20 04 64 20 04 64 C8 C0 28 D0 F2 20 41 64 90 EA 68 85 37	3AF4
63C0:	68 85 36 68 20 58 FB 68 85 24 60 2C 1E C0 10 12 20 3A 64 20 04 64 C8 C0 28 D0 F8 20 41 64 90 F0	A7A1
63E0:	B0 DB 20 3A 64 20 F4 63 C8 C0 28 D0 F8 20 41 64 90 F0 80 C9 B1 28 30 3F 29 3F C9 20 B0 02 09 40	838A
6400:	89 80 D0 24 B1 28 8D 54 C0 30 2C C9 20 90 17 C9 40 90 15 C9 60 80 11 8D 53 64 A2 F7 8D 58 63 20	55F8
6420:	ED FD E8 D0 F7 60 69 40 8D 58 64 A2 F9 BD 5F 63 20 ED FD E8 D0 F7 60 4C ED FD A0 00 A5 25 4C C1	5FCE
6440:	FB 20 BE FD A5 25 C9 17 E6 25 60 5F 64 AF 65 00 00 1B 27 00 1B 24 00 00 00 1B 58 00 1B 59 00 89	DA83
6460:	B2 B5 B4 CE 1B 21 00 00 00 89 5A 1B 2D 1B 49 40 48 38 FC FC 7A FD FC 48 00 41 48 38 C4 84 42 85	CB77
6480:	B4 48 00 42 48 00 FC 78 30 60 40 80 00 43 48 81 C3 A5 D9 A5 C3 81 00 44 48 10 20 C0 20 10 80 84	1138
64A0:	00 45 48 10 20 C0 20 10 80 04 80 46 48 20 70 FB 20 20 3E 00 47 48 95 95 95 95 95 95 95 48	D2EC
64C0:	48 00 1C 2A 49 00 00 88 00 49 48 00 80 00 00 00 00 00 4A 48 10 20 40 FF 40 20 10 00 4B 48 00	F314
64E0:	00 04 02 FF 02 04 00 4C 48 01 01 01 01 01 01 01 4D 48 20 70 FB 20 20 20 3F 00 4E 48 FF FF FF	5386
6500:	FF FF FF FF FF 4F 48 81 89 9C BE 7F 88 88 89 50 48 89 88 88 7F BE 9C 89 81 51 48 C3 88 18 38 7E	8754
6520:	38 18 C8 52 48 C3 10 18 1C 7E 1C 18 D3 53 48 00 00 00 00 00 00 00 00 54 48 FF 00 00 00 00 00	4887
6540:	80 55 48 00 00 00 49 2A 1C 00 56 48 AA 55 AA 55 AA 55 AA 55 57 48 AA 55 AA 55 AA 55 AA 55 58	8261
6560:	48 00 00 FC 82 82 82 82 86 59 48 84 84 84 84 F8 00 00 5A 48 00 00 00 00 00 00 00 FF 5B 48 00	D448
6580:	1C 3E 7F 3E 1C 88 00 5C 48 81 81 81 81 81 81 5D 48 24 24 E7 00 E7 24 24 00 5E 48 81 81 99	DE2B
65A0:	99 81 81 FF 00 5F 48 FF 00 00 00 00 00 00 00 04 D0 C1 D3 A0 C4 C5 A0 C3 C1 D2 D4 C5 A0 D3 CF D5	9E77
65C0:	D2 C9 D3 A0 A1 C9 CE D3 D4 CF CE A0 EF F5 A0 C9 CE D3 D4 CF C6 8D 00 00	CF8E

SDUMP/PRO.FP,A#0300,L#009E

0300:	20 F8 BE A9 9E 8D 07 BE A9 BE 8D 88 BE A9 03 8D F6 83 A9 BE 8D F7 03 A9 88 85 68 A2 81 86 67 CA	A141
0320:	8E 00 00 8A A2 12 9D 58 BF CA D0 FA A9 3F 8D 6B BF A9 CF 8D 58 BF A9 F0 8D 10 BE A9 FD 8D 11 BE	E0CD
0340:	8D 21 BE A9 1B 8D 20 BE A9 00 8D 12 BE 8D 16 BE 8D 22 BE 8D 26 BE A9 C1 8D 13 BE 8D 23 BE A9 C3	3882
0360:	8D 17 BE 8D 27 BE A9 95 20 ED FD 20 84 FE 20 2F FB 20 58 FC A9 00 8D F2 03 A9 BE 8D F3 03 49 A5	947F
0380:	8D F4 03 A2 01 BE 9A 03 20 91 03 E8 E0 85 90 F5 60 78 20 00 BF 41 99 03 60 01 01	1A4E

Turbo Pascal 4.0 dynamisé

Rares sont les programmeurs qui ne sont pas tentés par un langage aussi convivial que Pascal. Nombreux sont ceux qui se sont tournés vers celui de Borland, le fameux **Turbo Pascal**, réputé pour sa rapidité de compilation et la simplicité de son utilisation.

*Turbo Professionnel** séduira sans doute les adeptes de Turbo Pascal. Il contient, en effet, des routines optimisées, indispensables à tout programmeur. Il tourne avec la version 4.0 de Turbo Pascal. Avec lui, vous écrirez sans problème des programmes résidents en mémoire, vous générerez des menus déroulants à volonté, jonglerez avec les fonctions DOS sans être un as de l'assembleur... et bien d'autres choses encore.

Voulez-vous travailler avec des tableaux de plus de 64 K ? C'est possible.

Souhaitez-vous manipuler des chaînes de plus de 255 caractères ? Sachez que toutes les routines standard de *Turbo Professionnel* sont fournies pour des chaînes ASCII de 1 à 65 535 caractères !

Notez que *Turbo Professionnel* est fourni avec une documentation française de plus de 400 pages, rédigée avec clarté et donnant, pour chaque routine, la syntaxe de l'appel et une indication des problèmes possibles.

Un détail encore : les programmes de démonstration sont non seulement instructifs, mais

utiles : processeur résident de macros, table de référence résidente (codes ASCII, couleurs, etc.), calculatrice, utilisateur de gestion de temps, générateur de menu... ouf !

CHEZ LE MÊME ÉDITEUR

TDebugPlus, même éditeur, même efficacité, peut être conseillé à tout programmeur en Pascal soucieux de mettre rapidement au point les programmes les plus complexes.

Ce logiciel détecte, en effet, les erreurs cachées, montre le code source en *Turbo Pascal 4.0* en mettant en valeur l'instruction en cours.

Avec *TDebugPlus*, vous passerez de l'écran de mise au point à celui de programmation en un clin d'œil.

Documentation française, bien sûr, et réalisée dans le même esprit que celle de *Turbo Professionnel*.

Des produits sérieux et indispensables.

Clément RENARD.

* **ATEA EDITION**, 98, rue Giraudeau 37012 TOURS CEDEX
Tél. : 47.39.57.13.

/ **LES LIVRES** /

- **PC ET COMPATIBLES, MODE D'EMPLOI**
(J. Lasselle et C. Ramsay)

Ce livre est destiné à tous les nouveaux utilisateurs d'un IBM PC ou compatible dans le cadre d'un usage professionnel ou personnel.

Supposant que le lecteur ne possède aucune expérience de l'informatique, les auteurs ont voulu commencer par

les explications de base de l'utilisation d'un ordinateur pour ensuite poursuivre progressivement avec des explications plus approfondies. L'utilisateur étudiera, entre autres : le chargement du système d'exploitation, l'utilisation du clavier et des unités de disques, l'utilisation des commandes du système d'exploitation pour communiquer avec l'ordinateur, l'organisation du stockage des informations dans le disque dur,

le chargement et l'utilisation des programmes d'applications tels que Word ou Multiplan... etc.

Pour tirer le meilleur profit de ce livre, l'utilisateur devra se servir d'un ordinateur PC ou compatible, du système d'exploitation MS-DOS/PC-DOS (version 3.3) et mettre en pratique les exercices proposés.

COLLECTION MICRO-POCHE — 196 pages — 49 F
SYBEX, 6-8, impasse du Curé, 75018 PARIS,
tél. : (16-1) 42.03.95.95.

ALPHA Lexis

vous facilitera la vie

Un produit
BORLAND-LAROUSSE

Vous avez lu la pub : Plus de 400 000 formes et 40000 synonymes, directement accessibles par une seule touche... et vous avez pensé que l'ALPHA Lexis était exclusivement destiné aux heureux possesseurs du SPRINT de Borland. Tout faux !

Ce produit est compatible avec les meilleurs logiciels du moment, y compris *Multiplan* ou *Wordperfect*. Il permet réellement à tout utilisateur de micro personnel de vérifier (en temps réel ou différé) l'orthographe d'un mot... ou d'un texte.

Mieux, il est assorti d'un dictionnaire de synonymes qui ravira les écrivailleurs au vocabulaire anémié !

Coexistence pacifique

Quand ALPHA Lexis est installé en mémoire, il passe totalement inaperçu, mais un bip sonore

vous signale sa présence dès que vous faites une faute de frappe. Pour le reste, pas de problème : il cohabite aussi bien avec *Quattro* qu'avec un programme personnel en Turbo Basic par exemple. Je me suis amusé, pour le plaisir, à taper l'embryon de programme ci-dessous et j'ai constaté, non sans une certaine satisfaction, je l'avoue, que mon ALPHA Lexis jouait pleinement son rôle.

Quelques idées

Vous avez écrit une œuvre impérissable que vous conservez soigneusement sur une disquette dorée sur tranche, dans un fichier texte intitulé *TOTO*. Lisez-le gentiment en utilisant `more < toto.txt` qui envoie *toto.txt* à l'écran à raison d'une page à la fois.

Enfonchez alors ALT-F8 et ALPHA Lexis vous indiquera en surbrillance tous les mots qu'il ne connaît pas. Pour le reste, je vous fais confiance !

(suite page 38)

Lexis.bas

```

'*****
'*   EXEMPLE D'UTILISATION D'ALPHA LEXIS A PARTIR D'UNE ROUTINE BASIC   *
'*****
CLS
DIM phrase$(99)
DO
    i% = i% + 1
    v% = v% + 1
    LOCATE v%, 1
        PRINT SPACE$(80)
    LOCATE v%, 1
        PRINT USING "##"; i%;
        LINE INPUT ". ", phrase$(i%)
    IF v% = 23 THEN v% = 0
LOOP WHILE phrase$(i%) <> ""
i% = i% - 1
'Suite du programme

```

'On arrête en entrant une phrase vide

'Turbo Basic autorise incr i%

'Turbo Basic autorise decr i%

'*****

Précision intéressante :

Les touches de fonction d'ALPHA Lexis sont paramétrables à tout moment et il ne peut donc pas y avoir d'interférence avec les commandes des autres logiciels... sauf cas très particuliers.

Inutile de s'étendre sur le manuel accompagnant ce logiciel. Il est succinct, mais plus que

suffisant, étant donné la facilité d'emploi d'ALPHA Lexis.

Domage que cet excellent outil ne fonctionne pas sur nos vieux Apple II ! Il est vrai que sur le GS, avec le *PC transporter*, on bénéficie des mêmes prestations que sur un compatible XT..., mais oui !

Nestor.

LES LIVRES

• **LE LIVRE D'APPLETALK**
(Jean-Pierre Curcio
et Jean Robberecht)

Auteurs : Jean-Pierre Curcio est responsable des Techniques avancées et des télécommunications dans le département informatique de la BFCE. Il est l'auteur de "Boîte à outils de l'Apple IIGS" (des Editions P.S.I.) et signe régulièrement les "Passeports Mac" dans *Décision Informatique*.

Jean Robberecht est Directeur Technique au niveau des affaires nationales appelées également "grands comptes", chez Apple.

Thème : A ses débuts, Macintosh a été accusé d'être un ordinateur fermé. En 1985, Apple a lancé le réseau AppleTalk. Ce réseau servait principalement à partager le service d'une imprimante LaserWriter entre plusieurs Macintosh. C'est d'ailleurs le réseau utilisé par l'équipe des Editions P.S.I.

AppleTalk est une architecture complète de réseau local répondant aux six premières couches de la norme internationale ISO (définition normalisée selon sept niveaux d'une communication entre ordinateurs). AppleTalk sert donc à relier des postes isolés (pour partager des fichiers, pour échanger des messages) et à communiquer avec l'extérieur (autres réseaux : IBM, DEC, HP, Bull, etc.).

Description :

Ce livre très complet et abondamment illustré comporte deux parties :

- La première, destinée à l'utilisateur présente :
 - les concepts : principe du réseau local, principe d'AppleTalk ;

- les "produits" : la connectique AppleTalk, la connexion sur Ethernet, la connexion inter-réseaux, Appleshare, Lasershare, Tops, la messagerie électronique ;

- La seconde est plus particulièrement destinée au technicien réseau :

- les différents types de réseau ;
- aspects techniques sur AppleTalk.

Le livre d'AppleTalk est, à notre connaissance, le premier ouvrage sur ce sujet, et nul doute qu'il est d'une qualité exceptionnelle tant du point de vue rédactionnel que technique.

Renseignements au public :

Editions P.S.I. 496 pages — 310 F
6-10, boulevard Jourdan — 75014 PARIS
Tél. : (16-1) 45.65.06.06.

• **AUTOCAD
PAR LA PRATIQUE**
(George Omura)

Ce livre représente le guide essentiel d'AutoCAD destiné à tous ceux qui abordent le dessin assisté par ordinateur. Il traite de la version 9 d'AutoCAD pour IBM PC, AT/XT ou compatible. Toutefois, bon nombre d'exemples peuvent être réalisés à l'aide des versions 2.5 et 2.6.

L'ouvrage se développe autour d'un exercice d'architecture : la construction d'un immeuble. Les exercices proposés permettent à l'utilisateur d'acquérir les bases à partir desquelles il peut établir ses propres méthodes d'utilisation d'AutoCAD. De plus, ils sont développés de manière progressive.

Parmi les sujets principaux, l'utilisateur trouvera :

- La présentation du programme, la création et l'édition d'un dessin simple.
- Les notions de base de DAO et d'informatique.
- La description du matériel nécessaire pour utiliser AutoCAD.
- La gestion d'un grand dessin.
- L'utilisation des cotations.
- L'utilisation des attributs.
- L'utilisation du texte pour annoter les dessins.
- Le transfert dans AutoCAD de dessins réalisés à la main.
- La récupération d'informations contenues dans un dessin pour les transférer vers d'autres programmes.
- Le dessin en 3 dimensions et l'utilisation d'AutoShade.
- Comment augmenter la vitesse de saisie et de traitement.
- La personnalisation d'AutoCAD et les bases d'AutoLISP.

Le livre se termine par quatre annexes abordant l'installation d'AutoCAD et d'AutoShade, les commandes courantes du DOS, les programmes AutoLISP et enfin les options accessibles par Modifvar.

AutoCAD par la pratique ne nécessite pas de connaissances particulières en informatique et offre à l'utilisateur ou au gestionnaire la possibilité d'acquérir une connaissance rapide et pratique du logiciel. Les exemples sont nombreux et illustrent chacune des étapes de développement traitées tout au long de l'ouvrage, apportant ainsi une vision détaillée d'AutoCAD dans le contexte d'une situation réelle.

SYBEX, 6-8 Impasse du Curé, 75018 PARIS
Tél. : (16-1) 42.03.95.95 974 pages — 485 F

FRANÇAIS-ANGLAIS

Collins On-Line™

Pour PC et compatibles équipés d'un disque dur

Produit suisse de MARCTECH, ce dictionnaire électronique reprend le contenu d'un dictionnaire de poche COLLINS. Il est distribué en France par SOFTISSIMO*. Une disquette de programmes plus cinq disquettes de dictionnaire : on ne regarde pas à la dépense du côté de Neufchâtel ! Nous n'allons pas nous en plaindre.

L'installation de *Collins On-Line* est automatique et relativement longue, mais le plus ignorant en matière de PC devrait pouvoir la mener à bien sans problème. D'ailleurs, si besoin était, le guide d'installation pourrait remplir son office : il est clair et en français.

Ce logiciel requiert un disque dur offrant plus de 3,5 Mo d'espace libre. Il nécessite aussi le système d'exploitation PC/MS DOS (dans une version 2.1 ou ultérieure). *Collins On-Line* est gourmand et mobilise, quand il est actif, 150 K de la mémoire centrale.

COMPATIBILITÉ

Ce traducteur s'accommode plus ou moins bien de la présence d'autres logiciels. Tout se passe très bien quand ceux-ci ne chargent pas leur propre programme de commande de clavier, lors de l'initialisation. Dans l'autre cas, il y a risque de conflit avec celui de COLLINS.

Il est dommage que le menu ne permette pas une redéfinition des touches de commande (comme c'est le cas dans l'ALPHA Lexis de Borland). Ceci étant, le COLLINS est automatiquement reconfiguré pour *Microsoft Word*, par exemple, lors de l'installation. Le manuel explique, en outre, comment s'en servir avec l'*IBM DisplayWrite3*.

Peut-on tirer parti de *Collins On-Line* à travers un programme personnel en Turbo Basic, permettant par exemple de saisir une liste de mots français, puis leur traduction en anglais ? Assurément. Collins placera automatiquement le terme anglais sélectionné dans votre buffer, en réponse à un input, simplement en actionnant ALT-I... comme insertion.

Enfantin, pas vrai ? Naturellement, au préalable, vous aurez pris la peine de taper ALT-L pour localiser votre mot dans le dictionnaire. Tout cela est instantané, dans de belles fenêtres colorées (reconfigurables).

Un regret, pourtant. Dans le cadre d'une petite routine de saisie comme celle évoquée plus haut, il vous faut taper deux fois votre mot français : une pour le saisir... dans votre liste... une à la suite d'ATL-L.

A quand une option autorisant la consultation du dernier mot entré... comme dans l'ALPHA Lexis de Borland ?

La difficulté provient sans doute du fait que le COLLINS permet d'obtenir la traduction de locutions simples...

En conclusion, un produit très intéressant, bien fini, capable de rendre d'incontestables services à toute personne désirant améliorer son anglais ou l'enseigner... mais aussi à tous les utilisateurs moyens que nous sommes.

NESTOR

* 129, boulevard Sébastopol — 75002 PARIS

• **VENTURA PUBLISHER
À LA LOUPE**
(Ted Nace)

L'auteur tient une chronique de micro-édition dans la revue américaine *PC-World*. Spécialiste de PAO sur Ventura, il suit depuis ses tous débuts l'évolution de ce logiciel.

Thème : Ventura est l'un des logiciels de PAO sur PC les plus répandus avec PageMaker. Il bénéficie de la part de son diffuseur en France, Xérox, d'un important soutien.

Description : Afin d'aider l'utilisateur à mieux comprendre et utiliser toute la puissance de ce logiciel, l'auteur de *Ventura à la loupe* propose, étape par étape, la préparation et la réalisation de livres, manuels, bulletins, formulaires et rapports. Il apprendra entre autres :

- L'intégration rapide et efficace des textes et images.
- Les commandes cachées et les raccourcis permettant un formatage plus efficace et plus subtil des documents.
- Les méthodes d'archivage qui lui feront éviter le chaos sur son disque dur.
- Les différentes stratégies et l'esthétique de la typographie.
- Comment intégrer les tableaux de Lotus 1.2.3. et autres tableurs.
- Ce qu'il faut savoir des graphiques et des images obtenues par balayage.
- Comment choisir votre système, y compris le moniteur ou l'imprimante laser.

Ventura Publisher à la loupe est un guide exceptionnel pour Ventura Publisher, le logiciel 5 étoiles de la micro-édition sur ordinateurs XT/AT/386.

Renseignements au public :

Editions P.S.I. 224 pages — 250 F
6-10, boulevard Jourdan — 75014 PARIS
Tél. : (16-1) 45.65.06.06.

• **MISE AU POINT DE
PROGRAMMES EN
LANGAGE C**
Collection *QUE* (Robert Ward)
Texte français de D. Stoquart

Les développeurs d'applications en lan-

gage C ont sans cesse à résoudre des problèmes de mise au point. Cette tâche est ingrate, grande consommatrice de temps et d'énergie. Etrangement, il n'existait pas d'ouvrage qui ait pour objectif de les aider à mener à bien cette phase. Ce livre devrait donc faire le bonheur de nombreux professionnels. En effet, il a pour objectif non seulement le «débogage», mais aussi l'amélioration des techniques de mise au point, indispensable à la progression du développeur qui doit sans cesse faire face à des défis de plus en plus grands. Une mise au point réussie aide à l'exploration et facilite la compréhension. Plus grande est l'aptitude à découvrir les erreurs, plus facilement l'on maîtrise l'environnement dans lequel on travaille.

Dans ce livre, Robert Ward adopte une approche du «débogage» spécifique au C, car lorsque l'on travaille dans cet environnement, on ne peut se fier aux techniques empruntées à d'autres langages de programmation.

L'ouvrage se divise en quatre parties :

- Fondements et méthodes de mise au point.
- Localisation des erreurs et techniques de test de programmes.
- Méthodes de «traçage» lors de l'exécution des programmes.
- Programmes de mise au point au niveau source.

Toutes ces méthodes et techniques sont illustrées à l'aide de nombreux exemples. Cet ouvrage est l'outil indispensable du développeur d'applications en C, qu'il soit débutant ou expérimenté.

La lecture du livre présuppose la connaissance de la syntaxe et de la sémantique du C, ainsi que l'accès à un bon compilateur ou interpréteur C.

Robert Ward, ingénieur de formation, enseigne l'informatique à l'université et développe des programmes pour l'industrie, notamment sous UNIX. Son intérêt pour la programmation en C sur ordinateur personnel l'a conduit à devenir coordinateur international du C User's Group.

344 pages — 270 F

INTEREDITIONS
87, avenue du Maine 75014 PARIS
Tél. : (16-1) 43.27.74.50.

• **POSTSCRIPT
PAR L'EXEMPLE**
Outils pour l'édition
électronique
Adobe Systems, Inc.
Texte français d'Alain Kadé

L'édition électronique, ou publication assistée par ordinateur (PAO), se développe de plus en plus dans de nombreux secteurs d'activité. Elle permet, en effet, aux cadres d'entreprises de présenter des rapports de façon professionnelle, aux dirigeants de PME d'éditer eux-mêmes leurs catalogues, aux secrétaires d'associations de préparer leurs bulletins de liaison, etc. Toutes ces tâches peuvent aujourd'hui être accomplies, grâce à l'édition électronique, à bien moindres frais et avec des résultats tout à fait comparables à ceux de l'imprimerie classique. Pour pouvoir éditer des textes et des graphiques de qualité, il est indispensable, afin d'aller plus loin, de pouvoir en fait programmer son imprimante, de se familiariser avec le langage PostScript. C'est un langage de description de page, qui est très vite devenu le standard en matière d'édition électronique. Un programme PostScript permet de décrire efficacement, à l'intention de l'imprimante qui l'exécutera, l'aspect des textes, des images et des objets graphiques que l'on veut imprimer sur une page.

PostScript est déjà implanté, notamment, sur la LaserWriter d'Apple ainsi que sur les ordinateurs de la nouvelle gamme IBM, les Systèmes Personnels/2.

Cet ouvrage se compose de deux parties :

• Dans la première, les auteurs, qui ne sont autres que les membres de l'équipe américaine ayant développé le langage, proposent une "visite guidée" de PostScript, qui permet au lecteur de se familiariser avec les principales caractéristiques de ce langage véritablement exceptionnel par sa puissance.

• La deuxième partie propose des programmes courts, concrets, de difficulté croissante et abondamment commentés.

C'est une véritable "boîte à outils", qui permettra au lecteur de développer ses propres applications. Il apprendra ainsi, par exemple, à dessiner son logo, à obtenir des effets de relief, à imprimer un texte verticalement ou en arrondi, etc. Une mine d'idées et de ressources pour les mettre en pratique.

INTEREDITIONS 260 pages — 232 F

Langage C: Text Tool

les fonctions de l'outil texte (\$0C)

Nous avons vu dans Tremplin Micro numéro 21 la possibilité d'utiliser certaines fonctions du Text Tool, mais dans l'optique de fonctions prédéfinies. Nous vous proposons ici la possibilité d'utiliser directement les fonctions de l'outil texte (Text Tool).

Nous allons utiliser deux types de chaînes. Elles se différencient par le premier et le dernier octet les composant. Dans un cas (chaîne à la Pascal - PString) le premier octet représente le nombre de caractères qui compose la chaîne, et dans l'autre (chaîne à la "C" - CString-) la chaîne se termine par un octet nul. Pour des raisons de simplicité nous appellerons les chaînes PString pour les chaînes à la Pascal et CString pour les chaînes avec un octet nul terminateur. Le compilateur "C" probablement à des fins de sécurité, ajoute un octet nul à la fin des chaînes PString.

Comment définir une PString ? C'est très simple:

a) on insère la séquence `\p` en début de la chaîne de caractères. Lorsque le compilateur "C" rencontre cette séquence, il calcule la longueur de la chaîne et remplace `\p` par la valeur trouvée.

b) on insère le caractère `\` puis un nombre qui représente le nombre de caractères composant la chaîne. Ce nombre sera en hexadécimal ou en octal.

Il apparaît évident que la première solution est la plus pratique à l'utilisation, c'est donc celle que nous allons utiliser.

Comment définir une CString ? C'est très simple:

Il suffit de faire suivre le dernier caractère de la chaîne par la séquence `\0`. Le compilateur remplacera cette séquence par un octet nul (\$00).

Tout d'abord, nous allons décrire chaque appel avec leur équivalent (ou approchant) en Applesoft.

1 - Les fonctions d'écriture:

a) WriteChar (Char);

Cette fonction est l'équivalent du PRINT CHR\$(n) de l'Applesoft. Attention, l'utilisateur est responsable de la valeur du caractère Char. En effet, si PRINT CHR\$(21) provoque le passage en 40 colonnes (carte 80 colonnes inactive), en utilisation ProDOS 16, WriteChar(21) provoque en plus du passage en 40 colonnes le plantage de la machine.

La valeur de Char sera comprise entre 0 et 255. L'affichage du caractère sera fonction de la valeur des masques AND et OR.

Char = valeur ASCII du caractère à afficher.

b) WriteLine (PStringPtr);

Cette fonction est l'équivalent du PRINT "Texte à afficher" de l'Applesoft. La chaîne PStringPtr sera affichée et suivie d'un CR (pour un RAM based driver ou Basic driver) ou d'un CR + un LF (pour un Pascal driver).

PStringPtr = adresse de la PString à afficher.

c) WriteString (PStringPtr);

Cette fonction est l'équivalent du PRINT "Texte à afficher"; de l'Applesoft. La chaîne sera une PString. Il n'y a pas de CR envoyé à la suite de la chaîne. Les coordonnées horizontales et verticales correspondent à la position du dernier caractère affiché + 1.

PStringPtr = adresse de la PString à afficher.



d) WriteCString (CStringPtr);

Cette fonction est l'équivalent du PRINT "Texte à afficher"; de l'Applesoft. La chaîne est une CString. Il n'y a pas de CR envoyé à la fin de la chaîne. Les coordonnées horizontales et verticales correspondent à la position du dernier caractère affiché + 1.

CStringPtr = adresse de la CString à afficher.

e) TextWriteBlock(textPtr, offset, Count);

Il n'y a pas d'équivalent Applesoft à cette fonction dotée de trois arguments:

textPtr = adresse du texte ASCII à afficher,
Offset = position du premier caractère à afficher. Si offset = 0, l'affichage débutera au premier caractère du texte.

Count = nombre d'octets à afficher.

2 - Les fonctions de lecture:

a) Char = ReadChar (echoFlag);

Cette fonction est l'équivalent du GET variable : PRINT variable de l'Applesoft. La valeur de Char peut être comprise entre 0 et 255. L'affichage du caractère sera fonction de la valeur du drapeau echoFlag. Attention, le programmeur est responsable de la valeur du drapeau echoFlag (possibilité d'afficher des caractères de contrôle).

Char = valeur du caractère entré (entre 0 et 255),

echoFlag = drapeau définissant l'affichage (1) ou non (0) du caractère lu.

b) charCount = ReadLine (bufferPtr, maxCount, eolChar, echoFlag);

Cette fonction se rapproche de l'instruction INPUT "";A\$ de l'Applesoft mais elle est plus puissante. En effet, voyons les paramètres:

charCount = nombre de caractères enregistrés,

bufferPtr = adresse où stocker le texte,
maxCount = nombre maximum de caractères à enregistrer,

eolChar = terminateur d'entrée. La valeur possible varie entre 0 et 255.

echoFlag = drapeau de commande de l'affichage (1) ou non (0) du texte.

L'utilisateur est responsable de la valeur du drapeau echoFlag.

c) TextReadBlock (bufferPtr, offset, blockSize, echoFlag);

Il n'y a pas d'équivalent Applesoft à cette fonction. Il y a ici quatre arguments:

bufferPtr = adresse du block à lire,
Offset = position du premier caractère à lire. Si offset = 0, la lecture débutera au premier caractère du texte,

blockSize = taille du block à lire,
echoFlag = drapeau de commande de l'affichage (1) ou non (0) du texte.

```
1 /*          TEXT.DEMO1.CC
2 **
3 **          Utilisation de l'outil Text Tool
4 **
5 **          (c) Juillet/Septembre 1988
6 **
7 **          par Emile SCHWARZ & Tremplin Micro
8 */
9
10 #include    <locator.h>    /* Fichiers nécessaires pour utiliser */
11 #include    <memory.h>    /* les outils. */
12 #include    <texttool.h>
13
14 /*          Constantes utilisées par ce programme */
15
16 int      MonID;          /* Identificateur de ce programme */
```

```

17 char *Salut = { "\pSalut!" }; /* Message */
18 char *Attends = { "\pPressez une touche SVP " }; /* Invite */
19 char Touche; /* On stockera le caractère ici */
20
21 main() /* Main program */
22
23 {
24 TLStartUp (); /* Initialise le Tool Locator */
25 MonID = MMStartUp (); /* puis le Memory Manager */
26 TextStartUp (); /* et enfin le Text Tool */
27
28 WriteLine (Salut); /* Affiche le message de salutation */
29 WriteString (Attends); /* Affiche un message de sollicitation */
30 Touche = ReadChar ( (word) 0); /* Lit un caractère au clavier */
31
32 /* Teste la valeur du caractère */
33 if ( Touche > 0x1F && Touche < 0x7E)
34 { WriteChar ( Touche); }
35
36 DisposeAll (MonID); /* Libère la mémoire réservée */
37 TextShutDown (); /* Clos le Text Tool */
38 MMShutDown (MonID); /* Clos le Memory Manager */
39 TLShutDown (); /* " " Tool Locator */
40}

```

3 - L'initialisation des entrées/sorties:

a) SetInputDevice (DeviceRec);

Cette fonction permet de spécifier l'entrée standard de caractères à l'aide de la structure DeviceRec.

b) SetOutputDevice (DeviceRec);

Cette fonction permet de spécifier la sortie standard des caractères à l'aide de la structure DeviceRec.

c) SetErrorDevice (DeviceRec);

Cette fonction permet de spécifier la sortie standard des caractères en cas d'erreur à l'aide de la structure DeviceRec.

Définition commune aux fonctions a, b et c:

La structure DeviceRec est définie dans le fichier texttool.h et comprend deux paramètres:

```

typedef struct DeviceRec
{ word deviceType;
  long word ptrOrSlot;
}

```

deviceType = trois valeurs possibles:

0 = Basic,
1 = Pascal,
2 = Ram based,

ptrOrSlot = trois valeurs possibles:

ptr = adresse d'une table dans le cas d'un RAM driver,
Slot = numéro de slot dans les deux autres cas.

d) InitTextDev (deviceNum);

Cette fonction doit être utilisée conjointement aux trois instructions ci-dessus. En effet, elle initialise le périphérique désigné en a, b ou c en validant les différents paramètres.

L'argument deviceNum a trois valeurs possibles:

0 = périphérique d'entrée (input),
1 = périphérique de sortie (output),
2 = périphérique d'erreur (errorOutput).

4 - Mise en place des différents masques:

a) SetInGlobals (TxtMaskRec);

Cette fonction fixe les paramètres pour l'entrée standard des caractères à l'aide de la structure ➔

TxtMaskRec.

b) SetOutGlobals (TxtMaskRec);

Cette fonction fixe les paramètres pour la sortie standard des caractères à l'aide de la structure TxtMaskRec.

c) SetErrGlobals (TxtMaskRec);

Cette fonction fixe les paramètres pour la sortie standard en cas d'erreur à l'aide de la structure TxtMaskRec.

Définitions communes aux fonctions a, b et c:

La structure TxtMaskRec est définie dans le fichier texttool.h et comprend deux paramètres:

```
typedef struct TxtMaskRec
    { word andMask;
      word orMask; }
```

andMask = Valeur du masque AND. Cette valeur varie entre 0 et 255.

orMask = Valeur du masque OR. Cette valeur

varie entre 0 et 255.

Attention: le programmeur est responsable de la valeur des masques. C'est à lui de déterminer chaque masque en fonction de ses besoins.

5 - L'impression de messages en cas d'erreur:

a) ErrWriteChar, ErrWriteLine, ErrWriteString, ErrWriteBlock, ErrWriteCString.

Ces cinq fonctions donnent la possibilité au programmeur d'afficher sur le périphérique défini par SetErrorGlobals un caractère, une ligne ou un texte, ceci sans modifier l'état des périphériques d'entrée ou de sortie standard.

Se reporter à la définition des fonctions d'écriture pour les paramètres de ces fonctions.

Voir également le paragraphe 3 qui traite des entrées/sorties.

```
1 /*          TEXT.DEMO2.CC
2 **
3 **          Utilisation de l'outil Text Tool
4 **
5 **          (c) Juillet/Septembre 1988
6 **
7 **          par Emile SCHWARZ & Tremplin Micro
8 */
9
10 #include    <locator.h>
11 #include    <memory.h>
12 #include    <texttool.h>
13
14 int      MonID;          /* Identificateur de ce programme */
15 char     *Salut = {"\pSalut!" };          /* Message */
16 char     *Attends = {"\pPressez une touche SVP " }; /* Invite */
17 char     *Errtxt = {"\pCe texte est affiché sur le périphérique d'erreur"};
18 char     *Printer = {"\pVeuillez brancher votre imprimante en slot 1 SVP" };
19 char     Touche;        /* On stockera le caractère ici */
20 TxtMaskRec  OldMask;    /* Sauvegarde de l'environnement */
21 TxtMaskRec  NewMask;    /* Mise en place de mes masques */
22 DeviceRec   OldErrDev;  /* Sauvegarde de l'environnement */
23 DeviceRec   NewErrDev;  /*Mise en place des nouvelles valeurs*/
24
25 main()          /* Programme principal */
26
27 {
28 TLStartUp      ();          /* Initialise le Tool Locator */
```

```

29 MonID = MMStartUp ();          /* puis le Memory Manager      */
30 TextStartUp ();              /* et enfin le Text Tool      */
31
32 WriteLine ( Salut );
33
34 OldErrDev = GetErrorDevice (); /* Sauvegarde l'ancien port   */
35 NewErrDev.deviceType = OldErrDev.deviceType;
36 NewErrDev.ptrOrSlot = (long) 1; /* Slot 1 pour imprimante    */
37 SetErrorDevice ( NewErrDev ); /* Fixe le nouveau "error device" */
38 InitTextDev ( errorOutput ); /* Valide les nouveaux paramètres */
39 OldMask = GetErrGlobals (); /* Récupère les anciennes valeurs */
40 NewMask.andMask = 0xFF; /* Fixe la valeur du masque AND */
41 NewMask.orMask = 0x80; /* Fixe la valeur du masque OR */
42 SetErrGlobals ( NewMask );
43 WriteLine ( Printer ); /* Affiche une invite */
44 WriteLine ( Attends ); /* Message de sollicitation */
45 Touche = ReadChar ( (word) 0 ); /* Lit un caractère clavier */
46 ErrWriteLine ( Errtxt ); /* Affiche sur périph. d'erreur */
47 SetErrGlobals ( OldMask ); /* Restaure les anciennes valeurs */
48 SetErrorDevice ( OldErrDev ); /* " " */
49 InitTextDev ( errorOutput ); /* Valide les nouveaux paramètres */
50
51 WriteString ( Attends ); /* Affiche un message sollicitation */
52 Touche = ReadChar ( (word) 0 ); /* Lit un caractère clavier */
53 if (Touche > 0x1F && Touche < 0x7F) /* Teste la valeur du caractère: */
54 { WriteChar ( Touche ); } /* Si > 31 et < 127 : on affiche */
55 /* Autrement, on ne fait rien */
56
57 /* On ferme les outils que l'on a ouverts */
58
59 DisposeAll ( MonID ); /* Libère la mémoire réservée */
60 TextShutDown (); /* Clos le Text Tool */
61 MMShutDown ( MonID ); /* Clos le Memory Manager */
62 TLShutDown (); /* Clos le Tool Locator */
63 }

```

6 - Lecture de l'état de l'environnement:

a) DeviceRec = GetInputDevice();

Cette fonction stocke l'état des masques d'entrée existant au moment de l'appel dans la structure DeviceRec.

b) DeviceRec = GetOutputDevice();

Cette fonction stocke l'état des masques de sortie existant au moment de l'appel dans la structure DeviceRec.

c) DeviceRec = GetErrorDevice();

Cette fonction stocke l'état des masques d'erreur existant au moment de l'appel dans la structure DeviceRec.

Définition commune aux fonctions a, b et c:

La structure DeviceRec est définie dans le fichier texttool.h et comprend deux paramètres:

```

typedef struct DeviceRec
{
    word deviceType;
    long word ptrOrSlot;
}

```

deviceType = trois valeurs possibles:
0 = Basic, 1 = Pascal,
2 = Ram based,

ptrOrSlot = deux possibilités:
ptr = adresse d'une table dans le cas d'un RAM driver,

Slot = numéro de slot dans les deux autres cas.

d) `TxtMaskRec = GetInGlobals();`

Cette fonction stocke l'état des masques d'entrée existant au moment de l'appel dans la structure `TxtMaskRec`.

e) `TxtMaskRec = GetOutGlobals();`

Cette fonction stocke l'état des masques de sortie existant au moment de l'appel dans la structure `TxtMaskRec`.

f) `TxtMaskRec = GetErrGlobals();`

Cette fonction stocke l'état des masques d'erreur existant au moment de l'appel dans la structure `TxtMaskRec`.

Définition commune aux fonctions d, e et f:

La structure `TxtMaskRec` est définie dans le fichier `texttool.h` et comprend deux paramètres:

```
typedef struct TxtMaskRec
{
    word andMask;
    word orMask;
}
```

`andMask` = valeur du masque AND. Cette valeur varie entre 0 et 255.

`orMask` = valeur du masque OR. Cette valeur varie entre 0 et 255.

Attention: le programmeur est responsable de la valeur des masques. C'est à lui de déterminer chaque masque en fonction de ses besoins.

7 - Les fonctions de contrôle:

a) `CtlTextDev(deviceNum, controlCode);`

Cette fonction permet d'envoyer un code de contrôle à un périphérique.

Les paramètres sont définis comme suit:

`deviceNum` = périphérique à contrôler. Il y a trois valeurs possibles:

0 = input,
1 = output,
2 = error output.

`controlCode` = code de contrôle. Doit avoir une valeur comprise entre 0 et 255.

L'utilisation de cet appel outil ne doit se faire qu'avec une grande connaissance du driver de périphérique à tester.

b) `StatusTextDev(deviceNum, requestCode);`

Cette fonction renvoie un code à l'entrée STATUS du périphérique spécifié. Une erreur sera envoyée si le périphérique n'est pas prêt.

Les paramètres sont définis comme suit:

`deviceNum` = Périphérique à contrôler. Il y a trois valeurs possibles:

0 = input,
1 = output,
2 = error output.

`requestCode` = Code de contrôle. Doit avoir une valeur comprise entre 0 et 255.

Définition commune aux fonctions a et b:

Voir `SetErrorDevice`, `SetInputDevice`, `SetOutputDevice` et `InitTextDev` pour l'initialisation des périphériques.

NB: Les périphériques initialisés Basic ne supportent pas cette fonction.

Utilisation des exemples:

- 1 - Démarrer APW.SYS16,
- 2 - taper: EDIT TEXT.DEMO1.CC,
- 3 - Recopier un source,
- 4 - lorsqu'il est recopié, taper Pomme-Q pour quitter le texte édité,
- 5 - presser S pour sauver le source,
- 6 - presser E pour revenir au shell d'APW,
- 7 - taper: CML TEXT.DEMO1.CC KEEP=\$,
- 8 - taper: COMPACT TEXT.DEMO1 -S,
- 9 - taper: FILETYPE TEXT.DEMO1 S16,
- 10 - taper: DELETE TEXT.DEMO.ROOT.

Vous avez maintenant un fichier application (du type S16 ou \$B3) que vous pouvez exécuter sous FINDER, LAUNCHER, ProSEL, BASIC.SYSTEM patché etc...

```

1 /*          TEXT.DEMO3.CC
2 **
3 **          Utilisation de l'outil Text Tool
4 **
5 **          (c) Juillet/Septembre 1988
6 **
7 **          par Emile SCHWARZ & Tremplin Micro
8 */
9
10 #include    <locator.h>
11 #include    <memory.h>
12 #include    <texttool.h>
13 #include    <intmath.h>
14
15 int    MonID;                /* Identificateur de ce programme */
16 char    *Salut    = { "\pSalut!" }; /* Message */
17 char    *Attends  = { "\pPressez une touche SVP " }; /* Invite */
18 char    Touche;            /* On stockera le caractère ici */
19 char    Valeur[] = { 0x20, 0, 0, 0, 0x20, 0x24, 0, 0, 0, 0 };
20                                /* Valeur décimale et hexadécimale */
21
22 main()    /* Main program */
23
24 {
25 TLStartUp    ();            /* Initialise le Tool Locator */
26 MonID = MMStartUp ();        /* puis le Memory Manager */
27 TextStartUp  ();            /* suivi du Text Tool */
28 IMStartUp    ();            /* et enfin l'Integer Math */
29
30 WriteLine    (Salut);
31 WriteString  (Attends); /*Affiche message de sollicitation */
32 Touche = ReadChar ( (word) 0); /* Lit un caractère clavier */
33 if(Touche > 0x1F && Touche < 0x7F) /* Teste la valeur du caractère:*/
34 { WriteChar  (Touche); /* Si > 31 et < 127 : on affiche */
35                                /* Autrement, on ne fait rien */
36     Int2Dec   (Touche, Valeur+1, 3,0); /* Conversion */
37     Int2Hex   (Touche, Valeur+6,2); /*Conversion */
38     WriteCString (Valeur); } /* On affiche le tout */
39
40 DisposeAll   (MonID); /* Libère la mémoire réservée */
41 IMShutDown   ();      /* Clos l'Integer Math */
42 TextShutDown ();      /* Clos le Text Tool */
43 MMShutDown   (MonID); /* Clos le Memory Manager */
44 TLShutDown   ();      /* Clos le Tool Locator */
45 }

```

```

1 /*          TEXT.DEMO4.CC
2 **
3 **          Utilisation de l'outil Text Tool
4 **
5 **          (c) Juillet/Septembre 1988
6 **
7 **          par Emile SCHWARZ & Tremplin Micro
8 */

```

```

9
10 #include      <locator.h>
11 #include      <memory.h>
12 #include      <texttool.h>
13 #include      <intmath.h>
14
15 char  *Valeur  = { "\pValeurs des masques      " };
16 char  *Device  = { "\pDevice                " };
17 char  *Entree  = { "\pEntrée                " };
18 char  *Sortie  = { "\pSortie                " };
19 char  *Erreur  = { "\pErreur                " };
20 char  *And      = { "\pAND $                " };
21 char  *Or       = { "\pOR $                 " };
22
23 char  *DevType  = { "\pType de "driver"      " };
24 char  *Slot     = { "\pNuméro de slot"        };
25 char  *SlotNum  = { "\p$                " };
26 char  *Basic    = { "\pBasic Type            " };
27 char  *Pascal   = { "\pPascal Type           " };
28 char  *Ram      = { "\pRam based driver      " };
29 char  *Attends  = { "\pVeuillez presser une touche SVP" };
30 char  Touche;   /* On stockera le caractère ici      */
31 int   MonID;    /* Identificateur de programme      */
32
33 TxtMaskRec  InputMask; /* Structure de réception des masques */
34 TxtMaskRec  OutputMask; /* Structure de réception des masques */
35 TxtMaskRec  ErrMask; /* Structure de réception des masques */
36 DeviceRec   InputDev; /* Structure de réception des */
37 DeviceRec   OutputDev; /* paramètres */
38 DeviceRec   ErrDev;
39
40 main()      /* Programme principal */
41 {
42 {
43 TLStartUp      (); /* Initialise le Tool Locator */
44 MonID = MMStartUp (); /* puis le Memory Manager */
45 TextStartUp    (); /* suivi du Text Tool */
46 IMStartUp      (); /* et enfin l'Integer Math */
47
48 WriteString    ( Device ); /* Affiche ligne présentation */
49 WriteString    ( Valeur );
50 WriteString    ( DevType );
51 WriteLine     ( Slot );
52 WriteChar     ( 0x0A );
53
54 /*      1 - Récupération et affichage des paramètres d'entrée */
55
56 InputDev  = GetInputDevice (); /* Récupère le type et les */
57 InputMask = GetInGlobals   (); /* masques de ce "device" */
58
59 WriteString    ( Entree ); /*Affiche les paramètres */
60 /* correspondant à l'input device */
61 Int2Hex        ( (byte) InputMask.andMask, And+6, 2);
62 Int2Hex        ( (byte) InputMask.orMask, Or+6, 2);
63 WriteString    ( And ); /* Ecrit la valeur des masques */

```

```

64 WriteString      ( Or      );
65 if ( InputDev.deviceType == 0 ) /* Dans ce cas: Basic driver */
66     { WriteString ( Basic );
67       Int2Hex      ((word) InputDev.ptrOrSlot, SlotNum+2,2); };
68 if ( InputDev.deviceType == 1 ) /* Ici: Pascal driver */
69     { WriteString ( Pascal );
70       Int2Hex      ((word) InputDev.ptrOrSlot, SlotNum+2,2); };
71 if ( InputDev.deviceType == 2 ) /* Ici: Ram based driver */
72     { WriteString ( Ram );
73       Long2Hex     ( InputDev.ptrOrSlot, SlotNum+2, 6); };
74 WriteLine        ( SlotNum ); /* Ecrit la ligne */
75 WriteChar        ( 0x0A ); /* Un saut de ligne */
76 WriteChar        ( 0x0D ); /* + un CR */
77 /*      2 - Récupération et affichage des paramètres d'entrée */
78
79 OutputDev = GetOutputDevice (); /* Récupère le type et les */
80 OutputMask = GetOutGlobals (); /* masques de ce "device" */
81
82 WriteString      ( Sortie ); /* Affiche les paramètres */
83                  /*correspondant à l'output device */
84 Int2Hex          ( (byte) OutputMask.andMask, And+6, 2);
85 Int2Hex          ( (byte) OutputMask.orMask, Or+6, 2);
86 WriteString      ( And ); /* Ecrit la valeur des masques */
87 WriteString      ( Or );
88 if ( OutputDev.deviceType == 0 ) /* Dans ce cas: Basic driver */
89     { WriteString ( Basic );
90       Int2Hex      ((word) OutputDev.ptrOrSlot, SlotNum+2,2); };
91 if ( OutputDev.deviceType == 1 ) /* Ici: Pascal driver */
92     { WriteString ( Pascal );
93       Int2Hex      ((word) OutputDev.ptrOrSlot, SlotNum+2,2); };
94 if ( OutputDev.deviceType == 2 ) /* Ici: Ram based driver */
95     { WriteString ( Ram );
96       Long2Hex     ( OutputDev.ptrOrSlot, SlotNum+2, 6); };
97 WriteLine        ( SlotNum ); /* Ecrit la ligne */
98 WriteChar        ( 0x0A ); /* Un saut de ligne */
99 WriteChar        ( 0x0D ); /* + un CR */
100 /*      3 - Récupération et affichage des paramètres d'erreur */
101
102 ErrDev = GetErrorDevice (); /* Récupère le type et les */
103 ErrMask = GetErrGlobals (); /* masques de ce "device" */
104
105 WriteString      ( Erreur ); /* Affiche les paramètres */
106                  /* correspondant à l'error device */
107 Int2Hex          ( ErrMask.andMask, And+6, 2);
108 Int2Hex          ( ErrMask.orMask, Or+6, 2);
109 WriteString      ( And ); /* Ecrit la valeur des masques */
110 WriteString      ( Or );
111 if ( ErrDev.deviceType == 0 ) /* Dans ce cas: Basic driver */
112     { WriteString ( Basic );
113       Int2Hex      ((word) ErrDev.ptrOrSlot, SlotNum+2, 2); };
114 if ( ErrDev.deviceType == 1 ) /* Ici: Pascal driver */
115     { WriteString ( Pascal );
116       Int2Hex      ((word) ErrDev.ptrOrSlot, SlotNum+2, 2); };
117 if ( ErrDev.deviceType == 2 ) /* Ici: Ram based driver */
118     { WriteString ( Ram );
119       Long2Hex     ( ErrDev.ptrOrSlot, SlotNum+2, 6); };

```

```

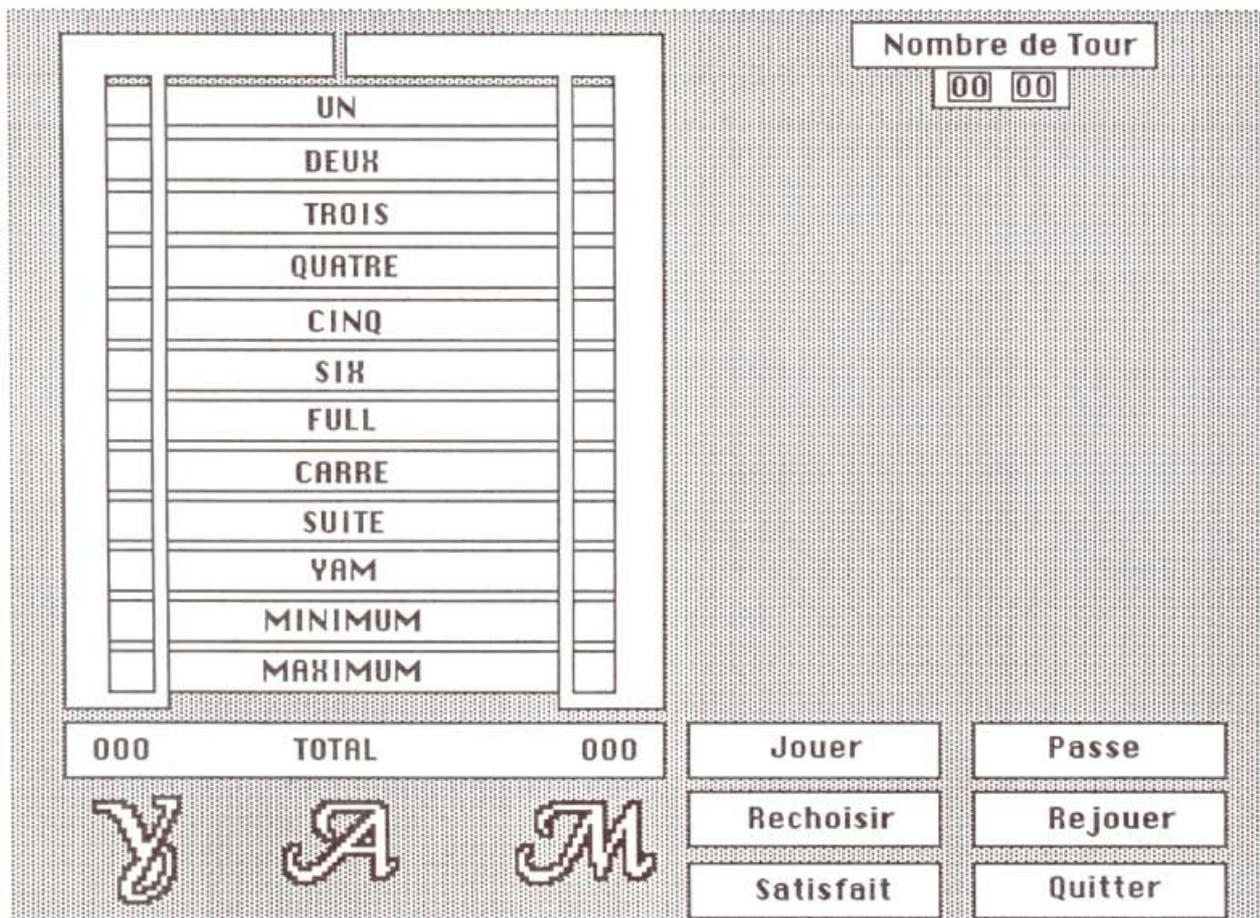
120 WriteLine      ( SlotNum );          /* Ecrit la ligne */
121 WriteChar      ( 0x0A );            /* Un saut de ligne */
122 WriteChar      ( 0x0D );            /* + un CR */
123 /* 4 - Attente d'une touche au clavier pour faciliter la lecture */
124
125 WriteString     ( Attends );          /*Affiche un message */
126 Touche = ReadChar ( (word) 0);        /* Lit un caractère clavier */
127
128 /*              On ferme les outils que l'on a ouverts */
129
130 IMShutDown      ();                  /* Clos l'Integer Math */
131 DisposeAll      ( MonID );           /*Libère la mémoire réservée */
132 TextShutDown    ();                  /* Clos le Text Tool */
133 MMShutDown      ( MonID );           /* Clos le Memory Manager */
134 TLShutDown      ();                  /* Clos le Tool Locator */
135 }

```

Bibliographie:

Apple IIgs Toolbox Reference Volume 1 & Volume 2 Addison-Wesley
 Apple IIgs Programmer's Workshop APDA
 Apple IIgs Programmer's Workshop "C" APDA

Ci-dessous la représentation en noir et blanc de l'image du jeu de YAM, paru dans T.M. n°21.



Vous avez dit IMPOSSIBLE...

bizarre, bizarre !

Lorsque l'on appelle le tableau de bord par Ctrl-pomme-ESC, le GS sauve consciencieusement diverses choses importantes, à savoir :

L'écran 00/0400	en E0/0C00
L'écran 01/04000	en E0/1000
L'écran E0/0400	en E0/1400
L'écran E1/0400	en E0/1800
La pile 00/0100	en E0/0300
La page 00/0000	en E0/1C00

ainsi que les trous d'écran utilisés par les routines *control panel*.

Il semblait donc parfaitement possible de tirer parti de cette manipulation pour doter le GS d'une recopie d'écran texte accessible à tout moment pour tous les programmes travaillant bien évidemment en mode texte.

C'est ce que fait l'accessoire de bureau *TextScreenDump*.

Ctrl-pomme-ESC vous amène au menu du tableau de bord, sélectionne *TextScreenDump* et exécute la recopie.

Il s'agit d'un outil tout à fait simple, qui suppose que le port imprimante est le port 1. Afin de permettre l'utilisation d'imprimantes anciennes, on emploie le mode *BASIC* qui ne permet pas de déterminer si l'imprimante appelée est effectivement en ligne. C'est pourquoi vous devez vous assurer qu'elle est bien connectée, allumée et alimentée en papier. Avec une IMW2 éteinte, on ressort de l'accessoire. Il suffit d'allumer l'imprimante et de résélectionner *TextScreenDump*. Avec une EPSON sur carte parallèle, *SDUMP* attend patiemment que vous allumiez la bête et effectue alors la recopie demandée.

Dans sa version actuelle, le programme fonctionne sur IMW2. Il reproduit alors les *INVERSE(S)* en caractères soulignés et les icônes grâce aux icônes internes de l'imprimante.

Si vous disposez d'une IMW1 ou d'une DMP, il vous suffira de changer un octet (BMI Icone devient BMI Write). Les icônes seront alors remplacées par le caractère majuscule qui les désigne.

Si vous employez un autre modèle d'imprimante, remplacez également les deux *NOP(S)* de *ChifInv* par *BRA Write* et les *INVERSE(S)* par leur équivalent *normal*.

En ce qui concerne la structure du programme, peu de choses à signaler. Tous les appels outils effectués ont déjà été utilisés dans *FREEZER*.

Il était tentant d'employer *WriteLn* au lieu de *__WriteChar*, mais je ne l'ai pas fait à cause du traitement des icônes et *INVERSE(S)*. Leur nombre étant par définition inconnu, il aurait fallu prévoir un tampon de 80x5 caractères, ce qui ne me plaisait guère.

On récupère en *e01d1e-1f* la configuration initiale sauvegardée lors de l'entrée dans le tableau de bord : *e01d1e* contient l'ancienne valeur de *ALTCHARSET*, *e01d1f* l'ancienne valeur de *RD80VID*. Cela permet de traiter correctement les écrans 40/80 colonnes et les éventuels chiffres en *FLASH*.

Pourquoi reproduire ce qui se trouvait dans les bancs *E0/E1* plutôt que ce qui figurait dans les bancs *00/01* ? La réponse est liée à une suggestion faite par APPLE dans l'annexe H au *Firmware Reference Manual*.

Pour afficher des informations sans avoir à reconstruire l'écran en service, on propose en effet :

- de travailler normalement en mode *shadowing on*,
- de supprimer provisoirement le *shadowing*,
- d'écrire directement dans l'écran *E0-E1*,
- de rétablir le *shadowing* pour revenir à l'écran initial.

(suite page 52)

C'est donc bien l'écran E0-E1 qu'il faut reproduire si l'on cherche à garder une trace de ce qui figure sur l'écran du moniteur.

A signaler que tant que l'on n'a pas quitté le tableau de bord, le contenu des zones de stockage reste inchangé. Il est donc parfaitement possible d'effectuer plusieurs recopies papier du même écran.

Si vous n'avez pas APW (MERLIN.8/16 est utilisable avec quelques modifications du source et d'ailleurs, SDUMP a été écrit sous MERLIN puis transféré sous APW), faites :

bsave SDUMP.06,a\$2000,l\$1bf

vérifications (avec SIGNATURE si possible) si tout est OK, on peut sauver définitivement mais auparavant, pensons aux variantes :

bload SDUMP.06,a\$2000

vous imprimez depuis le port 2, \$2079 : 02
vous utilisez IMW1 ou DMP, \$2129 : 3a
vous utilisez une autre imprimante, \$2129 : 3a
 \$2132 : 80 30

create SDUMP,t\$b9

bsave SDUMP,t\$b9,a\$2000,l\$1bf

N'oubliez pas de transférer ensuite SDUMP dans le sous-catalogue /*/system/desk.accs.

Si vous avez APW, créez, en mode EXEC, le fichier MAKE contenant les 4 lignes suivantes :

```
ASSEMBLE SDUMP.SRC
LINK SDUMP KEEP = SDUMP
COMPACT SDUMP
FILETYPE SDUMP CDA
```

Taper MAKE lancera l'exécution des commandes nécessaires à l'assemblage complet de SDUMP.

Note à l'intention des audacieux :

Si vous souhaitez aller plus loin (recopie d'écran GR, HGR, DHGR ou SHGR), je vous signale que l'état de certains commutateurs logiques est sauvegardé lors de l'entrée au tableau de bord.

```
c018 —» e01d04 | c01a —» e01d1a | c01b —» e01d1b
c01c —» e01d1c | c01d —» e01d1d | c01e —» e01d1e
c01f —» e01d1f | c029 —» e01d18 | c068 —» e01d03
```

Impossible... bizarre !

Yvan KOENIG — 1^{er} Septembre 1988.

```

1
2 *-----
3 *           TextScreenDump
4 *
5 *           quand un CDA est actif,
6 *           l'écran Texte est sauvé
7 *           en E0/1400... et E0/1800...
8 *           assembleur APW
9 *
10 * Y. KOENIG   01 Septembre 1988
11 *-----
12
13           Case ON
14           absaddr off
15           keep SDUMP
16
17 DASStart  START
18           KIND $1E      ; interdit 'Special Memory'
19
20 Bmain    gequ 0
21 B_aux    gequ 4
22 Flags    gequ 8
23
24           dc    i1'Tend-Tbeg'
25 Tbeg     dc    c'TextScreendump'
26 Tend     anop
27           dc    i4'StartDA'
28           dc    i4'ShutDA'
29
30 StartDA  anop          ; ici Natif Vrai
31           PHB          ; sauve banc données
32           PHK
33           PLB          ; banc données = banc
34           PHA          programme
35           PHA
36           PHA
37           LDX    £$130C ; _GetOutputDevice
38           JSL    $E10000 ; empile état initial
39           PHA
40           PHA
41           LDX    £$0D0C ; _GetOutGlobals
42           JSL    $E10000 ; empile état initial
43
44           PEA    0      ; BASIC device
45           PEA    i0-16 ; port 1 (ou 2)
46           PEA    1      ; .....
```

```

47     LDX  £$100C      ; _SetOutputDevice
48     JSL  $E10000
49     PEA  $FF        ; AND mask
50     PEA  $80        ; ORA mask
51     LDX  £$0A0C     ; _SetOutGlobals
52     JSL  $E10000
53     PEA  1          ; output
54     LDX  £$150C     ; _InitTextDevice
55     JSL  $E10000
56     BCS  exit       ; P.....C..M....
57
58     PEI  Bmain      ; Sauve 8 octets
59     PEI  Bmain+2    ; de page 0
60     PEI  B_aux
61     PEI  B_aux+2
62     PEI  Flags
63     LDA  >$E01D1E   ; valeurs initiales de
64     STA  Flags      ;          CO1E et CO1F
65
66     LDA  £$E01400ù-16
67     STA  Bmain+2
68     STA  B_aux+2
69     LDX  £0
70 V1up PHX
71     LDA  bases,X
72     STA  Bmain      ; base sur écran Main
73     CLC
74     ADC  £$400
75     STA  B_aux      ; base sur écran Aux
76     LDY  £0
77
78 H1up PHY
79     BIT  Flags
80     BPL  main       ; mode 40 cols
81     LDA  *B_aux$5,Y
82     JSR  CharOut
83     PLY
84     PHY
85 main LDA  *Bmain$5,Y
86     JSR  CharOut
87     PLY
88     INY
89     CPY  £40        ; dernière colonne ?
90     BCC  H1up       ; non
91
92     LDA  £$8D
93     JSR  Write
94     PLX
95     INX
96     INX
97     CPX  £24*2      ; dernière ligne ?
98     BCC  V1up       ; non
99
100    PLA

101    STA  Flags
102    PLA
103    STA  B_aux+2     ; restaure 10 octets
104    PLA              ; page zéro
105    STA  B_aux
106    PLA
107    STA  Bmain+2
108    PLA
109    STA  Bmain
110
111 exit LDX  £$0A0C     ; _SetOutGlobals
112     JSL  $E10000
113     LDX  £$100C     ; _SetOutputDevice
114     JSL  $E10000     ; restaure état initial
115     PLB              ; ..... banc données
116 ShutDA RTL         ; au revoir
117
118 CharOut AND  £$00FF
119     CMP  £$80
120     BCS  Write      ; Carac 'Normal'
121     CMP  £$60
122     BCC  NoMinInv
123     BIT  Flags-1
124     BMI  MinInv     ; AltCharset actif
125     SBC  £$40      ; ChifFlash ->
126     BRA  ChifInv   ChifInverse
127 NoMinInv CMP  £$40
128     BCC  NoIcône
129     BIT  Flags-1
130     BPL  MinInv     ; MajFlash -> MajInv
131     BMI  Icône     ; Icône (BMI Write
132 ;                  ;          si pas IMW2)
133 NoIcône CMP  £$20
134     BCS  ChifInv   ; Chiffre... inverse
135 ;                  ; Majuscules Inverse
136
137 MajInv ADC  £$40
138 MinInv anop
139 ChifInv dc  i1'$EA,$EA' ; BRA Write si pas IMW
140     PHA              ; sauve
141     JSR  ESC_out   ; Inverse -> souligné
142     LDA  £'X'
143     JSR  Write
144     PLA              ; récup
145     JSR  Write
146     JSR  ESC_out
147     LDA  £'Y'
148     BRA  Write
149
150 Icône PHA          ; sauve
151     JSR  ESC_out   ; mode icône IMW2
152     LDA  £'&'
153     JSR  Write
154     PLA              ; récup

```

155	JSR	Write		164	RTS
156	JSR	ESC_out		165	
157	LDA	£'\$'		166 bases	dc i2'\$1400,\$1480,\$1500,\$1580'
158	BRA	Write		167	dc i2'\$1600,\$1680,\$1700,\$1780'
159				168	dc i2'\$1428,\$14A8,\$1528,\$15A8'
160 ESC_out	LDA	£\$1B	; ESCape	169	dc i2'\$1628,\$16A8,\$1728,\$17A8'
161 Write	PHA			170	dc i2'\$1450,\$1400,\$1550,\$1500'
162	LDX	£\$180C	; _WriteChar	171	dc i2'\$1650,\$16D0,\$1750,\$17D0'
163	JSL	£E10000		172	END

SDUMP.CDA/TEXTES/SDUMP.06,A\$2000,L\$01BF

2000:	BF 01 00 00 00 00 00 00 00 57 01 00 00 FF 0A 04 02	F527
2010:	00 00 01 00 00 10 00 00 00 00 00 00 00 00 00 00	6C11
2020:	00 00 01 00 00 00 00 00 2C 00 40 00 20 20 20 20	B6ED
2030:	20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	1400
2040:	F2 57 01 00 00 0E 54 65 78 74 53 63 72 65 65 6E	A05D
2050:	64 75 6D 70 17 00 00 00 C3 00 00 00 8B 4B AB 48	AF59
2060:	48 48 A2 0C 13 22 00 00 E1 48 48 A2 0C 0D 22 00	60C1
2070:	00 E1 F4 00 00 F4 00 00 F4 01 00 A2 0C 10 22 00	099E
2080:	00 E1 F4 FF 00 F4 00 00 A2 0C 0A 22 00 00 E1 F4	37F7
2090:	01 00 A2 0C 15 22 00 00 E1 B0 5E D4 00 D4 02 D4	C353
20A0:	04 D4 06 D4 08 AF 1E 1D E0 85 08 A9 E0 00 85 02	5821
20B0:	85 06 A2 00 00 DA BD 27 01 85 00 18 69 00 04 85	027B
20C0:	04 A0 00 00 5A 24 00 10 07 B7 04 20 C4 00 7A 5A	90B4
20D0:	B7 00 20 C4 00 7A C0 C0 20 00 90 E8 A9 0D 00 20	FA93
20E0:	1E 01 FA E8 E8 E0 30 00 90 C8 68 05 00 68 05 06	433C
20F0:	68 05 04 68 05 02 68 05 00 A2 0C 0A 22 00 00 E1	9B88
2100:	A2 0C 10 22 00 00 E1 AB 6B 29 FF 00 C9 00 00 B0	8DF8
2110:	52 C9 60 00 90 09 24 07 30 18 E9 40 00 00 13 C9	550C
2120:	40 00 90 06 24 07 10 0A 30 20 C9 20 00 B0 03 69	C770
2130:	40 00 EA EA 40 20 1B 01 A9 58 00 20 1E 01 68 20	5560
2140:	1E 01 20 1B 01 A9 59 00 00 19 48 20 1B 01 A9 26	D449
2150:	00 20 1E 01 68 20 1E 01 20 1B 01 A9 24 00 00 03	E472
2160:	A9 1B 00 48 A2 0C 10 22 00 00 E1 60 00 14 00 14	84DD
2170:	00 15 00 15 00 16 00 16 00 17 00 17 20 14 AB 14	39FC
2180:	28 15 AB 15 20 16 AB 16 20 17 AB 17 50 14 D0 14	C03C
2190:	50 15 D0 15 50 16 D0 16 50 17 D0 17 F5 04 00 0F	EAEC
21A0:	00 17 00 F5 04 00 13 00 C3 00 F7 0F 00 00 00 00	39EC
21B0:	07 72 87 8E 9B F1 F7 FB FE 03 07 0D 11 14 00	4846

COMMODORE 64/128 ET 6502 D'APPLE

Bonjour à toute la rédaction de ma part, et félicitations pour la qualité, toujours excellente de votre revue.

Ne sachant où trouver réponse à ma question, c'est vers vous que je me tourne pour la seconde fois ; elle est la suivante :

- Le système GEOS développé sur COMMODORE 64/128 tourne-t-il sur APPLE II ? Une personne me l'a affirmé, mais je reste perplexe ; si oui, s'agit-il d'une version modifiée pour APPLE II ou bien peut-on acheter celle pour COMMODORE sans avoir de surprise ?

R. HACKERT (92230 GENEVILLIERS)

Oh le naïf ! Le seul point commun entre le C64 et les APPLE est la présence d'un microprocesseur 6502.

GEOS est effectivement disponible pour APPLE mais n'est pas identique à la version Commodore.

Pour plus de précisions, veuillez consulter *Call APPLE*, mai 88, pages 14/15 et (ou) contacter :

BERKELEY SOFTWAREWORKS

2150 Shattuck Avenue

Berkeley, CA 94704 U.S.A.

Yvan KENIG.

Notez que le numéro 23 de

TREMPIN MICRO

sera mis en vente le 25 Janvier 1989

QUESTIONS-RÉPONSES

Questions :

Dans le cadre de vos réponses aux questions des lecteurs de Tremplin Micro, je souhaite vous soumettre ces deux problèmes :

1. Disposant d'un APPLE IIGS, d'un MEG, comment remédier au "NO BUFFERS AVAILABLE" que provoque, sous PRODOS 8, l'ordre BLOAD P16, A\$2000, T\$F9 ?
2. Ayant copié sur le disque virtuel /RAM5, les fichiers P8, renommé PRODOS, et BASIC.SYSTEM, puis ayant configuré le tableau de bord sur "démarrage /RAM DISK", je n'obtiens, lors du boot, que le message "Check startup device". Que faire pour démarrer sur disque virtuel ?

C. VAN FRACHEM (75012 PARIS).

Réponses :

Il est impossible de charger P16 en entier depuis P8 et BASIC.SYSTEM. Réfléchissez un peu. P16 contient \$9450

octets. Vous essayez de le charger en \$2000. \$2000 + \$9450 = \$B450. Heureusement que BASIC.SYSTEM vérifie ce qu'on lui demande de faire avant d'obéir car ses tampons de travail commencent au mieux en \$9600 ! En l'absence de contrôle, BASIC.SYSTEM serait écrasé par le programme chargé. De toute façon, P16 ne pourrait s'exécuter, car le fichier stocké sur disque n'est pas directement exécutable.

Vous avez omis une étape. Le système d'exploitation crée /RAM 5, mais n'y installe pas le bloc 0 indispensable à un disque de démarrage. Il faudrait : soit FORMATER /RAM5 à l'aide d'un utilitaire, soit utiliser le petit programme ci-dessous qui lira le bloc 0 du disque magnétique slot5,drive1 et le copiera dans /RAM5 en slot5, drive2.

Attention ! il faut que votre disque magnétique ait été normalement formaté (bloc0 standard).

Yvan KOENIG

```

1
2 *****
3 * Installe Bloc0
4 *****
5
6         ORG     $300
7
8 0300: A9 50      8         LDA     £$50          ; slot 5, drive 1
9 0302: 8D 1C 03   9         STA     Where
10 0305: 20 00 BF  10        JSR     MLI
11 0308: 80        11        DFB     $80          ; ReadBloc
12 0309: 1B 03    12        DW     table
13 030B: B0 12    13        BCS     erreur
14 030D: A9 D0    14        LDA     £$D0          ; slot 5, drive 2
15 030F: 8D 1C 03 15        STA     Where
16 0312: 20 00 BF 16        JSR     MLI
17 0315: 81      17        DFB     $81          ; WriteBloc
18 0316: 1B 03   18        DW     table
19 0318: B0 05   19        BCS     erreur
20 031A: 60      20        RTS
21
22 031B: 03      22        table  DFB     3
23 031C: 50      23        Where  DFB     $50
24 031D: 00 20  24        DW     $2000
25 031F: 00 00  25        erreur DW     $00          ; Bloc Num 00
26
27 MLI      =    $BF00

```


Chasseur d'Images

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



**Chez votre
marchand de journaux !**