

pom's



3

Sommaire

Editorial		3
Apprentissage de l'assembleur (I)	<i>Jean-Louis Meillaud</i>	5
Contrôlez le nettoyage-mémoire	<i>Guy Mathieu</i>	17
Un exemple de HELLO	<i>Guy Mathieu</i>	19
Analyse du directory Pascal (II)	<i>Michel Crimont</i>	21
Notions de base	<i>Gérard Michel</i>	32
Conversion Pascal/BASIC/Pascal	<i>Gilles Mauffrey</i>	35
Communication grâce à l'APPLE	<i>Bruno Estrangin</i>	44
Copie d'écran-texte	<i>Christian Guérin</i>	51
Routine de présentation graphique	<i>Jean-François Duvivier</i>	53
Les fichiers EXEC	<i>Bruno Rives</i>	55
Personnalisez vos disquettes	<i>Pierre Boutrèux</i>	61
Les clubs ont la parole		63
Courrier des lecteurs		64

EDITORIAL

De nombreux lecteurs nous téléphonent ou nous écrivent pour nous poser des questions sur l'Apple et ses utilisations possibles. Il ne nous est malheureusement pas possible de répondre à tous. D'une part, cela demande souvent du temps, et il s'agit là de notre ressource la plus rare. D'autre part, il y a encore beaucoup de choses que nous ne savons pas, ou pour lesquelles l'équipement nous fait défaut. Les questions les plus fréquentes ou les plus intéressantes peuvent aussi trouver leur réponse dans le Courrier des Lecteurs. Nous vous prions de nous excuser de ne pas pouvoir faire plus en ce moment, mais nos moyens ne nous permettent pas encore d'assurer ce genre de service.

Nous vous offrons quatre longs articles dans ce numéro. Tout d'abord, l'article d'initiation à l'Assembleur, de Jean-Louis Meillaud, annoncé dès le numéro 1... Ensuite, la seconde partie de l'analyse du directory, de Michel Crimont. Enfin, le programme de transfert BASIC-Pascal-BASIC réalisé par Gilles Mauffrey pour la disquette Pom's N° 2, et un article de Bruno Estrangin sur les joies de la communication à l'aide d'un Apple.

De nombreux autres articles viennent compléter ce numéro 3, y compris (et c'est là une première pour nous) des contributions de lecteurs. N'hésitez pas à poursuivre dans cette voie : le prosélytisme et la participation sont les clés de la réussite de votre revue. C'est vous qui faites votre revue ; nous ne sommes que des catalyseurs !

Les lecteurs nous demandent souvent s'il est encore possible de commander le numéro 1, ou si l'on peut acquérir les disquettes séparément. La réponse est affirmative dans les deux cas. N'hésitez pas non plus à nous faire parvenir des petites annonces. Elles seront gratuites, sous réserve qu'il s'agisse effectivement d'annonces de particuliers. La périodicité trimestrielle est en effet quelque peu un frein pour les annonces, mais cela n'empêche pas le rapport performance/prix d'être excellent...

Personne n'ayant trouvé la bonne réponse à notre concours publié dans Pom's N° 2, nous vous donnons page 60 une seconde chance de prouver votre sagacité et de gagner ainsi un abonnement d'un an.

Nous vous donnons rendez-vous au salon Micro-Expo, qui se tiendra à Paris du 14 au 19 juin 1982. Nous y partagerons un stand avec les revues La Commode (Commodore) et Trace (TRS-80).

Hervé Thiriez

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

Apprentissage de l'assembleur (I)

Jean-Louis Meillaud

IL N'Y A PAS DE LANGAGE UNIVERSEL

Le possesseur d'un Apple qui veut se lancer dans la programmation est tout de suite sollicité par une pléiade de langages, BASIC, Pascal, Forth etc, sans parler de quelques "vieilleseries" comme le Cobol et le Fortran. Ces langages ont sans doute beaucoup d'attrait pour ceux qui disposent à la fois du temps et du budget nécessaires! Mon propos n'est pas d'entamer une polémique avec ceux qui vantent les mérites de Pascal et du pari qu'il représente... A mon niveau, que je pense être celui de nombreux utilisateurs, l'Apple, tel qu'il est livré, avec son BASIC et son MONITEUR, a de quoi contenter les plus exigeants. Les softs, je devrais dire les logiciels, proposés par les sociétés spécialisées (une bonne centaine rien qu'aux Etat-Unis) sont écrits à 99% dans un mélange de BASIC et de langage-machine, et ils marchent en effet très bien.

Prenons comme référence l'Apple II+ avec Applesoft et Autostart, qui représente le Standard actuel. Ce BASIC est un langage d'une souplesse et d'une facilité d'emploi inégalées. Le fait que le BASIC Applesoft soit interprété et soumis à un contrôle syntaxique au moment de cette interprétation rend nos programmes immédiatement opérationnels; ce n'est pas le cas avec des langages compilés.

Les limitations mêmes de l'Applesoft me paraissent plutôt un avantage. Il est facile, quand le besoin s'en fait sentir, de trouver des sous-programmes simulant le PRINT USING (voir POM'S numéro 4, en juin) ou les instructions de traitement de matrices qu'on trouve généralement dans les BASICs plus "étendus". Sans goût immodéré pour le paradoxe, je dirais que l'Applesoft est efficace en proportion de sa simplicité.

LES VRAIES LIMITES

Le BASIC, quelque soit son parfum, Applesoft ou Integer, se montre invivable dans certaines applications. Pour des calculs très répétitifs, sa lenteur devient exaspérante. Qui n'a jamais "piqué une crise" devant son écran à attendre qu'une boucle en BASIC finisse de s'exécuter, ou encore qu'un segment de programme soit chargé ? Ceux qui ont essayé d'utiliser un Mailing List tel qu'APPLEPOST me comprendront. Et pourtant ce progiciel a été écrit par les gens de PERSONAL SOFTWARE, les auteurs de VISICALC. Certaines solutions employées sont certes ingénieuses et de nature à inspirer le programmeur en quête de routines élégantes. Mais, en la circonstance, le BASIC se montre d'une telle gourmandise qu'il a fallu segmenter le programme et que la majeure partie du temps d'exécution se passe en appel de ces segments de programme.

LA SOLUTION HARMONIEUSE

Le BASIC est irremplaçable pour les petites tâches qui s'exécutent en une fois, comme les affichages d'écran. Il faut le compléter par des sous-programmes en langage-machine pour tous les travaux répétitifs tels que les conversions de codes.

Il faut remarquer que le langage-machine, comme le fait d'ailleurs le BASIC, peut utiliser des sous-programmes fort efficaces qui se trouvent dans le Moniteur. D'autre part il n'est pas interdit de brancher un programme en langage-machine sur

une fonction Applesoft dont on connaît le point d'entrée.

L'ARTICULATION

La communication entre Applesoft et langage-machine est prévue dans l'Applesoft grâce aux quatre instructions CALL, PEEK, POKE et USR. En outre, différentes techniques sont répertoriées pour effectuer ces mixages (cf S.H.LAM Routine, expliquée dans POM'S numéro 2).

LES OUTILS DE TRAVAIL

La "Bible", c'est le "REFERENCE MANUAL", le bouquin que vous avez soigneusement mis de côté le jour où vous avez pris livraison de votre Apple ! Il est grand temps de le faire sortir de sa cachette et de se familiariser avec les commandes du Moniteur; les exercices sont obligatoires pour une bonne compréhension. Il vous faut également un manuel de programmation en assembleur et ... un assembleur.

L'ASSEMBLEUR

Je commence par lui car il constitue une option fondamentale. Son choix entraîne en partie celui du manuel.

PREMIERE HYPOTHESE : Vous avez un ancien APPLE II, que nous appellerons APPLE II MOINS. Vous êtes vernis car vous disposez du "MINI-ASSEMBLER"; le "Vieux" Moniteur et le BASIC Entier, moins voraces que l'Applesoft Autostart, laissent la place à ce petit bijou, ainsi qu'aux sous-programmes Step et Trace, bien utiles pour les mises au point.

DEUXIEME HYPOTHESE : Vous avez un APPLE II +. Vous n'avez plus ni Trace ni Step et le MINI-ASSEMBLER a également disparu. Pour le récupérer, il faut que votre APPLE soit muni soit de la carte "Integer", (TRACE et STEP réapparaissent dans la foulée), soit, (plus normalement, car l'intérêt de l'INTEGER en PROM est limité), de la Carte Langage qui vous redonne le Mini-Assembleur, mais cette fois sans le pas-à-pas ni le traçage.

Si vous êtes réduit à l'APPLE II + sans plus... pas de mini-assembleur, ce qui est bien gênant car même si un "grand" Assembleur est pratiquement indispensable, le "mini" rend bien des services pour les programmes courts et les modifications rapides. Le mini est très pratique car il vous permet de voir ce que vous faites au fur et à mesure que vous écrivez votre programme. Aussitôt écrite, votre instruction est assemblée, à l'endroit même où elle devra être exécutée. L'écriture est particulièrement simplifiée et rapide. Il est par contre impossible d'assortir vos lignes de commentaires, pourtant bien pratiques pour savoir ce que vous aviez voulu faire, quand vous redécouvrez votre programme quelque temps plus tard.

LES GRANDS ASSEMBLEURS

L'inconvénient majeur du mini-assembleur pour les programmes importants est que, à la différence des grands, il ne permet pas les étiquettes symboliques. Ces étiquettes (Symbolic Labels) sont le principal atout du langage assembleur. Elles vont vous permettre, dans un assembleur adulte, de dire par exemple "Branchez-moi sur les TRAVAUX D'HERCULE", ou plus exactement sur "TVXDHERC", car il y a quand même une limite dans le nombre de caractères permis (4 à 8 suivant les modèles); pendant la phase d'assemblage, le programme assembleur ira chercher l'adresse de votre routine TRAVAUX D'HERCULE quand celle-ci sera définitive. Le Programme d'assemblage, qui a pour mission de transformer votre programme source (langage

assembleur) en programme objet (langage-machine) déterminera automatiquement cette adresse en début d'exécution lors du premier passage. C'est quand même une bonne manière de s'affranchir du GOTO X du BASIC, qui vous obligera à mettre impérativement votre routine en ligne X, ou si vous changez d'avis, à modifier toutes vos références à X dans les GOTOs et autres GOSUBs.

Ces étiquettes symboliques et bien d'autres merveilles ne sont disponibles que sur les assembleurs complets. Il y en a une demi-douzaine de connus sur le marché et il n'est pas toujours facile de les départager. Ils sont en pleine crise de croissance et chaque nouvelle version représente un progrès sur la dernière version du concurrent... Dans un état larvaire il y a deux ans, d'aspect encore très "bricolé" il y a seulement un an, ils s'améliorent de trimestre en trimestre et on peut considérer aujourd'hui qu'ils ne sont pas loin d'atteindre leur majorité. En outre les assembleurs actuels s'accompagnent d'une foule d'utilitaires : désassembleur pour transformer un programme en langage-machine en programme en assembleur, traçage pour aide à la mise au point et table de références pour retrouver toutes les instructions qui utilisent une étiquette donnée. L'existence de ces utilitaires vous simplifie la vie.

Nous prendrons comme assembleur de référence le LISA car, d'une part, c'est le plus répandu en France et, d'autre part, son auteur, Randy Hyde, est bourré d'idées qu'il a le bon goût de nous faire partager, non seulement en les incorporant dans ses logiciels, mais aussi en nous les commentant dans différentes publications (revue Applesauce et livre de programmation du 6502) et même à l'intérieur des listings-source de ses programmes.

LISA - prononcé "Lailleza" comme Minelli- Interactive Assembleur est un assembleur conversationnel de format libre. Il a le grand avantage, comme l'INTEGER, de signaler vos erreurs de syntaxe au fur et à mesure de l'écriture sans avoir à attendre le premier passage. En outre il met à votre disposition un large éventail de "pseudo-codes", commandes qui s'exécutent pendant l'assemblage. Il y en a pour chaque situation !

MANUELS DE PROGRAMMATION EN ASSEMBLEUR

Dans ce domaine également, l'évolution est rapide. Il y a encore un an nous n'avions à nous mettre sous la dent que des livres sur l'assembleur pour le microprocesseur 6502 en général, ce qui permet de programmer l'Apple mais ne tient aucun compte de la mine de programmes tout faits que sont le Moniteur et le DOS, sans parler de l'Interpréteur BASIC. Toutes les routines de gestion de l'écran, du clavier et des autres périphériques existent déjà. Il convient simplement d'avoir la clé d'accès.

Ces livres sur le 6502 sont d'ailleurs excellents. En dehors du Rodney Zaks, que je ne connais que de réputation (bonne, merci et vous!), mon coeur balance entre le R. Findley (6502 Software Cookbook, chez Scelbi) et le L.A. Leventhal (6502 Assembly Language Programming, Osborne/McGraw Hill).

L'Apple-o-mane (ou Pommolâtre) français est désormais comblé. Il peut apprendre l'assembleur spécifique de son P.S.I. favori et dans sa propre langue. En effet, Nicole Bréaud-Pouliquen a publié son Tome III de la Pratique de l'Apple (Edition du P S I). En collaboration avec Daniel-Jean David, elle nous dissèque le langage-machine et l'assembleur du 6502. Il est important d'avoir un livre spécialement orienté APPLE, car le programmeur est souvent appelé à utiliser directement des routines du Moniteur. Il serait titanesque d'avoir à les ré-écrire, sans compter qu'il serait difficile de faire aussi bien que Steve Wozniac. Comme le disent (aussi) les Américains à tout bout d'article : "Il est inutile de réinventer la roue!".

Le Don & Kurt Inman , "Apple Machine Language" édité par Reston/Prentice Hall est lui aussi spécifique. Le pas-à-pas très "enseignement programmé" est agréable à suivre . Pour les amateurs de sons et de graphiques, il est fait de larges incursions dans ces domaines. Malheureusement le livre s'arrête au moment où ça devient vraiment intéressant, c'est à dire au moment d'aborder le véritable assembleur, avec étiquettes symboliques et pseudo-codes. A conseiller néanmoins au débutant, s'il lit l'Anglais.

Le bouquin que j'attendais personnellement avec le plus d'impatience, le Randall Hyde, est enfin sorti. Il s'agit de "How anyone can program the Apple II - Using 6502 assembly Language". Il est parfait pour les utilisateurs de LISA, car il donne un tas de tuyaux complémentaires (utilisation des pseudo-codes LISA et du TRACE 65 pour la mise au point des programmes). Mais bien qu'il s'intitule de lui-même "Best Seller", son manque d'universalité empêche de le recommander à des gens qui n'auraient pas l'intention d'utiliser le LISA. Ces derniers par contre ne pourront s'empêcher d'avoir l'eau à la bouche à l'annonce des futurs volumes de la collection "Advanced 6502 L/M programming (Graphics, Disk I/O...)" et "Z80 and 6809 programming on Apple".

Enfin, pour rester dans les nouvelles alléchantes, signalons la sortie aux U.S. de "Everyone's Guide to Assembly Language". C'est la réunion des exceptionnelles chroniques parues dans SoftTalk depuis plus d'un an. Elles sont signées Roger Wagner, auteur d'Appledoc (voir Pom's numéro 2) et patron de la société de Software S D S . Ce livre constitue l'initiation la plus progressive et la plus complète au langage-machine de l'Apple et à l'assembleur.

LES SOFTS UTILES

L'amateur est noyé sous les logiciels mis à sa disposition . C'est peut-être cette multiplicité qui lui a fait préférer l'APPLE . Mais ces logiciels, ainsi que les dispositifs "hardware" mis sur le marché, ne sont pas tous d'une égale utilité. Dans chaque numéro de Pom's, des articles nous permettrons d'y voir plus clair et de sélectionner, par exemple, les aides à la programmation les plus efficaces.

Dès qu'il nous faut écrire, à titre professionnel ou privé, un système de traitement de texte est le premier pas vers la rentabilisation de notre "micro". Personnellement, j'ai un gros faible pour "Applewriter". C'est un bon soft, dont l'acquisition est peu onéreuse. Attention à Applewriter II, annoncé par Apple Computer ! Il paraît qu'il "tire dans les coins". Mais il sortira en version protégée... et il ne sera pas donné. Notre Applewriter actuel, s'il est un peu frustré, en ce sens qu'il lui manque un certain nombre d'options, disponibles sur SUPER SCRIBE et autres MAGIC WINDOWS, est perfectible. La plupart des options ou gadgets qui lui font défaut peuvent lui être greffés. Il suffit pour cela de parler sa langue, le langage-machine.

A mes yeux, la plus belle qualité d'Applewriter est d'être un système ouvert. Il existe nombre de "softs sophistiqués" qui vous rendent malade. Aussi perfectionnés soient-ils, ils ne peuvent toujours correspondre à ce que vous en attendez et il arrive fatalement qu'un jour ou l'autre vous souhaitiez leur faire subir une petite opération de chirurgie esthétique. Il y a un truc en trop, une commande dont l'exécution vous paraît exaspérante, ou un truc qui manque, une idée à vous qui vous paraît géniale et dont la mise en oeuvre vous simplifierait bien la vie. Bref, vous voulez apporter des modifications. Hélas pas de "modif" possible. Ces softs sont généralement cadenassés et leurs auteurs se sont évertués, par psychose du piratage, à vous empêcher d'en trouver la clé.

Avec Applewriter, tout est possible ! Non seulement on trouve dans le commerce des programmes tout faits (AppleWriter Extended de chez Brillig Systems, Proportional

Text Formatter pour les Centronics 737/9 de Micro-Media, dont les produits n'ont pas l'air d'être très diffusés, ou encore "Go-Between", un programme qui, d'après Apple-Europe, serait disponible en Angleterre), mais encore n'importe qui peut s'écrire les modifs, à condition de posséder suffisamment le langage-machine. C'est même la meilleure manière de se familiariser avec le langage-machine, (cf. la modification qu'a écrite Sheila Widnall pour utiliser son adaptateur de minuscules "Paymar Chip" avec l'Apple-Writer (Call-A.P.P.L.E. de Septembre 80), alors qu'un mois auparavant elle ignorait l'existence du LDA et les joies de la gestion de pile!).

Je suis résolument hostile aux adaptateurs de minuscules qui donnent de bien vilains caractères. De même je pense que les cartes 80 colonnes ont un intérêt très limité. Je me satisfais parfaitement de ce que l'Apple affiche sur l'écran. Par contre je tiens à ce que mon imprimante écrive en français.

La modification que je vous propose aujourd'hui et qui nous servira de tremplin pour nous familiariser avec le langage assembleur permet de disposer de la quasi-totalité des lettres françaises accentuées que l'imprimante est capable de produire. Il faut bien entendu que l'imprimante dispose d'un jeu de caractères français, mais c'est le cas le plus répandu et c'est le cas en particulier des Centronics 730-2 et 737/739. Les seuls caractères pour lesquels nous n'apportons pas de solution immédiate sont le tréma et l'accent circonflexe, ces caractères qu'Applewriter sait éditer, mais ne sait pas positionner sur les lettres correspondantes. Pour le moment, nous non plus, nous ne savons pas les imprimer et nous ne saurons pas le faire avant d'avoir résolu le délicat problème de l'espacement arrière sur les imprimantes à lignes. Rassurez-vous, la solution existe et elle fera l'objet d'un prochain article.

CARACTERES FRANCAIS

Sans la moindre modification il est possible d'obtenir (manuel Apple Writer, extraordinary characters, p.31) les caractères suivants:

é, °, è, §, °, à, ^, `

é CTRL N

° ESC CTRL N

è SHIFT M

§ ESC SHIFT M

°° SHIFT N

` CTRL SHIFT P

à ESC CTRL SHIFT P

^ ESC SHIFT N

La modification que nous proposons est directement inspirée du "patch" proposé par Lou Rivas pour utiliser le dispositif "Lower Case Adaptor" avec AppleWriter (Apple Orchard, Automne 1980). Cette modification permet d'imprimer le ù, le ç et le trait mode soulignage, en appuyant simultanément sur trois touches (CTRL A et, suivant le cas, la touche Barre oblique ou la touche Moins):

ù CTRL A /

ç ESC CTRL A /

- ESC CTRL A -

Après cette modification, la totalité des caractères ASCII 224 à 255 peut être affichée sur l'écran (il faudra se familiariser avec les nouveaux symboles!). Plus important, il est désormais possible de produire sur l'imprimante la quasi-totalité des lettres françaises accentuées.

Il est impératif que le programme modifié soit appelé TEDITOR car après l'impression (par "PRINTER") le retour à l'éditeur se fait par "BRUN TEDITOR" (en langage-machine... of course). Si vous désirez conserver le programme original sur

la même disquette (on ne sait jamais!), prenez le soin, avant toute modification, de changer son nom.

MODE D'EMPLOI

Il faut commencer par écrire le "patch" (moins français, mais plus joli que "verruie") en langage-machine. Nous allons utiliser LISA. Tout autre Assembleur fera l'affaire. Si on utilise le Mini-Assembleur il faut prendre particulièrement soin des adresses où doivent s'exécuter les branchements. Mais en l'occurrence, la tâche est grandement facilitée par la présence du listing d'assemblage LISA.

Il reste encore la solution qui consiste à taper directement les instructions en HEXA avec la fonction "CHANGE MEMORY" du Monitor, ultime recours de celui qui n'a pas encore d'assembleur.

Nous prendrons comme origine \$1848, qui se trouve en dehors du programme TEDITOR. Le pseudocode "ORG" s'occupera de tout. Une fois le programme assemblé, il convient de le sauver sur disquette. Pour cela faire "BSAVE CARFRAN.OBJ, A\$1848, L\$50". Le patch fait moins de 80 octets, mais il vaut mieux prévoir trop large que trop juste. N'oubliez pas que sur la disquette vous enregistrez par multiples de 256 octets (100 en hexadécimal) et qu'en conséquence l'avarice n'est pas payante! Tellement peu payante que, entre membres d'Apple-holic Anonymous, je peux bien vous confesser qu'il m'est arrivé de perdre trois heures à rechercher une erreur dans un programme en langage-machine, que j'avais tout simplement amputé du dernier octet. En effet, pour calculer la longueur de mon "BSAVE", j'avais retranché l'adresse d'origine de l'adresse de fin et j'avais ainsi escamoté un octet... C'est un grand classique, moins grave que d'oublier d'éteindre quand on retire une carte... mais tout aussi populaire !

Le côté farce de cette panne ne vous échappera pas quand je vous aurai dit que, aux premiers essais, mon programme "marchait comme sur des roulettes". Pour le tester, j'avais en effet fait suivre mon BSAVE d'un BRUN, sans prendre la précaution de vider la mémoire et du coup j'avais récupéré mon octet final, sagement resté à sa place en la circonstance.

PRECAUTIONS ET SOINS

Pour vous éviter une pareille mésaventure, je vous conseille la manoeuvre prophylactique suivante : avant de recharger un programme à tester, utilisez le "MOVE MEMORY" du Monitor pour mettre à zéro une zone aussi grande que possible. Par exemple, en supposant que votre programme de quelques centaines d'octets débute en \$800, passez en Moniteur (CALL-151). Puis tapez :

```
800:0  
801<800.2000M
```

Le TEDITOR prend chaque caractère du texte que l'on est en train d'éditer et lui fait subir un certain nombre de traitements pour voir de quoi il retourne (Lettre Majuscule, si précédée d'un Escape, minuscule, ou caractère de contrôle correspondant à une commande du programme). Puis, s'il s'agit d'un caractère du texte proprement dit, il lui fait subir un changement radical de code. L'Apple II modifie légèrement le code ASCII pour son usage interne. Il lui ajoute un bit à la position la plus significative de l'octet (on parle alors d'ASCII négatif). Applewriter le secoue bien davantage : les majuscules sont descendues dans la zone des caractères en INVERSE (\$01 à \$1A), les minuscules prennent la place des majuscules en NORMAL (\$C1 à \$DA), quant aux signes de ponctuation et aux chiffres, ils viennent se placer avant les majuscules (\$A1 à \$BF).

Quand Applewriter a fini de tester le caractère à écrire (au moyen d'instructions CMP) et lui a fait subir les conversions voulues (avec des AND et des ORA), il le range dans une chaîne, dont l'adresse de départ se trouve dans la mémoire \$28 et qu'il fait progresser en incrémentant le registre -Y. Il s'agit de l'adressage indirect post-indexé par Y que nous retrouverons plus loin.

Les Modes d'adressage très variés sont un des points forts du 6502, mais aussi un sujet de perplexité pour le néophyte. Pour s'y retrouver, il suffit d'appliquer les conventions classiques suivantes:

\$ renvoie à l'adresse
, ajoute à l'adresse la valeur du registre nommé après la virgule
() donne l'adresse de l'adresse. C'est une boîte aux lettres.
£\$ n'indique pas une adresse, mais une constante à utiliser immédiatement.
(Ne vous étonnez pas si, sur certaines imprimantes, le jeu de caractères français change le signe "pound" -notre signe mathématique sensiblement égal- en signe Livre Sterling).

MISE EN OEUVRE

A l'aide de l'assembleur de votre choix vous avez préparé sur une disquette le fichier binaire CARFRAN.OBJ, ou encore vous possédez la disquette Pom's qui contient ce fichier.

Vous disposez également d'une copie d'Applewriter. Il est recommandé de ne pas utiliser l'original ...

1) Vous chargez la Modif:

BLOAD CARFRAN.OBJ

2) Vous mettez votre disquette Applewriter et vous chargez l'Editeur:

BLOAD TEDITOR

Pour garder une copie de sécurité, vous changez le nom :

UNLOCK TEDITOR

RENAME TEDITOR, VIEUX TEDITOR

LOCK VIEUX TEDITOR

3) Vous passez en Moniteur (CALL-151) et vous utilisez l'instruction pour changer la mémoire (Reference Manual p.44), afin d'introduire les nouvelles valeurs suivantes :

*811: 8D 10 C0 4C 48 18

*3DOG ou 9DBFG (retour à l'Applesoft)

4) Vous enregistrez votre nouvel Editeur sur disquette:

BSAVE TEDITOR, A\$803, L\$186C

5) Après l'avoir essayé, vous verrouillez le nouvel éditeur :

LOCK TEDITOR

REMARQUE: Si vous ne disposez ni d'un assembleur ni de la disquette Pom's, vous pouvez encore vous en sortir : vous commencez directement au pas 2) et quand vous arrivez au pas 3) vous enregistrez également la modif en HEXA à partir de *1848: Bien entendu vous vous limitez au code objet qui est dans la partie gauche du listing d'assemblage.

LIGNES SOURCE POUR ASSEMBLAGE AVEC LISA

TTL "CARACTERES FRANCAIS"
ORG \$1848

Le pseudocode TTL permet, en LISA Version 2.5, d'imprimer le titre au début de chaque page de l'Assemblage. En outre, si vous disposez d'une Horloge Temps Réel, du type Mountain Computers (née Mountain Hardware), la date et l'heure s'imprimeront à côté du titre, ce qui est une façon agréable d'assurer la pagination d'un long listing.

Le pseudocode ORG est lui indispensable; il permet à l'Assembleur de calculer l'adresse des branchements. Il est de la responsabilité du programmeur de placer son programme objet à l'adresse indiquée au moment de l'exécution (à moins que son programme ne soit écrit de façon à pouvoir être déplacé -"relocatable" disent les anglophiles-, c'est-à-dire ne comporte aucun saut à une adresse absolue. Si ce n'est le cas, il faut faire appel à un programme de déplacement (cf. J.-F. Duvivier Déplacement des programmes en Assembleur, Pom's N 1)

C'est une bonne habitude que d'écrire, autant que faire se peut, des programmes recasables ("relocatable"). Il n'en est évidemment pas question dans notre exemple puisqu'il s'agit d'une modification qui doit se raccorder à une adresse bien définie (par JMP \$1848 que nous avons placé en \$814 dans AppleWriter).

Le pseudocode OBJ, qui indique l'adresse de départ de l'Assemblage du code-objet est inutile ici car, à moins d'en décider autrement et de le spécifier expressément avec OBJ, ORG génère automatiquement un OBJ à la même adresse (en LISA).

2. EQUATE

Le pseudocode Equate (EQU) permet d'affecter une étiquette à une cellule de mémoire où se trouve une valeur ou un début de programme. S'il s'agit d'un programme, il suffira par la suite de faire un saut inconditionnel (JMP) ou un branchement conditionnel (BEQ) à cette étiquette. C'est beaucoup plus gai que la solution du BASIC, qui consiste à définir un numéro de ligne fixe au point d'entrée du sous-programme.

Les "equates" servent beaucoup pour les sous-programmes du Moniteur. Si le sous-programme est en ligne (à l'intérieur du programme principal), vous faites figurer simplement l'étiquette avant la première instruction (cf. BAROBL infra). L'Assembleur LISA sait qu'il s'agit d'une étiquette parce que vous ne la faites pas précéder d'un espace, alors qu'il en faut au moins un avant un code instruction.

Dans le dialecte spécifique à LISA, un equate à une mémoire en page zéro (\$00 à \$FF) s'écrit EPZ (equate page zero) et ne sera suivi, bien entendu, que de deux chiffres hexa au maximum.

3. TEXTE EPZ \$0028

Il s'agit de la mémoire où le programme va chercher l'adresse de la chaîne de caractères qu'il est en train de constituer à partir des entrées au clavier.

4. CLAVIER EQU \$C000

Quand le caractère prélevé par un LDA à cette adresse est "négatif" (bit 7, le plus significatif, positionné), c'est qu'une touche du clavier a été enfoncée. Cet

état sera constaté par l'échec du test BPL.

5. STROBE EQU \$C010

Remet le bit le plus significatif à zéro pour pouvoir accepter une nouvelle entrée clavier. Le programme va effectuer une série de tests afin de voir si les touches qui nous intéressent ont été activées, pour se rebrancher finalement au point d'où est partie la dérivation (au moyen d'un RTS). Si les caractères testés sont bien ceux que nous recherchons, ils doivent être stockés entre temps dans la chaîne de caractères qu'utilise A/W. Ceci est effectué par STA (\$28), Y suivi d'un INY pour mettre à jour le pointeur.

6. VALEUR IMMEDIATE - ADRESSE - ADRESSE DE L'ADRESSE

LDA : Load Accumulateur permet de placer une valeur dans l'accumulateur.

Cette valeur peut être contenue dans l'instruction : LDA £\$Valeur en hexa, indique qu'il n'y a pas à chercher loin l'adresse de la valeur puisqu'il faut prendre littéralement la valeur contenue dans l'instruction. On parle alors d'adressage immédiat.

La valeur peut être recherchée directement dans une cellule de mémoire dont l'adresse est contenue dans l'instruction: LDA \$ adresse (en Hexa). C'est l'adressage direct.

On peut encore indiquer dans l'instruction l'adresse de la cellule où se trouve l'adresse de la valeur : LDA (\$ adresse de l'adresse où se trouve la valeur). Ce système de boîte aux lettres s'appelle l'adressage indirect. Il permet d'utiliser la même instruction pour toute une série de valeurs. L'adresse de la boîte aux lettres est fixe, seul son contenu varie.

Une autre manière de faire varier l'adresse consiste à associer un registre (X ou Y) dont le contenu vient s'ajouter à l'adresse. Il s'agit de l'adressage indexé. L'adresse indiquée dans l'instruction ne varie pas, mais on fait progresser le registre.

Si on marie l'indexage avec l'adressage indirect on obtient l'adressage indirect indexé. On peut de la sorte faire varier l'adresse de la "boîte aux lettres", adressage indirect pré-indexé par X, ou faire varier l'adresse trouvée dans la boîte aux lettres, adressage indirect post-indexé par Y. Ici le rôle des deux registres d'index est nettement différencié, ce qui est bien agréable pour éviter les confusions ! Ainsi LDA (\$28), Y permet de charger l'accumulateur avec le contenu de l'adresse A+Y, A étant l'adresse contenue dans la mémoire 40 (28 en hexadécimal). Les beautés de l'adressage indirect post-indexé par Y résident dans le fait qu'il n'a pas besoin de toucher au contenu de la cellule 40 et qu'il suffit d'incrémenter le registre-Y (INY) pour faire progresser l'adresse où l'on va lire où stocker le caractère suivant.

7. LIGNES 9 A 11

```
CMP £$81           ;TOUCHES CTRL-A?  
BEQ BAROBL  
RTS
```

Les branchements conditionnels se font en deux temps. D'abord, on met en présence les quantités à comparer. Ici l'accumulateur et la valeur 129 (81 en hexa). Ensuite, on indique la condition et la branche à prendre si cette condition est remplie.

En BASIC cela donnerait:

```
10 LET A = ACCUMULATEUR : LET B = 129 : REM 129=$81
20 IF A = B THEN GOTO 1000
30 REM SOUS-PROGRAMME BAROBL COMMENCE EN LIGNE 1000
40 RETURN
```

Remarque : la comparaison n'est pas à proprement parler une comparaison, mais une soustraction : avec CMP (et ses variantes) on soustrait la valeur à comparer de l'accumulateur (ou du registre concerné). Un indicateur est positionné selon le résultat de cette soustraction :

C - pour Carry - s'il est positif
Z - pour Zero - s'il est nul
N - pour Negative - s'il est négatif

Tous ces indicateurs (flags) sont combinés dans un même registre, le Mot d'Etat (Status Word). L'instruction suivante de branchement conditionnel examine ensuite l'état de l'indicateur qui l'intéresse.

BEQ (Branch if equal), par exemple, effectuera le branchement si la soustraction a donné pour résultat 0, c'est-à-dire si l'indicateur Z = 1. Par contre le branchement ne sera pas effectué si l'indicateur Z = 0. Surprenant, mais logique... Surtout si l'on considère que "0" veut dire Faux et que "1" veut dire Vrai.

Si la comparaison ne donne pas l'égalité c'est qu'il ne s'agit pas de CTRL-A. Retour au programme principal (RTS). Par contre si c'est un CTRL-A il convient de déterminer si celui-ci est accompagné d'une barre oblique ou d'un moins.

Chaque fois que la bonne séquence de touches aura été décelée, nous transformerons le code ASCII dans le code qui donnera le caractère désiré à l'impression. Cela est effectué par un LDA avec adressage immédiat, c'est à dire que l'on charge directement l'accumulateur avec la valeur ASCII qui servira au programme PRINTER. La différenciation provoquée par l'emploi préalable de la touche ESC (majuscules) sera effectuée dans le tronc commun du programme principal (TEDITOR classique).

8. SOUS-PROGRAMME

BAROBL:

```
LDA CLAVIER
BPL BAROBL          ;RIEN! RELIRE CLAVIER.
CMP £$AF           ;TOUCHE "/"?
BNE TIRET?
LDA £$DC
```

ACCEPTTE:

```
STA STROBE         ;LE CLAVIER EST PRET A
RTS                ;ACCEPTER UN CARACTERE
```

TIRET?:

```
CMP £$AD           ; TOUCHE "-"?
BNE ACCEPTTE       ; RIEN AU NUMERO.FAITES VOS JEUX!
LDA £$DF           ; SOULIGNERA A L'IMPRESSION
BNE ACCEPTTE       ; STROBE MET BIT7 A 0 POUR LIRE
STA (TEXTE), Y     ; RANGE LE CARACTERE TRAITE
INY                ; DANS LA ZONE-TAMPON
RTS
```

END

JSR (Jump to subroutine) exécute un branchement inconditionnel. Il est au JMP ce que le GOSUB est au GOTO. C'est-à-dire qu'il s'attend à trouver ultérieurement un RTS (Return from subroutine), l'équivalent du RETURN du BASIC.

BNE (Branch if not equal) ne doit guère poser de problème si on a compris le BEQ. Le branchement sera effectué si l'indicateur Z (dans le Mot d'Etat) est égal à zéro.

STA (Store accumulateur) est la manoeuvre inverse de LDA. Il transfère le contenu de l'accumulateur dans la mémoire spécifiée. Ici nous utilisons donc l'adressage indirect post-indexé par Y. Nous rangeons le contenu de l'accumulateur à l'adresse obtenue en ajoutant le contenu de la boîte aux lettres étiquetée "TEXTE" (d'après l'EPZ on sait qu'il s'agit de la cellule \$28 en hexa, soit 40 en décimal, qui indique le point de départ de la chaîne de caractères) au contenu du registre Y (qui, lui, progressera d'une unité au prochain INY).

LISA 2.5

CARACTERES FRANCAIS

```

0800          1          TTL "CARACTERES FRANCAIS"
1848          2          ORG $1848
1848          3          ;
0028          4  TEXTE   EPZ $0028
1501          5  DISQUE  EQU $1501
C000          6  CLAVIER EQU $C000
C010          7  STROBE  EQU $C010
1848 C9 81    8          CMP  $$81          ;TOUCHES CTRL-A?
184A F0 01    9          BEQ  BAROBL
184C 60       10         RTS
184D          11  BAROBL;
184D AD 00 C0 12         LDA  CLAVIER
1850 10 FB    13         BPL  BAROBL          ;RIEN! RELIRE CLAVIER.
1852 C9 AF    14         CMP  $$AF          ;TOUCHE "/"?
1854 D0 06    15         BNE  TIRET?
1856 A9 DC    16         LDA  $$DC
1858          17  ACCEPTE;
1858 8D 10 C0 18         STA  STROBE          ;LE CLAVIER EST PRET A
1858 60       19         RTS          ;ACCEPTER UN CARACTERE
185C          20  TIRET?;
185C C9 AD    21         CMP  $$AD          ; TOUCHE "-"?
185E D0 FB    22         BNE  ACCEPTE          ;RIEN AU NUMERO,A VOS MISES!
1860 A9 DF    23         LDA  $$DF          ;SOULIGNERA A L'IMPRESSION
1862 D0 F4    24         BNE  ACCEPTE          ;STROBE MET BIT7 A 0 POUR LIRE
1864 91 28    25         STA  (TEXTE),Y      ;RANGE LE CARACTERE TRAITE
1866 C8       26         INY          ;DANS LA ZONE-TAMPON
1867 60       27         RTS
1868          28         END

```

```

JCALL-151

*1848.1868

1848- C9 81 F0 01 60 AD 00 C0
1850- 10 FB C9 AF D0 06 A9 DC
1858- 8D 10 C0 60 C9 AD D0 FB
1860- A9 DF D0 F4 91 28 C8 60
1868- 00

```



apple en province

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

Anancy/Favergeres

74210

Bayonne

64100

Bordeaux

33000

Clermont-Ferrand

63000

Epernay

51200

Lyon

69003

Marseille 2

13000

Montpellier

34000

Nancy/Laxou

54250

Orléans

45000

Perpignan

66000

Rouen

76100

Strasbourg

67000

Toulon

83100

Valenciennes

59300

EUROPROCESS

Siège social : Doussard
(50) 44.31.12

LE CALCUL INTEGRAL

3, rue Aristide-Briand
(59) 55.43.47

BOUTISOFT 33

9, rue de la Lande
(56) 91.55.08

NEYRIAL

5, Bd Desaix
(73) 35.02.70

MAGENTA GESTION

7, av. A.-Thévenet Magenta
(26) 53.19.93

CIRCE

9, rue P.-Florence
(78) 54.31.95

ORDITEL

Siège social : BELCODEN
(42) 04.44.00

IFI-MICRO INFORMATIQUE

9-12, rue Castilhon
(67) 58.58.28

SEMITEC

69, rue Mareville
(8) 340.43.38

AMC

13, rue des Minimes
(38) 62.62.58

MAB

2, place de Catalogne
(68) 34.04.46

CONSEIL COMPUTER

20, quai Cavalier-de-la-Salle
(35) 63.36.06

CILEC

18, quai Saint-Nicolas
(88) 37.31.61

S I A

Lepaillon, avenue de Brunet
(94) 23.74.30

MICROMEGA

38, rue de Famars
(27) 46.89.22

Si cette publicité vous intéresse, contactez le
CALCUL INTEGRAL

Contrôlez le nettoyage mémoire

Guy Mathieu

Un article de Gérard MICHEL paru dans le numéro 2 de POM'S étudie l'effet de différentes méthodes de "Garbage Collection" pour nettoyer la mémoire centrale des vieilles chaînes qu'elle contient :

- "garbage" automatique par Applesoft,
- FRE (0) qui force le "garbage",
- d'autres "FRE", par exemple FRE (16) avec un programme approprié en langage machine.

Il peut être intéressant de voir de plus près comment fonctionne le "garbage collection", en vue de choisir la méthode la plus convaincante pour un programme donné. Une façon de le faire est de visualiser sur écran graphique les stades successifs de l'occupation mémoire, ce que fait le programme listé ci-après.

L'image écran dont le fac-similé est reproduit ici montre que le "garbage" automatique par Applesoft est fait au dernier moment, quand de la place doit effectivement être dégagée : il nécessite alors un certain temps d'attente (selon les programmes, ce temps, souvent chiffré en secondes, peut dépasser la minute).

Un "garbage" forcé régularise le processus et ne demande que des temps unitaires faibles. Mais bien souvent, la somme de ces temps unitaires atteint, et dépasse même, parfois notablement, le temps total exigé par le "garbage collection" automatique.

Le choix entre diverses façons de nettoyer la mémoire se pose par conséquent en des termes différents, car on ne peut plus parler d'optimum général.

En mode conversationnel, on préférera généralement des attentes courtes, même si le temps total est plus long : psychologiquement, c'est une solution mieux acceptée; techniquement, le temps le plus long est de toute façon celui qui dépend de l'opérateur.

Il n'en va pas de même pour les travaux en temps différé ou semi-différé (par ex. : chargement en mémoire centrale d'un gros fichier d'index).

Quelques commentaires sur le programme

- Les instructions 60 à 67 étudient la composition et la position sur écran de la chaîne de caractères destinée à former les nombres de l'échelle.
- Les instructions 50 à 59 constituent un sous-programme de dessin de chiffres.
- Les instructions 1500 à 1530 visent à mettre à égalité les 3 cas étudiés : un "LOMEM" remet à 0 la zone de pointeurs cependant que la mémoire 40185 (indestructible par le DOS) contient les valeurs successives d'une variable Z destinée à réinitialiser la "boucle J".
- Enfin, chacun connaît les vertus des mémoires 111 et 112 (instruction 81) qui contiennent la valeur décimale de la plus basse mémoire occupée à chaque instant par un tableau de "strings".

```

0 HOME
5 D$ = CHR$(4)
10 REM INITIALISATION
11 LOMEM: 16385: REM PROTECTION
    ZONE HGR
13 POKE 40185,50
15 DIM A$(150)
20 REM DESSIN DU CADRE
21 HGR : HCOLOR= 2
22 HPLOT 50,0 TO 50,150 TO 270,1
    50 TO 270,0
23 HPLOT 86,145 TO 86,150
24 HPLOT 186,145 TO 186,150
25 FOR Y = 0 TO 150 STEP 50: HPLOT
    50,Y TO 55,Y: NEXT
26 REM ECHELLES
27 FOR J = 0 TO 3
29 ZY% = 50 * J + 8
30 ZX% = 36 - J * 8: IF ZX% < 20 THEN
    ZX% = 20
31 ZA$ = STR$(J * 50)
32 GOSUB 60
33 NEXT J
34 ZX% = 70:ZY% = 159:ZA$ = "2000
    0": GOSUB 60
35 ZX% = 170:ZY% = 159:ZA$ = "300
    00": GOSUB 60
36 HCOLOR= 3: VTAB 23: PRINT "NB
    ITER.: - OCCUP. MEM.
    ;"
37 GOTO 1000
50 HPLOT ZX% - 2,ZY% TO ZX% + 2,
    ZY% TO ZX% + 2,ZY% - 6 TO ZX
    % - 2,ZY% - 6 TO ZX% - 2,ZY%
    : RETURN
51 HPLOT ZX%,ZY% TO ZX%,ZY% - 6 TO
    ZX% - 2,ZY% - 4: RETURN
52 HPLOT ZX% - 2,ZY% - 6 TO ZX% +
    2,ZY% - 6 TO ZX% + 2,ZY% - 4
    TO ZX% - 2,ZY% TO ZX% + 2,Z
    Y%: RETURN
53 HPLOT ZX% - 2,ZY% TO ZX% + 2,
    ZY% TO ZX% + 2,ZY% - 6 TO ZX
    % - 2,ZY% - 6: HPLOT ZX% - 2
    ,ZY% - 3 TO ZX% + 2,ZY% - 3:
    RETURN
54 HPLOT ZX% - 2,ZY% - 6 TO ZX% -
    2,ZY% - 2 TO ZX% + 2,ZY% - 2
    : HPLOT ZX%,ZY% - 4 TO ZX%,Z
    Y%: RETURN
55 HPLOT ZX% - 2,ZY% TO ZX% + 2,
    ZY% TO ZX% + 2,ZY% - 3 TO ZX
    % - 2,ZY% - 3 TO ZX% - 2,ZY%
    - 6 TO ZX% + 2,ZY% - 6: RETURN
56 HPLOT ZX% - 2,ZY% - 6 TO ZX% -
    2,ZY% TO ZX% + 2,ZY% TO ZX% +
    2,ZY% - 3 TO ZX% - 2,ZY% - 3
    : RETURN

```

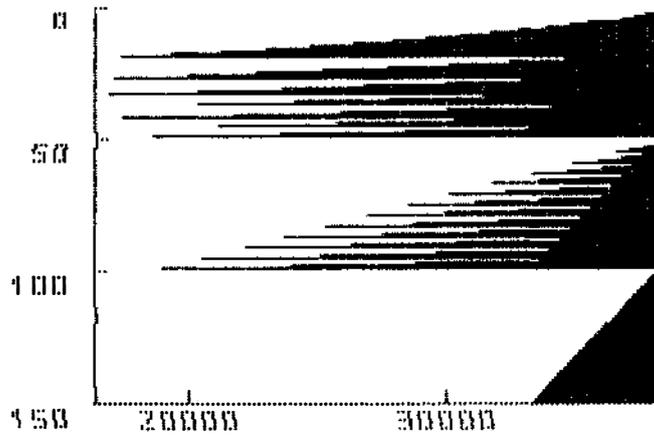
```

57 HPLOT ZX% - 2,ZY% TO ZX% + 2,
    ZY% - 6 TO ZX% - 2,ZY% - 6: RETURN
58 HPLOT ZX% - 2,ZY% TO ZX% - 2,
    ZY% - 6 TO ZX% + 2,ZY% - 6 TO
    ZX% + 2,ZY% TO ZX% - 2,ZY%: HPLOT
    ZX% - 2,ZY% - 3 TO ZX% + 2,Z
    Y% - 3: RETURN
59 HPLOT ZX% + 2,ZY% TO ZX% + 2,
    ZY% - 6 TO ZX% - 2,ZY% - 6 TO
    ZX% - 2,ZY% - 3 TO ZX% + 2,Z
    Y% - 3: RETURN
60 REM ECRITURE SUR ECRAN H.R.

61 Z2% = LEN (ZA$): FOR I = 1 TO
    Z2%
63 Z3% = ASC ( MID$ (ZA$,I,1)) -
    47
64 ON Z3% GOSUB 50,51,52,53,54,5
    5,56,57,58,59
65 ZX% = ZX% + 8
66 NEXT I
67 RETURN
80 REM TRACE CARTE MEMOIRE
81 MM = PEEK (112) * 256 + PEEK
    (111)
82 X = 50 + (MM - 16385) / 100
83 HPLOT X,J TO 270,J
84 VTAB 23: HTAB 12: PRINT J: VTAB
    23: HTAB 33: PRINT MM
85 RETURN
1000 REM CONSTITUTION DE STRINGS

1010 FOR J = 0 + Z TO 49 + Z
1020 A$(I) = ""
1030 FOR I = 1 TO 100
1040 A$(I) = A$(I) + "A"
1050 NEXT I
1060 IF J < 50 THEN VTAB 22: PRINT
    "'GARBAGE COLLECTION' AUTOMA
    TIQUE": GOSUB 80: GOTO 1500
1070 VTAB 1
1080 IF J > 99 THEN 1110
1090 IF J / 4 = INT (J / 4) THEN
    PRINT FRE (0): VTAB 22: PRINT
    "'FRE(0)' TOUTES LES 4 BOUCL
    ES
    "
1100 GOSUB 80: GOTO 1500
1110 PRINT FRE (0): VTAB 22: PRINT
    "'FRE(0) A CHAQUE BOUCLE ,
    "
1120 GOSUB 80
1500 NEXT J
1510 LOMEM: 16385
1520 DIM A$(101)
1530 Z = PEEK (40185): POKE 4018
    5,Z + 50: IF Z < = 100 THEN
    1000

```



Un exemple de HELLO

Guy Mathieu

Rien n'est aussi triste que le programme "HELLO" dont sont affligées tant de disquettes !

Un bon programme "HELLO" doit avoir deux caractéristiques :

- dire de quoi il s'agit (et ceci qu'il s'agisse d'un programme sophistiqué destiné à votre publicité, ou d'une simple disquette de travail utilisée chaque jour).
- permettre de faire ensuite... ce qu'il est logique de faire : un "RUN" sur un programme, ou l'obtention du catalogue.

Accessoirement, il n'est pas désagréable que le programme "HELLO" soit extrêmement simple à mettre au point.

En voici un exemple.

Le cadre est valable pour tous vos programmes : votre nom ou celui de votre société, un copyright, un sigle,...

Dans le corps du programme, 2 instructions seulement servent à personnaliser.

L'instruction 11 définira tous les titres :

DATA X\$, N, XI\$, X2\$, ...N[de 1 à 3]

Dans l'exemple numéro 1 ci-dessous :

X\$ = "DISQUE DE TRAVAIL"

N = 2

XI\$ = "PRESSER LA BARRE D'ESPACEMENT"

X2\$ = "POUR OBTENIR LE CATALOGUE"

L'instruction 300 lance la suite. Ici PRINT D\$ "CATALOG": END

Les choses ayant évolué, les programmes en cours de mise au point sont devenus opérationnels. Votre disquette est prête à être utilisée (peut-être à être commercialisée. Pourquoi pas ?).

L'instruction 11 devient :

DATA MON PROGRAMME, 3, PROGRAMME DESTINE, A MONTRER A TOUS, A QUEL POINT JE SAIS PROGRAMMER.

L'instruction 300 devient, elle :

PRINT D\$ "RUN MON PROGRAMME".

```

1 D$ = CHR$ (4)
10 REM DATA DANS L'ORDRE:NOM DE
    CODE DU PROGRAMME,NOMBRE DE
    LIGNES DE TEXTE,LIGNES DE TE
    XTE
11 DATA DISQUE DE TRAVAIL,2,PRE
    SSEZ LA BARRE D'ESPACEMENT,P
    OUR OBTENIR LE CATALOG
20 REM LECTURE DATA
21 READ T1$
22 L1 = LEN (T1$)
23 HT = INT (18 - L1 / 2)
24 T1$ = " " + T1$ + " "
25 T0$ = LEFT$ ("
                                ",L1 +
                                4)
30 READ NT
31 FOR I = 1 TO NT: READ T$(I): NEXT
50 EC = 3: IF NT = 4 THEN EC = 2
51 VT = 8: IF NT < = 2 THEN VT =
    10
100 HOME

```

```

110 INVERSE : FOR I = 1 TO 3 STEP
2: VTAB I: HTAB HT: PRINT T0
$: NEXT
120 VTAB 2: HTAB HT: INVERSE : PRINT
T1$
130 NORMAL
140 FOR I = 1 TO NT
150 VTAB VT: PRINT T$(I)
160 VT = VT + EC
170 NEXT
210 FOR I = 1 TO 4: PRINT : NEXT
220 VTAB 20: INVERSE : PRINT "CO
PIC"
240 NORMAL
250 PRINT "GROUPE"
260 PRINT "EUREQUIP
--> BARRE D'ESPAC."
270 PRINT "VAUCRESSON
    POUR CONTINUER"
280 IF PEEK ( - 16384) < = 127
    THEN 280
300 HOME : PRINT D$"CATALOG"

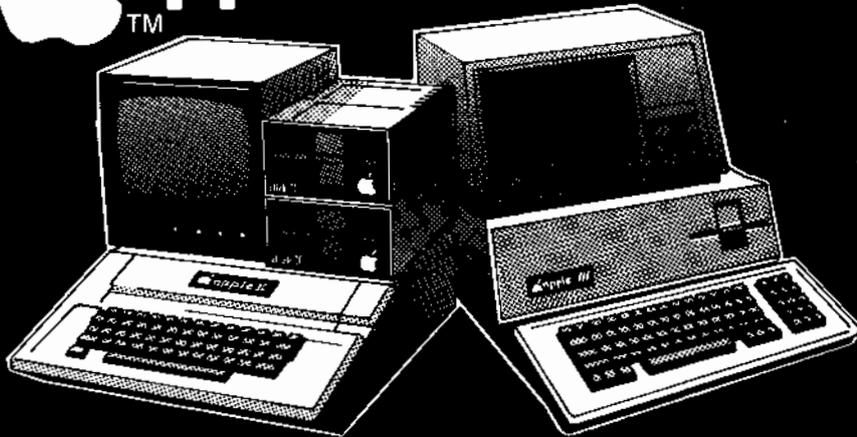
```

A TOULOUSE

100 M² D'EXPOSITION



apple II et III



SOUBIRON S.A.

En plein cœur de TOULOUSE
c'est :

- 100 m² d'exposition spécialisée
- Une équipe de techniciens à votre service pour :
 - Etudes
 - Programmes
 - Mise en place
 - Formation
 - Service après-vente

SOUBIRON S.A.

BOUTIQUE MICRO INFORMATIQUE LIBRAIRIE INFORMATIQUE

Tél. : (61) 21.64.39 - 21.04.57 . Telex LPS INF 521075 F

9, rue Kennedy . 31000 TOULOUSE

**COURS DE FORMATION
BASIC : 3 Jours**

CONTRAT D'ENTRETIEN : S.A.V. DEPANNAGE RAPIDE

Analyse du directory Pascal (II)

Michel Crimont

A. Présentation générale

Nous avons vu dans le précédent numéro comment lire directement le répertoire du disque Pascal sans passer par le "FILER". Nous allons voir maintenant comment tirer parti de ces procédures pour placer automatiquement dans un fichier unique tous les répertoires de nos disques Pascal, ce qui permettra une recherche beaucoup plus aisée; l'article du prochain numéro montrera en complément comment réaliser les procédures de sélection dans le catalogue.

Les types et variables précédemment définis restent valables, ainsi que les procédures et fonctions, excepté ECRIREP et LIREREP que nous n'utiliserons pas ici.

Il convient d'ajouter le type ELEMENT qui est une ligne du catalogue définie comme un RECORD comprenant :

GENRE : deux caractères servant à définir l'utilisation et le type de programme; par exemple J : Jeux, G : Graphisme, U : Utilitaire, M : Mathématiques -pour l'utilisation- et P : Programme, L : Librairie, U : Unit, etc. - pour le type -.
NOMFILE : nom du fichier.
NOMDISK : nom du disque comportant le fichier.
NUMBOITE : numéro de la boîte contenant le disque.
DATEFILE : date de création du fichier.

Les ELEMENTS sont placés dans un fichier nommé GARDE. L'item NUMBOITE de la fiche zéro conserve le nombre de fiches du fichier.

PAPIER : nom du fichier d'impression.
QUIT : Booléen d'arrêt du programme.
IMP : impression si Vrai.
NUMCAT : nombre de fiches du catalogue.
NUMLINE : numéro de la ligne courante sur l'écran.

Afin de rendre le programme très fiable, il est fait usage de diverses procédures filtrant tout ce qui est entré par l'utilisateur au clavier; par ailleurs, quelques procédures facilitent les affichages d'écran. C'est ainsi que :

MESSAGE : affiche une chaîne de caractères à la ligne Y et efface la fin de la ligne.
ALARME : efface l'écran, sonne, affiche un message d'erreur et attend un [Return].
PRENCHAINE et ENTIER : permettent de prendre au clavier en toute sécurité et avec correction éventuelle par la flèche arrière une chaîne de caractères ou un entier; enfin un [Return] à la place d'une entrée clavier renvoie la chaîne ou l'entier donné en Variable, ce qui permet de garder une valeur par défaut.
TRAIT et TRAITFIN : permettent l'ouverture et la fermeture du fichier impression avec édition d'un trait.
CONTINUE : permet la lecture d'un écran et attend un [Return] pour afficher la suite.
ZEROELEM : donne une valeur par défaut à Elément.
DECODATE : renvoie une chaîne de caractères standard correspondant au type

d'entrée "Date".

AFFICHE : place à l'écran une ligne du catalogue avec le numéro de la fiche.

IMPRIME : imprime une ligne du catalogue.

RAPPEL : rappelle à la fin du programme qu'il faut replacer le disque système dans le lecteur 1 [£4:].

TITRE : affiche ou imprime les têtes de colonne.

Le fonctionnement du programme est le suivant : une fois le programme chargé apparaissent 4 options :

E(nregistrer, L(ire, M(odifier, F(in.

Si l'on demande l'enregistrement, le disque dont le répertoire sera enregistré dans le catalogue sera placé dans le lecteur 1 [£4:], le disque contenant le catalogue général dans le lecteur 2 [£5:].

Afin d'éviter toute erreur, le disque du lecteur 2 devra porter un nom particulier, ici "CARNET", le plus simple étant de prendre un disque Pascal formaté, de lui donner le nom de "CARNET" et de placer dessus CATALOG.Code, si bien qu'il suffira lorsqu'on voudra ajouter un répertoire au catalogue de faire eX(ecute £5:CATALOG, le disque "CARNET" contenant ainsi à la fois le programme et le catalogue général.

La Lecture ne permet aujourd'hui que la lecture du catalogue général. Elle sera perfectionnée, pour permettre toutes les sélections imaginables, dans le prochain numéro.

La Modification permet de modifier les éléments d'une fiche; cette fiche apparaît à l'écran. Si un item ne doit pas être modifié, il suffit de taper le [Return] pour garder la même valeur, sinon on entre d'autres données.

B. Les procédures.

1. OUVERTURE

ouvre le fichier GARDE s'il existe et attribue à NUMCAT sa valeur (nombre de fiches du catalogue). Si le fichier n'existe pas, il est créé et la première fiche enregistrée (fiche non utilisée en dehors de NUMBOITE qui contient NUMCAT).

2. REMPLIR

est chargé de constituer une fiche à partir du répertoire du disque du £4: et des données complémentaires entrées au clavier. Il faut d'abord lire le répertoire avec LIREREP (4). Si ce répertoire ne peut être lu, la procédure envoie un message d'erreur.

S'il est lu, on prépare une fiche vierge avec ZEROELEM, puis avec ENTETE on place le nom du disque et l'on demande le numéro de la boîte où se range ce disque. Ces valeurs sont ensuite gardées dans LM pour tous les fichiers de ce disque, puis on fait appel à la procédure FILES qui propose un par un les fichiers et demande s'il faut les inclure dans le catalogue (ce qui permet d'inclure seulement les nouveaux fichiers d'un disque sans reprendre les anciens déjà enregistrés).

Si l'on garde un fichier, on garde son nom, sa date et l'on demande son type et son genre, puis on l'enregistre en place et l'on avance NUMCAT de 1. Quand le disque ne possède plus d'autre fichier, on ferme le catalogue pour se mettre à l'abri d'une coupure de courant ou d'un incident et on l'ouvre à nouveau.

3. ENREGISTRE

appelle simplement, de façon itérative la procédure REMPLIR et donne les messages d'écran nécessaires pour les manipulations.

4. LIRE et TRI

permettent simplement d'éditer le catalogue soit à l'écran, soit sur l'imprimante;

ces deux procédures seront par la suite développées pour permettre des sélections dans le catalogue.

5. MODIFIE

affiche un numéro quelconque du catalogue qu'il faudra donc noter auparavant par une lecture et permet de modifier un item quelconque dans le cas par exemple où l'on change un disque de boîte ou un fichier de disque, ou encore si un programme a été transformé en UNIT.

C. Le programme principal.

Le programme principal est très simple :

PRINFO : attribue les caractères de contrôle d'écran.

OUVERTURE : contrôle l'existence du disque "CARNET" dans le lecteur 2 et ouvre ou crée le fichier "DONNEES".

Ensuite, affichage du titre du programme et du nombre de lignes du catalogue, et attente des options.

A la fin, mise à jour de NUMCAT sur la fiche zéro de GARDE - fermeture du fichier et rappel d'insérer le disque système au cas où l'on aurait placé dans le lecteur 1 un autre disque.

Quelques perfectionnements sont possibles au gré des lecteurs :

1. Il est possible de placer dans le lecteur 2 un disque simplement formaté sans nom particulier, auquel cas, lors de l'ouverture, le programme demande un nom pour ce disque (avec PRECHAINED et 7 caractères maximum); grâce à la procédure ECRIREP, ce nom pourra être placé sur le répertoire du disque du lecteur 2. Le nom peut également être préalablement déclaré en constante dans le programme, mais attention si vous faites démarrer accidentellement votre programme avec un disque quelconque dans le lecteur 2 !

2. A la fin du programme, il est possible d'essayer d'ouvrir un fichier système du lecteur 1. Si IORESULT=0, le disque système est dans le lecteur 1, sinon l'on passe à RAPPEL.



LIBRAIRIE LA NACELLE

INFORMATIQUE • ÉLECTRONIQUE • AUTOMATISME • MICROPROCESSEUR

**TOUS OUVRAGES ET ABONNEMENTS
FRANÇAIS ET ÉTRANGERS**

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

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

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

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

```

PROGRAM CATALOG;

CONST MAXFILE      =77;

TYPE CRTCOMMAND    =(EFEOS,EFEOL,INVERSE,NORMAL,DROIT,GAUCHE,TETE);
CHOIDECA          =SET OF CHAR;
DATE              =PACKED RECORD
                  MOIS   :1..12;
                  JOUR   :1..31;
                  AN     :1..99
                  END;
DISQUE            =RECORD
                  A       :ARRAY[0..2] OF INTEGER;
                  NOMDISQUE :STRING[7];
                  B       :INTEGER;
                  NBRFICHE :INTEGER;
                  C       :INTEGER;
                  DATEDISQUE:DATE;
                  D,E     :INTEGER
                  END;
FICHIER          =RECORD
                  DEBFICHIER :INTEGER;
                  FINFICHIER :INTEGER;
                  TYPEFICHIER:0..7;
                  NOMFICHIER :STRING[15];
                  DERNBLOC   :INTEGER;
                  DATEFICHIER:DATE
                  END;
ELEMENT          =RECORD
                  GENRE      :ARRAY[1..2] OF CHAR;
                  NOMFILE    :STRING[15];
                  NOMDISK    :STRING[7];
                  NUMBOITE   :INTEGER;
                  DATEFILE   :DATE;
                  END;

VAR REPERTOIRE    :RECORD
                  VOL        :DISQUE;
                  CATALOG   :ARRAY[1..MAXFILE] OF FICHIER
                  END;
NUMLECTEUR       :INTEGER;
HOME,BS,EFL,EFB,SON,INV,NORM,CR:CHAR;
CH,COM           :CHAR;
PAPIER           :TEXT;
GARDE            :FILE OF ELEMENT;
NUMCAT,NUMLIGNE,I,J,K,L:INTEGER;
LM               :ELEMENT;
QUIT,IMP         :BOOLEAN;

(*!*)
(* PROCEDURES DIVERSES *)
(*#*)
FUNCTION LIREREP(NUM:INTEGER):BOOLEAN;
BEGIN
  (*$I-* )
  UNITREAD(NUM,REPERTOIRE,SIZEOF(REPERTOIRE),2);
  LIREREP:=IORESULT=0
  (*$I+* )
END;

PROCEDURE PRINFO;
BEGIN

```

```

HOME:=CHR(12);      EFL:=CHR(29);      BS :=CHR(8);      CR :=CHR(13);
SON :=CHR(7);      INV:=CHR(18);      NORM:=CHR(20);   EFB:=CHR(11)
END;

```

```

PROCEDURE PRENRETURN;
VAR SORT :CHAR;
BEGIN
  REPEAT
    READ(KEYBOARD,SORT)
  UNTIL EOLN(KEYBOARD)
END;

```

```

FUNCTION PRENCAR(BONSET:CHOIDECA):CHAR;
VAR CH :CHAR;
    BON :BOOLEAN;
BEGIN
  REPEAT
    READ(KEYBOARD,CH);
    IF EOLN(KEYBOARD) THEN CH:=CR;
    BON:=CH IN BONSET;
    IF NOT BON THEN WRITE(SON)
    ELSE IF CH IN [' ','.', '^'] THEN WRITE(CH)
  UNTIL BON;
  PRENCAR:=CH
END;

```

```

FUNCTION OUI:BOOLEAN;
BEGIN
  OUI:=PRENCAR(['O','N']) IN ['O']
END;

```

```

PROCEDURE MESSAGE(Y:INTEGER;S:STRING);
BEGIN
  GOTOXY(0,Y);WRITE(S,EFL)
END;

```

```

PROCEDURE ALARME(S:STRING);
BEGIN
  GOTOXY(0,1);WRITE(SON,EFB);
  MESSAGE(10,S);
  MESSAGE(12,'FAITES <RETURN> ');
  PRENRETURN
END;

```

```

PROCEDURE PRENCHaine(LONGMAX:INTEGER;BONSET:CHOIDECA;VAR S:STRING);
VAR S1 :STRING[1];
    CONT :STRING;
BEGIN
  S1:= ' ';
  CONT:='';
  REPEAT
    IF LENGTH(CONT)=0 THEN S1[1]:=PRENCAR(BONSET+[CR])
    ELSE IF LENGTH(CONT)=LONGMAX THEN S1[1]:=PRENCAR([CR,BS])
    ELSE S1[1]:=PRENCAR(BONSET+[CR,BS]);
    IF S1[1] IN BONSET THEN CONT:=CONCAT(CONT,S1)
    ELSE IF S1[1]=BS THEN
      BEGIN
        WRITE(BS,' ',BS);
        DELETE(CONT,LENGTH(CONT),1)
      END;
  UNTIL S1[1]=CR;
  IF LENGTH(CONT) <> 0 THEN S:=CONT

```

```

        ELSE WRITE(S)
    END;

PROCEDURE ENTIER(LONGMAX:INTEGER;VAR S:INTEGER);
VAR S1 :STRING[1];
    I :INTEGER;
    CONT :STRING;
    OKSET :CHOIDECA;
BEGIN
    OKSET:=[ '0'..'9' ];S1:= ' ';CONT:= '';
    REPEAT
        IF LENGTH(CONT)=0 THEN S1[1]:=PRENCAR(OKSET+[CR])
        ELSE IF LENGTH(CONT)=LONGMAX THEN S1[1]:=PRENCAR([CR,BS])
            ELSE S1[1]:=PRENCAR(OKSET+[CR,BS]);
        IF S1[1] IN OKSET THEN CONT:=CONCAT(CONT,S1)
        ELSE IF S1[1]=BS THEN
            BEGIN
                WRITE(BS,' ',BS);
                DELETE(CONT,LENGTH(CONT),1)
            END;
        UNTIL S1[1]=CR;
        IF LENGTH(CONT)<>0 THEN
            BEGIN
                S:=0;
                FOR I:=1 TO LENGTH(CONT) DO
                    BEGIN
                        S:=S*10;S:=S+(ORD(CONT[I])-ORD('0'))
                    END;
            END ELSE WRITE(S)
        END;
END;

```

```

PROCEDURE TRAIT;
VAR I :INTEGER;
BEGIN
    REWRITE(PAPIER,'PRINTER:');
    FOR I:=1 TO 79 DO WRITE(PAPIER,'=');
    WRITELN(PAPIER)
END;

```

```

PROCEDURE TRAITFIN;
VAR I:INTEGER;
BEGIN
    WRITELN(PAPIER);
    FOR I:=1 TO 79 DO WRITE(PAPIER,'=');
    FOR I:=1 TO 6 DO WRITELN(PAPIER);
    CLOSE(PAPIER);IMP:=FALSE
END;

```

```

PROCEDURE CONTINU;
BEGIN
    MESSAGE(22,'FOUR CONTINUER TAPEZ <RETURN> ');
    PRENRETURN;
    NUMLIGNE:=6;
    GOTOXY(0,5);WRITE(EOF)
END;

```

```

(*!*)
(* PROCEDURES GENERALES SPECIFIQUES DE MASTER CATALOGUE *)
(*!*)

```

```

PROCEDURE ZEROELEM(VAR L:ELEMENT);
BEGIN
    WITH L DO

```

```

BEGIN
  GENRE[1]:='Z';      GENRE[2]:='Z';
  NOMFILE :='';      NOMDISK :='';      NUMBOITE:=0;
  WITH DATEFILE DO
  BEGIN
    MOIS:=1;JOUR:=1;AN:=1
  END
END
END;

PROCEDURE DECODE(DT:DATE;VAR STDATE:STRING);
VAR ST      :STRING;

PROCEDURE FAITSTRING(I:INTEGER;VAR S:STRING);
BEGIN
  S:='';
  STR(I,S);
  IF LENGTH(S)=1 THEN S:=CONCAT('0',S)
END;

BEGIN
  STDATE:='';ST:='';
  WITH DT DO
  BEGIN
    FAITSTRING(JOUR,ST);
    STDATE:=CONCAT(STDATE,ST,'/');
    FAITSTRING(MOIS,ST);
    STDATE:=CONCAT(STDATE,ST,'/');
    FAITSTRING(AN,ST);
    STDATE:=CONCAT(STDATE,ST)
  END
END;

PROCEDURE AFFICHE(INDEX:INTEGER;VAR ELM:ELEMENT);
VAR S      :STRING;
BEGIN
  IF NUMLIGNE=21 THEN CONTINU;
  GOTOXY(0,NUMLIGNE);
  WITH ELM DO
  BEGIN
    WRITE(INDEX:4,GENRE[1]:2,GENRE[2],NOMFILE:16,NOMDISK:8,NUMBOITE:4);
    DECODE(DATEFILE,S);WRITE(S:9)
  END;
  NUMLIGNE:=NUMLIGNE+1
END;

PROCEDURE IMPRIME(INDEX:INTEGER;VAR ELM:ELEMENT);
VAR S      :STRING;
BEGIN
  WITH ELM DO
  BEGIN
    WRITE(PAPIER,INDEX:4,GENRE[1]:2,GENRE[2],NOMFILE:16,NOMDISK:8,NUMBOITE:4);
    DECODE(DATEFILE,S);WRITELN(PAPIER,S:9)
  END
END;

PROCEDURE RAPPEL;
BEGIN
  GOTOXY(0,3);WRITE(EFB);
  MESSAGE(10,'LE CATALOGUE A ETE PLACE DANS LE #5:');
  WRITE(INV,SON);
  MESSAGE(12,'REMETTRE LE DISQUE SYSTEME DANS #4:');

```

```

WRITE(NORM);
MESSAGE(14,'FAITES <RETURN> ');PRENRETURN
END;

PROCEDURE TITRE;
VAR S:STRING;
BEGIN
  S:=' ENR GT      NOM      DISQUE  BT  DATE';
  IF IMP THEN WRITELN(PAPIER,S) ELSE MESSAGE(4,S)
END;

(*!*)
(* PROCEDURES PROPRES AU PROGRAMME CATALOGUE *)
(*!*)
(*****
(* O U V E R T U R E                               *)
(* OUVRE LE CATALOGUE OU LE CREE SINON *)
(*****

PROCEDURE OUVERTURE;
VAR BON      :BOOLEAN;
BEGIN
  REPEAT
    BON:=FALSE;
    (*$I-*)
    RESET(GARDE,'CARNET:DONNEES');
    IF IORESULT=0 THEN
      BEGIN
        NUMCAT:=GARDE^.NUMBOITE;
        BON:=TRUE
      END ELSE
      BEGIN
        REWRITE(GARDE,'CARNET:DONNEES');
        ZEROLEM(GARDE^);
        GARDE^.NUMBOITE:=1;NUMCAT:=1;
        PUT(GARDE);
        CLOSE(GARDE,LOCK);
        RESET(GARDE,'CARNET:DONNEES');
        BON:=(IORESULT=0)
      END;
    IF NOT(BON) THEN ALARME('PLACER  ''CARNET''  DANS LE #5:');
  UNTIL BON
  (*$I+*)
END;
(*****
(* R E M P L I R                               *)
(* TRANSFERE LE REPERTOIRE DANS LE CATALOGUE *)
(*****

PROCEDURE REMPLIR;

PROCEDURE ENTETE;
BEGIN
  L:=LENGTH(REPERTOIRE.VOL.NOMDISQUE);
  IF L>0 THEN
    BEGIN
      LM,NOMDISK:=REPERTOIRE.VOL.NOMDISQUE;
      MESSAGE(7,'NOM DU DISQUE      : ');WRITE(LM,NOMDISK);
      L:=0;

```

```

    MESSAGE(8,'NUMERO DE LA BOITE ? ');ENTIER(4,LM.NUMBOITE);
END
END;

PROCEDURE FILES;
VAR TEMP    :ELEMENT;

PROCEDURE ENREGISTRE(L:ELEMENT);
BEGIN
    (*$I-* )
    SEEK(GARDE,NUMCAT);
    GARDE^:=L;
    PUT(GARDE);
    NUMCAT:=NUMCAT+1;
    IF IORESULT<>0 THEN
        BEGIN
            ALARME('ERREUR A LA COPIE...');
            EXIT(FILES)
        END
    (*$I+* )
END;

BEGIN
IF REPERTOIRE.VOL.NBRFICHE>0 THEN
BEGIN
    FOR I:=1 TO REPERTOIRE.VOL.NBRFICHE DO
    BEGIN
        TEMP:=LM;
        WITH REPERTOIRE.CATALOG[I] DO
        BEGIN
            GOTOXY(0,9);WRITE(EFB);
            MESSAGE(12,'INCLURE LE FICHER ');WRITE(NOMFICHER,' (O/N) ? ');
            IF OUI THEN
                BEGIN
                    TEMP.NOMFILE :=NOMFICHER;
                    TEMP.DATEFILE:=DATEFICHER;
                    MESSAGE(13,'TYPE    (A..Y) ? ');TEMP.GENRE[1]:=PRENCAR(['A'..'Y']);
                    MESSAGE(16,'GENRE   (A..Y) ? ');TEMP.GENRE[2]:=PRENCAR(['A'..'Y']);
                    ENREGISTRE(TEMP)
                END
            END;
        END
    END ELSE ALARME('PAS DE FICHER !')
END;

3BEGIN(* REMPLIR *)
    IF LIREREP(4) THEN
        BEGIN
            ZEROELEM(LM);
            ENTETE;
            FILES;
            CLOSE(GARDE,LOCK);RESET(GARDE,'CARNET:DONNEES')
        END ELSE ALARME('ERREUR LECTEUR 1.')
END;

PROCEDURE ENREGISTRE;
3BEGIN
    QUIT:=FALSE;
    WHILE NOT QUIT DO

```

```

BEGIN
  GOTOXY(0,2);WRITE(EFB);
  MESSAGE(3,'PLACER LA DISQUETTE A INCLURE DANS');
  MESSAGE(4,'LE CATALOGUE DANS LE LECTEUR 1 (#4:)'');
  MESSAGE(6,'QUAND TOUT EST PRET, FAITES <RETURN>');PRENRETURN;
  GOTOXY(0,5);WRITE(EFB);
  REMPLIR;
  MESSAGE(22,'CONTINUER ? (O/N) ');
  QUIT:=NOT OUI
END
END;

(*****
(* L I R E   &   T R I                               *)
(* PROCEDURES A AMELIORER                               *)
(*****

PROCEDURE TRI;
VAR I      :INTEGER;
BEGIN
  GOTOXY(0,3);WRITE(EFB);
  MESSAGE(3,'          EDITION DU CATALOGUE GENERAL');
  IF IMP THEN
    BEGIN
      MESSAGE(10,'          ***** SUR L''IMPRIMANTE *****');
      WRITELN(PAPIER,'EDITION DU CATALOGUE GENERAL':55);WRITELN(PAPIER)
    END;
  TITRE;
  FOR I:=1 TO NUMCAT-1 DO
    BEGIN
      GET(GARDE);
      IF IMP THEN IMPRIME(I,GARDE^) ELSE AFFICHE(I,GARDE^)
    END;
  MESSAGE(22,'C''EST TERMINE, TAPEZ <RETURN> ');PRENRETURN
END;

PROCEDURE LIRE;
VAR CHOIX  :CHAR;
BEGIN
  QUIT:=FALSE;
  GOTOXY(0,2);WRITE(EFB);
  MESSAGE(3,'LE CATALOGUE DOIT ETRE DANS LE DRIVE 2');
  WHILE NOT QUIT DO
    BEGIN
      GOTOXY(0,4);WRITE(EFB);NUMLIGNE:=6;
      MESSAGE(4,'EDITION COMPLETE');
      MESSAGE(6,'VOULEZ-VOUS IMPRIMER ? ');IMP:=OUI;
      IF IMP THEN TRAIT;
      SEEK(GARDE,1);
      TRI;
      IF IMP THEN TRAITFIN;
      MESSAGE(22,'UNE AUTRE RECHERCHE (O/N) ? ');QUIT:=NOT OUI
    END
  END;
END;
(*#*)
(*****
(* M O D I F I E                               *)
(* MISE A JOUR D'UN NUMERO DU CATALOGUE*)
(*****

```

```

PROCEDURE MODIFIE;
VAR N :INTEGER;
    CH :CHAR;
BEGIN
    GOTOXY(0,4);WRITE(EOF);N:=0;
    REPEAT
        MESSAGE(4,'NUMERO DU CATALOGUE A MODIFIER:');
        ENTIER(4,N);
    UNTIL (N<NUMCAT);
    IF N=0 THEN EXIT(MODIFIE);
    SEEK(GARDE,N);GET(GARDE);LM:=GARDE^;
    NUMLIGNE:=6;TITRE;AFFICHE(N,LM);
    MESSAGE(8,'MODIFIEZ OU <RETURN>');
    MESSAGE(9,'TYPE :');CH:=PRENCAR(['A'..'Y',CR]);
    IF(CH<>CR) THEN LM.GENREC[1]:=CH
        ELSE BEGIN GOTOXY(7,9);WRITE(LM.GENREC[1]) END;
    MESSAGE(10,'GENRE:');CH:=PRENCAR(['A'..'Y',CR]);
    IF (CH<>CR) THEN LM.GENREC[2]:=CH
        ELSE BEGIN GOTOXY(7,10);WRITE(LM.GENREC[2]) END;
    MESSAGE(11,'NOM DU FICHIER :');
    (*$V-*);
    PRENCHaine(15,['A'..'Z','0'..'9','.'],LM.NOMFILE);
    MESSAGE(12,'NOM DU DISQUE :');
    PRENCHaine(7,['A'..'Z','0'..'9','.'],LM.NOMDISK);
    (*$V+*);
    MESSAGE(13,'NUMERO DE LA BOITE:');
    ENTIER(4,LM.NUMBOITE);
    GARDE^:=LM;SEEK(GARDE,N);PUT(GARDE);
    CLOSE(GARDE,LOCK);RESET(GARDE,'CARNET:DONNEES');
END;
(*XXXXXXXXXXXXXXXXXXXXXXXXX PROGRAMME PRINCIPAL XXXXXXXXXXXXXXXXXXXXXXX*)
BEGIN
    PRINFO;
    OUVERTURE;
    REPEAT
        PAGE(OUTPUT);WRITE(INV);
        MESSAGE(0,'          *** SUPER CATALOGUE ***');
        WRITE(NORM);
        MESSAGE(5,' ');WRITE('IL Y A <',NUMCAT-1,> TITRES DISPONIBLES. ');
        MESSAGE(22,'E(NREGISTRER, L(IRE, M(ODIFIER, F(IN ');
        COM:=PRENCAR(['E','L','M','F']);
        CASE COM OF
            'E':ENREGISTRE;
            'L':LIRE;
            'M':MODIFIE
        END;
        UNTIL COM='F';
        SEEK(GARDE,0);GARDE^.NUMBOITE:=NUMCAT;PUT(GARDE);
        CLOSE(GARDE,LOCK);
    RAPPEL
END.

```

Notions de base

Gérard Michel

Depuis la parution du premier numéro de POM'S, de nombreux lecteurs ont pris contact avec la rédaction pour exprimer leur embarras face au décalage (provisoire ...) existant parfois entre le niveau de leurs connaissances et celui de certains articles publiés dans la revue. Fort logiquement, ils ont aussi formulé un souhait : trouver dans POM'S quelques informations de base susceptibles de les guider dans l'exploration de leur APPLE et de faciliter l'exploitation des articles "techniques".

Cette revendication est légitime ! S'étant donné pour objectif d'informer, si possible, TOUS les utilisateurs d'APPLE, votre revue ne pouvait y rester insensible. Aussi tenterons nous désormais d'assurer une rubrique destinée à répondre aux questions fondamentales posées par les lecteurs dans cette optique. Nous prions simplement les "experts" de bien vouloir sauter allègrement ces quelques lignes, témoignages d'une époque pour eux révolue ...

Pour ce coup d'essai, nous traiterons des problèmes touchant à la saisie des programmes en langage-machine et à la manipulation des fichiers correspondants. Vous avez été nombreux, en effet, à regretter de ne pouvoir utiliser les programmes de ce type listés dans la revue, faute de pouvoir les rentrer en machine. Pour clarifier la suite de l'exposé, il peut être utile de rappeler ici quelques notions (très élémentaires) concernant le système numérique à base 16 (le fameux HEXAdécimal !).

En fait, il nous suffit de savoir que :

- 1) Il comporte 16 'chiffres' : 0, 1, ..9, A(10), B(11), C(12), D(13), E(14), F(15).
- 2) WXYZ en Hexa est égal à : $(W*4096)+(X*256)+(Y*16)+Z$ en Décimal (exemple : 927C en Hexa égale 37500 en Décimal).
- 3) A l'inverse, 37500 en Décimal vaut en Hexa :
 - . $INT(37500/4096) = 9$
 - . $INT((37500 - (9*4096))/256) = 2$
 - . $INT((37500 - (9*4096) - (2*256))/16) = 7$
 - . $37500 - (9*4096) - (2*256) - (7*16) = 12$, soit C
- 4) Par convention, les nombres HEXA sont précédés d'un "\$".

1. Saisie d'un programme en langage-machine (codes hexadécimaux).

Prenons l'exemple du petit programme présenté par J.-L. Meillaud. La marche à suivre se décompose ainsi :

a) Sortir du BASIC pour rentrer dans le moniteur. Pour cela, tapez CALL-151 [Return]. Une "*" apparait, vous êtes en mode moniteur. Attention, cet animal étrange ne comprend que l'Hexadécimal !

b) Pour saisir les instructions, la séquence est "Adresse (en HEXA, bien sûr!)" puis ":" puis "codes hexadécimaux des instructions". Vous taperez ainsi :

```
1848: C9 81 F0 01 60 AD 00 C0 [Return]
1850: 10 FB ...
.
.
1860: A9 DF ..... 60 [Return]
```



L'ORDINATEUR PERSONNEL DE M^R LE DIRECTEUR DES VENTES.

Apple, c'est d'abord un nouveau concept. Celui de la relation individuelle et directe entre l'homme et sa machine. Un concept qui ouvre une troisième voie entre rien du tout et l'informatique de millionnaire. Apple, c'est un outil personnel aussi facile à utiliser qu'une voiture, aussi léger et peu encombrant qu'une simple machine à écrire et aussi peu coûteux qu'une photocopieuse. Un outil qui vous simplifie la vie en diminuant la paperasse qui vous encombre, efficace et rapide dès qu'une décision s'impose.

Apple vous aide à répondre à des questions de type "qu'arriverait-il si?", traite vos problèmes de gestion (planification, prévisions, budget) et met en forme et imprime vos lettres et vos rapports. Apple est un outil versatile qui fonctionne sur une vaste gamme de programmes que vous soyez avocat ou médecin, responsable des ventes ou du marketing, comptable ou financier.

Intéressé, un peu sceptique ou dubitatif... Lorsque votre revendeur Apple local aura résolu devant vous quelques-uns de vos problèmes, ou vous aura fait utiliser vous-même un programme de planification ou une analyse sous forme de graphique couleur, vous aurez compris l'intérêt d'avoir à portée de la main la puissance de l'ordinateur personnel.

Et vous commencerez à penser à votre Apple.

Pour obtenir une information complète sur les systèmes d'ordinateur Apple II et Apple III, veuillez s'il vous plaît retourner ce coupon à : SEEDRIN SARL.

Avenue de l'Océanie, Z.I. de Courtabœuf, 91940 Les Ulis

Nom

Société

Fonction

Adresse

Téléphone

EXN1



L'ordinateur personnel

Si vous voulez lister votre programme, tapez :

1848.1867 [Return] (de façon générale, adresse début.adresse fin).

c) Pour retourner au BASIC, tapez 3DOG [Return]. Le "prompt" du BASIC réapparaît.

2. Sauvegarde d'un programme sur disquette après retour au BASIC.

La syntaxe est alors la suivante :

- . BSAVE NOM,A\$X,L\$Y où X désigne l'adresse de départ du programme en HEXA, Y la longueur du programme (nombre d'octets) exprimée en HEXA, et NOM tout simplement le nom que vous désirez donner au programme.
- . ou encore BSAVE NOM,AX,LY où X et Y sont exprimés en décimal.

L'adresse de départ ne pose pas de problème; seule la longueur réclame quelque attention. Si vous ne voulez pas vous livrer aux joies de l'arithmétique hexadécimale, le plus simple est encore de raisonner en décimal. Ainsi, dans notre exemple (trivial) :

- . Adresse de départ : 1848 HEXA = 6216 en décimal
- . Adresse de fin : 1867 HEXA = 6247 en décimal
- . Longueur : 6247-6216+1=32 (vous pouvez aussi compter les octets ligne à ligne ...). N'oubliez pas de rajouter un à la différence : c'est le fameux problème des intervalles entre les arbres...
- . Sauvegarde : BSAVE NOM,A6216,L32

3. Chargement d'un programme à partir de la disquette.

Il vous suffit de taper BLOAD NOM. L'adresse de départ et la longueur ont été mémorisées au moment de la sauvegarde. Si vous désirez les retrouver, tapez, après le BLOAD :

- . Pour l'adresse : PRINT PEEK(43634) + 256 * PEEK(43635)
- . Pour la longueur : PRINT PEEK(43616) + 256 * PEEK(43617)

Les deux valeurs vous seront données en décimal.

De même, BRUN NOM permet de charger le programme NOM en mémoire et de lancer son exécution (comme le RUN du BASIC).

Une fois le programme chargé, si vous voulez vérifiez les codes hexadécimaux, entrez dans le moniteur et tapez, comme ci-dessus, "adresse début.adresse fin" (adresses hexadécimales toujours).

Enfin, supposons que vous ayez sur disquette un programme NOM en langage-machine (routine de nettoyage mémoire, routine de tri ...), commençant à l'adresse 37000 (en décimal), et que vous souhaitiez l'utiliser dans le cours d'un programme BASIC. Les choses pourraient se présenter ainsi :

```
10 D$=CHR$(4)
20 HIMEM:36500 (pour protéger la routine en langage-machine)
30 PRINT D$"BLOAD NOM,S6,D1"
.
.
1000 CALL 37000 (exécution de la routine en langage-machine)
.
```

Suite du programme BASIC

Conversion Pascal/BASIC/Pascal

Gilles Mauffrey

1. Principes généraux.

Il était utile de pouvoir regrouper sur une même disquette pour les lecteurs de Pom's des programmes d'origines diverses : Pascal ou BASIC. Le programme présenté ici répond à cet objectif. Avant de décrire brièvement le programme, nous allons mettre en évidence les principales différences entre le DOS 3.3 et le Pascal Operating System (P.O.S.).

Tout d'abord le DOS raisonne en piste et secteur, tandis que le POS ne connaît théoriquement que les blocs de 512 octets. Un bloc correspond en fait à deux secteurs consécutifs. De plus la sectorisation soft des pistes est différente (voir "Beneath Apple DOS"):

- pour le DOS 3.3, les secteurs sont numérotés 0, 13, 11, 9, 7, 5, 3, 1, 14, 12, 10, 8, 6, 4, 2, 15.
- pour le POS, la numérotation est 0, 2, 4, 6, 8, 10, 12, 14, 1, 3, 5, 7, 9, 11, 13, 15.

Toutefois, le DOS 3.3 permet de lire une disquette Pascal, puisque la table de sectorisation se trouve aux adresses \$BFB8 et suivantes.

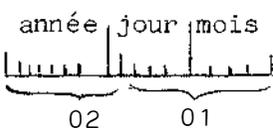
L'implantation des "directorys" est également différente. Sur une disquette Pascal, le directory commence au bloc 2 et se termine au bloc 6; chaque enregistrement (fichier ou en-tête) occupant 26 octets. Pour le DOS 3.3 le directory débute en piste 17 secteur 0, chaque fichier occupant 35 octets.

Une autre différence fondamentale concerne l'implémentation des fichiers eux-mêmes. En DOS 3.3, les fichiers ne sont pas nécessairement sur des secteurs consécutifs, tandis que le POS écrit les fichiers sur des blocs consécutifs, ce qui simplifie beaucoup la lecture de ces fichiers en assembleur.

2. Description des enregistrements "fichier" du directory Pascal.

Comme il a été dit plus haut, chaque enregistrement du directory comprend 26 octets; nous ne décrivons ici que les enregistrements concernant les fichiers. Traduisant la description de M. Crimont (Pom's 2), nous obtenons:

- Bloc de début de fichier : 2 octets (exemple bloc 6 : 06 00)
- Bloc de "fin" de fichier : 2 octets (en fait c'est le bloc de début du fichier suivant)
- Type du fichier : 2 octets
- Nom du fichier : 16 octets (1 octet pour la longueur du nom et 15 octets pour le nom)
- Nombre d'octets inutilisés dans le dernier bloc: 2 octets (toujours 00 02 car ce bloc est le début du fichier suivant)
- Date de création du fichier : 3 octets (1 octet pour la longueur 02 et 2 octets pour la date elle-même 01 02)



7 bits pour l'année, 5 pour le jour et 4 pour le mois

3. Déroulement du programme.

Pour simplifier l'écriture du fichier sur la disquette BASIC, nous avons utilisé la commande BSAVE du DOS. Ceci a pour inconvénient de limiter la longueur des fichiers transportables par ce programme; cependant ceci ne s'est pas avéré gênant pour l'utilisation que nous en avons faite.

Après avoir demandé le nom du fichier Pascal à transférer et le nom du fichier "BASIC" transféré, le programme vérifie l'existence du fichier sur la disquette Pascal; en cas de fichier inexistant, le programme s'arrête. Une fois trouvé, le fichier est chargé en mémoire à partir de l'adresse \$2000, puis chargé sur la disquette "BASIC".

Quatre sous-programmes sont utilisés :

- Le sous-programme PRERWT est appelé avant chaque appel à RWTS, pour initialiser la table.
- Le sous-programme TABPS charge la sectorisation Pascal.
- Le sous-programme TABDO charge la sectorisation DOS.
- Le sous-programme CALPSE transforme le numéro de bloc en numéros de piste et de secteur.

Nous n'analyserons pas ici le programme permettant de transférer le fichier "BASIC" sur une disquette Pascal dans la mesure où la démarche suivie est pratiquement la même. Les lecteurs courageux pourront cependant l'améliorer en permettant d'introduire la date, puisque ce programme prend la date du dernier fichier existant sur la disquette Pascal, ce qui peut poser des problèmes en cas de disquette vierge.

```
SOURCE FILE: PASCBAS
----- NEXT OBJECT FILE NAME IS PASCBAS.OBJO
0900:          1          ORG  $900
FD8E:          2 CROUT  EQU  $FD8E      * <RETURN>
FDED:          3 COUT   EQU  $FDED      * ECRITURE ECRAN
FD35:          4 RDCHAR EQU  $FD35      * LECTURE CLAVIER
FC9C:          5 CLREOL EQU  $FC9C
FC58:          6 HOME   EQU  $FC58
03F2:          7 RESET  EQU  $3F2
0900:          8 *
0900:          9 * PARAMETRES POUR RWTS
0900:         10 *
0000:         11          DSECT
0000:         12 RPLIOB  DS    1
0001:         13 RPLSLT  DS    1
0002:         14 RPLDRV  DS    1
0003:         15 RPLVOL  DS    1
0004:         16 RPLTRK  DS    1
0005:         17 RPLSEC  DS    1
0006:         18 RPLDCT  DS    2
0008:         19 RPLBUF  DS    2
000A:         20 RPLSIZ  DS    2
000C:         21 RPLCMD  DS    1
0001:         22 READ   EQU  $01
0002:         23 WRIT   EQU  $02
000D:         24 RPLCRD  DS    1
000E:         25 RPLTVL  DS    1
000F:         26 RPLPSL  DS    1
0010:         27 RPLPDR  DS    1
0900:         28          DEND
0900:         29 *
0900:         30 * UTILISATION DE LA PAGE ZERO
```

```

0900:      31 *
0000:      32 PTR      EQU   $0          * POINTEUR GENERAL
0002:      33 PISTE    EQU   $2
0003:      34 SECTEUR  EQU   $3
0004:      35 CMD      EQU   $4
0005:      36 DRIVE   EQU   $5
0007:      37 HBUF     EQU   $7
0008:      38 TEMP     EQU   $8
0048:      39 FREG     EQU   $48        * A REMETTRE A 0 APRES RWTS
0900:      40 *
0900:      41 * AUTRES ADRESSES
0900:      42 *
2000:      43 BUFFER   EQU   $2000
USE3:      44 LOCPRL   EQU   $3E3        *CHERCHE L'ADRESSE DE LA TABLE POUR RW
D3D9:      45 RWTS    EQU   $3D9
BFB8:      46 SECDES   EQU   $BFB8      *DEFINITION "SOFT" DES SECTEURS DOS 3.

```

```

D900:      48 *
0900:      49 * DEBUT PROGRAMME
D900:      50 *
0900:20 58 FC      51 CONVER   JSR   HOME
D903:A9 20        52           LDA   #<BUFFER
D905:85 07        53           STA   HBUF
D907:A2 00        54           LDX   #00
D909:BD 16 0C     55 ECRI     LDA   MESS,X
D90C:C9 BE       56           CMP   #' ;
D90E:F0 06       57           BEQ   ENTREE
D910:20 ED FD     58           JSR   COUT
D913:E8          59           INX
D914:D0 F3       60           BNE   ECRI
D916:A2 00       61 ENTREE   LDX   #0
D918:20 35 FD     62 MAJF    JSR   RDCHAR
D91B:C9 88       63           CMP   #$88
D91D:D0 0B       64           BNE   INBUF
D91F:E0 00       65           CPX   #00
D921:F0 F5       66           BEQ   MAJF
D923:20 ED FD     67           JSR   COUT
D926:CA          68           DEX
D927:4C 18 09     69           JMP   MAJF
D92A:C9 8D       70 INBUF    CMP   #$8D
D92C:F0 18       71           BEQ   FINF
D92E:C9 E0       72           CMP   #$E0
D930:90 02       73           BCC   ADDBUF
D932:29 DF       74           AND   #$DF
D934:49 80       75 ADDBUF   EOR   #$80
D936:9D F0 0C     76           STA   FICH,X
D939:49 80       77           EOR   #$80
D93B:20 ED FD     78           JSR   COUT
D93E:E8          79           INX
D93F:E0 0F       80           CPX   #15
D941:D0 D5       81           BNE   MAJF
D943:4C 73 0B     82           JMP   ERR1
D946:8A          83 FINF     TXA
D947:8D FF 0C     84           STA   LFIC
D94A:20 9C FC     85           JSR   CLREOL
D94D:20 8E FD     86           JSR   CROUT
D950:A2 00       87           LDX   #0
D952:BD 2F 0C     88 SUIDO    LDA   MEDO,X
D955:C9 BE       89           CMP   #' ;
D957:F0 07       90           BEQ   SADO
D959:20 ED FD     91           JSR   COUT
D95C:E8          92           INX
D95D:4C 52 09     93           JMP   SUIDO
D960:A2 00       94 SADO    LDA   #0
D962:20 35 FD     95 ENDOS   JSR   RDCHAR
D965:C9 88       96           CMP   #$88
D967:D0 0B       97           BNE   ACDOS
D969:E0 00       98           CPX   #$0
D96B:F0 F5       99           BEQ   ENDOS
D96D:20 ED FD    100          JSR   COUT
D970:CA          101          DEX
D971:4C 62 09    102          JMP   ENDOS
D974:C9 8D       103 ACDOS   CMP   #$8D
D976:F0 14       104          BEQ   FINDO
D978:C9 E0       105          CMP   #$E0
D97A:90 02       106          BCC   ADDOF
D97C:29 DF       107          AND   #$DF
D97E:9D 00 0D    108 ADDOF   STA   FIDO,X
D981:20 ED FD    109          JSR   COUT
D984:E8          110          INX
D985:E0 1E       111          CPX   #30
D987:D0 D9       112          BNE   ENDOS
D989:4C 73 0B    113          JMP   ERR1
D98C:8A          114 FINDO   TXA
D98D:8D 1F 0D    115          STA   LFID
D990:20 9C FC    116          JSR   CLREOL
D993:20 8E FD    117          JSR   CROUT
D996:A2 00       118          LDX   #0
D998:BD 59 0C    119 ECRMO   LDA   MODEMP,X
D99B:C9 BE       120          CMP   #' ;
D99D:F0 07       121          BEQ   FINDE
D99F:20 ED FD    122          JSR   COUT
D9A2:E8          123          INX
D9A3:4C 98 09    124          JMP   ECRMO
D9A6:20 8E FD    125 FINDE   JSR   CROUT
D9A9:A9 00       126 FIN2    LDA   #0
D9AB:20 35 FD    127          JSR   RDCHAR
D9AE:C9 8D       128          CMP   #$8D
D9B0:D0 F7       129          BNE   FIN2
D9B2:           130 *

```

```

09B2:      131 * SAUVEGARDE DE LA TABLE DOS
09B2:      132 *
09B2:A2 00 133          LDX #0
09B4:BD B8 BF 134 LOOP  LDA SECDES,X
09B7:9D 35 0D 135          STA DOSTAB,X
09BA:E8      136          INX
09BE:E0 10 137          CPX #16
09BD:D0 F5 138          BNE LOOP
09BF:      139 *
09BF:      140 * RECHERCHE DES POINTEURS PASCAL
09BF:      141 *
09BF:20 C5 0B 142          JSR TABPS
09C2:A9 00 143          LDA #0
09C4:85 02 144          STA PISTE
09C6:A9 04 145          LDA #4
09C8:85 03 146          STA SECTEUR
09CA:A9 01 147          LDA #1
09CC:85 05 148          STA DRIVE
09CE:A9 01 149          LDA #READ
09D0:85 04 150          STA CMD
09D2:20 94 0B 151          JSR PRERWT
09D5:      152 *
09D5:      153 * TRAITEMENT DU CATALOG PASCAL
09D5:      154 *
09D5:20 E3 03 155          JSR LOCPRL
09D8:20 D9 03 156          JSR RWTS
09DB:A9 00 157          LDA #0
09DD:85 48 158          STA PREG
09DF:AD 02 20 159          LDA BUFFER+2
09E2:0A      160          ASL A
09E3:85 08 161          STA TEMP
09E5:E6 03 162 LOOP1  INC SECTEUR
09E7:A5 03 163          LDA SECTEUR
09E9:C9 10 164          CMP #10
09EB:D0 06 165          BNE PLBU
09ED:A9 00 166          LDA #0
09EF:85 03 167          STA SECTEUR
09F1:E6 02 168          INC PISTE
09F3:E6 07 169 PLBU  INC HEUF
09F5:A9 01 170          LDA #READ
09F7:85 04 171          STA CMD
09F9:20 94 0B 172          JSR PRERWT
09FC:20 E3 03 173          JSR LOCPRL
09FF:20 D9 03 174          JSR RWTS
0A02:A9 00 175          LDA #0
0A04:85 48 176          STA PREG
0A06:C6 08 177          DEC TEMP
0A08:A5 08 178          LDA TEMP
0A0A:D0 D9 179          BNE LOOP1
0A0C:A9 20 180          LDA #<BUFFER
0A0E:85 07 181          STA HEUF
0A10:A9 00 182          LDA #>BUFFER
0A12:85 06 183          STA HEUF-1
0A14:AE 10 20 184          LDX BUFFER+#10
0A17:8A      185          TXA
0A18:18      186          CLC
0A19:A5 06 187          LDA HEUF-1
0A1B:69 06 188          ADC #06
0A1D:85 06 189          STA HEUF-1

```

A0 00	190 LP2	LDY #0
18	191	CLC
A9 1A	192	LDA #26
65 06	193	ADC HBUF-1
85 06	194	STA HBUF-1
90 02	195	BCC CNF
E6 07	196	INC HBUF
B1 06	197 CNF	LDA (HBUF-1),)
CD FF 0C	198	CMP LFIC
D0 13	199	BNE PASTR
AD FF 0C	200	LDA LFIC
85 00	201	STA PTR
C8	202 LP3	INY
B1 06	203	LDA (HBUF-1),'
D9 EF 0C	204	CMP FICH-1,Y
D0 06	205	BNE PASTR
C6 00	206	DEC PTR
F0 1E	207	BEQ TROUVE
D0 F2	208	BNE LP3
CA	209 PASTR	DEX
D0 D6	210	BNE LP2
20 8E FD	211	JSR CROUT
A2 00	212	LDX #0
BD C5 0C	213 ECPA	LDA PASFI,X
C9 BB	214	CMP #' ;
F0 0A	215	BEQ FIN1
20 ED FD	216	JSR COUT
E8	217	INX
4C 4E 0A	218	JMP ECPA
20 D3 0B	219	JSR TABDO
6C F2 03	220 FIN1	JMP (RESET)
38	221 TROUVE	SEC
A5 06	222	LDA HEUF-1
E9 06	223	SBC #6
85 06	224	STA HBUF-1

```

0A69:B0 02      225      ECS  ADR1
0A6B:C6 07      226      DEC  HBUF
0A6D:A0 00      227  ADR1  LDY  #0
0A6F:B1 06      228      LDA  (HBUF-1),Y
0A71:8D 21 0D   229      STA  DEFFA
0A74:C8         230      INY
0A75:B1 06      231      LDA  (HBUF-1),Y
0A77:8D 22 0D   232      STA  DEFFA+1
0A7A:C8         233      INY
0A7B:B1 06      234      LDA  (HBUF-1),Y
0A7D:8D 23 0D   235      STA  FIFPA
0A80:C8         236      INY
0A81:B1 06      237      LDA  (HBUF-1),Y
0A83:8D 24 0D   238      STA  FIFPA+1
0A86:         239 *
0A86:         240 * LES POINTEURS DE DEBUT ET FIN DE FICHIERS SONT CONSERVES
0A86:         241 *
0A86:18         242      CLC
0A87:A5 06      243      LDA  HBUF-1
0A89:69 16      244      ADC  #22
0A8B:85 06      245      STA  HBUF-1
0A8D:90 02      246      BCC  REST
0A8F:E6 07      247      INC  HBUF
0A91:A0 00      248  REST  LDY  #0
0A93:B1 06      249      LDA  (HBUF-1),Y
0A95:8D 45 0D   250      STA  RESPA
0A98:C8         251      INY
0A99:B1 06      252      LDA  (HBUF-1),Y
0A9B:8D 46 0D   253      STA  RESPA+1
0A9E:         254 *
0A9E:         255 * CALCUL DES SECTEURS ET PISTES
0A9E:         256 *
0A9E:A2 00      257      LDX  #0
0AA0:20 E1 0B   258      JSR  CALPSE
0AA3:A2 02      259      LDX  #2
0AA5:20 E1 0B   260      JSR  CALPSE
0AA8:38         261      SEC
0AA9:AD 23 0D   262      LDA  FIFPA
0AAC:ED 46 0D   263      SBC  RESPA+1
0AAF:8D 23 0D   264      STA  FIFPA
0AB2:B0 0B      265      ECS  PLULO
0AB4:CE 24 0D   266      DEC  FIFPA+1
0AB7:A9 0F      267      LDA  #$0F
0AB9:2D 23 0D   268      AND  FIFPA
0ABC:8D 23 0D   269      STA  FIFPA
0ABF:         270 *
0ABF:         271 * ECRITURE DU PROGRAMME A PARTIR DE $2000
0ABF:         272 *
0ABF:A9 20      273  PLULO  LDA  #<BUFFER
0AC1:85 07      274      STA  HBUF
0AC3:AD 21 0D   275      LDA  DEFFA
0AC6:85 03      276      STA  SECTEUR
0AC8:AD 22 0D   277      LDA  DEFFA+1
0ACB:85 02      278      STA  PISTE
0ACD:A9 01      279  L003  LDA  #READ
0ACF:85 04      280      STA  CMD
0AD1:20 94 0B   281      JSR  PRERWT
0AD4:20 E3 03   282      JSR  LOCPRL
0AD7:20 D9 03   283      JSR  RWTS
0ADA:A9 00      284      LDA  #0

```

0ADC:85 48	285	STA	PREG	0B63:A9 C4	344	LDA	#'D
0ADE:E6 03	286	INC	SECTEUR	0B65:20 ED FD	345	JSR	COUT
0AE0:A5 03	287	LDA	SECTEUR	0B68:A9 B2	346	LDA	#'2
0AE2:C9 0F	288	CMP	##0F	0B6A:20 ED FD	347	JSR	COUT
0AE4:D0 06	289	BNE	LABA	0B6D:20 8E FD	348	JSR	CROUT
0AE6:A9 00	290	LDA	#0	0B70:6C F2 03	349	JMP	(RESET)
0AE8:85 03	291	STA	SECTEUR	0B73:	351 *		
0AEA:E6 02	292	INC	PISTE	0B73:	352 *	SOUS ROUTINE	ERREUR
0AEC:A5 02	293	LDA	PISTE	0B73:	353 *		
0AEE:CD 24 0D	294	CMP	FIFPA+1	0B73:A2 00	354	ERR1	LDX #0
0AF1:90 0D	295	BCC	PREFI1	0B75:20 8E FD	355		JSR CROUT
0AF3:F0 02	296	BEQ	VERSEC	0B78:20 8E FD	356		JSR CROUT
0AF5:B0 0F	297	BES	PREBAS	0B7E:A9 87	357		LDA ##87
0AF7:A5 03	298	LDA	SECTEUR	0B7D:20 ED FD	358		JSR COUT
0AF9:CD 23 0D	299	CMP	FIFPA	0B80:BD 4A 0C	359	ERRC	LDA ERMES,X
0AFC:90 02	300	BCC	PREFI1	0B83:C9 BB	360		CMP #' ;
0AFE:B0 06	301	BES	PREBAS	0B85:F0 07	361		BEQ FINER
0B00:E6 07	302	INC	HBUF	0B87:20 ED FD	362		JSR COUT
0B02:90 C9	303	BCC	LOO3	0B8A:E8	363		INX
0B04:B0 C7	304	BES	LOO3	0B8B:4C 80 0B	364		JMP ERRC
0B06:20 D3 0B	305	JSR	TABDO	0B8E:20 8E FD	365	FINER	JSR CROUT
0B09:20 8E FD	306	JSR	CROUT	0B91:6C F2 03	366		JMP (RESET)
0B0C:A2 00	307	LDX	#0	0B94:	368 *		
0B0E:BD 47 0D	308	LDA	BLO,X	0B94:	369 *	PREPARATION DE RWTE	
0B11:C9 BB	309	CMP	#' ;	0B94:	370 *		
0B13:F0 07	310	BEQ	SUI1	0B94:20 E3 03	371	PRERWT	JSR LOCPRL
0B15:20 ED FD	311	JSR	COUT	0B97:84 00	372		STY PTR
0B18:E8	312	INX		0B99:85 01	373		STA PTR+1
0B19:4C 0E 0B	313	JMP	LOO5	0B9B:A5 02	374		LDA PISTE
0B1C:A2 00	314	LDX	#0	0B9D:A0 04	375		LDY #RPLTRK
0B1E:BD 00 0D	315	LDA	FIDO,X	0B9F:91 00	376		STA (PTR),Y
0B21:20 ED FD	316	JSR	COUT	0BA1:A5 03	377		LDA SECTEUR
0B24:E8	317	INX		0BA3:A0 05	378		LDY #RPLSEC
0B25:E0 1E	318	CPX	#30	0BA5:91 00	379		STA (PTR),Y
0B27:90 F5	319	BCC	LOO6	0BA7:A0 08	380		LDY #RPLBUF
0B29:A9 AC	320	LDA	#' ,	0BA9:A9 00	381		LDA #>BUFFER
0B2B:20 ED FD	321	JSR	COUT	0BAB:91 00	382		STA (PTR),Y
0B2E:A9 C1	322	LDA	#'A	0BAD:C8	383		INX
0B30:20 ED FD	323	JSR	COUT	0BAE:A5 07	384		LDA HBUF
0B33:A9 A4	324	LDA	#' \$	0BB0:91 00	385		STA (PTR),Y
0B35:20 ED FD	325	JSR	COUT	0BB2:A5 04	386		LDA CMD
0B38:A9 20	326	LDA	#<BUFFER	0BB4:A0 0C	387		LDY #RPLCMD
0B3A:20 DA FD	327	JSR	\$FDDA	0BB6:91 00	388		STA (PTR),Y
0B3D:A9 00	328	LDA	#>BUFFER	0BB8:A9 00	389		LDA #0
0B3F:20 DA FD	329	JSR	\$FDDA	0BEA:A0 03	390		LDY #RPLVOL
0B42:A9 AC	330	LDA	#' ,	0BBC:91 00	391		STA (PTR),Y
0B44:20 ED FD	331	JSR	COUT	0BBE:A5 05	392		LDA DRIVE
0B47:A9 CC	332	LDA	#'L	0BC0:A0 02	393		LDY #RPLDRV
0B49:20 ED FD	333	JSR	COUT	0BC2:91 00	394		STA (PTR),Y
0B4C:A9 A4	334	LDA	#' \$	0BC4:60	395		RTS
0B4E:20 ED FD	335	JSR	COUT				
0B51:38	336	SEC					
0B52:A5 07	337	LDA	HBUF				
0B54:E9 20	338	SBC	#<BUFFER				
0B56:20 DA FD	339	JSR	\$FDDA				
0B59:A9 FF	340	LDA	##FF				
0B5B:20 DA FD	341	JSR	\$FDDA				
0B5E:A9 AC	342	LDA	#' ,				
0B60:20 ED FD	343	JSR	COUT				

```

0BC5:          397 *
0BC5:          398 * MISE EN PLACE DE LA SECTORISATION PASCAL
0BC5:A2 00     399 TABPS   LDX  #0
0BC7:BD 25 0D  400 B1     LDA  PASTAB,X
0BCA:9D B8 BF  401         STA  SECDES,X
0BCD:E8        402         INX
0BCE:E0 10     403         CPX  #16
0BD0:D0 F5     404         BNE  B1
0BD2:60        405         RTS

0BD3:          407 *
0BD3:          408 * MISE EN PLACE DE LA SECTORISATION DOS 3.3
0BD3:          409 *
0BD3:A2 00     410 TABDO   LDX  #0
0BD5:BD 35 0D  411 B2     LDA  DOSTAB,X
0BD8:9D B8 BF  412         STA  SECDES,X
0BDB:E8        413         INX
0BDC:E0 10     414         CPX  #16
0BDE:D0 F5     415         BNE  B2
0BE0:60        416         RTS

0BE1:          418 *
0BE1:          419 * CONVERSION SECTEUR-BLOC
0BE1:          420 *
0BE1:A0 00     421 CALPSE  LDY  #0
0BE3:18        422         CLC
0BE4:A9 02     423         LDA  #02
0BE6:7D 21 0D  424         ADC  DEFFA,X
0BE9:9D 21 0D  425         STA  DEFFA,X
0BEC:90 03     426         BCC  BOUC0
0BEE:FE 22 0D  427         INC  DEFFA+1,X
0BF1:1E 22 0D  428 BOUC0   ASL  DEFFA+1,X
0BF4:1E 21 0D  429         ASL  DEFFA,X
0BF7:90 03     430         BCC  BOUC1
0BF9:FE 22 0D  431         INC  DEFFA+1,X
0BFC:BD 21 0D  432 BOUC1   LDA  DEFFA,X
0BFF:29 F0     433         AND  #$F0
0C01:18        434         CLC
0C02:6A        435         ROR  A
0C03:6A        436         ROR  A
0C04:6A        437         ROR  A
0C05:6A        438         ROR  A
0C06:18        439         CLC
0C07:7D 22 0D  440         ADC  DEFFA+1,X
0C0A:9D 22 0D  441         STA  DEFFA+1,X
0C0D:BD 21 0D  442         LDA  DEFFA,X
0C10:29 0F     443         AND  #$0F
0C12:9D 21 0D  444         STA  DEFFA,X
0C15:60        445         RTS

0C16:          447 *
0C16:          448 * MESSAGES ET CONSTANTES
0C16:          449 *

```

450 MESS ASC 'NOM DU FICHER PASCAL : ;
 451 MEDO ASC 'NOM DU FICHER "NORMAL" : ;
 452 ERMES ASC 'NOM TROP LONG;
 453 MODEMP ASC 'INSERER LA DISQUETTE PASCAL DANS LE DRIVE 1, LA
 DISQUETTE BASIC DANS LE DRIVE 2, PUIS APPUYER SUR "RETURN" ;
 454 PASFI ASC 'CE FICHER N EST PAS SUR CETTE DISQUETTE ;
 455 FICH ASC '
 456 LFIC DFB 0
 457 FIDO ASC '
 458 LFID DFB 0
 459 NFPAS DFB 0
 460 DEFPA DFB 0,0
 461 FIFPA DFB 0,0
 462 PASTAB DFB 0, 2, 4, 6, 8, 10, 12, 14, 1, 3, 5, 7, 9, 11, 13, 15
 463 DOSTAB DFB 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 464 RESPA DFB 0, 0
 465 BLO ASC 'BSAVE;

*900.D4D

RECAPITULATION

0900- 20 58 FC A9 20 85 07 A2
 0908- 00 BD 16 0C C9 BB F0 06
 0910- 20 ED FD E8 D0 F3 A2 00
 0918- 20 35 FD C9 88 D0 0B E0
 0920- 00 F0 F5 20 ED FD CA 4C
 0928- 18 09 C9 8D F0 18 C9 E0
 0930- 90 02 29 DF 49 80 9D F0
 0938- 0C 49 80 20 ED FD E8 E0
 0940- 0F D0 D5 4C 73 0B 8A 8D
 0948- FF 0C 20 9C FC 20 8E FD
 0950- A2 00 BD 2F 0C C9 BB F0
 0958- 07 20 ED FD E8 4C 52 09
 0960- A2 00 20 35 FD C9 88 D0
 0968- 0B E0 00 F0 F5 20 ED FD
 0970- CA 4C 62 09 C9 8D F0 14
 0978- C9 E0 90 02 29 DF 9D 00
 0980- 0D 20 ED FD E8 E0 1E D0
 0988- D9 4C 73 0B 8A 8D 1F 0D
 0990- 20 9C FC 20 8E FD A2 00
 0998- BD 59 0C C9 BB F0 07 20
 09A0- ED FD E8 4C 98 09 20 8E
 09A8- FD A9 00 20 35 FD C9 8D
 09B0- D0 F7 A2 00 BD B8 BF 9D
 09B8- 35 0D E8 E0 10 D0 F5 20
 09C0- C5 0B A9 00 85 02 A9 04
 09C8- 85 03 A9 01 85 05 A9 01
 09D0- 85 04 20 94 0B 20 E3 03
 09D8- 20 D9 03 A9 00 85 48 AD
 09E0- 02 20 0A 85 08 E6 03 A5
 09E8- 03 C9 10 D0 06 A9 00 85
 09F0- 03 E6 02 E6 07 A9 01 85
 09F8- 04 20 94 0B 20 E3 03 20
 0A00- D9 03 A9 00 85 48 C6 08
 0A08- A5 08 D0 D9 A9 20 85 07
 0A10- A9 00 85 06 AE 10 20 8A
 0A18- 18 A5 06 69 06 85 06 A0
 0A20- 00 18 A9 1A 65 06 85 06
 0A28- 90 02 E6 07 B1 06 CD FF

0A30- 0C D0 13 AD FF 0C 85 00
 0A38- C8 B1 06 D9 EF 0C D0 06
 0A40- C6 00 F0 1E D0 F2 CA D0
 0A48- D6 20 8E FD A2 00 BD C5
 0A50- 0C C9 BB F0 0A 20 ED FD
 0A58- E8 4C 4E 0A 20 D3 0B 6C
 0A60- F2 03 38 A5 06 E9 06 85
 0A68- 06 B0 02 C6 07 A0 00 B1
 0A70- 06 8D 21 0D C8 B1 06 8D
 0A78- 22 0D C8 B1 06 8D 23 0D
 0A80- C8 B1 06 8D 24 0D 18 A5
 0A88- 06 69 16 85 06 90 02 E6
 0A90- 07 A0 00 B1 06 8D 45 0D
 0A98- C8 B1 06 8D 46 0D A2 00
 0AA0- 20 E1 0B A2 02 20 E1 0B
 0AA8- 38 AD 23 0D ED 46 0D 8D
 0AB0- 23 0D B0 0B CE 24 0D A9
 0AB8- 0F 2D 23 0D 8D 23 0D A9
 0AC0- 20 85 07 AD 21 0D 85 03
 0AC8- AD 22 0D 85 02 A9 01 85
 0AD0- 04 20 94 0B 20 E3 03 20
 0AD8- D9 03 A9 00 85 48 E6 03
 0AE0- A5 03 C9 0F D0 06 A9 00
 0AE8- 85 03 E6 02 A5 02 CD 24
 0AF0- 0D 90 0D F0 02 B0 0F A5
 0AF8- 03 CD 23 0D 90 02 B0 06
 0B00- E6 07 90 C9 B0 C7 20 D3
 0B08- 0B 20 8E FD A2 00 BD 47
 0B10- 0D C9 BB F0 07 20 ED FD
 0B18- E8 4C 0E 0B A2 00 BD 00
 0B20- 0D 20 ED FD E8 E0 1E 90
 0B28- F5 A9 AC 20 ED FD A9 C1
 0B30- 20 ED FD A9 A4 20 ED FD
 0B38- A9 20 20 DA FD A9 00 20
 0B40- DA FD A9 AC 20 ED FD A9
 0B48- CC 20 ED FD A9 A4 20 ED
 0B50- FD 38 A5 07 E9 20 20 DA
 0B58- FD A9 FF 20 DA FD A9 AC
 0B60- 20 ED FD A9 C4 20 ED FD
 0B68- A9 B2 20 ED FD 20 8E FD

0B70- 6C F2 03 A2 00 20 8E FD
 0B78- 20 8E FD A9 87 20 ED FD
 0B80- BD 4A 0C C9 BB F0 07 20
 0B88- ED FD E8 4C 80 0B 20 8E
 0B90- FD 6C F2 03 20 E3 03 84
 0B98- 00 85 01 A5 02 A0 04 91
 0BA0- 00 A5 03 A0 05 91 00 A0
 0BA8- 08 A9 00 91 00 C8 A5 07
 0BB0- 91 00 A5 04 A0 0C 91 00
 0BB8- A9 00 A0 03 91 00 A5 05
 0BC0- A0 02 91 00 60 A2 00 ED
 0BC8- 25 0D 9D E8 BF E8 E0 10
 0BD0- D0 F5 60 A2 00 ED 35 0D
 0BD8- 9D E8 BF E8 E0 10 D0 F5
 0BE0- 60 A0 00 18 A9 02 7D 21
 0BE8- 0D 9D 21 0D 90 03 FE 22
 0BF0- 0D 1E 22 0D 1E 21 0D 90
 0BF8- 03 FE 22 0D BD 21 0D 29
 0C00- F0 18 6A 6A 6A 6A 18 7D
 0C08- 22 0D 9D 22 0D BD 21 0D
 0C10- 29 0F 9D 21 0D 60 CE CF
 0C18- CD A0 C4 D5 A0 C6 C9 C3
 0C20- C8 C9 C5 D2 A0 D0 C1 D3
 0C28- C3 C1 CC A0 BA A0 BB CE
 0C30- CF CD A0 C4 D5 A0 C6 C9
 0C38- C3 C8 C9 C5 D2 A0 A2 CE
 0C40- CF D2 CD C1 CC A2 A0 BA
 0C48- A0 BB CE CF CD A0 D4 D2
 0C50- CF D0 A0 CC CF CE C7 BB
 0C58- A0 C9 CE D3 C5 D2 C5 D2

0C60- A0 CC C1 A0 C4 C9 D3 D1
 0C68- D5 C5 D4 D4 C5 A0 D0 C1
 0C70- D3 C3 C1 CC A0 C4 C1 CE
 0C78- D3 A0 CC C5 A0 C4 D2 C9
 0C80- D6 C5 A0 B1 AC A0 CC C1
 0C88- A0 C4 C9 D3 D1 D5 C5 D4
 0C90- D4 C5 A0 C2 C1 D3 C9 C3
 0C98- A0 C4 C1 CE D3 A0 CC C5
 0CA0- A0 C4 D2 C9 D6 C5 A0 B2
 0CA8- AC A0 D0 D5 C9 D3 A0 C1
 0CE0- D0 D0 D5 D9 C5 D2 A0 D3
 0CE8- D5 D2 A0 A2 D2 C5 D4 D5
 0CC0- D2 CE A2 A0 BB C3 C5 A0
 0CC8- C6 C9 C3 C8 C9 C5 D2 A0
 0CD0- CE A0 C5 D3 D4 A0 D0 C1
 0CD8- D3 A0 D3 D5 D2 A0 C3 C5
 0CE0- D4 D4 C5 A0 C4 C9 D3 D1
 0CE8- D5 C5 D4 D4 C5 A0 BB A0
 0CF0- A0 A0 A0 A0 A0 A0 A0 A0
 0CF8- A0 A0 A0 A0 A0 A0 A0 00
 0D00- A0 A0 A0 A0 A0 A0 A0 A0
 0D08- A0 A0 A0 A0 A0 A0 A0 A0
 0D10- A0 A0 A0 A0 A0 A0 A0 A0
 0D18- A0 A0 A0 A0 A0 A0 A0 00
 0D20- 00 00 00 00 00 00 02 04
 0D28- 06 08 0A 0C 0E 01 03 05
 0D30- 07 09 0B 0D 0F 00 00 00
 0D38- 00 00 00 00 00 00 00 00
 0D40- 00 00 00 00 00 00 00 84
 0D48- C2 D3 C1 D6 C5 BB

logma

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

ÉTUDIE

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

FORME

- formation à l'utilisation de la micro-informatique

RÉALISE

- réalisation de programmes à la demande

LIVRE

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

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

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

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

Communication grâce à l'Apple

Bruno Estrangin

1-Introduction

Nouveau venu à l'informatique, mais ancien radio-amateur, j'ai été très vite tenté par les possibilités de communication de l'APPLE. Parmi des objectifs variés : accéder à des moyens informatiques supplémentaires, échanger des textes ou des programmes, interroger des banques de données..., ce qui m'intéresse le plus vivement est la perspective d'éviter toute reprise manuelle afin de profiter de toute la souplesse de l'informatique.

D'un point de vue non technique, communiquer avec un micro-ordinateur, c'est aussi expérimenter de nouvelles relations sociales. Ainsi, avec une certaine naïveté de néophyte, j'ai entrepris de communiquer. C'est passionnant et ça marche ! J'espère que les informaticiens spécialistes feront connaître ce domaine avec autant de moyens que pour le BASIC, le Pascal...

Je présente ci-dessous les réalisations suivantes :

échange de textes en environnement Pascal entre un APPLE et une machine bureautique RANK XEROX 850 pleine page,
utilisation de la même machine bureautique en tant que terminal, l'APPLE servant d'ordinateur,
accès à un réseau de temps partagé et au réseau MICRODIAL.

Je n'ai pas eu l'occasion de communiquer directement avec un autre APPLE, mais cela ne présente a priori aucune difficulté (entre APPLE, il est plus facile de se comprendre !).

2-Matériel utilisé

2-1 Coté APPLE

- Apple II, 48K avec carte langage,
- deux drives (il est possible de n'en utiliser qu'un),
- carte de communication : APPLE II COMMUNICATIONS INTERFACE CARD [A2B0003X],
- modem Anderson Jakobson AJ 311.

2-2 Coté bureautique

- Rank Xerox 850 pleine page,
- modem Anderson Jakobson AJ 211.

Une explication sur les MODEMS :

La transmission s'effectue par modulation de part et d'autre d'un canal (+/- 100hz). Par convention, le modem appelant émet dans le canal inférieur (1080 hz) et le modem appelé émet dans le canal supérieur (1750 hz).

Le modem AJ 211 ne fonctionne qu'en mode "APPEL" (ORIGINATE) : il "écoute" sur le canal supérieur et "émet" dans le canal inférieur. Le second modem utilisé devra donc "écouter" sur le canal inférieur et émettre dans le canal supérieur : on dit qu'il doit être en mode "REPONSE" (ANSWER).

Pour communiquer avec une machine munie d'un modem ne disposant que du mode "APPEL" (ORIGINATE), il faudra mettre le modem AJ 311 en mode "REPONSE" (ANSWER), soit :

- MODEM AJ 211 : mode "APPEL"
- MODEM AJ 311 : mode "REPONSE"

3-Echange de texte avec une machine bureautique

Plusieurs procédures sont disponibles. J'utilise une procédure dissymétrique du coté de l'APPLE :

émission sans logiciel particulier, en utilisant la fonction TRANSFER du FILER du POS de l'APPLE Pascal,

réception au moyen d'un logiciel de communication écrit en Pascal : le programme LIAISON dont on trouve le listing dans "LE PASCAL PAR LA PRATIQUE", pages 516 à 518, Pierre LEBEUX et Henri TAVERNIER, SYBEX.

Voici la procédure utilisée :

préparation du texte à l'aide de l'EDITOR de l'APPLE Pascal, et stockage sur disquette en notant son titre (facile à retrouver avec le FILER)

préparation de l'APPLE pour la communication :

- carte de communication dans le slot 2,
- liaison selon la norme RS 232 avec le modem acoustique,
- mise en service du modem, mode ORIGINATE, FULL DUPLEX.

appel du poste téléphonique de la machine bureautique pour vérifier que tout est également prêt du coté bureautique.

- Il suffit ensuite de passer au niveau du FILER les commandes :

- TRANSFER (nom du fichier)
- à la question TO WHERE, répondre "REMOUT:"; ceci envoie le fichier par la carte de communication au modem.
- Durant la transmission, rien n'apparaît sur l'écran; les commandes au clavier sont sans effet. A la fin de la transmission du fichier, on voit sur l'écran la flèche habituelle à l'issue d'un transfert.

Avec l'APPLE Pascal, la transmission d'un fichier n'est donc pas plus difficile que sa copie entre deux disquettes.

La préparation de la machine bureautique à la réception est spécifique de cette machine. J'indique ci-dessous la gamme opératoire propre à la machine RANK XEROX 850 pleine page.


```

=====
XOUTIQUE numero:003          REVISION numero:01 du 21/10/81 FOLIO: 2/2  X
X
X      -faire <CODE PRGR>                                           X
X      -enlever <4 MODE COM>                                         X
X      -mettre <LOCAL>                                               X
X      -faire <ACCEPT>                                               X
X      -creer un document (titre propose: FORMATCOM)                X
X      -remplacer le format standard par le format ci-dessus       X
X      -stocker le document                                          X
X      -revenir en MODE COM:                                         X
X          -<CODE PRGR>                                              X
X          -effacer <LOCAL>; mettre: <4 MODE COM>                    X
X          -touche <ACCEPT>                                          X
X
X      On peut aussi appeler un document comportant ce format, faire X
X      apparaitre ce format sur l'ecran, puis re-stocker le document X
X      Ce format sera ensuite mis sur tout nouveau document         X
X
X      13-Creer un nouveau document qui recevra la communication    X
X          -donner un titre: <TITRE>                                  X
X          -faire 2 fois <ACCEPT>; le curseur se met apres le format; X
X          l'ecran indique <COM PRET>                                 X
X
X      14-Appeler le poste distant                                   X
X
X      15-Disposer le combine sur le modem: cordon du cote de CORD  X
X
X      16-Verifier la reception de la porteuse: lampe verte allumee X
X
X      17-En cas de <PROBLEME TRANSMISSION>, appuyer sur la touche STOP X
X
X      18-A la fin de la transmission: stocker le document: touche STOCK X
X
X      19-Pour imprimer le document:                                  X
X          -depuis le sommaire disque, mettre le curseur sur le document X
X          -mettre le curseur qui est sur <TITRE>, sur <IMPRIMER>    X
X          (par la touche MOT)                                        X
X          -touche <ACCEPT>, une nouvelle ligne apparait           X
X          -avec la touche <MOT>, venir sur <ECHAPPEMENT>          X
X          -touche <ACCEPT>, faire apparaitre <10>                   X
X
X      20-Touche <CARAC>, acceder au sommaire de travail; le document X
X          est indique sous <IMPRIMANTE>                             X
X
X      21-Lancer l'impression par les touches <CODE PARA>;         X
X
=====

```

Cette procédure me permet d'imprimer les majuscules et minuscules et de garder la présentation du texte tel que je l'ai écrit sous l'EDITOR. Pour cela, il faut que la 850 P mette un code "LINE FEED" après chaque retour marge. En effet, la carte COMMUNICATION ne transmet pas les codes LINE FEED.

Si l'on transmet ce code (ASCII 10) au moyen d'un programme de communication comme le programme LIAISON, on observe sur la 850 P des effets inattendus : par exemple le code ASCII 10 (ou code J) est utilisé par la 850 P comme préfixe aux instructions de mise en page. Ainsi, la 850 P comprend (ASCII 10,E) comme signifiant un "retour marge code" et non comme un LINE FEED suivi de la lettre E.

La communication avec une machine bureautique, nécessite par conséquent que l'on s'informe sur les conventions utilisées.

Je prends une autre précaution en disposant les marges sur la 850 P : le chiffre indiquant la marge de gauche pour la 850 P est augmenté d'une unité par rapport à celui donné par SET ENVIRONNEMENT à RIGHT MARGIN.

Enfin, pour garder sur la 850 P son aspect au texte écrit sur l'APPLE, je dois supprimer l'"espacement proportionnel" de la 850 P en choisissant comme "échappement" sur la 850 P : 10 (pour mémoire, ceci signifie que les caractères seront séparés par 1/10 de pouce).

Toutes les surprises sont-elles évitées ? Oui, tant que l'on ne transmet que des lettres, des chiffres et des signes de ponctuation. J'indique ci-dessous les signes de l'APPLE qui sont interprétés différemment par la 850 P :

APPLE (Pascal)		850 P (FRANCAIS)
M = Majuscule		m = minuscule
S = Shift +		C = Ctrl + E = Esc +
R = Rept		CR = retl
clavier.....écran.....écran		
M+S+3	⌘	£ .
m+S+3	⌘	£ .
M+S+:	*	- .
m+S+:	*	- .
M+S+p	@	à .
m+S+p	@	à .
M+R		P .
M+S+R		à .
m+R		P .
m+S+R		à .
M+S+M]	\$.
m+S+M]	CR .
M+S+	<	“ .
m+S+	,	“ .
M+S+	>	^ .
m+S+	>	^ .

Récupération du texte sur l'APPLE

1- Du côté de la machine bureautique, il faut la préparer à l'émission en suivant les étapes 1 à 11 ci-dessus. Ensuite, on déplace le curseur à l'aide de la touche PARA pour surligner TRANSMISSION. Avec la touche LIGNE, on surligne le titre du document à transmettre, puis touche ACCEPT. Le titre du texte apparaît avec la mention EMISSION.

Il faut créer un fichier provisoire qui contiendra toute la communication :

donner un titre

ACCEPT, ACCEPT

On est devant une page blanche et il est possible de transmettre des caractères directement du clavier (l'écran indique "émission"). On lance l'émission du texte par la touche PAGE.

Il est indispensable d'émettre depuis la machine bureautique le code qui indiquera la fin de l'émission : code D pour le programme LIAISON, code C pour une réception sous la fonction TRANSFER du FILER.

2- Du côté de l'APPLE, il suffit de faire exécuter le programme LIAISON et de suivre les explications qu'il donne.

3- Une autre méthode de réception peut être utilisée en l'absence du programme LIAISON :

procéder comme ci-dessus pour la machine bureautique

sur l'APPLE, utiliser la fonction TRANSFER DU FILER ainsi :

-- TRANSFER

-- FROM REMIN:

-- TO WHERE: nom de fichier

le transfert est clos à la réception de CTRL-C

Toutefois, je ne sais pas actuellement saisir ce fichier avec le FILER : les informations contenues dans les deux premiers BLOCKS sont absentes. Ce fichier peut être transmis à l'imprimante (TRANSFER TO PRINTER:) et à l'écran (TRANSFER TO CONSOLE:).

4. Utilisation d'une machine bureautique comme terminal d'un APPLE

CONVERSATION TEMPS REEL ENTRE MACHINES

Du côté de l'APPLE, on se met en environnement BASIC et on suit les indications du manuel de la carte de communication. Plusieurs combinaisons sont possibles :

4-1 Conversation entre machines :

sur l'APPLE :

-- IN&2

-- CTRL A, CTRL F

sur la 850 P : suivre les étapes 1 à 18 de la procédure en encadré.

Tout ce qui est frappé sur un clavier apparaît sur l'écran de la machine émettrice ainsi que de la machine réceptrice; il n'y a aucune manoeuvre à faire pour passer du rôle d'émetteur à celui de récepteur.

4-2 Commande par la 850 P d'un programme exécuté par l'APPLE :

sur l'APPLE

-- PR&0

-- IN&2
-- CTRL A, CTRL F
sur la 850 P
-- CODE R pour accéder à l'unité centrale de l'APPLE et CODE T pour la libérer
- L'APPLE reprend la main par CTRL A, CTRL X.
La 850 P peut ainsi faire exécuter tout programme sur l'APPLE (EN BASIC), à condition de ne pas utiliser de fonction graphique. L'affichage s'effectue à la fois sur les deux écrans.

4-3 Montrer sur la 850 P le déroulement d'un programme commandé de l'APPLE :
sur l'APPLE :

-- PR&2
sur la 850 P : rien

Un programme exécuté sur l'APPLE est visualisé sur les deux machines.

5. Accès à un réseau

5-1 Cas d'un réseau de temps partagé :

On prépare l'APPLE par :

- IN&2
- CTRL A
- CTRL F

Il n'y a plus qu'à appeler le réseau, et à suivre la procédure de connexion propre à ce réseau (identification, mot de passe, nom du projet...). L'APPLE fonctionne comme un terminal. Toutefois, je ne sais pas encore garder une trace écrite des informations apparaissant sur l'écran.

5-2 Cas de MICRODIAL :

MICRODIAL est un ensemble de services pour micro-ordinateurs implanté sur le réseau EURODIAL de TELESYSTEME.

On peut se renseigner :

9 rue Huisman - 75006 - tel : 544 70 23

MICRODIAL est ouvert aux professionnels, mais également aux "hobystes" pour la somme de 30F/heure de connexion, tout compris. Les services sont en cours de développement. Il est déjà possible d'utiliser la MESSAGERIE pour recevoir et émettre des messages avec un autre membre du réseau. Autres fonctions implantées :

petites annonces
fichiers et édition de textes - jeux
bibliothèque de logiciels - annuaire

Une caractéristique très intéressante est de pouvoir envoyer sur MICRODIAL un fichier provenant d'une disquette d'un APPLE et réciproquement de réceptionner sur l'APPLE un fichier provenant de MICRODIAL. Il est ainsi possible de communiquer entre des micros différents (APPLE, TRS, GOUPIL).

La connexion se fait par TRANSPAC en utilisant un logiciel de communication fourni par MICRODIAL. Les utilisations de ce type de réseau ne sont limitées que par l'imagination des utilisateurs. Pour les lecteurs de POM'S actuels ou futurs "hobystes" sur MICRODIAL, mon numéro est : ECB 15086. A bientôt !

Copie d'écran-texte

Christian Guérin

Le programme en assembleur présenté ici vous permettra de faire de la copie d'écran-texte sur imprimante. Il transforme également les caractères INVERSE et FLASH pour l'impression, mais ne peut être utilisé avec une carte-imprimante qui affiche en même temps les données à l'écran.

Le petit programme BASIC ci-joint donne un exemple d'utilisation.

```
0800      1  ;
0800      2  ;
0000      3      ORG $0000
0000      4      OBJ $0800
0000      5  ;
0000      6  ;
0000      7  ;
0000      8  ;
0000      9  ;
0000     10  ;
0000     11  ;
0000     12  ;
0000     13  ;
0000     14  ;
0000     15  ADL   EPZ $6           ;ADRESSE DE DEBUT DE LIGNE
0000     16  LIGNE EPZ $8         ;NUMERO DE LIGNE
0000     17  ;
0000     18  OUT   EQU $FDED      ;ECRITURE D'UN CARACTERE
0000     19  ;
0000     20  ;
0000     21  ;
0000     22  ;-----
0000     23  ;
0000     24  ;
0000     25  ;
0000 A900  26  HCT   LDA #0         ;DEBUT HARD COPIE LIGNE 0
0002 48    27  HCT.1 PHA          ;SAUVEGARDE NUMERO LIGNE
0003      28  ;
0003      29  ;-----
0003      30  ;
0003      31  ;
0003      32  ;
0003 48    33          PHA
0004 4A    34          LSR
0005 2903  35          AND #$3
0007 0904  36          ORA #$4           ;ROUTINE VTAB
0009 8507  37          STA ADL+1
000B 68    38          PLA
000C 2918  39          AND #$18
000E 9002  40          BCC HCT.2
0010 697F  41          ADC #$7F
0012 8506  42  HCT.2  STA ADL
0014 0A    43          ASL
0015 0A    44          ASL
```


Routine de présentation graphique

Jean-François Duvivier

Si vous êtes passé sur notre stand au Sicob l'an dernier, ou si vous êtes abonné à nos disquettes, vous avez certainement remarqué la présentation de la disquette Pom's N 1.

Pour ceux qui ne l'on pas vue, disons qu'il s'agit d'un dédoublement d'image HGR et d'un "scrolling" d'une des deux images dédoublées. Le résultat est particulièrement esthétique pour des images peu chargées ou des logos, comme dans le cas de la disquette Pom's N 1.

Après de multiples améliorations, nous vous proposons ici un programme qui vous permettra de définir vous-même les effets que vous souhaitez. Ecrit en Assembleur, pour des raisons évidentes de rapidité, il occupe environ 150 octets.

Avant d'exécuter ce programme, il faut lui définir la page-écran sur laquelle il va travailler et dans laquelle vous avez chargé votre image, cette variable doit être placée en \$303 (771) et sa valeur est :

\$20 (32) pour la page 1
\$40 (64) pour la page 2

Il faut ensuite définir le séquençement du programme grâce à une zone située entre \$304 (772) et \$317 (791). La séquence définie peut avoir une longueur quelconque (mais dans les limites de la zone prévue), paire puisque chaque "action" élémentaire est définie par deux octets. La séquence est terminée par un zéro.

Le premier octet définit le nombre de déplacements effectués. Le second indique l'ampleur du déplacement.

Vite, un exemple, car je sens que ce n'est pas clair ! Une paire \$05-\$02 indique que le programme va faire 5 fois un déplacement de l'écran d'une ligne sur deux, ce qui correspond à un dédoublement parfait de l'image. Si, au lieu de \$02, vous mettez \$FE, c'est-à-dire son complément, le déplacement s'effectuera toujours une ligne sur deux, mais dans l'autre sens.

A noter qu'une paire \$XX-\$01 ou \$XX-\$FF fait un déplacement de XX lignes, 1 ligne sur 1, c'est-à-dire de toute l'image.

Vous pouvez essayer diverses valeurs pour obtenir le meilleur effet, mais vous avez toujours intérêt après un mouvement dans un sens à effectuer le mouvement complémentaire dans l'autre sens, du même nombre de lignes afin de retrouver le dessin original.

0800	1	;	XXXXXXXXXXXXXXXXXXXX	0800	12	NE	EPZ	\$01		
0800	2	;	X	*	0800	13	LIGNF	EPZ	\$02	
0800	3	;	X	GRAPHIQUE	*	0800	14	SAVX	EPZ	\$04
0800	4	;	X		*	0800	15	SENS	EPZ	\$05
0800	5	;	X	COPYLEFT	*	0800	16	LIGN	EPZ	\$26
0800	6	;	X		*	0800	17	HLIGN	EPZ	\$E2
0800	7	;	X	POM'S	*	0800	18	HPAG	EPZ	\$E6
0800	8	;	X		*	0800	19	HPOSN	EQU	\$F411
0800	9	;	XXXXXXXXXXXXXXXXXXXX	0800	20	HIRES1	EQU	\$C050		
0800	10	;		0800	21	HIRES2	EQU	\$C052		
0800	11		NOLIGN EPZ \$00	0800	22	HIRES3	EQU	\$C057		

```

0800      23 PAGE1 EQU $C054
0800      24 PAGE2 EQU $C055
0800      25 ;
0300      26          ORG $300
0300      27          OBJ $800
0300      28 ;
0300      29 ; DEBUT DU PROGRAMME
0300      30 ; =====
0300      31 ;
0300 4C1803 32          JMP DEBUT
0303      33 ;
0303      34 ; DONNEES
0303      35 ; =====
0303      36 ;
0303 20     37 PAGE    HEX 20
0818      38 SEQU    DFS 20
0318      39 ;
0318      40 ; PROGRAMME
0318      41 ; =====
0318      42 ;
0318 AD50C0 43 DEBUT   LDA HIRES1
0318 AD52C0 44         LDA HIRES2
031E AD57C0 45         LDA HIRES3
0321      46 ;
0321 A200   47         LDX $$00
0323 8604   48         STX SAVX
0325      49 ;
0325 AD54C0 50         LDA PAGE1
0328 AD0303 51         LDA PAGE
0328 C940   52         CMP  $$40
032D D003   53         BNE BOUCLE
032F AD55C0 54         LDA PAGE2
0332      55 BOUCLE:
0332 A900   56         LDA  $$00
0334 8500   57         STA NOLIGN
0336 85E2   58         STA HLIGN
0338 2011F4 59         JSR HPOSN
033B      60 ;
033B A604   61         LDX SAVX
033D E604   62         INC SAVX
033F E604   63         INC SAVX
0341 BD0403 64         LDA SEQU,X
0344 F051   65         BEQ FIN
0346 8501   66         STA NB
0348 E8     67         INX
0349 BD0403 68         LDA SEQU,X
034C 8505   69         STA SENS
034E      70 ;
034E A505   71 BOUCL2 LDA SENS
0350 1004   72         BPL BOUCL3
0352 A9C0   73         LDA  $!192
0354 8500   74         STA NOLIGN
0356      75 ;
0356 206203 76 BOUCL3 JSR TOUR
0359 C601   77         DEC NB
035B A501   78         LDA NB
035D D0EF   79         BNE BOUCL2
035F      80 ;
035F 4C3203 81         JMP BOUCLE
0362      82 ;
0362      83 ; REALISATION D'UN TOUR DE
0362      84 ; L'ECRAN SUIVANT LE SENS
0362      85 ; =====

```

```

0362      86 ;
0362 207303 87 TOUR   JSR CLIGN
0365      88 ;
0365 A027   89         LDY  $$27
0367 B126   90 TOUR1  LDA (LIGN),Y
0369 9102   91         STA (LIGNP),Y
036B 88     92         DEY
036C 10F9   93         BPL TOUR1
036E      94 ;
036E A500   95         LDA NOLIGN
0370 D0F0   96         BNE TOUR
0372 60     97         RTS
0373      98 ;
0373      99 ; CALCUL DE LA LIGNE
0373     100 ; =====
0373     101 ;
0373 A526   102 CLIGN  LDA LIGN
0375 8502   103         STA LIGNP
0377 A527   104         LDA LIGN+1
0379 8503   105         STA LIGNP+1
037B      106 ;
037B A500   107         LDA NOLIGN
037D 18     108         CLC
037E 6505   109         ADC SENS
0380 8500   110         STA NOLIGN
0382 C9C0   111         CMP  $!192
0384 D004   112         BNE LIGN1
0386 A900   113         LDA  $$00
0388 8500   114         STA NOLIGN
038A      115 ;
038A AD0303 116 LIGN1  LDA PAGE
038D 85E6   117         STA HPAG
038F A500   118         LDA NOLIGN
0391 85E2   119         STA HLIGN
0393 2011F4 120         JSR HPOSN
0396 60     121         RTS
0397      122 ;
0397      123 ; FIN
0397      124 ; ===
0397      125 ;
0397 60     126 FIN   RTS
                                127 END

```

0300-	4C	18	03	20	00	00	00	00
0308-	00	00	00	00	00	00	00	00
0310-	00	00	00	00	00	00	00	00
0318-	AD	50	C0	AD	52	C0	AD	57
0320-	C0	A2	00	86	04	AD	54	C0
0328-	AD	03	03	C9	40	D0	03	AD
0330-	55	C0	A9	00	85	00	85	E2
0338-	20	11	F4	A6	04	E6	04	E6
0340-	04	BD	04	03	F0	51	85	01
0348-	E8	BD	04	03	85	05	A5	05
0350-	10	04	A9	C0	85	00	20	62
0358-	03	C6	01	A5	01	D0	EF	4C
0360-	32	03	20	73	03	A0	27	B1
0368-	26	91	02	88	10	F9	A5	00
0370-	D0	F0	60	A5	26	85	02	A5
0378-	27	85	03	A5	00	18	65	05
0380-	85	00	C9	C0	D0	04	A9	00
0388-	85	00	AD	03	03	85	E6	A5
0390-	00	85	E2	20	11	F4	60	60

Les fichiers EXEC

Bruno Rives

INTRODUCTION

Sur APPLE II et APPLE ///, les langages BASIC et Pascal donnent la possibilité à l'utilisateur d'employer des fichiers d'un type un peu particulier, les fichiers EXEC : ces fichiers sont EXECutables.

Un fichier EXEC consiste en une liste de commandes sauvées en format ASCII (ou TEXT). Quand un tel fichier est exécuté, chaque commande qu'il contient est prise en compte par le système comme si elle avait été tapée au clavier par l'utilisateur. Ne marcheront donc que les commandes que l'utilisateur peut donner en 'direct' au clavier et non celles qui ne peuvent s'utiliser qu'à l'intérieur d'un programme : les instructions de gestion de fichiers, les GOTO ...

L'utilisation la plus rationnelle est comparable à celle que l'on trouve sur les gros systèmes sous l'appellation de J.C.L. (Job Control Language) et qui consiste à enchaîner plusieurs programmes en y mêlant des données. On crée une suite de RUNS de programmes suivis chacun des données dont ils ont besoin. On gagne ainsi du temps en lançant une série de travaux sans devoir exécuter un par un chacun des programmes et attendre parfois la fin d'impressions interminables pour lancer les programmes suivants. Cela permet aussi de faire tourner un même programme avec plusieurs jeux de données différents sans avoir à les lancer chaque fois.

Sur APPLE, on peut non seulement lier l'exécution de programmes que l'on tient à garder séparés les uns des autres, mais aussi et surtout rendre automatique toute une suite de commandes et d'ordres que l'on tape généralement un à un au clavier : fabrication et modification automatiques de programmes, copie et formatage de disquettes, démonstrations automatiques ...

UTILISATION DES FICHIERS EXEC EN BASIC.

Un fichier EXEC est donc une suite de commandes BASIC saisies dans un fichier de type TEXT. Ce fichier est ensuite exécutable par la commande EXEC suivie du nom du fichier. Plusieurs moyens sont envisageables pour le créer, le meilleur étant d'utiliser un programme de traitement de texte dont les fichiers sont sous forme TEXT (APPLEWRITER n'en fait pas partie, par contre APPLEWRITER II et /// sont utilisables). Pour ceux qui n'en disposent pas, voici un petit programme BASIC pour APPLE II qui fait très bien l'affaire :

```
0 D$ = CHR$(4) : REM CTRL-D
10 HOME
20 PRINT "DONNEZ LE NOM DU FICHIER EXEC A CREER"
30 INPUT N$
40 PRINT D$;"OPEN";N$
50 PRINT
60 PRINT "DONNEZ LES COMMANDES QUE VOUS VOULEZ EXECUTER"
70 PRINT "QUIT POUR TERMINER"
80 PRINT
90 INPUT C$
100 IF C$="QUIT" THEN 200
110 PRINT D$;"WRITE";N$
120 PRINT G$
130 PRINT D$
```

```

140 GOTO 80
200 PRINT D$;"CLOSE";N$
210 PRINT
220 PRINT "VOULEZ VOUS TESTER CE FICHER EXEC (O/N)"
230 GET A$:IF A$ = "N" THEN 260
240 PRINT
250 PRINT D$;"EXEC";N$
260 END

```

Sauvez ce programme sous le nom TEST, par exemple, et exécutez le. Amusez vous à donner les ordres les plus variés possibles (en évitant l'ordre INIT ...):

```

CATALOG
LOAD TEST
SPEED = 100
LIST
SPEED = 255
NEW
10 HOME
20 FOR I=1 TO 100
30 PRINT I
40 NEXT I
50 END
RUN
RUN TEST
QUIT

```

Pour APPLE ///, le principe est exactement le même; seuls quelques ordres BASIC sont différents. Si vous terminez votre série par RUN TEST, cela permet de retourner au programme initial de création.

En partant de ce mode d'emploi, il est possible d'utiliser les fichiers EXEC pour une foule d'applications différentes.

UTILISATION DES FICHIERS EXEC EN BASIC

1) Liste des commandes usuelles.

C'est l'utilisation la plus courante du fichier EXEC. Lorsqu'on programme en BASIC, il arrive très souvent que l'on ait à rentrer des suites de commandes toujours identiques et il est beaucoup plus facile de taper EXEC qu'une suite de commandes, même si l'on n'a que deux ou trois ordres à donner. On peut se créer un véritable catalogue de groupes de commandes que l'on utilise fréquemment. Par exemple :

```

EXEC P où P contient : PR&1
                      CTRL-I 80N
                      POKE 36,80
                      LIST
                      POKE 36,40
                      PR&0

EXEC L où L contient : OPEN &1, ".PRINTER"
                      OUTPUT &1
                      LIST

EXEC C où C contient : CATALOG S6,D1
                      CATALOG S6,D2 ...

EXEC M où M contient : CALL -151
                      300L
                      3DOG

```

Il faut noter que l'ordre 'EXEC nom de fichier' peut être donné à l'intérieur d'un fichier EXEC.

2) La capture d'un programme BASIC.

Cette manipulation est très intéressante car elle donne la possibilité, en saisissant un programme BASIC sous forme TEXT, de lui faire subir de nombreuses transformations :

- Par l'intermédiaire d'un traitement de texte comme APPLEWRITER II ou ///, il est possible de faire des modifications rapides en se promenant dans le programme comme dans un texte continu ou en procédant à la recherche et au remplacement automatique ou non. Il est donc possible de remplacer tous les noms de variables par d'autres, une certaine catégorie de lignes par d'autres ...

- Par l'intermédiaire d'un programme BASIC qui lit ce fichier EXEC comme un fichier séquentiel classique, on peut aussi faire des modifications automatiques, compter les variables, changer les numéros de lignes, produire un listing sur imprimante qui sépare automatiquement les boucles, les sous-programmes ...

- A l'aide de TEXT FILE CONVERTER ou de APPLEWRITER /// FILE CONVERTER, il est facile de faire passer un programme BASIC d'APPLE II sur APPLE /// et réciproquement.

- On peut également réaliser un traducteur BASIC permettant de passer de INTEGER à APPLESOFT et à BUSINESS BASIC.

- On peut très simplement transmettre ce fichier EXEC avec un programme de communication tel que VISITERM ou ACCES ///.

Pour 'capturer' un programme BASIC, rien de plus facile ! Tout d'abord, faire un LOAD du programme en mémoire centrale et repérer les numéros de la première et de la dernière ligne (par exemple, le programme va de 10 à 2000). Ajouter ensuite en début de programme les lignes suivantes :

```
0 D$ = CHR$(4) : REM CTRL-D
1 PRINT D$;"OPEN EX"
2 PRINT D$;"WRITE EX"
3 POKE 33,30 : REM PERMET DE COMPACTER LE PROGRAMME
4 LIST 10,2000
5 POKE 33,40
6 PRINT D$;"CLOSE"
7 END
```

Après exécution, le programme initial sera saisi en mode TEXT sous le nom EX. Pour le retrouver, il suffit de taper EXEC EX, et il sera de nouveau en mémoire centrale. Essayez !!!

(Pour APPLE ///, seuls changent les ordres BASIC).

3) La génération automatique de programmes BASIC.

Il s'agit là de l'utilisation la plus performante, mais aussi la plus délicate, des fichiers EXEC. Elle est employée par des concepteurs de logiciel qui doivent théoriquement écrire des programmes différents pour chacun de leurs clients. Ils ont effectivement commencé par adapter chacun des programmes, mais se sont vite aperçus que certaines parties étaient toujours les mêmes : ouverture, lecture et écriture, fermeture de fichiers, éditions, traitement de chaînes de caractères ... Alors, plutôt que de retaper à chaque fois tous les programmes, il est plus efficace de se confectionner un générateur de programmes !

Tout d'abord il faut organiser un peu ses programmes de telle manière que les ordres courants soient toujours à la même place : par exemple, les ouvertures de fichiers à partir de la ligne 2000, les fermetures à partir de 3000 ..., et en

utilisant les mêmes noms de variables pour recevoir les mêmes types d'information (par exemple, F1\$ sera le nom du premier fichier, R1\$ à Rn\$ pour les réponses des utilisateurs ...). Ensuite il faut créer des petits programmes répondant à ces fonctions, susceptibles d'être inclus dans le programme principal final, et saisis sous forme de fichiers EXEC, comme indiqué précédemment. Enfin, il faut créer un programme générateur de programmes qui, en fonction des traitements choisis, va engendrer un programme particulier composé d'un certain nombre de sous-programmes différents, appelés par EXEC et liés entre eux par des lignes BASIC toujours identiques.

4) La saisie d'un programme en langage-machine.

Il est des cas où l'on veut implanter un programme en langage-machine directement à partir du BASIC sans passer par l'intermédiaire de BSAVE et BLOAD et en évitant de rentrer des lignes de POKES difficiles à contrôler et à taper. Les fichiers EXEC permettent aussi de faire cela.

Imaginez que vous ayez construit un programme en langage-machine qui se trouve aux adresses 768 à 900 en décimal, et que vous vouliez l'implanter à partir d'un programme BASIC à l'aide de POKES. Le programme BASIC suivant le permet :

```
10 D$ = CHR$(4) : REM CTRL-D
20 PRINT D$;"OPEN EX"
30 PRINT D$;"WRITE EX"
40 FOR I=768 TO 900
50 PRINT "10";I;" POKE ";I;" ";PEEK(I)
60 NEXT I
70 PRINT "10910 RETURN"
80 PRINT D$;"CLOSE"
```

Pour rajouter à votre programme cette phase d'implémentation du sous-programme en langage-machine, faites un LOAD de votre programme, puis un EXEC EX qui rajoutera les lignes 10768 à 10910, ajoutez ensuite une ligne 'GOSUB 10768' pour initialiser la routine, et re-sauvez le tout.

5) Les démonstrations et exécutions automatiques.

Cette utilisation des fichiers EXEC permet de faire exécuter un programme qui demande des données extérieures sans aucune intervention, sans modifier le programme, et même de faire tourner une même application avec plusieurs jeux d'essais définis à l'avance.

Prenons l'exemple d'un programme DATE qui demande à l'utilisateur son nom, puis son prénom, enfin sa date de naissance. Pour l'exécuter sans toucher au clavier, il suffit de créer un fichier EXEC comportant :

```
- RUN DATE
- DURAND
- BERNARD
- 23 JANVIER 1952
```

Ceci peut servir à des versions de démonstration de programmes dont on ne veut pas modifier toutes les lignes INPUT pour en rendre l'exécution automatique.

UTILISATION DES FICHIERS EXEC EN PASCAL.

En Pascal APPLE II ou APPLE ///, un fichier EXEC est également un fichier de type ASCII comprenant une liste de commandes que l'on donne habituellement au clavier. Comme en BASIC, on peut créer ce fichier par un programme Pascal. La marche à

suivre est alors la même et nous ne la décrirons pas ici.

Ce que l'on peut faire de plus, c'est déclarer au niveau du système Pascal que tout ce que l'on va taper au clavier sera désormais enregistré dans un fichier préalablement défini. La différence est de taille car il suffit de taper une fois la série de commandes pour la refaire ensuite autant de fois que l'on veut de manière automatique. Cela veut dire aussi qu'il faudra exécuter TOUTES les commandes que l'on veut avoir sous le fichier EXEC, ce qui n'est pas le cas lorsqu'on le crée à partir d'un programme.

La marche à suivre est alors la suivante :

- Au niveau du système, taper M pour MAKE EXEC.
- Il est alors demandé un nom de fichier et un signe qui, tapé deux fois par la suite, arrêtera l'enregistrement sur le fichier EXEC.
- A partir de ce moment là, TOUT ce que vous tapez est enregistré sous le nom que vous avez donné, et cela jusqu'à ce que vous tapiez deux fois de suite le caractère de fin d'enregistrement.
- Pour exécuter le fichier, tapez EXEC//, suivi du nom du fichier.

Par exemple,

- M (au niveau système Pascal)
- TEST (nom du fichier EXEC)
- % (acceptation de % comme caractère de fin)
- F (FILER)
- L (catalogue)
- : (du disque courant)
- Q (retour au niveau général)
- %% (fin de l'enregistrement)

L'exécution de ce fichier par EXEC//TEST va automatiquement dans le FILER et donne le directory du volume courant. Cette exécution peut aussi être lancée à partir d'un autre programme Pascal à l'aide de CHAINSTUFF (comme le programme exécuté lors du BOOT).

Les utilisations sont exactement les mêmes que celles précédemment définies pour le langage BASIC, à part celles qui ont trait à la modification et au transfert des programmes, puisque les programmes source Pascal sont déjà en format TEXT. De même, la saisie de programmes en assembleur est inutile.

Par contre, la fabrication de programmes générateurs de programmes et l'utilisation de toutes les commandes de gestion de fichiers et de volumes est très puissante. La partie de programme décrite ci-après montre l'utilisation des fichiers EXEC à partir d'un programme Pascal APPLE II pour formater des disquettes. Il est généralement inclu dans un programme principal sous forme de procédure mais ce petit exemple vous permettra de le tester immédiatement :

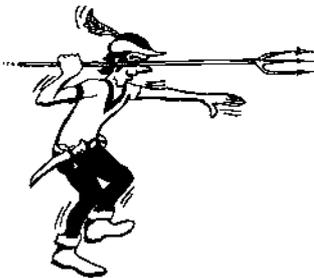
```
PROGRAM FORMAT;  
$ USES CHAINSTUFF;  
BEGIN  
  WRITELN  
    ('PLACEZ UNE DISQUETTE VIERGE DANS LE LECTEUR 5 ET TAPÉZ RETURN');  
  READ(KEYBOARD, REponse);  
  IF REponse = CHR(13) THEN SETCHAIN('EXEC//EXFORMAT');  
END
```

Le fichier EXEC EXFORMAT contient : - X
- FORMATTER
- 5
- (Return)

- X
- FORMAT

La dernière instruction, X FORMAT, permet de retourner au programme principal. Attention, comme toutes les instructions que vous entrez sous fichier EXEC doivent être exécutées une première fois, vous serez obligé de faire un faux programme FORMAT pour pouvoir stopper l'enregistrement, faute de quoi l'EXEC répètera fidèlement ou le formatage continu ou l'arrêt du programme.

Les fichiers EXEC ne sont généralement connus que de nom sur APPLE. Ils font pourtant partie des petites caractéristiques d'APPLE qui offrent de multiples ressources. Si l'accent n'a pas été mis dessus, c'est parce que ses concepteurs ont sans doute pensé que tout informaticien devait connaître ce genre de possibilités. Mais nous ne sommes pas tous des informaticiens ...



E . A . O .

Enseignement Assisté par Ordinateur pour Ordinateurs

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

COURS EN FRANCAIS

de BASIC

- . APPLESOFT - APPLE II (+ Carte APPLESOFT)
- . APPLE II +
- . APPLE III
- . MOPPE D'ANPICO (+ APPLESOFT en ROM)
- . PALSOFIT - ITT 2020 (+ Carte PALSOFIT)
- . SILROK - SILEX (de Léonard)

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

1/ COURS 1 (BASIC) - En Français

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

A LA FIN DE CE COURS, VOUS SAUREZ PROGRAMMER.

2/ COURS 2 (BASIC +) - En Français

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

3/ CONTRATS-LOCATION du COURS BASIC pour:

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

PRIX (T.V.A. comprise)

. COURS 2 (BASIC +) 350FF

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

ANDRÉ F. FINCT
8 Allée SUPPOT
91000 EVRY-COURCOURONNES

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

Ne ratez pas la cible !

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

```

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

Personnalisez vos disquettes

Pierre BOUTREUX

Vous souhaitez peut-être pouvoir personnaliser vos disquettes en faisant apparaître votre nom, par exemple, lors du CATALOG... Pour cela, il suffit de remplacer la mention "DISK VOLUME" affichée en tête du catalogue par un titre à votre convenance.

En effet, si vous examinez le contenu des adresses \$B3B0 à \$B3BA, vous trouverez :

B3B0- C5 CD D5 CC CF D6 A0 CB

B3B8- D3 C9 C4

En vous reportant à la page 7 du "APPLE II Reference Manual", vous constatez que ces codes sont les codes-écran (ou ASCII étendu) des caractères suivants :

B3B0- E M U L O V K

B3B8- S I D

soit DISK VOLUME à l'envers. Il faut donc charger dans les mémoires \$B3B0 à \$B3BA les codes-écran de votre titre en sens inverse, dans la limite des 11 caractères disponibles. C'est ce que fait le programme BASIC listé ci-après, avec quelques variantes selon que vous désirez un affichage normal, INVERSE ou clignotant.

Pour sauver le DOS contenant votre modification, il suffit ensuite d'initialiser une nouvelle disquette.

Remarque : si $C=ASC(C\$)$, le code écran du caractère $C\$$ est :

. $C+128$ en mode NORMAL, $C-64$ en mode INVERSE, C en mode FLASH, si $C \gg 64$.

. $C+128$ en mode NORMAL, C en mode INVERSE, $C+64$ en mode FLASH, si $C < 64$.

```
10 HOME
20 REM *****
   *                               *
   * PERSONNALISEZ                 *
   *                               *
   * VOTRE DISQUETTE              *
   *                               *
   *****

60 REM
65 REM COPYRIGHT P.BOUTREUX
70 REM ET POM'S
72 REM
75 HTAB 10: GOTO 90
80 HTAB 20 - LEN (Z$) / 2: PRINT
  Z$: RETURN
90 A$ = "*****":B$ =
  "*                               *":Z$ = A$
  : GOSUB 80:Z$ = B$: GOSUB 80
  :Z$ = "*PERSONNALISATION*": GOSUB
  80:Z$ = B$: GOSUB 80:Z$ = "*
  D'UNE DISQUETTE*"
100 GOSUB 80:Z$ = B$: GOSUB 80:Z
  $ = A$: GOSUB 80
150 PRINT

160 PRINT " CE PROGRAMME VOUS PE
  RMETTRA DE PERSON-"
170 PRINT "NALISER VOS DISQUETTE
  S EN REMPLACANT"
180 PRINT "L'INFORMATION DISK VO
  LUME PAR CELLE DE"
190 PRINT "VOTRE CHOIX.IL FONCTI
  ONNE PAR MODIFICA-"
200 PRINT "TION DU DOS.IL FAUDRA
  DONC SAUVER LE NOUVEAU DOS
  AINSI MODIFIE EN INITIALISAN
  T UNE NOUVELLE DISQUETTE AV
  EC LE PROGRAMME HELLO DE VOT
  RE CHOIX."
210 PRINT "L'INFORMATION AURA UN
  E LONGUEUR MAXIMA-"
220 PRINT "LE DE 11 CARACTERES."
```

Pour ne pas vous tromper



IMAGOL
informatique

vous apporte

ASSISTANCE et CONSEIL

pour la

MISE EN SERVICE

de votre

1^{er} MICRO-ORDINATEUR

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

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

- Contrats d'entretien de matériel



- Aide au choix de logiciels standards

Demandez une démonstration de nos Logiciels Standards Interactifs « Série S » pour APPLE :
(Tous compatibles entre eux et avec VISICALC)

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



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

```

240 PRINT " L'AFFICHAGE PEUT SE
FAIRE:"
250 HTAB 10: PRINT " 1-EN VIDEO
NORMALE"
260 HTAB 10: PRINT " 2-EN VIDEO
INVERSE"
270 HTAB 10: PRINT " 3-EN MODE C
LIGNOTANT "
280 PRINT
290 PRINT "ENTREZ VOTRE CHOIX :1
,2,3";: HTAB 30: INPUT " ? "
;AZ:A = AZ
295 PRINT : PRINT : PRINT
300 IF A < 1 OR A > 3 THEN 290
310 PRINT : PRINT
320 PRINT "ENTREZ VOTRE MODIFICA
TION"
330 PRINT "(MAXIMUM 11 CARACTERE
S)";: HTAB 27: INPUT A$
350 X = LEN (A$)
360 IF X > 11 THEN 320
370 B = 11 - X
380 IF A = 1 OR A = 3 THEN B$ =
" "
390 IF A = 2 THEN B$ = "^"
395 IF B = 0 THEN 430
400 FOR I = 1 TO B
410 A$ = B$ + A$
420 NEXT I
430 FOR J = 1 TO 11
440 B$ = MID$ (A$,J,1)
450 IF A = 1 THEN POKE 46011 -
J, ASC (B$) + 128
460 IF A = 2 THEN POKE 46011 -
J, ASC (B$) - 64 * ( ASC (B$
) > = 64)
470 IF A = 3 THEN POKE 46011 -
J, ASC (B$) + 64 * ( ASC (B$
) < 64)
480 NEXT J
485 HOME
490 D$ = "": REM CTRL-D
500 PRINT D$;"CATALOG"
505 PRINT : PRINT
510 PRINT "LA MODIFICATION VOUS
PLAIT-ELLE ?"
520 HTAB 10: PRINT "1=OUI"
530 HTAB 10: PRINT "2=NON";: HTAB
30: INPUT Z: PRINT : PRINT
540 HOME
545 IF Z < 1 OR Z > 2 THEN 510
550 IF Z = 2 THEN 240
560 PRINT "POUR SAUVEGARDER CETT
E MODIFICATION, CHARGER L
E PROGRAMME HELLO DE VOTRE"
570 PRINT "CHOIX ET INITIALISER
UNE NOUVELLE "
580 PRINT "DISQUETTE"
590 PRINT : PRINT : PRINT "
AU REVOIR"

```

Les clubs ont la parole

ASSOCIATION DE MICRO-INFORMATIQUE

A.M.I. 76

BOITE POSTALE 4059

76022 ROUEN CEDEX

TEL : 16 (35) 89 49 18

Créée en août 1981, cette association à but non lucratif se donne pour objectif de regrouper des utilisateurs et utilisatrices de micro-ordinateurs.

1. Premiers axes d'activité

- Etudes comparatives des matériels et logiciels présentés sur le marché.
- Essais et démonstrations de programmes professionnels.
- Initiation et formation à l'utilisation des micro-ordinateurs, à l'analyse et à la structuration des informations.
- Assistance et conseil, mutuels, entre les membres de l'association.

2. Projets en cours

- Etude comparative de plusieurs logiciels de comptabilité.
- Expérimentation en contrôle de processus, sur APPLE.
- Travail à long terme de réflexion sur la micro-informatique, avec des formateurs.
- Constitution d'une bibliothèque micro-informatique (déjà assez importante).

Nous avons déjà permis à des utilisateurs d'horizons très différents de se rencontrer : commerce, ophtalmologie, travail social, biologie, électronique, recherche en intelligence artificielle, marketing...

REJOIGNEZ-NOUS AU PLUS VITE, ECHANGEONS NOS EXPERIENCES !

MA POMME

6, rue Saunière - 75016 - Paris

La première Assemblée Générale a eu lieu récemment. Il y a été décidé de fixer la cotisation annuelle à 300 francs, les membres de moins de 18 ans bénéficiant d'une réduction de 50%. Le club tiendra une réunion les troisièmes mardis de chaque mois à 19H30, dans un local restant à déterminer.

Prochaines réunions : Visicalc le 20/4, les bases de données le 18/5 et CPM le 15/6. Pour tous renseignements, téléphoner tous les jours de 20H à 24H au (1) 558 0578.

NANTES APPLE CLUB

21 A, bd Gabriel GUIST'HAU

44000 NANTES

Personne à contacter : Monsieur BACHELIER

TEL : (40) 47 53 09

Ce club a pour but de mettre en relation les utilisateurs d'APPLE de la région nantaise, utilisateurs (ou futurs utilisateurs), professionnels ou personnels...

Courrier des lecteurs

Monsieur,

J'ai bien reçu le numéro 2 de POM'S avec pas mal de choses intéressantes.

Une idée : ne pourriez-vous pas à chaque numéro joindre dans la disquette 2 ou 3 graphiques H.R. utiles pour toute la famille, par exemple :

- carte de France avec départements
- carte de France avec voies navigables
- carte de l'Europe...etc

Cela montrerait à certaines familles que l'ordinateur peut être utile à tous.

Par ailleurs, pourriez-vous faire un article sur les multiples emplois de l'instruction EXEC ? Merci et bon courage

Docteur Pierre-Yves GOURIOU
08600 Givet

REPONSE :

1- Nous attendons les contributions graphiques (et autres) de la part de nos lecteurs.

2- Bruno RIVES vous offre justement dans le présent numéro un article sur EXEC. Vous êtes comblé !

pom's

BULLETIN D'ABONNEMENT

Je désire recevoir le N° de POM'S

- avec disquette _____ 85 F TTC
 sans disquette _____ 35 F TTC

Je désire m'abonner pour 4 numéros

à partir du N°

- avec disquette _____ 295 F TTC
 sans disquette _____ 120 F TTC

Nom _____

Adresse _____

Ces tarifs comprennent l'envoi postal en France Métropolitaine et CEE (voie aérienne exceptée)

Envoyez ce bon de commande et votre règlement à :

Editions MEV - 49 rue Lamartine - 78000 Versailles

C'est avec impatience que j'attendais le numéro 2 de POM'S, et dans l'ensemble je n'ai pas été déçu ! Quel bonheur d'avoir enfin une revue en langue française consacrée à l'APPLE II ! Bravo !

Cependant, le dicton dit : "Qui aime bien châtie bien !". Aussi, après mes encouragements les plus sincères, me permettrai-je de faire quelques critiques !

1- La parution trimestrielle ne pourrait-elle pas être ramenée à une parution bimestrielle ?

2- Beaucoup d'articles de fond que vous présentez proviennent de la revue "CALL A.P.P.L.E.". Il est vrai que vous l'avouez, mais tout de même, c'est un peu décevant pour ceux de vos lecteurs qui lisent "NIBBLE" ou "CALL A.P.P.L.E" !

3- Les programmes BASIC proposés sont d'un intérêt très restreint (AGENDA ou MULTIPLICATION). Pourquoi ne pas faire un article de fond sur un sujet (par exemple : fichier à accès séquentiel indexé) et proposer un programme BASIC illustrant cette étude ?

A présent, à titre de participation, je vous propose un petit programme BASIC pouvant servir de "MENU" et qui utilise les possibilités très complètes de tabulation sur l'écran. L'exemple proposé permet 12 options, mais il peut être étendu à l'infini.

Au niveau des possibilités méconnues de l'Applesoft, savez-vous qu'il est possible d'utiliser des données n'importe où dans un DATA à l'aide de POKE en page 0. Les pointeurs DATA sont : POKE 125,N : POKE 126,P

N : valeur décimale du Bit de poids faible de l'adresse DATA,

P : valeur décimale du Bit de poids haut de l'adresse DATA.

EX : 818...HEX 8 = DEC 8 HEX 18 = DEC 24

On tapera : POKE 125,24 : POKE 126,8.

Seules contraintes :

1- Posséder un utilitaire de DUMP, par exemple CRAE.

2- Placer les DATA en début de programme, de façon qu'aucune modification ne change l'adresse du DATA.

3- On ne peut placer le pointeur qu'en début de ligne BASIC d'où la nécessité d'une ligne par type de DATA.

Cette façon de procéder est plus simple que le "RESTORE NNN" proposé dans l'OI il y a quelque temps...

Maintenant une question pratique : enseignant, j'aimerais utiliser les programmes de l'Education Nationale en LSE. Je possède un APPLE II + avec carte langage et carte Z80. Existe-t-il un LSE pour APPLE ? Y a-t-il moyen d'utiliser un LSE d'un autre MICRO moyennant quelques modifications minimales ?

Edmond AUGIER

MONACO

REPONSE :

1 -Hélas, ce n'est pas encore possible...

2 -Nous ne pouvons pas nous taire sur tous les sujets évoqués par les revues américaines. Sinon, que pourrions-nous dire ? Notez que nous améliorons toujours ce dont nous avons eu l'idée en lisant les classiques. Vous pouvez constater qu'avec la conversion Pascal-BASIC de Gilles MAUFFREY, mise au point en novembre 1981, nous avons eu l'idée en même temps que "CALL A.P.P.L.E".

3 -Avec l'article sur EXEC, vous avez déjà un article de fond. En épluchant le programme aide-mémoire (Pom's N 1), vous avez une excellente occasion d'analyser l'utilisation de fichiers séquentiel et à accès direct.

4 -Malheureusement, il n'existe pas de LSE pour APPLE, et il n'est pas simple de modifier un LSE d'un autre micro. Par contre, il existe d'autres logiciels pour APPLE destinés à l'enseignement.

```

10 REM *****
   * *
   * EXEMPLE DE MENU *
   * *
   *****

70 POKE 214,255: TEXT : HOME : REM
   POKE=PAS DE LIST
80 DATA OPTION 1,OPTION 2,OPTIO
   N 3,OPTION 4,OPTION 5,OPTION
   6,SUITE MENU
90 DATA OPTION 7, OPTION 8, OPT
   ION 9, OPTION 10, OPTION 11,
   OPTION 12,RETOUR DEBUT MENU

100 DIM M$(14)
110 FOR I = 1 TO 14: READ M$(I):
   NEXT
120 REM *****
   * *
   * PAGE 1 DU MENU *
   * *
   *****

170 V = 1: GOSUB 690: FOR V = 1 TO
   22: HTAB 1: PRINT "*";: HTAB
   39: PRINT "*";: NEXT V: GOSUB
   690
180 VTAB 2: HTAB 2: INVERSE : PRINT
   " DEMONSTRATION D'UN LOGICIE
   L DE MENU ": NORMAL
190 VTAB 3: FOR I = 2 TO 39: PRINT
   "*";: NEXT I
200 FOR I = 1 TO 7: HTAB 3: VTAB
   4 + 2 * I: PRINT I;"..";M$(I
   ): NEXT I
210 HTAB 4: VTAB 18: PRINT ">>>"
220 VTAB 21: HTAB 15: PRINT "VOT
   RE CHOIX "; CHR$(91);" "; CHR$(
   93): VTAB 21: HTAB 28: GET
   C$: PRINT C$: IF C$ = "@" THEN
   670: REM @ = FIN DE PROGR
   AMME
230 V = VAL (C$): IF V = 0 THEN
   220
240 FOR I = 0 TO 7 - V: VTAB 18 -
   2 * I: HTAB 4: PRINT ">>>"; IF
   I > 0 THEN VTAB 20 - 2 * I:
   HTAB 4: PRINT ".,."
250 NEXT I: IF V = 6 THEN FOR I
   = 1 TO 200: NEXT
260 VTAB 4 + 2 * V: HTAB 6: INVERSE
   : PRINT M$(V): NORMAL
270 FOR I = 1 TO 1000: NEXT I
280 ON V GOTO 590,600,610,620,63
   0,640,340
290 REM *****
   * *
   * PAGE 2 DU MENU *
   * *
   *****

340 HOME :V = 1: GOSUB 690: FOR
   V = 1 TO 22: HTAB 1: PRINT "
   *";: HTAB 39: PRINT "*";: NEXT
   V: GOSUB 690

```

```

350 VTAB 2: HTAB 2: INVERSE : PRINT
   " DEMONSTRATION-PAGE 2
   ": NORMAL
360 VTAB 3: FOR I = 2 TO 39: PRINT
   "*";: NEXT I
370 FOR V = 8 TO 14: VTAB 2 * V -
   10: HTAB 3: PRINT (V - 7);".
   .";M$(V): NEXT V
380 HTAB 4: VTAB 18: PRINT ">>>"
390 VTAB 21: HTAB 15: PRINT "VOT
   RE CHOIX "; CHR$(91);" "; CHR$(
   93): VTAB 21: HTAB 28: GET
   C$: PRINT C$:V = VAL (C$)
400 IF C$ = "@" THEN POKE 214,0
   : END : REM @=FIN DU VERRO
   UILLAGE DE LIST
410 IF V = 0 THEN 390
420 FOR I = 0 TO 7 - V: VTAB 18 -
   2 * I: HTAB 4: PRINT ">>>"; IF
   I > 0 THEN VTAB 20 - 2 * I:
   HTAB 4: PRINT ".,."
430 NEXT I: IF V = 7 THEN FOR I
   = 1 TO 300: NEXT
440 VTAB 4 + 2 * V: HTAB 6: INVERSE
   : PRINT M$(V + 7): NORMAL
450 FOR I = 0 TO 1000: NEXT I
460 ON V GOTO 520,530,540,550,56
   0,570,580
470 REM *****
   * *
   * APPEL OPTIONS *
   * *
   *****

520 GOTO 650: REM OPTION 7
530 GOTO 650: REM OPTION 8
540 GOTO 650: REM OPTION 9
550 GOTO 650: REM OPTION 10
560 GOTO 650: REM OPTION 11
570 GOTO 650: REM OPTION 12
580 HOME : GOTO 170
590 GOTO 650: REM OPTION 1
600 GOTO 650: REM OPTION 2
610 GOTO 650: REM OPTION 3
620 GOTO 650: REM OPTION 4
630 GOTO 650: REM OPTION 5
640 REM OPTION 6
650 HOME : VTAB 12:A$ = "DEBUT D
   E VOTRE OPTION": HTAB 20 - LEN
   (A$) / 2: PRINT A$: VTAB 20:
   HTAB 25: INPUT "TAPEZ 'RETU
   RN' ";R$: IF R$ < > "" THEN
   650
660 HOME : GOTO 170
670 HOME : VTAB 12: HTAB 12: PRINT
   "AU MOIS PROCHAIN !"
680 NORMAL : VTAB 23: NEW
690 VTAB V: FOR H = 1 TO 3 : P
   "*";: NEXT H: RETURN
2200 FOR I = 1 TO 7: HTAB 3
   4 + 2 * I: PRINT I;"..";
   ): NEXT I

```

ILLEL

ESPACE ELECTRONIQUE

ILLEL CENTER PARIS 15^e: 143, av. Félix-Faure - 75015 Paris, Tél. 554.97.48, Métro : Balard.

ILLEL CENTER PARIS 10^e : 86 bd Magenta 75010 Paris, Tél. 201.94.68, Métro : Gare de l'Est, Parking : Magenta.



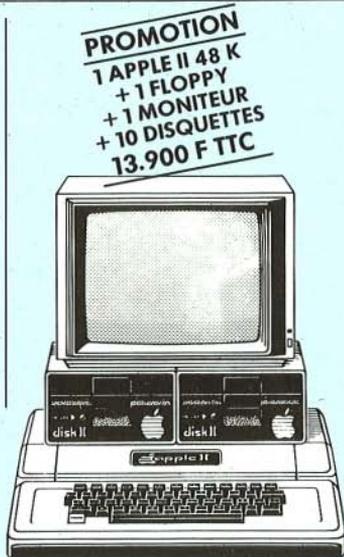
LES MICRO-ORDINATEURS

APPLE II

Un des micro-ordinateurs les plus fiables de sa génération, Apple II est utilisé dans de nombreux domaines : gestion, comptabilité, enseignement, utilisations scientifiques et industrielles, applications domestiques.

D'une très grande robustesse (garantie totale 1 an) Apple II n'ex-cède pas 5 kg et sa facilité de transport renforce encore sa souplesse d'utilisation.

Son extensibilité est remarquable : Apple II étant compatible avec la plupart des périphériques actuels, il bénéficie d'un large éventail de possibilités.



CONFIGURATION DE DÉVELOPPEMENT

Matériel	Langage			
	BASIC	PILOT	PASCAL	FORTRAN
Système	II Plus	II Plus	II Plus	II Plus
Mémoire utilisateur (RAM)	32 K	48 K	48 K	48 K
Micro-programmation	Cartes BASIC	Cartes BASIC	Carte Langage	Carte Langage
Unités Disk II	1	1 ou 2	1	1

APPLE III

L'Apple III est un système d'ordinateur de bureau puissant, faisant partie d'ensembles étudiés sur mesure et conçus pour résoudre vos besoins complexes en application. Pour les managers, les financiers, les analystes et tous ceux qui ont besoin d'organiser des faits et des chiffres, il existe le système d'Analyse de l'information Apple III.

Option A : 33.330 F TTC visuale 3 - S.O.S. buissness Basic - Moniteur 3 12"

Option B : 38.100 F TTC.

Idem A + Floppy supplémentaire

Option C : 41.100 F TTC

Idem B + Imprimante thermique graphique.



LES LOGICIELS

Pour APPLE II

PHANTOMS FIVE 48 K (DOS 3.3)	260,00 F TTC
SPACE EGGS 48 K (DOS 3.2 ou 3.3)	260,00 F TTC
RASTER BLASTER 48 K	295,00 F TTC
APPLE PANIC 48 K	380,00 F TTC
COMPUTER BISMARCK 48 K	395,00 F TTC
COMPUTER NAPOLEONICS 48 K	450,00 F TTC
COMPUTER AIR COMBAT 48 K	495,00 F TTC
VISICALC (DOS 3.3) 16 secteurs	1 764,00 F TTC
VISITREND + VISIPILOT	2 363,76 F TTC
VISIDEX	1 764,00 F TTC
VISIPILOT	1 640,52 F TTC
VISITERM	1 375,92 F TTC
DESK TOP PLAN II	1 764,00 F TTC
CCA/DMS (Gestion de Fichier)	1 105,44 F TTC
APPLE WRITER	576,24 F TTC
APPLE POST	352,80 F TTC
APPLE PILOT	1 293,60 F TTC
MINI-ASSEMBLEUR APPLE SOFT	235,20 F TTC
PROGRAMME	
COMPTABILITÉ GÉNÉRALE (SAARI)	3 410,40 F TTC
PROGRAMME PAYE (GIPSI)	2 587,20 F TTC

BON DE COMMANDE EXPRESS ILLEL

A retourner à : ILLEL Center Informatique : service vente par correspondance 143, avenue Félix Faure 75015 Paris.

MODE DE RÈGLEMENT CHOISI

- à la commande paiement comptant
 à crédit* à partir de 2000 F.

Je verse 20 % du montant total de mon achat : _____, _____ F
 ci-joint : Chèque bancaire C.C.P. Mandat carte

* Conditions de crédit CREG : • Être salarié
 • 20 % minimum au comptant, solde arrondi à la centaine supérieure.

Je soussigné : Nom _____ Prénom _____

N° _____ Rue _____

Code postal _____ Ville _____ Tél. _____

commande ferme et désire recevoir d'urgence

	Quantité	Prix unitaire	Prix total

Signature : _____

Montant net _____
 Frais de port pour envoi postal **3 0 0 0**
 TOTAL A PAYER _____

diagonale

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



Lisez

L'ORDINATEUR INDIVIDUEL

Vous y trouverez :

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

L'ORDINATEUR INDIVIDUEL

chez votre marchand de journaux

41 rue de la Grange aux Belles - 75010 Paris