

GESTION DU DISQUE DUR sur micro-ordinateur



• GESTION DU DISQUE DUR SUR MICRO-ORDINATEUR

Collection QUE

Don Berliner et Chris De Voney

Texte français d'Olivier Engler, Tellig

Vous possédez un micro-ordinateur — PC ou compatible ? Alors, un jour ou l'autre, vous aurez besoin... et envie d'un disque dur. Peut-être même en avez-vous déjà un, et envisagez-vous l'achat d'un second disque dur. Peut-être aussi désirez-vous tirer le meilleur parti possible du disque dur que vous possédez déjà.

Quelle que soit votre situation, ce livre est pour vous, débutants ou utilisateurs chevronnés. Il se divise en quatre parties :

- La première examine les avantages et inconvénients respectifs des disquettes et des disques durs, les principes de base, la technologie, les fonctions et les différentes caractéristiques d'un disque dur. Puis on apprend à l'installer, à le structurer et à le mettre en service.
- La deuxième partie présente les connaissances de base à acquérir avant de commencer à travailler avec un disque dur. Les auteurs y analysent les fonctions du DOS.
- La troisième partie a pour but de vous aider à améliorer l'organisation de votre disque dur et à la rendre plus performante.
- La dernière partie illustre les différentes méthodes possibles pour améliorer la gestion du disque dur. Les techniques et les programmes présentés permettent d'organiser le disque et de faire des copies de sauvegarde.

Reliure souple, 158 X 230, 624 pages — 395 F

Peter Norton

IBM PC Structure et environnement

DEUXIÈME ÉDITION



• IBM PC, STRUCTURE ET ENVIRONNEMENT Deuxième édition

Peter Norton

Texte français de Dimitri Stoquart

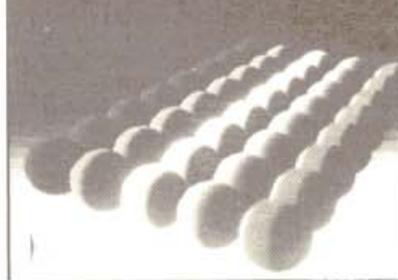
La plupart des utilisateurs de la gamme IBM PC connaissent déjà bien l'ouvrage de Peter Norton, qui est devenu une véritable "bible". L'évolution du DOS, l'apparition des nouveaux modèles de la gamme ont rendu indispensable une nouvelle édition entièrement refondue et remise à jour. Peter Norton nous donne aujourd'hui une version nouvelle de son livre, tenant compte, en particulier, des nouvelles versions du DOS (jusqu'au DOS 3), et des machines de type PC XT et AT.

Les lecteurs qui connaissent déjà l'ouvrage trouveront donc une mine d'informations complémentaires. Ceux qui ne le connaissent pas encore découvriront un précieux guide de travail. Le livre permet en effet d'acquérir une compréhension totale de la structure, du fonctionnement et de l'environnement des machines de la gamme IBM PC. C'est une initiation aux caractéristiques et aux possibilités qu'elles offrent. Il permet en outre d'acquérir une bonne maîtrise du système d'exploitation MS-DOS/PC-DOS. Cette maîtrise permettra aux utilisateurs d'obtenir de leur matériel les résultats les plus performants.

Les commandes, les fonctions et les utilitaires du MS-DOS /PC-DOS sont présentés, illustrés et discutés. L'auteur s'attache à faire découvrir, pas à pas, les différentes capacités du micro-ordinateur et de son environnement. Il propose de nombreux exemples, à travers des programmes que le lecteur pourra tester lui-même.

Reliure souple, 180 X 240, 408 pages — 260 F

dBASE III Plus PROGRAMMATION AVANÇÉE



• dBASE III Plus Programmation avancée

Collection QUE

Joseph-David Carrabis

Texte français d'Eric Terrier, Groupe ESC, Pau

dBASE III Plus s'est imposé comme le standard des systèmes de gestion de base de données sur micro-ordinateur — PC ou compatible. Son principal attrait est de fonctionner en réseau. Il dispose en outre d'un générateur d'applications intégré.

Avec cet ouvrage, écrit dans un style très convivial, par un spécialiste chevronné de la programmation, l'utilisateur pourra tirer de dBASE III Plus le meilleur profit possible. Il s'agit en effet d'un outil logique de programmation, et non pas d'un livre contenant des recettes qu'il suffirait de copier. Toutes les techniques utilisées sont expliquées de façon très pédagogique et illustrées de nombreux exemples. Une fois assimilées, elles permettront au lecteur de les reproduire et de les adapter à ses besoins.

Le livre est divisé en trois parties. La première est consacrée aux quatre principes de la programmation avancée en dBASE (écriture de programmes compacts ; développement de programmes minimisant le temps d'exécution et l'espace disque ; écriture de programmes modulaires ; développement d'une connaissance avancée des spécificités de dBASE). La deuxième examine les techniques à utiliser pour développer les applications professionnelles. La troisième est consacrée à l'utilisation de dBASE III Plus couplé à des logiciels complémentaires permettant de développer des applications complètes.

Reliure souple, 158 X 230, 384 pages — 295 F
Ces 3 ouvrages sont édités chez :
INTER EDITIONS, 87, av. du Maine 75014 PARIS

tremplin micro 21

SOMMAIRE

Avec la collaboration de :

André BROSSAUD, CHARLY et F. SIGAL, Marc FREFOYE, Stephan HADINGER, Yvan KOENIG, NESTOR, Jean PERROT, Emile SCHWARZ.

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

BIMESTRIEL

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

Tous nos prix sont indiqués TTC.

EDITIONS JIBENA

Direction-Rédaction :

Editions JIBENA

Guy-HACHETTE

La Petite Motte — Senillé —
86100 CHÂTELLERAULT.

Téléphone :
49.93.66.66

PUBLICITÉ :

Raymond JULLIEN
(1) 45.75.41.81

Commission Paritaire :

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

TREMPLIN MICRO — Bimestriel — C'est une publication des Editions JIBENA, 4, rue de la Cour-des-Noues, 75020 PARIS — S.A. au capital de 3 600 000 F — Imprimé par CITÉ-PRESS/PARIS — Service de vente : Presse-Promotion, tél. : 49.93.65.03. — Dépôt légal à la date de parution — Inscription à la Commission Paritaire des Publications et Agences de Presse : en cours — Directeur de la Publication : Guy-Clément COGNÉ — Diffusion N.M.P.P.

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

Yvan KOENIG s'interroge... **TREMPLIN MICRO** aussi ! 2

APPLESOFT

La France et le Monde FRANCE 3
sur votre Apple (Jean PERROT) MONDE 7

FACTARIF Un programme complet de facturation,
(Yvan KOENIG) 37

Nombres premiers (NESTOR) 62

Poursuivez votre initiation au langage C

Apprendre le C dans les livres 11

DE L'APPLESOFT] Une contribution majeure
au langage C] de CHARLY et F. SIGAL 26

GS Basic

LE JEU DE YAM (André BROSSAUD) 20

! Ce logiciel inédit, entièrement écrit en GS Basic exploite les possibilités graphiques de l'Apple // GS.

Initiez-vous à l'assembleur

Tout le monde comprendra comment fonctionnent les trois versions du GET SPÉCIAL de NESTOR 13

LE TML BASIC

Il existe et Emile l'a essayé. Très puissant, il compile les programmes écrits en GS Basic 17

MS-DOS sur APPLE

TURBO BASIC] Appel de sous-programmes 49

sur Apple // GS] Utilisation de DEF SEG 50

(avec PC Transporter)] Encadrez vos listages 51

Prodos.NDA

Un nouvel accessoire de bureau pour votre Apple // GS,
(Stephan HADINGER) 53

Yvan KOENIG s'interroge... TREMPLIN MICRO AUSSI !

Après de longs mois de flottement, APPLE France a enfin un Directeur Général. Nous attendons avec impatience que celui-ci veuille bien nous dire si, comme l'affirme J. SCULLEY, le GS est encore un produit d'avenir, ou si, comme de nombreuses sirènes cherchent à nous en convaincre, ce n'est (en France tout au moins) que l'APPLE III et le LISA réunis. Sans doute n'est-il pas inutile de déballer quelques échantillons du fonctionnement quasi suicidaire de ce qui était, il y a peu, un réseau de distribution...

«Si vous m'achetez un MAC, j'assurerai le service après-vente moi-même, mais si vous prenez un GS, il faudra vous dém... avec Les Ulis».

Signé : un concessionnaire agréé Apple.

Un autre, non moins agréé : «Je n'arrive pas à me procurer les composants pour la mise à niveau du GS que je vous ai vendu il y a six mois».

Le client : «Dommage : j'envisageais d'en acheter trois exemplaires supplémentaires».

L'agréé : «Ah bon ! alors revenez dans huit jours, j'aurai le nécessaire.»

Le pire, c'est que, huit jours plus tard, la mise à jour attendue depuis des semaines était effectivement réalisée.

Y.K. : «Allo ! dans le magazine T... vous publiez une page de pub proposant, entre autres, des logiciels pour le GS. L'un d'eux n'étant pas compatible avec la nouvelle Rom, pouvez-vous me dire ce qu'il en est des autres ?»

Réponse : Bof ! je n'en sais rien.

Y.K. : «Allo ! dans le défunt *Echo des Apple*, vous proposiez les disques durs CMS. Je sais que CMS propose pour ceux-ci, aux U.S.A., une carte interface SCSI permettant un partitionnement du disque, ce qui

permet — en particulier — d'utiliser à fond le disque 60 mégas. Puisque Apple France ne semble pas pressé de diffuser la révision C de sa rom SCSI (et son logiciel de partitionnement), pensez-vous pouvoir importer la carte CMS ?»

Réponse : Bof !

Y.K. : «Allo ! il paraît que vous êtes CLARIS-France, est-il possible d'obtenir, en France, la version nouvelle APPLEWORKS 2.0 (américaine) aux mêmes conditions qu'aux U.S.A., lorsqu'on est OFFICIELLEMENT en possession d'APPLEWORKS ?»

Réponse : Bof ! vous savez, ce qui nous intéresse, c'est le Mac.

Y.K. : «Ah oui ! pourtant, CLARIS fait la plus grosse part de son bénéfice avec APPLEWORKS et fait de la pub à tout va pour ce programme ?»

Réponse : C'est vrai, mais c'est aux U.S.A.

X, Y, Z et compagnie : Attendez-vous à voir bientôt un GS performant grâce au 65c832...

Réponse de W. MENSCH (père des 6502, 65c02, 65c816, 65c832 et W65c265)

«Nous prévoyons que les prototypes du 65c832 seront disponibles dans 18 mois et que la fabrication de la version 8Mhz commencera en 1990 (c'était le 25 avril 1988 sur GENIE).

Comme vous le voyez, le moral est au plus bas. Envolé l'optimisme des lendemains d'AppleExpo ! Cependant, je suis têtue et j'aimerais qu'on nous explique pourquoi la gamme II est si mal traitée en France, alors qu'après avoir remodelé le IIe, APPLE US se prépare à rajeunir le IIc (d'aucuns, bien informés, annonçaient l'abandon de ces deux machines). D'accord : Apple France est en avance sur les U.S.A., puisque la firme livre désormais le GS en version 1,2 mégas, alors que les U.S.A. introduisent enfin la version 512 k, mais pourquoi n'ont-ils pas utilisé leur fichier clients pour informer directement les possesseurs de GS de la procédure de mise à niveau ? Pour proposer un abonnement à *l'Echo des Apple* quatre mois avant sa disparition, ON avait été capable d'ouvrir ces fichiers, *l'Echo* était-il plus important que le GS ?

Dernier détail avant de vous quitter : acheter du matériel aux U.S.A. peut conduire à des situations délicates... en cas de panne. Par contre, dans l'état de décrépitude actuel du réseau de distribution français (non, je ne parle pas de SIVEA Cannes), n'hésitez pas à commander directement aux U.S.A. Pour moins de \$30 (port inclus) PBI software vous enverra MONTE-CARLO. Ce n'est pas de la pub, juste un exemple pour rappeler que, lorsque les prix sont corrects, mieux vaut un original qu'une copie. Bien sûr, la doc est en anglais, mais il en est de même chez les rares importateurs.

Yvan KOENIG — 5 juin 1988.

La FRANCE et le MONDE

sur votre Apple !

Désirez-vous savoir quelle distance sépare Angers de Toulon ? Utilisez le programme FRANCE de Jean Perrot. Le petit cousin de votre grand-mère passe un week-end à Guadalajara et vous désirez lui téléphoner, mais en étant sûr de ne point le réveiller. Autrement dit : quelle heure est-il là-bas et (ça ne coûte pas plus !) combien de kilomètres vous séparent ? Lancez le programme MONDE !

Un conseil : lisez d'abord les explications de Jean Perrot. Mieux : tapez-les et n'hésitez pas à les revoir de temps en temps. C'est un peu longuet, mais cela vous évitera, par la suite, si vous donnez une copie de votre disquette à un copain, d'avoir à lui photocopier deux pages de TREMLIN MICRO !

Second conseil : créez d'abord un sous-catalogue intitulé DISTANCES (ou HEURES), puis commencez par y mettre le micro-menu que nous avons baptisé START. Suivront EXPLICATIONS, FRANCE et MONDE. Bonne copie !



START

100 D\$ = CHR\$ (4): PRINT D\$"PR £3": PRINT D\$"PREFIX/T21/DI STANCES"	275D	"NR\$" le programme 'MONDE'" : PRINT	68D3
105 IV\$ = CHR\$ (15):NR\$ = CHR \$ (14)	449C	155 POKE 1403,10: PRINT IV\$" 3 "NR\$" le programme 'FRANCE' ": PRINT	791D
120 T\$ = "Par Jean PERROT 27 Ru e du Clos Beaumois 14000 CA EN": GOSUB 130	4287	160 POKE 1403,10: PRINT IV\$" 4 "NR\$" abandonner les recher ches": PRINT	F771
125 GOTO 135	2144	165 VTAB 19: PRINT "Votre choix ";: GET H\$: PRINT H\$	2E8E
130 L = LEN (T\$): POKE 1403,40 - L / 2: PRINT T\$: RETURN	767C	170 IF H\$ = "1" THEN PRINT D\$" RUN EXPLICATIONS"	773D
135 PRINT : PRINT :T\$ = " M E N U " " D I S Q U E T T E " : I NVERSE : GOSUB 130: NORMAL : PRINT	4D82	175 IF H\$ = "2" THEN PRINT D\$" RUN MONDE"	3011
140 PRINT "Voulez-vous ": PRIN T	15AB	180 IF H\$ = "3" THEN PRINT D\$" RUN FRANCE"	424E
145 POKE 1403,10: PRINT IV\$" 1 "NR\$" le programme 'EXPLICA TIONS': PRINT	24F2	185 IF H\$ = "4" THEN HOME : PR INT D\$"PREFIX/T21": PRINT D \$"RUN STARTUP"	4BF8
150 POKE 1403,10: PRINT IV\$" 2		190 CALL - 198: GOTO 165	F378



EXPLICATIONS

Ce programme de présentation pourrait aussi être mémorisé sous forme de pages ou bien utiliser une courte routine lisant un fichier texte. Jean Perrot a choisi la formule BASIC "totale".

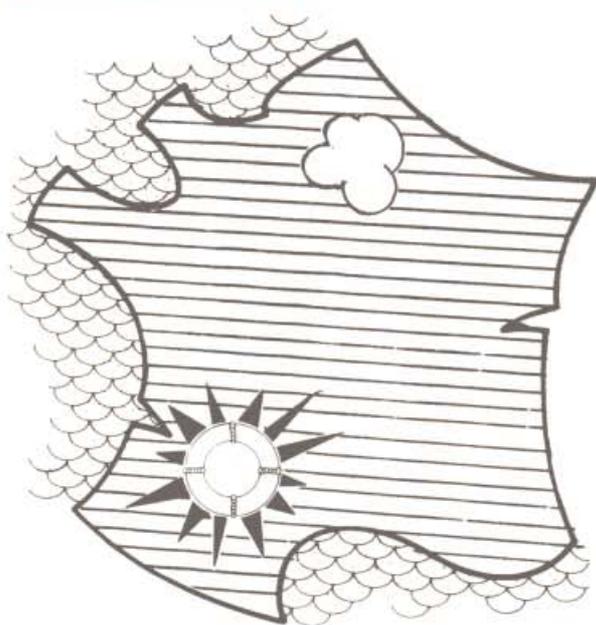
100 REM -----(EXPLICATIONS)-----	
105 D\$ = CHR\$(4): PRINT D\$"PRÉ3": PRINT	BD30
110 IV\$ = CHR\$(15):NR\$ = CHR\$(14)	449C
115 PRINT "-Voici quelques explications concernant les programmes "IV\$ " MONDE "NR\$" et "IV\$" FRANCE "NR\$	F78E
120 PRINT : PRINT IV\$" PROGRAMME 'FRANCE' "NR\$ ":" PRINT	D612
125 PRINT "-Il permet de déterminer les distances entre 46 villes de F RANCE (2 à 2)"	3761
130 POKE 1403,10: PRINT "Le calcul est basé sur le fait que :"	989C
135 PRINT IV\$" 1 "NR\$" la longueur (MR) d'une portion de méridien de 1 minute d'angle est égale": PRINT " ""à 1,852 Km (équivalent au m ille marin international)"	2179
140 PRINT IV\$" 2 "NR\$" la longueur (PR) d'une portion de parallèle de 1 minute est égale (en Km)": PRINT " ""à 1.852 * COS (LA) "-LA ét ant la latitude du "parallèle-."	DB8B
145 PRINT IV\$" 3 "NR\$" sur une faible distance (quelques centaines de kilomètres) la distance": PRINT " ""entre 2 points est sensibleme nt l'hypoténuse d'un triangle rectangle dont"	2709
150 PRINT " ""les 2 cotés de l'angle droit sont ":" PRINT "- 1 portio n de "parallèle (PR) égale à la différence des longitudes des 2 po ints";	0504
155 PRINT "- 1 portion de méridien (MR) égale à la différence des lati tudes des 2 points": PRINT	110F
160 PRINT "-Le théorème de PYTHAGORE permet le calcul de cette hypotén use"	4439
165 VTAB 23: POKE 1403,60: PRINT IV\$" TOUCHE S.V.P. "NR\$ " ";; GET H\$: HOME	1A92
170 PRINT IV\$" PROGRAMME 'MONDE' "NR\$ ":"	B1EE
175 POKE 1403,25: PRINT "-Ce programme détermine :"	DF18
180 PRINT IV\$" 1 "NR\$" l'heure d'un lieu de la terre, soit par rapport au méridien de GREENWICH": PRINT " ""(temps universel TU), soit par rapport à l'heure légale de PARIS (aussi"	E40E
185 PRINT " ""bien heure d'été que d'hiver)."	D275
190 PRINT : PRINT IV\$" 2 "NR\$" la distance entre 2 points de la terre, ainsi que les heures en ces 2": PRINT " ""points (TU ou heure l égale de PARIS) donnés par longitudes et latitudes"	DF21
195 PRINT : PRINT IV\$" 3 "NR\$" On peut avoir la liste des 102 villes e ntrées en DATAs": PRINT	7078
200 PRINT IV\$" 4 "NR\$" les distances (2 à 2) entre 102 principales vil les du monde, ainsi que les": PRINT " ""heures dans ces 2 villes (TU ou heure de PARIS)"	CCCE
205 PRINT : PRINT "Il donne aussi l'heure dans l'une des villes lorsqu 'on a l'heure dans l'autre."	6637
210 PRINT : PRINT " "-A noter que le programme ne tient pas compte des heures d'été ou d'hiver qui": PRINT "peuvent exister dans d'autre s pays que la FRANCE, mais il tient compte des dates";	DC55

```

215 PRINT "en précisant, à la suite de chaque heure, s'il s'agit du jour J ou J-1 ou J+1."
220 :
225 PRINT : PRINT " --L'utilisateur pourra, à son gré, augmenter le nombre des villes du monde"
230 PRINT "entrées en DATA. Il suffira de modifier la valeur de NV à la ligne 50 de 'MONDE'"
235 VTAB 23: POKE 1403,60: PRINT IV$ " TOUCHE S.V.P. "NR$ " ";: GET H$: HOME
240 PRINT "--Le principe du calcul est basé sur la résolution des triangles sphériques": PRINT "Le triangle envisagé est ici l'intersection du tétraèdre dont le sommet est le"
245 PRINT "centre de la terre, et dont les arêtes passent par le pôle NORD de la terre et": PRINT "chacun des 2 points envisagés."
250 PRINT : PRINT "Si a, b, c sont les faces du tétraèdre (cotés du triangle sphérique) et A, B et C, les angles dièdres opposés on a notamment la FORMULE suivante : "
255 PRINT : POKE 1403,15: PRINT "COS(c)=COS(a) * COS(b) + SIN(a) * SIN(b) * COS(C)"
260 PRINT : PRINT "--Les cotes du triangle spherique sont alors des portions de GRANDS CERCLES de la terre supposee spherique (longueur d'un grand cercle --> 40000 Km)"
265 VTAB 23: PRINT IV$ " R "NR$ " = Revoir explications "IV$ " M "NR$ " = MENU disquette "IV$ " F "NR$ " = Abandonner ";
270 GET H$: PRINT H$: IF H$ = "R" THEN RUN
275 IF H$ = "M" THEN PRINT CHR$ (4)"RUN START"
280 IF H$ = "F" THEN HOME : PRINT CHR$ (4)"PREFIX/T21": PRINT CHR$ (4)"RUN STARTUP"
285 CALL - 998: GOTO 265

```

FRANCE



```

100 REM -----( PROGRAMME VILLE.FRANCE )-----
105 D$ = CHR$ (4): PRINT D$"PRÉ 3": PRINT
110 PI = 3.1415926535:R = 20000 / PI:A = 0:B = 0
115 RESTORE :IV$ = CHR$ (15):NR$ = CHR$ (14)
120 GOSUB 320
125 PRINT : PRINT "Voulez-vous voir les villes concernées ? (taper "IV$" 0 "NR$" pour OUI) ";: GET H$: PRINT H$
130 IF H$ = "0" THEN GOSUB 325
135 RESTORE : PRINT
140 PRINT : INPUT "Nom de la première ville DANS L'ORDRE ALPHABETIQUE (RETURN = FIN) ";V1$: PRINT : IF V1$ = "" THEN HOME : END
145 INPUT "Nom de la seconde : ";V2$: IF V2$ < V1$ THEN 225
150 FOR I = 1 TO 46 * 3 STEP 3
155 READ V$,LO,LA

```

FRANCE

160 IF V\$ = V1\$ THEN LA(1) = LA: LO(1) = LO:V1\$ = V\$:A = 1	AC3F	0,2951,CANNES,420,2433	86E8
165 IF V\$ = V2\$ THEN LA(2) = LA: LO(2) = LO:V2\$ = V\$:B = 1	9744	260 DATA CHERBOURG,-97,2980,CLER MONT-FERRAND,187,2747,DIJON, 302,2960	B015
170 NEXT	0582	265 DATA DUNKERQUE,142,3062,GREN OBLE,343,2711,LE HAVRE,10,29 70	4F06
175 IF A = 0 OR B = 0 THEN 355	CC5F	270 DATA LE MANS,12,2880,LILLE,1 81,3040,LIMOGES,76,2750	7A16
180 A = 0:DN = ABS (LO(1) - LO(2)):DP = 1.855 * DN * COS (LA(1) * PI / 10800)	9330	275 DATA LOURDES,1,2587,LYON,290 ,2744,MARSEILLE,325,2598	36C6
185 DL = ABS (LA(1) - LA(2)):DM = 1.852 * DL	E660	280 DATA MONACO,447,2623,MONTPEL LIER,233,2616,MULHOUSE,440,2 864	479C
190 DS = SQR (DP ^ 2 + DM ^ 2): DS = INT (DS)	00D7	285 DATA NANCY,370,2921,NANTES,9 0,2834,NICE,438,2622	0851
195 FOR I = 1 TO 30:BZ = PEEK (49200): NEXT	6A83	290 DATA NIMES,259,2630,ORLEANS, 115,2875,PARIS,140,2930	5D2D
200 PRINT : PRINT "A vol d'oisea u, la distance entre ";IV\$V1 \$NR\$;" et ";IV\$V2\$NR\$;" est de :";IV\$;DS;NR\$;" Km"	087E	295 DATA PERPIGNAN,174,2563,POIT IERS,21,2791,REIMS,242,2955	2385
205 VTAB 22: POKE 1403,1: PRINT IV\$ " C "NR\$;" pour continuer ";IV\$ " M "NR\$;" pour ME NU de disquette "IV\$" F "NR\$" = FIN "": GET H\$: P RINT : IF H\$ = "C" THEN HOM E : RUN	707C	300 DATA RENNES,-102,2887,ROUBAI X,182,3041,ROUEN,65,2965	2A8E
210 IF H\$ = "F" THEN HOME : PRI NT CHR\$ (4)"PREFIX/T21": PRI NT CHR\$ (4)"RUN STARTUP"	3312	305 DATA ST-ETIENNE,262,2725,ST- QUENTIN,196,2990,STRASBOURG, 465,2915	6611
215 IF H\$ = "M" THEN PRINT CHR \$ (4)"RUN START"	DF4B	310 DATA TOULON,356,2588,TOULOUS E,88,2603,TOURS,42,2844	ABC9 4B45
216 IF H\$ = CHR\$ (27) THEN HOM E : END	479F	315 DATA VERSAILLES,128,2928	
220 GOTO 205	2D42	320 POKE 1403,14: PRINT IV\$ " DIS TANCE A VOL D'OISEAU DE DEUX VILLES FRANCAISES "NR\$: RET URN	32F7
225 CALL - 198: PRINT : PRINT : PRINT : PRINT IV\$"J'AI DIT ' PAR ORDRE ALPHABETIQUE'...RE COMMENCEZ... TOUCHE S.V.P."N R\$;" "": CALL - 198: GET H \$:A = 0:B = 0: PRINT : HOME : GOTO 135	6FB0 0180	325 HOME :H = 1:V = 0: FOR I = 1 TO 46: READ V\$,LO,LA:V = V + 1	BD44
230 END		330 VTAB V: POKE 1403,H: PRINT V \$:BZ = PEEK (49200)	9D6D
235 CALL - 198: PRINT "L'une de s villes ne figure pas dans ma liste...RECOMMENCEZ...To uche S.V.P. "": CALL - 198 : GET H\$: GOTO 135	3416	335 IF I = 17 OR I = 34 THEN V = 0:H = H + 20	925A
240 DATA ANGERS,-34,2850,AMIENS, 154,3053,AVRANCHES,-82,2921	3128	340 NEXT : VTAB 23: PRINT "Touch e SVP pour la suite.... "":	DFCC
245 DATA BASTIA,568,2563,BAYONNE , -90,2610,BESANCON,362,2834	8A2A	345 GET H\$: RESTORE : HOME : GOS UB 320	8A62 63B1
250 DATA BEZIERS,198,2600,BORDEA UX,-36,2650,BOURGES,144,2824	A99E	350 RETURN	
255 DATA BREST,-270,2903,CAEN,-2		355 REM -----(EN CAS D'ERREURS)-----	
		360 IF A = 0 THEN PRINT : PRINT IV\$ " ";V1\$;" N'EST PAS SUR M A LISTE... "NR\$" "":	2A07
		365 IF B = 0 THEN PRINT : PRINT : PRINT IV\$ " ";V2\$;" N'EST P AS SUR MA LISTE... "NR\$" "":	D8FD
		370 CALL - 198: GET H\$: HOME : RUN	0812



MONDE

(suite)

```

280 IF NOT LA(I) OR ABS (LA(I)
) > 90 THEN CALL - 198: GO
TO 270
285 N = LA(I):Z = 0: GOSUB 460:P
A(I) = PI / 2 - NR
290 VTAB 8 + I * K: POKE 1403,45
: IF K = 1 THEN PRINT "Long
itude du lieu ";: GOTO 300
295 PRINT "Longitude de ";V$(I)
;
300 INPUT LO$(I):LO(I) = VAL (L
O$(I)): IF LO$(I) = "0" THEN
310
305 IF NOT LO(I) OR ABS (LO(I)
) > 180 THEN CALL - 198: G
OTO 290
310 N = LO(I):Z = 1: GOSUB 460:G
O(I) = NR:H(I) = HG + HL:H(I
) = INT (H(I) * 100 + 0.5)
/ 100
315 H(I) = H(I) - CR
320 NEXT : POKE 216,0: IF K = 2
THEN 355
325 IF H(1) > 23 THEN H(1) = H(1
) - 24:J$(1) = "(jour J+1)":
GOTO 340
330 IF H(1) < 0 THEN H(1) = H(1)
+ 24:J$(1) = "(jour J-1)": G
OTO 340
335 J$(1) = "(jour J)"
340 VTAB 12: PRINT "S'il est "IV
$" ";HG;" "NR$" heures ";K$;
" le jour J, il est "IV$" "H
(1);" "NR$" heures au lieu":
PRINT "de longitude "IV$" "L
O(1);" "NR$;" degrés ";J$(1)
345 FOR II = 1 TO 20:BZ = PEEK
(49200): NEXT : GOTO 435
350 REM =====
=====
355 DL = ABS (GO(1) - GO(2)): R
EM DIFFERENCE DE LONGITUDE
360 C = COS (PA(1)) * COS (PA(

```

B325

CE91

8DEB

391C

D487

F973

C021

6AF2

BCFF

AA13

77E1

2CFF

2F83

804C

B07F

```

2)) + SIN (PA(1)) * SIN (P
A(2)) * COS (DL) 4804
365 IF C = 0 THEN THETA = PI / 2
: GOTO 375 0314
370 S = SQR (1 - C ^ 2):T = S /
C:THETA = ATN (T) 944E
375 IF C < 0 THEN THETA = THETA
+ PI A402
380 RT = INT (R * THETA + 0.5):
L = LEN (V$(1)) + LEN (V$(
2)) + 26 1136
385 VTAB 14: POKE 1403,40 - L /
2: PRINT "Distance entre "IV
$V$(1)NR$" et "IV$V$(2)NR$;"
-> ";RT;" Km" 810B
390 FOR I = 1 TO 2: IF H(I) < 0
THEN H(I) = H(I) + 24:J$(I)
= "(jour J-1)": GOTO 405 153B
395 IF H(I) > 23 THEN H(I) = H(I
) - 24:J$(I) = "(jour J+1)":
GOTO 405 6675
400 J$(I) = "(jour J)" A017
405 NEXT 0582
410 PRINT : PRINT "Il est ";IV$;
" ";H(1);" ";NR$;" heures à
";IV$;V$(1);NR$ SPC( 1)J$(1)
; A95A
415 PRINT "... et ";IV$;" ";H(2)
;" ";NR$;" heures à ";IV$V$(
2)NR$; SPC( 1)J$(2) ED95
420 FOR II = 1 TO 30:BZ = PEEK
(49200): NEXT AFCC
425 GOTO 615 FA47
430 REM =====
=====
435 VTAB 23: POKE 1403,1: PRINT
IV$" C "NR$" pour (C)ontinue
r "IV$" M "NR$" pour
retour au (M)enu de disquet
te ";: GET H$: CALL - 958 A747
440 IF H$ = "C" THEN RUN E9E0
445 IF H$ = "M" THEN PRINT CHR
$(4)"RUN START" DF4B
450 IF H$ = CHR$(27) THEN END A4CE
455 GOTO 435 1347
460 N1 = N:N = N * SGN (N):MIN
= N - INT (N):MIN = MIN * 1
0 / 6 54F3
465 N = INT (N) + MIN:NR = N.*
PI / 180 2AFB
470 N = N * SGN (N1):NR = NR *
SGN (N1): IF Z = 0 THEN RET
URN 2944
475 FOR J = - 12 TO 12 CDEB

```

<pre> 480 IF LO(I) > - 7.5 + J * 15 A ND LO(I) < 7.5 + J * 15 THEN HL = J: RETURN 485 NEXT : RETURN 490 HOME :H = 1:V = 0: FOR I = 1 TO NV - 4: READ V\$,LO,LA:V = V + 1 495 V\$TAB V: POKE 1403,H: PRINT V \$:BZ = PEEK (49200) 500 IF I = 21 OR I = 42 OR I = 6 3 OR I = 84 THEN V = 0:H = H + 16 505 NEXT : FOR I = 1 TO 10:BZ = PEEK (49200): NEXT : GOTO 43 5 510 PRINT : CALL - 198: PRINT I V\$ " J'AI DIT..PAR ORDRE ALPH ABETIQUE..."NR\$:; CALL - 19 8: PRINT " ""Touche S.V.P. "";: GET H\$ 515 : 520 REM -----(DISTANCES ET HEU RES DES PRINCIPALES)----- 525 REM -----(VILLES DU MONDE)----- 530 HOME : GOSUB 195 535 INPUT "Nom de la première vi lle dans l'ordre alphabétiqu e :";V\$(1): PRINT 540 INPUT "Nom de la seconde :"; V\$(2): IF V\$(2) < V\$(1) THE N 510 545 PRINT "----- -----" 550 POKE 1403,32: V\$TAB 14: PRINT IV\$ " P A T I E N C E "NR\$ 555 A = 0:B = 0: FOR S = 1 TO NV : READ V\$,LO,LA:Z = 0 560 IF V\$ = V\$(1) THEN LA(1) = L A:N = LA:I = 1:A = 1: GOSUB 460:PA(1) = PI / 2 - NR:LO(1) = LO:N = LO:Z = 1:: GOSUB 460:GO(1) = NR:H(1) = HG + H L - CR 565 IF V\$ = V\$(2) THEN LA(2) = L A:N = LA:I = 2:B = 1: GOSUB 460:PA(2) = PI / 2 - NR:LO(2) = LO:N = LO:Z = 1:: GOSUB 460:GO(2) = NR:H(2) = HG + H L - CR:Z = 0: GOTO 575 570 NEXT 575 IF V\$(1) = "CAEN" OR V\$(1) = </pre>	<pre> 2E62 6E6D 0F7B 9D6D 25FC C9BE 312E 003A 1620 8B1A 0FE6 4B5A D57B 1D80 958F 1EB1 0582 </pre>	<pre> "PARIS" THEN H(1) = H(1) + C R 580 IF V\$(2) = "CAEN" OR V\$(2) = "PARIS" THEN H(2) = H(2) + C R 585 IF A = 1 AND B = 1 THEN 355 590 IF A = 0 AND B = 0 THEN PRI NT "Ni "IV\$" "V\$(1)" "NR\$" n i "IV\$" "V\$(2)" "NR\$" ne fig urent sur ma liste...RETURN S.V.P. "";: GOTO 605 595 IF A = 0 THEN PRINT IV\$" "V \$(1)" "NR\$;" n'existe pas su r ma liste...RETURN SVP "";: GOTO 605 600 IF B = 0 THEN PRINT IV\$" "V \$(2)" "NR\$" n'existe pas sur ma liste...RETURN S.V.P. "" ; 605 GET H\$: RUN 610 REM ----- ----- 615 V\$TAB 18: CALL - 958: PRINT "-Si on donne l'heure d'un l ieu (taper 1 pour le premier et 2 pour le second)" 620 PRINT "on obtiendra l'heure de l'autre lieu..."IV\$" M "N R\$" revient au menu" 625 V\$TAB 22: PRINT "Si l'heure e st en TU il faut ajouter 1 h eure en HIVER et 2 en ETE po ur avoir ""l'heure légale e n FRANCE." 630 V\$TAB 19: POKE 1403,61 635 GET K\$:K = VAL (K\$): CALL - 958: PRINT : IF K\$ = "M" T HEN RUN 640 IF K < 1 OR K > 2 THEN 615 645 V\$TAB 21: CALL - 958: PRINT "Si heure de ";V\$(K);" = ";: INPUT " ";K(K): V\$TAB 21: POK E 1403,15 + LEN (V\$(K)): PR INT IV\$" ";K(K);" ";NR\$;" "; "<jour J)"; 650 IF K(K) < 0 OR K(K) > 24 THE N 645 655 K(3 - K) = K(K) + H(3 - K) - H(K) 660 IF K(3 - K) < 0 THEN K(3 - K) = K(3 - K) + 24:J\$(3 - K) = "<jour J-1)"; GOTO 675 665 IF K(3 - K) > 23 THEN K(3 - </pre>	<pre> 88B6 E58A 4460 7714 7288 2359 F610 CFA2 6DD9 73C7 7B5A 6177 8D74 BA09 48E2 9DEF 0A4D </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------



MONDE

(suite)

```

K) = K(3 - K) - 24:J$(3 - K)
= "(jour J+1)": GOTO 675
670 J$(3 - K) = "(jour J)"
675 PRINT " alors il est ";IV$
";K(3 - K);" ";NR$;" heures
a ";V$(3 - K) SPC( 1)J$(3 -
K)
680 VTAB 22: POKE 1403,11: PRINT
"Taper "IV$" A "NR$" si on v
eut entrer une autre heure "
;: GET L$: PRINT L$
685 IF L$ = "A" THEN 615
690 CALL - 198: RUN
695 DATA ABIDJAN,-4.30,5.21,ADDI
S-ABEBA,38.48,8.45,ALEXANDRI
E,29.50,31.06
700 DATA ALGER,3,36.38,AMSTERDAM
,4.50,52.22,ANDORRE,1.30,42.
30
705 DATA ANKARA,32.47,39.56,ATHE
NES,23.45,38,BAGDAD,44.27,33
.07
710 DATA BARCELONE,2.10,41.21,BA
SSETERRE,-61.20,16.36,BELGRA
DE,20.30,44.45
715 DATA BERLIN,13.25,52.32,BERN
E,7.29,46.57,BEYROUTH,35.21,
33.48
720 DATA BIRMINGHAM,-1.53,52.27,
BOGOTA,-73.48,4.22,BONN,7.05
,50.45
725 DATA BRAZZAVILLE,14.22,-4.12
,BRUXELLES,4.19,50.50,BUCARE
ST,26.07,44.25
730 DATA BUDAPEST,19.05,47.30,CA
EN,-0.20,49.10,CARACAS,-66.5
2,10.39
735 DATA CAYENNE,-52.40,5,CHICAG
O,-87.45,41.45,COPENHAGE,12.
30,55.42
740 DATA DAKAR,-17.30,14.35,DELH
I,77.09,28.09,DETROIT,-83.02
,42.20

```

B97F
9415

FBFA

A4E9
1AD2
C7DD

0608

9EA5

BB90

985F

1B3D

1E59

35DF

314B

40DA

EBOC

```

745 DATA DJAKARTA,106.49,-6.05,D
JIBOUTI,43.09,11.59,DUBLIN,-
6.16,53.18 1858
750 DATA FORT DE FRANCE,-61,14.4
0,GUADALAJARA,-102.44,20.50,
HAMBOURG,10,53.32 552A
755 DATA HANOI,105.47,21,HELSINK
I,25,60.08,HONG KONG,114,22.
10 24A2
760 DATA HONOLULU,-157,21,HOUSTO
N,-95.33,29.20,ISTANBUL,28.5
0,41.03 C135
765 DATA JERUSALEM,35.12,31.45,K
ABOUL,69.15,34.30,LA HAVANE,
-82.30,22.30 EE85
770 DATA LE CAIRE,31.50,30,LE CA
P,18.30,-34,LENINGRAD,30.20,
60 1852
775 DATA LIMA,-77.23,-12.06,LISB
ONNE,-9.10,38.43,LONDRES,-0.
10,51.30 75A1
780 DATA LOS ANGELES,-118.15,34,LUXEMBOURG
,6.07,49.39,MADRID,-3.40,40.27 1C2C
785 DATA MANILLE,120.54,14.50,MELBOURNE,14
5.02,-37.30,MEXICO,-99.10,19.22 ECF1
790 DATA MILAN,9.08,45.27,MONACO,7.25,43.4
3,MONTREAL,-73.30,45.30 D42C
795 DATA MOSCOU,37.30,55.50,MUNICH,11.35,4
8.08,NANKIN,118.50,32 F8E0
800 DATA NAPLES,14.18,40.50,NEW-YORK,-74,4
0.40,NICOSIE,33.20,35 92EB
805 DATA NOUMEA,167,-22.20,ODESSA,30.40,46
.30,OSAKA,135.30,34.40 B9A8
810 DATA OSLO,10.40,60,OTTAWA,-75.40,45.25
,PAPEETE,-149,-17.30 9F67
815 DATA PARIS,2.20,48.50,PEKIN,117.36,40,
PORT AU PRINCE,-72.30,18.27 9431
820 DATA PYONGYANG,125.45,39,QUEBEC,-71.50
,47.18,RANGOON,96.44,16.26 7A2F
825 DATA REYJKJAVIK,-22,64.05,ROME,12.25,4
1.53,SAINT-MARIN,12.30,43.55 F9A7
830 DATA SALISBURY,31,-17.40,SAN FRANCISCO
,-122.30,37.45,SANTIAGO,-70.40,-38.30 E8BF
835 DATA SEOUL,127,37.30,SHANGAI,122,31,SI
NGAPOUR,104,1.22 0304
840 DATA SOFIA,23.20,42.39,STOCKOLM,18.20,
59.20,TAIPEH,121.30,25 24F4
845 DATA TANANARIVE,47.30,-18.57,TEHERAN,5
1.27,35.46,TIRANA,19.50,41.24 9EB1
850 DATA TOKYO,139.37,35.41,TRIPOLI,13.09,
32.47,TUNIS,9.03,36.40 941D
855 DATA TURIN,7.42,45.05,VARSOVIE,21,52.1
3,VIENNE,16.20,48.11 36BF
860 DATA WASHINGTON,-77,39.02,WELLINGTON,1
74.50,-41.23,YAOUNDE,11.37,3.49 60B0

```

Apprendre le C dans les livres ?

Tremplin Micro s'intéresse à tous les langages, mais notamment à celui qui est appelé à devenir LE langage... et vous l'avez reconnu : il s'agit du C. On ne peut pas dire que les applemaniques soient particulièrement gâtés avec la version dont ils disposent sur l'Apple IIGS. Il leur manque un TURBO C signé Borland, ou un QUICK C signé Microsoft (mais que nous ne connaissons pas).

Reste que le Turbo C est utilisable sur GS par le truchement du PC Transporter diffusé en France par Bréjoux. Et alors, quel régal ! D'autant plus que les ouvrages consacrés à TURBO C sont non seulement nombreux, mais enrichissants en diable. Nous les décrivons tous (lors de la parution) dans nos pages Biblio.

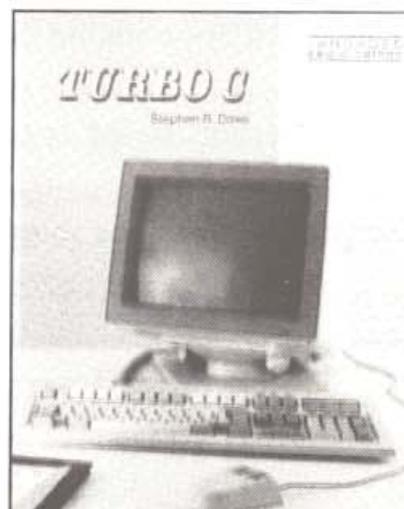
Au fait, peut-on apprendre le C dans les livres ? Assurément oui, mais en prenant le soin de tout lire... c'est-à-dire de commencer par le commencement. D'une manière générale, les habitués du Pascal ne font qu'une bouchée de l'Abc du C. Lorsqu'on arrive du BASIC, c'est un peu plus compliqué, mais on s'y fait très vite. Parlons des deux derniers (d'autres arrivent !).

TURBO C de Stephen R. Davis

Avec ce manuel, on peut parler d'ouvrage de référence accessible à des programmeurs de bon niveau, sur PC et compatibles. Turbo C (de Borland) est devenu un standard, mais s'écarte parfois de la norme Kernighan et Ritchie — avec bonheur d'ailleurs. Le livre de Stephen R. Davis fait le point là-dessus. Mais l'auteur ne se contente pas de disserter sur les mérites respectifs du C classique et du Turbo C. Il invite le lecteur à programmer. Les exemples abondent dans ce bouquin... du plus simple au plus complexe (et, parmi ces derniers, la création de programmes résidents).

Un livre qui fait honneur à la collection Cedic-Nathan.

475 pages, broché — 265 F PCV Diffusion BP86 77402 LAGNY/sur/MARNE



PLUS LOIN AVEC TURBO C de Daniel-Jean David

Enseignement complet de Turbo C sur IBM PC et compatibles, le livre traite les aspects les plus avancés du langage et notamment les pointeurs et la gestion dynamique de la mémoire, ainsi que les fichiers en accès séquentiel direct, et la récursivité.

Il comble par ailleurs une lacune des compilateurs C en fournissant un ensemble complet de graphiques haute résolution, permettant de réaliser tous les dessins possibles. Ce package graphique est disponible sur disquette, mais nous ne l'avons pas testé. Nous supposons que cette même disquette regroupe les corrigés des 50 exercices proposés dans l'ouvrage.

Edimicro, 121 avenue d'ITALIE 75013 PARIS

250 pages, Broché — 245 F

A essayer sur votre GS avec PC-Transporter

PRIM TEXT

Vous avez eu la bonne idée de transformer votre Apple IIGS en compatible grâce au PC-Transporter... mais vous hésitez à investir dans un nouveau traitement de texte : j'ai peut-être découvert un logiciel capable de vous intéresser.

Il est diffusé par *Internationale Système Machine* (ISM)*, une société lyonnaise qui développe une gamme de logiciels destinés à améliorer et optimiser l'emploi des ordinateurs.

Il s'agit de PRIM-TEXT (480 F HT, mais avec timer et alarme). Il m'a paru d'une réelle simplicité parce que convivial en diable. Il autorise la manipulation de blocs, la justification automatique, la frappe kilométrique, la recherche et le remplacement, la césure automatique, etc.

Il est auto-documenté en français et peut être utilisé de deux manières différentes :

1. Sans le Driver et sans le Chrono (impossibilité d'employer les différents caractères et non disponibilité du Timer/Alarme/Bloc-note/Table ASCII).
2. Avec ses accessoires et l'on a alors accès (avec les touches de fonctions F2 à F10) aux différents styles de caractères, ainsi qu'au chrono (ALT-1 à 8).

Un traitement de texte sympa, peu ambitieux, mais suffisant dans bien des cas, pour réaliser des impressions soignées.

Nestor.

* ISM, 2, Quai du Commerce 69009 LYON — Tél. : 78.47.76.08.

Chez le même éditeur :

ILICO, accélérateur logiciel pour PC, permettant de multiplier la vitesse d'exécution de vos applications (Dbase, bases de données diverses). ILICO assure une réécriture des routines d'entrées/sorties du DOS totalement compatible, permettant des temps d'accès et de chargement des fichiers de 2 à 20 fois plus courts. C'est un logiciel très facile à utiliser puis qu'il suffit de le lancer par un seul mot : ILICO !

GET SPÉCIAL

Comment afficher la mention "PRESSEZ UNE TOUCHE" sur la ligne 24 d'un écran 40 colonnes, en mode inverse, puis retrouver le contenu éventuel de cette ligne ?

Le contenu de la touche pressée est mémorisé en 86. La démo en Basic montre exactement le fonctionnement de cette courte routine.

```

10 TEXT : PRINT CHR$(21): HOME
15 PRINT CHR$(4)"BLOAD GETS/GETS.LM": VTAB 3
20 FOR I = 1 TO 879: PRINT "A";: NEXT
25 POKE 2039,193: VTAB 1: CALL - 868
30 CALL 768
35 PRINT : VTAB 1: CALL - 868: PRINT "Vous avez pressé
   la touche ";
40 T = PEEK (6) - 128: IF T < 32 THEN PRINT "CTRL-";:T
   = T + 64
45 PRINT CHR$(T)
50 IF PEEK (6) <> 155 THEN 30
55 VTAB 1: PRINT "(1) ENCORE (2) APPLESOFT (3) MENU
   DISK ": CALL 768
60 R = PEEK (6) - 176: IF R = 1 THEN 25
65 IF R = 2 THEN HOME : END
70 IF R = 3 THEN PRINT CHR$(4)"RUN STARTUP"
75 GOTO 55
  
```

```

300 : 20 18 03 JSR $0318
303 : 2C 10 C0 BIT $C010
306 : 2C 30 C0 BIT $C030
309 : A9 80 LDA £$80
30B : 20 A8 FC JSR $FCA8
30E : AD 00 C0 LDA $C000
311 : 10 F3 BPL $0306
313 : 2C 10 C0 BIT $C010
316 : 85 06 STA $06
  
```

Vers Bascule.
Clavier initialisé.
Clic haut-parleur.

Boucle d'attente (WAIT) en fonction de A.

Si aucune touche n'a été pressée, à refaire.

Clavier à zéro.
Contenu de l'Accumulateur dans \$6.

BASCULE

```

318 : AD 2C 03 LDA $032C
31B : 48 PHA
31C : AD 2F 03 LDA $032F
31F : 8D 2C 03 STA $032C
322 : 68 PLA
323 : 8D 2F 03 STA $032F
  
```

On inverse alternativement
le contenu des adresses \$32C et \$32F

MÉMO/AFFICHAGE

1. La partie de ligne-écran concernée est mémorisée à partir de \$34C et remplacée par le texte (\$338).
2. Quand une touche a été pressée, c'est le contraire qui est réalisé et l'écran retrouve son aspect.

*338.34B

```

20 10 12 05 13 13 05 1A
20 15 0E 05 20 14 0F 15
03 08 05 20
  
```

" PRESSEZ UNE TOUCHE " (en mode inverse)

GET SPÉCIAL

Cette seconde version est une petite démonstration qui vous montre la rapidité de l'affichage en langage machine. La routine d'affichage est pokée à la fin du programme en BASIC (ligne data 90). Les adresses du désassemblage obtenu pourront être différentes si vous avez modifié une seule des lignes en Basic, mais c'est sans importance.

GETS.BAS1

```

10 LOMEM: 29 + ( PEEK (105) + PEEK (106) * 256)
15 TEXT : PRINT CHR$ (21): HOME : GOSUB 75
20 PRINT CHR$ (4)"BLOAD GETS/GETS.LM": POKE 6,193
25 PRINT : CALL AD
30 VTAB 1: CALL 768: PRINT : VTAB 1: CALL - 868: PRINT "Vous avez pressé la touche ";
35 T = PEEK (6) - 128: IF T < 32 THEN PRINT "CTRL-";:T = T + 64
40 PRINT CHR$ (T)
45 IF PEEK (6) < > 155 THEN 25
50 VTAB 1: PRINT "(1) ENCORE (2) APPLESOFT (3) MENU DISK ": CALL 768
55 R = PEEK (6) - 176: IF R = 1 THEN 25
60 IF R = 2 THEN HOME : END
65 IF R = 3 THEN PRINT CHR$ (4)"RUN STARTUP"
70 GOTO 50
75 AD = ( PEEK (105) + PEEK (106) * 256) - 29
80 FOR I = AD TO AD + 28: READ R: POKE I,R: NEXT
85 RETURN
90 DATA 165,6,201,32,176,2,9,128,162,2,72,128,32,71,248,104,160,39,145,38,136,16,251,
232,224,24,144,238,96.
    
```

A01 :	A5	06	LDA	\$06	On prend la valeur du caractère dans \$6.
A03 :	C9	20	CMP	£\$20] Si ce n'est pas un contrôle, on saute.
A05 :	B0	02	BCS	\$0A09	
A07 :	09	80	ORA	£\$80	Sinon on le transforme en caractère affichable.
A09 :	A2	02	LDX	£\$02	X = 2 pour commencer à cette ligne.
A0B :	48		PHA		Valeur de A empilée.
A0C :	8A		TXA] X dans A pour que GBASCAL calcule l'adresse de base de la ligne.
A0D :	20	47	JSR	\$F847	
A10 :	68		PLA		Récupération de A.
A11 :	A0	27	LDY	£\$27	Longueur de ligne dans Y.
A13 :	91	26	STA	(\$26),Y] Une ligne du caractère dont le code est dans A.
A15 :	88		DEY		
A16 :	10	FB	BPL	\$0A13] + une ligne.
A18 :	E8		INX		
A19 :	E0	18	CPX	£\$18] Si on a atteint la limite on a fini. Sinon...
A1B :	90	EE	BCC	\$0A0B	
A1D :	60		RTS		Retour au Basic.

DÉMO MIXTE (GR OU TEXT)

GETS.BAS2

```

10 LOMEM: 29 + ( PEEK (105) + PEEK (106) * 256)
15 PRINT CHR$ (4)"BLOAD GETS/GETS.LM": POKE 6,193
20 TEXT : PRINT CHR$ (21): HOME : RESTORE : GOSUB 75
25 PRINT : CALL AD
30 VTAB 01: CALL 768: PRINT : VTAB 01: CALL - 868: PRINT "Vous avez pressé la touche ";
35 T = PEEK (6) - 128: IF T < 32 THEN PRINT "CTRL-";:T = T + 64
40 PRINT CHR$ (T)
45 IF PEEK (6) < > 155 THEN 25
50 VTAB 01: PRINT "(1) ENCORE (2) APPLESOFT (3) MENU DISK ": CALL 768
55 R = PEEK (6) - 176: IF R = 1 THEN 20
60 IF R = 2 THEN TEXT : HOME : END
65 IF R = 3 THEN PRINT CHR$ (4)"RUN STARTUP"
70 GOTO 50
75 AD = ( PEEK (105) + PEEK (106) * 256) - 29
80 VTAB 21: PRINT "TAPER (1) POUR MODE GRAPHIQUE ";; GET R$: PRINT
85 FOR I = AD TO AD + 28: READ R: POKE I,R: NEXT : IF R$ < > "1" THEN POKE
  2150,48: POKE 2151,49: POKE 2161,48: POKE 2162,49: POKE 2273,48: POKE
  2274,49: RETURN
90 GR : POKE AD + 9,0: POKE AD + 25,20: POKE 2150,50: POKE 2151,50: POKE
  2161,50: POKE 2162,50: POKE 2273,50: POKE 2274,50: RETURN
95 DATA 165,6,201,32,176,2,9,128,162,2,72,138,32,71,248,104,160,39,145,38,136,16,
  251,232,224,24,144,238,96

```

En modifiant quelques lignes de GETS.BAS1 ... et en ajoutant quelques POKES bien ajustés, il est possible d'observer l'effet de GETS en mode TEXT ou en mode GR.

Ces modifications concernent :

- 1.** L’AFFICHAGE du texte qui passe de la ligne 01 à la ligne 22.
- 2.** LE REMPLISSAGE de l’écran qui va de 1 à 19 au lieu de 3 à 24.

Attention ! *Le programme BASIC s’automodifie, ce qui comporte bon nombre de risques. Qu’un seul espace de VOTRE programme soit différent du NÔTRE... et rien ne va plus !*

Notez qu’il est facile, en passant en mode moniteur, puis en examinant la mémoire, de vérifier si les bonnes valeurs se trouvent bien aux bonnes adresses. Considérez en tout cas cet ajout comme un exercice amusant... et rien de plus.

C’est aussi cela l’informatique personnelle !

NESTOR.



GUIDE MICROSOFT DE MS-DOS 3.1 à 3.3 (Ray Duncan)

Difficile de trouver plus compétentes que les Editions Microsoft Press pour parler de MS-DOS ! On sait que ce système d'exploitation, mis au point par Microsoft Corp, est actuellement le plus utilisé dans le monde. Il tourne en effet sur une dizaine de millions de micro-ordinateurs et il est à la base de plus de 20 000 applications.

C'est aussi le mieux connu et l'on ne compte plus les ouvrages consacrés à ses diverses versions.

Celui de Microsoft Press est destiné à tous les utilisateurs et passe en revue, d'une manière très détaillée, toutes les commandes internes et externes du DOS sur PC, PS et compatibles.

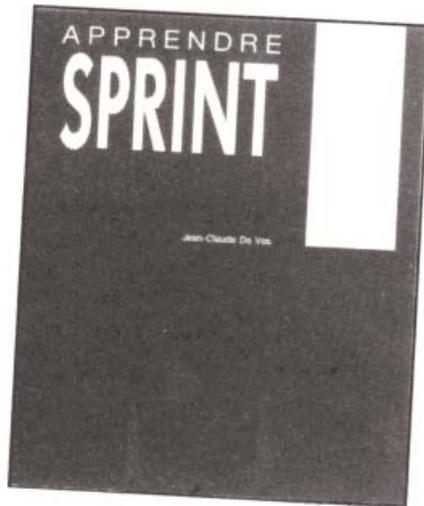
Il traite aussi les options de configuration du système, ainsi que les directives spécifiques aux fichiers de commandes, à l'éditeur EDLIN et aux fichiers de gestion de périphériques (DRIVER.SYS).

Un guide complet, à garder à portée de la main.

P.C.V. Diffusion — BP 86 77402 LAGNY/SUR/MARNE
290 pages sous couverture cartonnée — 285 F

APPRENDRE SPRINT (Jean-Claude De Vos)

Manuel de formation de la collection Cedic/Nathan, bien rédigé et très bien présenté, comme tous les ouvrages de cette série. Le traitement de textes de Borland est un produit séduisant et a déjà conquis bon nombre d'utilisateurs de PC et compati-



bles. Sa documentation est copieuse, mais ce manuel d'initiation ne fait pas double emploi. C'est une approche différente, intéressante à bien des égards, un tour d'horizon complet du logiciel, avec des informations indispensables au néophyte... et parfois peu connues des habitués.

A offrir à toute personne désirant découvrir rapidement et méthodiquement ce traitement de textes de la nouvelle génération.

P.C.V. (360 pages — 235 F)

LE LIVRE DE L'AT (Jörg Schieb)

L'AT est une machine ouverte, particulièrement configurable et disposant d'un grand nombre de périphériques. Auteur de nombreux ouvrages et logiciels, Jörg Schieb nous propose de découvrir les outils internes de l'AT pour une exploitation plus puissante et plus conviviale de la machine.

LE LIVRE DE L'AT est une manne d'informations sur le fonctionnement interne de la machine, ses périphériques, ses différentes configurations... Le lecteur découvrira au fil des pages comment créer ses propres outils de développement ou d'exploitation : savoir par exemple installer un RAM disque, augmenter les performances de son clavier ou gérer les interruptions du microprocesseur. Autant d'astuces fort précieuses pour une totale harmonie entre l'utilisateur et sa machine.

Chacun des thèmes traités dispose d'une panoplie complète d'exemples (programmes, instructions, adresses-clés du système...) destinés à élargir l'horizon du lecteur dans ses développements ultérieurs.



Voici un aperçu des principaux sujets étudiés :

- Description général du système.
- Les différences essentielles entre le XT et l'AT.
- L'installation et l'utilisation approfondie du DOS (version 3.3 incluse).
- La programmation des cartes vidéo (CGA, EGA, VGA, PGA, AGA, HERCULES (plus)).
- L'affichage à l'écran : modes texte et graphique.
- Le fonctionnement et la personnalisation du clavier (macros, accélérateur du clavier...).
- L'exploitation des ports série et parallèle : brochage des câbles, connexion des imprimantes, périphériques leurre, redirection des entrées/sorties.
- Les extensions du système y compris la gestion des cartes d'extension mémoire : EMS, EEMS, LIMS.
- Les systèmes d'exploitation et environnements particuliers : XENIX, WINDOWS, GEM.
- Le système d'exploitation de demain : OS/2.

Bénéficiant d'un style agréable tant dans l'écriture que dans la lisibilité des schémas, la valeur des informations du LIVRE DE L'AT permettra au lecteur d'assimiler très rapidement les informations jusqu'à les intégrer parfaitement dans ses travaux quotidiens.

ÉDITIONS MICRO APPLICATION
13, rue Sainte-Cécile 75009 PARIS
(330 pages — 199 F)

LE TML BASIC

existe... Emile l'a essayé !

Le très attendu TML BASIC est enfin arrivé. Pour la modique somme de \$125.00 (prix catalogue), vous obtiendrez une disquette et le manuel.

Le logiciel est livré avec le système 3.1, soit ProDOS 16, version 1.3 évidemment équipé du FINDER anglais.

Ce manuel (environ 500 pages au format 17,5 cm X 21 cm) vous donne tous les renseignements utiles sur le fonctionnement du logiciel. Il est divisé en quatre parties :

1. Le guide de l'utilisateur ;
2. La référence du langage ;
3. Programmer en utilisant la boîte à outils ;
4. Les annexes.

Ce logiciel fonctionne en mode "Desktop", c'est-à-dire qu'il utilise les menus déroulants et les fenêtres. Chaque programme — chaque source — est affiché dans une fenêtre. Vous pouvez avoir quatre fenêtres — quatre sources — visibles à l'écran, mais il est bien entendu qu'il n'y a qu'un seul programme actif.

Dans l'annexe D, vous trouverez les différentes instructions inconnues en TMLBASIC ou utilisées différemment, plus une liste des nouvelles. En fait, pour les instructions non reconnues, il s'agit essentiellement de celles fonctionnant en mode immédiat, plus quelques autres telles que : BASICà, CONT, COPY, DIR, INVOKE/PERFORM, TRACE/NOTRACE, LIBFIND, PROGNAME\$ et QUIT. Presque toutes ces instructions sont évidemment inutiles ou inutilisables dans l'environnement TMLBASIC car véritablement dépendantes d'un interpréteur.

Il s'agit là d'un Basic 16 bits, très puissant, capable de compiler sur disque, sous forme d'un fichier application, le source que vous avez écrit. Comme il est précisé dans la publicité, vous avez la possibilité de récupérer les programmes écrits en GS Basic. Pour cela, utilisez le petit EXEC "TRANSFERT".

En mode immédiat, tapez le programme ci-dessous, puis "RUN" pour créer le fichier EXEC :

```

10      REM MAKE.EXEC.TML
20  START:PRINT "Fait un EXEC de transfert de fichier .GSB en .TML"
30      OPEN "TRANSFERT", FILTYP = TXT FOR OUTPUT AS£1
40      PRINT£1;"HOME":REM Texte du fichier EXEC
50      PRINT£1;"REM Syntaxe: EXEC,";CHR$(34);"PROG.GSB,SOURCE.TML"
60      PRINT£1;"LOAD é1è":REM Charge le programme GS Basic
70      PRINT£1;"OPEN é2è,FILTYP = TXT FOR OUTPUT AS£1"
80      PRINT£1;"A = LISTTAB :OUTREC = 0 :LISTTAB = 128 + A"
90      PRINT£1;"OUTPUT £1 : LIST : OUTPUT £0"
100     PRINT£1;"OUTREC = 80 : LISTTAB = A":REM Restaure les valeurs
110     PRINT£1;"CLOSE £1":REM Ferme le fichier de destination
120     PRINT£1;"REM Transfert terminé":REM Fin du transfert
130     CLOSE£1:REM Ecriture de l'EXEC terminée
140  FIN  :END

```

Vous pouvez également utiliser votre traitement de texte habituel et taper le texte situé entre les deux REMarques, puis le sauver dans un fichier du type TXT sous le nom de "TRANSFERT".

La syntaxe d'appel de cet EXEC est :

EXEC TRANSFERT,"PROGRAMME.GSB,PROGRAMME.TML"
 TRANSFERT est le nom du fichier EXEC,
 PROGRAMME.GSB est le programme en GS Basic,
 PROGRAMME.TML est le fichier source en TML BASIC.

LE TML BASIC n'utilise pas les numéros de ligne, ce qui nous oblige à employer LISTTAB = 128+A (ancienne valeur). En effet, dans ce cas le programme sera listé sans numéro de ligne. OUTREC=0 permet d'avoir des lignes sans aucune tabulation ni CR intempestifs. Une fois cette opération effectuée, il vous est possible de récupérer le fichier texte avec le TML BASIC. Vous pouvez également vous servir des instructions décrites à la page 482 du manuel de l'utilisateur du TML BASIC.

Une fois votre "source" écrit, vous avez la possibilité d'appeler le "debugger" intégré en cliquant dans le menu **Compile** l'option **Check Syntax**, ou bien en utilisant l'équivalent clavier Pomme-Y. Si la syntaxe est correcte, vous pouvez, grâce à **To Memory & Run** (Pomme-M) ou **To Disk** (Pomme-D), exécuter ou sauver sur disque, sous forme de fichier \$B3 (ou S16), le résultat de la compilation.

Je ne m'étendrai pas davantage sur la description de ce Basic, mais je vous propose un petit programme que j'ai repris du GS Basic

et qui fonctionne sans presque rien changer. Reportez-vous à *Tremplin Micro numéro 19*, pour le programme d'origine.

Voici les modifications apportées (mise à part l'absence de numéros de lignes) :

- a. L'apostrophe peut remplacer le **REM** ou **:REM** ;
- b. L'instruction **LIBRARY** n'est pas suivie par **APPEND** ;
- c. L'utilisation du **IF... THEN... ELSE ... END IF** ;
- d. J'ai été amené à passer **\$E1C029 NEWVIDEO** en mode linéaire par un poke.

Une remarque s'impose :

IL EST IMPÉRATIF DE NE PAS TENTER DE CHARGER UNE IMAGE COMPACTÉE.

En effet, pour des raisons de compacité, je n'ai pas fait de test sur le type de fichier en cause. Vous pouvez donc charger un écran 32k du type BIN ou PIC (\$C1 aux \$0000).

Vous trouverez sur la disquette d'accompagnement (qui est indispensable !) en plus du source TMLBASIC, le fichier application qui est utilisable sous le Finder ou le Launcher.

Et pour les amateurs, voici l'adresse où vous procurer ce TML BASIC :

TML Systems, Inc.
8837-B Goodbys Executive Drive
Jacksonville, Florida 32217
United States of America

P.S. : Pour les amateurs de Pascal, il existe aussi un TML PASCAL pour Apple IIGS.

Il existe un TML PASCAL pour le Macintosh...

```
'-----
'
'                               UTIL.TML
'-----
' Charge et affiche une image au format ecran 32k
' Exemple de programme TML BASIC
'
'-----
'Copyright (c) Mai 1988 Emile SCHWARZ et Tremplin Micro
'-----

Debut:
    LIBRARY "Memory"
```

```

LIBRARY "QuickDraw"

DIM pic!(32767)                                ' Taille tableau = image 32K

Input "Nom de l'image a charger: ";Pic$
IF File(Pic$) THEN
  OPEN Pic$, AS #11,32767                       ' Ouvre le fichier image
  GET #11;Pic!(0)                               ' Stocke l'image dans le tableau
  CLOSE #11                                     ' Ferme le fichier
ELSE
  PRINT "L'image "Pic$" est inconnue."
  Pic$=""
END IF

ON Pic$="" GOTO Fin                             ' Image inconnue = fin programme

POKE 14794793,97                               ' Passe en mode lineaire

destà=TEN("E12000")                            ' Stocke l'adresse de destination
sourceà=VARPTR(Pic!(0))                        ' Stocke l'adresse de depart
longà=32767                                     ' Nombre d'octets à recopier
_BlockMove(sourceà,destà,longà)                ' Recopie l'image dans le buffer
Graf On                                         ' Affiche l'image

Fin:
PRINT "Pressez une touche pour quitter"
GET$ a$                                        ' Attente clavier
Home
Graf Off                                       ' Repasse en ecran texte.
END

```

TURBO PROLOG*

LE LANGAGE DE 5^e GÉNÉRATION ORIENTÉ APPLICATIONS

BORLAND INTERNATIONAL annonce la disponibilité de la version américaine de TURBO PROLOG 2.0. Cette nouvelle version apporte un support externe de bases de données, un interpréteur, un guide tuteur de 350 pages et un support graphique étendu. TURBO PROLOG 2.0 permet également de développer des bases de données plus importantes grâce à une nouvelle série d'outils.

Avec plus de 100 000 utilisateurs, TURBO PROLOG est le langage de 5^e génération sur micro le plus utilisé dans le monde. A la différence des langages procéduraux traditionnels, TURBO PROLOG offre un code compact très lisible et facile à gérer. Utilisant seulement le dixième de lignes codées normalement nécessaires pour une application écrite dans un langage procédural, il autorise une très grande rapidité de programmation.

"Avec la version 2.0, TURBO PROLOG apporte les outils nécessaires pour développer de puissantes applications commerciales dans les domaines tels que la gestion de bases de données, les systèmes experts, les interfaces de langage naturel, les bases de connaissances et les

systèmes d'informations intelligents." déclare Philippe KAHN, Président et Fondateur de BORLAND INTERNATIONAL.

Rappelons que BORLAND INTERNATIONAL a été fondée en 1983 par Philippe KAHN à Scotts Valley en Californie. Le siège européen a été créé en juillet 1986 par Alain BLANCQUART, Vice Président Europe. Aujourd'hui, BORLAND INTERNATIONAL compte un effectif de plus de 650 personnes dont 75 sur la France. Le chiffre d'affaires pour ses trois premiers trimestres fiscaux (du 1^{er} avril au 31 décembre 1987) de 47,3 Millions de Dollars fait apparaître une progression de 110% par rapport à la même période l'année précédente (22,5 Millions de Dollars).

* Uniquement sur PC et compatibles disposant de 384/640 K de mémoire. (Disque dur conseillé). Prix : 1495 F HT.

LE JEU DE YAM

en GS Basic

Voici le premier jeu programmé en GS Basic. Il utilise le principe (cher à Apple) de la souris. Le programme commence par demander si l'on désire consulter la règle. Si oui, il affiche les trois écrans d'aide. Dans le cas contraire, il donne immédiatement l'image du jeu et attend une action du joueur, celle-ci pouvant se déclencher à partir du clavier (pression d'une touche) ou bien de la souris (un ou plusieurs clicks).

Avant d'entrer dans le jeu, le joueur doit décliner son identité et indiquer s'il est seul ou accompagné d'un partenaire.

En ce qui concerne la programmation, on constatera que l'image utilisée est une image compactée de type \$C0 aux \$0001. Le petit module de décompactage est extrait du programme d'Emile SCHWARZ (que vous découvrirez dans ce même numéro).

- Bibliographie :**
- La boîte à outils de l'Apple IIGS de J.-P. Curcio (Editions PSI)
 - Les clés pour l'Apple IIGS de N. Bréaud-Pouliquen
 - Le Manuel du GS Basic.

```
10      REM ***** YAM par André Brossaud (c) Mars 1988 *****
20      REM
50      CLEAR 256200:HOME:GRAF INIT 320
60      ON ERR GOTO erreur
70      DIM pic!(11000),myam!(32767),event!(16),msg!(256),r(7),b!(5)
80      DIM t(1,12),b(12),des(5),dc(5):nt=1
110     FOR i=1 TO 12:READ b(i):NEXT
120     DATA 1,2,3,4,5,6,20,30,40,50,1,1
140  TOOL: LIBFIND "QDStartUp",q1%,q2%,q3%
150     IF q1% GOTO prelude
160     LIBRARY "tdfs/quickdraw.tdf"
170     LIBRARY APPEND "tdfs/memory.tdf"
180     LIBRARY APPEND "tdfs/misctool.tdf"
190     LIBRARY APPEND "tdfs/event.tdf"
200  PRELUDE: PROC loadim:GRAF ON:_ShowCursor:PROC debut
220  DEBJEU:  PROC NbreJo:PROC affnom
300     REM * Jeu de 1 à 3 coups *
310  JEU:    c=1:h%=218:v%=32:sbc%=4:IF n=1 THEN h%=244:sbc%=8
320     IF nt>9 THEN h%=h%-8
330     PROC msg(h%,v%,sbc%,sfc%,STR$(nt))
345     ON nf(n)=12 GOTO passe
350  COUPS:  IF c>3 THEN GOSUB replace:GOTO figure
360     REM * Joue, Satisfait, Rechoisir *
```

/ LE JEU DE YAM EN GS BASIC /

```

370 CHOIX:   PROC event
410         ON y%>150 AND y%<163 AND x%>160 AND x%<236 GOTO joue
420         IF y%>180 AND y%<193 AND x%>160 AND x%<236 THEN GOSUB replace:
           GOTO figure
430         IF y%>165 AND y%<178 AND x%>160 AND x%<236 THEN GOSUB replace:
           GOTO Choix
440         IF y%>180 AND y%<193 AND x%>244 AND x%<310 THEN GOSUB quitte
450         REM * Choix des dés *
460         FOR i%=1 TO 5:_SetRect(r,131+30*i%,58,155+30*i%,80)
470         ON EXFN_PtInRect(Ptà+10,r)=-1 GOTO deplace
480         NEXT:GOTO choix
520 DEPLACE: ON dc(i%)>0 GOTO Choix
530         dc(i%)=1:PROC effdes(0):PROC affdes(46)
540         GOTO Choix
550 REPLACE: FOR i%=1 TO 5:ON dc(i%)=0 GOTO 570
560         dc(i%)=0:PROC effdes(46):PROC affdes(0)
570         NEXT:RETURN
580 JOUE:   FOR i%=1 TO 5:PROC effdes(0):NEXT
590         FOR i%=1 TO 5:ON dc(i%)>0 GOTO 610
600         des(i%)=INT(RND(1)*6)+1:PROC affdes(0)
610         NEXT:c=c+1:GOTO coups
620         REM * Choix des figures *
630 FIGURE: PROC event
640         ON y%>150 AND y%<163 AND x%>244 AND x%<310 GOTO passe
650         IF y%>180 AND y%<193 AND x%>244 AND x%<310 THEN GOSUB quitte
660         FOR f%=1 TO 12:_SetRect(r,42,17+10*f%,118,26+10*f%)
670         ON EXFN_PtInRect(Ptà+10,r)>-1 GOTO 700
680         ON f%<7 GOTO unasix
690         ON f%-6 GOTO full,carre,suite,yam,mini,maxi
700         NEXT:GOTO figure
820         REM * Validations des Figures *
830 UNASIX: ON T(n,f%)>0 GOTO figure
840         FOR i%=1 TO 5:IF des(i%)=f% THEN T(n,f%)=T(n,f%)+1
850         NEXT:ON T(n,f%)=0 GOTO figure
860         GOTO resultat
870 FULL:  ON T(n,i%)>0 GOTO figure
880         k=des(1):k1=k:j=0
890         FOR i%=2 TO 5:IF des(i%)<>k THEN k1=des(i%)
900         NEXT
910         FOR i%=2 TO 5:ON des(i%)<>k AND des(i%)<>k1 GOTO figure
920         NEXT
930         FOR i%=2 TO 5:IF k=des(i%) THEN j=j+1
940         NEXT:ON j=0 OR j=3 GOTO figure
950         T(n,7)=1:GOTO resultat
960 CARRE: j=0:ON T(n,8)>0 GOTO figure
970         FOR i%=2 TO 5:IF des(i%)=des(1) THEN j=j+1
980         NEXT:ON j>2 GOTO 1010
990         FOR i%=3 TO 5:ON des(i%)<>des(2) GOTO figure
1000        NEXT
1010        T(n,8)=1:GOTO resultat
1020 SUITE: ON T(n,9)>0 GOTO figure
1030        FOR i%=2 TO 5
1040        IF des(i%-1)>des(i%) THEN j=des(i%):des(i%)=des(i%-1):des(i%-1)
           =j:GOTO 1030
1050        NEXT:j=0

```

/ LE JEU DE YAM EN GS BASIC /

```

1060      FOR i%=2 TO 5:IF des(i%)-des(i%-1)<>1 THEN j=j+1
1070      NEXT:ON j>0 GOTO figure
1080      T(n,9)=1:GOTO resultat
1090 YAM:  ON T(n,10)>0 GOTO figure
1100      FOR i%=2 TO 5:ON des(1)<>des(i%) GOTO figure
1110      NEXT:T(n,10)=1:GOTO resultat
1120 MINI: j=0:ON T(n,11)>0 GOTO figure
1130      FOR i%=1 TO 5:j=j+des(i%):NEXT
1140      ON T(n,12)>0 AND T(n,12)<=j GOTO figure
1150      T(n,11)=j:GOTO resultat
1160 MAXI: j=0:ON T(n,12)>0 GOTO figure
1170      FOR i%=1 TO 5:j=j+des(i%):NEXT
1180      ON T(n,11)>0 AND T(n,11)>=j GOTO figure
1190      T(n,12)=j:GOTO resultat
2000      REM * Resultat *
2010 RESULTAT:nf(n)=nf(n)+1:s%=b(f%)*T(n,f%):IF f%>10 THEN s%=T(n,f%)
2020      h%=31:v%=25+10*f%:IF n=1 THEN h%=137
2040      IF s%>9 THEN h%=h%-8
2050      PROC msg(h%,v%,12,0,STR$(s%))
2060      s%=0:FOR i%=1 TO 12:s%=s%+b(i%)*T(n,i%):NEXT:T(n,0)=s%
2070      h%=31:v%=159:IF n=1 THEN h%=137
2080      IF s%>99 THEN h%=h%-16:GOTO 2100
2090      IF s%>9 THEN h%=h%-8
2100      PROC msg(h%,v%,13,0,STR$(s%))
2120      REM * Joueur suivant *
2130 PASSE: FOR i%=1 TO 5:PROC effdes(0):NEXT
2140      n=n+1:ON n<nj GOTO jeu
2150      nt=nt+1:ON nt>20 GOTO fpartie
2160      n=0:GOTO jeu
2170      REM * Fin de la partie *
2180 FPARTIE: PROC event
2220      ON y%>165 AND y%<178 AND x%>244 AND x%<310 GOTO rejouer
2230      IF y%>180 AND y%<193 AND x%>244 AND x%<310 THEN GOSUB quitte
2240      GOTO fpartie
2300 QUITTE: PROC msg(174,90,7,0,"En "+CHR$(144)+"tes vous s"+CHR$(158)+"r")
2310      PROC reponse:IF l%=0 THEN RETURN
2320      _ClearMouse:GRAF OFF:END
2400      REM * Rejouer *
2410 REJOUER: _BlockMove(sourceà,destà,longà):nt=1:n=0:nf(0)=0:nf(1)=0
2420      FOR i%=0 TO 1:FOR j%=1 TO 12:T(i%,j%)=0:NEXT j%,i%
2430      PROC msg(168,90,7,0,"Les m"+CHR$(144)+"mes joueurs")
2440      PROC reponse:IF l%=1 THEN PROC affnom:GOTO jeu
2450      GOTO debjeu
2500      REM * Affiche les dés *
2510 AFFDES: DEF PROC affdes(v%):h%=30*i%
2520      _Set.Rect(r,131+h%,58+v%,155+h%,80+v%)
2530      _SetSolidPenPat(9):_PaintRect(r):_SetSolidPenPat(0)
2540      v1%=61+v%:v2%=68+v%:v3%=75+v%:h1%=135+h%:h2%=142+h%:h3%=149+h%
2550      ON des(i%) GOTO del,de2,de3,de4,de5,de6
2560      END PROC affdes
2570 DE1:  y%=v2%:x%=h2%:PROC pts:GOTO 2560
2580 DE2:  y%=v1%:x%=h1%:PROC pts:y%=v3%:x%=h3%:PROC pts:GOTO 2560
2600 DE3:  y%=v1%:x%=h1%:PROC pts:y%=v3%:x%=h3%:PROC pts
2620 DE4:  y%=v2%:x%=h2%:PROC pts:GOTO 2560
2630 DE4:  y%=v1%:x%=h1%:PROC pts:y%=v3%:x%=h3%:PROC pts

```

```

2650          y%=v3%:x%=h1%:PROC pts:y%=v1%:x%=h3%:PROC pts:GOTO 2560
2670 DE5:      y%=v1%:x%=h1%:PROC pts:y%=v3%:x%=h3%:PROC pts
2690          y%=v2%:x%=h2%:PROC pts:y%=v3%:x%=h1%:PROC pts
2710          y%=v1%:x%=h3%:PROC pts:GOTO 2560
2720 DE6:      y%=v1%:x%=h1%:PROC pts:y%=v3%:x%=h3%:PROC pts
2740          y%=v3%:x%=h1%:PROC pts:y%=v1%:x%=h3%:PROC pts
2760          y%=v2%:x%=h1%:PROC pts:y%=v2%:x%=h3%:PROC pts:GOTO 2560
2774 POINTS:  DEF PROC pts
2780          _SetRect (r,x%,y%,x%+2,y%+2):_PaintRect (r)
2790          END PROC pts
3000 EFFDES:  DEF PROC effdes (v%):h%=30*i%
3020          _SetRect (r,131+h%,58+v%,155+h%,80+v%)
3030          _SetSolidPenPat (7):_PaintRect (r)
3040          END PROC effdes
4200 EVENT:   DEF PROC event
4210          _FlushEvents (-1,0):Ptà=VARPTR (event! (0))
4220          _GetNextEvent (-1,Ptà):ON event! (0)+event! (1)*256=0 GOTO 4220
4230          y%=event! (10)+event! (11)*256:x%=event! (12)+event! (13)*256
4240          END PROC event
4399          REM * Nombre de Joueurs & leurs noms *
4400 NBREJO:  DEF PROC nbrejo:PROC msg (160,82,7,0,"Nombre de joueurs: ")
4410          PROC capte (288,82,0):nj=ASC (n$)-48
4420          IF nj<1 OR nj>2 THEN PROC msg (296,82,7,7," "):GOTO 4410
4450          PROC msg (160,98,7,0,"Votre Nom :"):PROC capte (232,98,7):n1$=n$
4460          ON nj=1 GOTO 4480
4470          PROC msg (160,110,7,0,"Votre Nom :"):PROC capte (232,110,7):
n2$=n$
4480          PROC fen (155,68,310,130,7)
4490          END PROC nbrejo
4600 CAPTURE: DEF PROC capte (h1%,v%,ca%):h%=0:i=0:n$=""
4610          GET$X$:X$=CHR$ (ASC (X$)-128):ON X$=CHR$ (13) GOTO 4670
4620          ON X$=CHR$ (127) GOTO 4650
4630          ON i>ca% GOTO 4610
4632          IF X$=CHR$ (123) THEN X$=CHR$ (ASC (X$)+19)
4634          IF X$=CHR$ (125) THEN X$=CHR$ (ASC (X$)+18)
4636          IF X$=CHR$ (92) THEN X$=CHR$ (ASC (X$)+49)
4638          IF X$=CHR$ (64) THEN X$=CHR$ (ASC (X$)+72)
4640          h%=h%+8:i=i+1:n$=n$+X$:PROC msg (h1%+h%,v%,7,0,X$):GOTO 4610
4650          ON n$="" GOTO 4610
4660          PROC msg (h1%+h%,v%,7,0," "):I=I-1:n$=MID$ (n$,1,i):h%=h%-8:GOTO
4610
4670          ON n$="" GOTO 4610
4680          END PROC capte
4800 AFFMSG:  DEF PROC msg (h%,v%,sbc%,sfc%,z%)
4810          SET (msg! (0))=^z$:_SetBackColor (sbc%):_SetForeColor (sfc%)
4820          _MoveTo (h%,v%):_DrawString (VARPTR (msg! (0)))
4830          END PROC msg
5000 AFFNOM:  DEF PROC affnom
5010          PROC msg (14,21,4,0,n1$):ON nj=1 GOTO 5040
5030          PROC msg (84,21,8,0,n2$)
5040          END PROC affnom
5300 DEBUT:   DEF PROC debut
5302          PROC msg (178,101,7,0,"Andr"+CHR$ (142)+" Brossaud")
5303          PROC msg (170,110,7,0,"et Tremplin Micro")
5304          PROC msg (186,119,7,0,"(c) juin 1988")

```

/ LE JEU DE YAM EN GS BASIC /

```

5310      FOR i%=1 TO 5:des(i%)=INT(RND(1)*6)+1:PROC affdes(0):NEXT
5320      FOR i=1 TO 600:ON EXFN_Button(0) GOTO 5340
5330      NEXT:FOR i%=1 TO 5:PROC effdes(0):NEXT:GOTO 5310
5340      FOR i%=1 TO 5:PROC effdes(0):NEXT:PROC fen(155,68,310,140,7)
5350      PROC msg(160,78,7,0,"Voulez-vous lire la")
5355      PROC msg(188,88,7,0,"R"+CHR$(143)+"gle du jeu")
5360      PROC reponse:PROC fen(155,68,310,130,7)
5375      ON l%=0 GOTO 5410
5380      GOSUB aide:GRAF INIT 320:_BlockMove(sourceà,destà,longà)
5390      GRAF ON:_ShowCursor
5410      END PROC debut
6000 FENETRE: DEF PROC fen(h1%,v1%,h2%,v2%,ssp%)
6010      _SetRect(r,h1%,v1%,h2%,v2%):_SetSolidPenPat(ssp%):_PaintRRect(
        r,8,8)
6020      END PROC fen
6100 REPONSE: DEF PROC reponse:_SetSolidPenPat(15)
6110      _SetRect(r,190,92,228,101):_PaintRRect(r,8,8)
6120      _SetRect(r,235,92,273,101):_PaintRRect(r,8,8)
6130      PROC msg(198,100,15,0,"Oui"):PROC msg(242,100,15,0,"Non")
6140      PROC event
6150      IF y%>92 AND y%<101 AND x%>190 AND x%<228 THEN l%=1:GOTO 6180
6160      IF y%>92 AND y%<101 AND x%>235 AND x%<273 THEN l%=0:GOTO 6180
6170      GOTO 6140
6180      PROC fen(164,82,298,102,7)
6190      END PROC reponse
6500 LOADIM: DEF PROC loadim:l%=10284
6510      OPEN "imyamc", FILTYP=192 AS#l,l%:GET#l;pic!(0):CLOSE#l
6520      SET(b!(0))=VARPTR(imyam!(0)):SET(b!(4))=32768
6530      _UnPackBytes(VARPTR(pic!(0)),l%,VARPTR(b!(0)),VARPTR(b!(4)))
6540      sourceà=VARPTR(imyam!(0)):destà=14753792:longà=32768
6550      _BlockMove(sourceà,destà,longà)
6560      END PROC loadim
7000 ERREUR: OFF ERR:NOTRACE:GRAF OFF
7010      PRINT "Erreur n°"ERR " "ERRTXT$(ERR) " à la ligne "ERRLIN:PRINT
7020      END
8000 AIDE:   PROC fen(0,0,320,200,1):PROC fen(2,2,318,198,0)
8010      PROC fen(4,4,316,196,4)
8020      GRAF ON:_ShowCursor
8030      PROC msg(86,20,4,3,"R"+CHR$(143)+"gles du jeu de Yam")
8040      PROC msg(18,36,4,0,"Ce jeu se joue "+CHR$(136)+" un ou deux
        joueurs et une")
8060      PROC msg(10,46,4,0,"partie est compos"+CHR$(142)+"e de vingt
        tours.")
8070      PROC msg(18,64,4,0,"On joue de un "+CHR$(136)+" trois coups par
        tour en")
8080      PROC msg(10,74,4,0,"gardant si on veut certains d"+CHR$(142)
        +"s.")
8090      PROC msg(18,90,4,0,"Le but est de former des figures, groupes")
8100      PROC msg(10,100,4,0,"de 1, 2, ..., "+CHR$(136)+" 6, FULL (paire
        + triplet),")
8110      PROC msg(10,110,4,0,"CARRE (4 d"+CHR$(142)+"s "+CHR$(142)+
        "gaux), SUITE (12345, 23456),")
8130      PROC msg(10,120,4,0,"YAM (5 d"+CHR$(142)+"s "+CHR$(142)+
        "gaux).")
8140      PROC msg(18,136,4,0,"On a droit "+CHR$(142)+"galement "+CHR$

```

```

(136)+" deux figures")
8160 PROC msg(10,146,4,0,"donnant la somme des d"+CHR$(142)+"s mais
l'une,")
8170 PROC msg(10,156,4,0,"MINIMUM, doit "+CHR$(144)+"tre inf"+CHR$(
142)+"rieur "+CHR$(136)+" l'autre,")
8190 PROC msg(10,166,4,0,"MAXIMUM.")
8200 n$="SUITE":PROC btn
8220 PROC msg(18,36,4,0,"Cliquez sur 'JOUER' pour lancer les d"+
CHR$(142)+"s,")
8240 PROC msg(10,46,4,0,"puis sur ceux qui amorcent une figure et "+
CHR$(136))
8250 PROC msg(10,56,4,0,"nouveau sur 'JOUER'.")
8260 PROC msg(18,76,4,0,"Si vous avez une figure avant la fin des")
8270 PROC msg(10,86,4,0,"trois coups, cliquez sur 'SATISFAIT' puis")
8280 PROC msg(10,96,4,0,"sur une des figures 'UN' "+CHR$(136)+"
'MAXIMUM'.")
8290 PROC msg(18,116,4,0,"A la fin des trois coups, cliquez sur
une")
8300 PROC msg(10,126,4,0,"des figures ou sur 'PASSE'.")
8310 PROC msg(18,146,4,0,"Pendant la phase des trois coups, vous")
8320 PROC msg(10,156,4,0,"pouvez remettre les d"+CHR$(142)+"s en
place en cli-")
8340 PROC msg(10,166,4,0,"quant sur 'RECHOISIR'.")
8350 n$="SUITE":PROC btn
8380 PROC msg(18,36,4,0,"Les vingt tours termin"+CHR$(142)+"s vous
pouvez")
8390 PROC msg(10,46,4,0,"cliquer sur 'REJOUER' pour une nouvelle")
8400 PROC msg(10,56,4,0,"partie avec les m"+CHR$(144)+"mes joueurs
ou d'autres")
8410 PROC msg(10,66,4,0,"joueurs.")
8420 PROC msg(18,89,4,0,"Enfin, pour sortir du jeu cliquez sur")
8430 PROC msg(10,99,4,0,"'QUITTER'.")
8440 n$=" FIN ":PROC btn
8450 RETURN
8460 BOUTON: DEF PROC btn
8470 _SetRect(r,106,173,210,190):_SetSolidPenPat(0):_PaintRRect
(r,12,12)
8480 _SetRect(r,108,175,208,188):_SetSolidPenPat(15):_PaintRRect
(r,12,12)
8482 PROC msg(139,185,15,0,n$)
8484 PROC event:ON y%<173 OR y%>190 OR x%<106 OR x%>210 GOTO 8484
8486 PROC fen(4,24,316,196,4)
8488 END PROC btn

```

Pour devenir un as du GS Basic, offrez-vous :

**À LA DÉCOUVERTE
DU GS Basic**

Un ouvrage

TREMLIN MICRO

260 pages + disquette.

Utilisez le bulletin de commande, à la fin de la revue

DE L'APPLESOFT AU C

La majorité des lecteurs de *Tremplin Micro* connaissent l'AppleSoft et parmi eux, certains possèdent un GS. Le langage **C** apporte de nets avantages par rapport au Basic, entre autres la vitesse et la puissance de la programmation structurée. Voilà pourquoi nous vous proposons d'adapter vos programmes Basic grâce à quelques fonctions d'entrées/sorties et de conversions en **C**.

Ces fonctions utilisent trois outils du GS : TextTool, IntegerMath et SANE (Standard Apple Numerics Environment).

DESCRIPTION DES FONCTIONS

• **initIO :**

- Initialisation et sélection de l'écran comme périphérique de sortie et du clavier comme périphérique d'entrée.
- Equivalent Basic : **PR£3** suivi de **IN£3**.

• **selectScreen :**

- Sélection de l'écran comme périphérique de sortie.
- Equivalent Basic : **PR£3**.

• **selectPrinter :**

- Sélection de l'imprimante comme périphérique de sortie.
- Equivalent Basic : **PR£1**.

• **newLine :**

- Envoie un saut de ligne sur le périphérique de sortie courant.
- Equivalent Basic : **PRINT**.

• **clearScreen :**

- Efface l'écran et positionne le curseur en haut et à gauche.
- Equivalent Basic : **HOME**.

• **gotoXY :**

- Positionne le curseur à la position désignée sur l'écran.
- Equivalent Basic : **HTAB X%** suivi de **VTAB Y%**.

Paramètres :

X : type int, représente la position horizontale.

Y : type int, représente la position verticale.

• **printWord :**

- Envoie une chaîne de caractères sur le périphérique courant.
- Equivalent Basic : **PRINT A\$;**
- Paramètre :
C : type pointer, représente la chaîne de caractères de type **C** (chaîne se terminant par le code ASCII 0) à envoyer.

• **printLine :**

- Envoie une chaîne de caractères suivie d'un retour à la ligne sur le périphérique courant.
- Equivalent Basic : **PRINT A\$.**
- Paramètre :
C : type pointer, représente la chaîne de caractères de type **C** à envoyer.

• **inputString :**

- Entrée d'une chaîne de caractères au clavier.
- Equivalent Basic : **INPUT A\$.**
- Paramètres :
C : type pointer, représente l'adresse où sera stockée la chaîne entrée.
L : type int, représente le nombre maximal de caractères à entrer au clavier.

- Valeur de retour :
N : type int, représente le nombre de caractères réellement entrés au clavier.

- **printInt :**

- Conversion d'un nombre entier puis envoi sur le périphérique courant.
- Equivalent Basic : **PRINT I%**.
- Paramètres :
I : type int, représente la valeur à convertir et à envoyer.
L : type int, représente le nombre de caractères à justifier à droite.
F : type int, drapeau pour le format (0 non-signé, 1 signé).

- **InputInt :**

- Entrée d'un nombre entier au clavier puis conversion.
- Equivalent Basic : **INPUT I%**.
- Valeur de retour :
I : type int, valeur du nombre entré.

- **printLong :**

- Conversion d'un nombre entier long puis envoi sur le périphérique courant.
- Equivalent Basic : aucun.
- Paramètres :
I : type long, représente la valeur à convertir et à envoyer.
L : type int, représente le nombre de caractères à justifier à droite.
F : type int, drapeau pour le format (0 non-signé, 1 signé).

- **InputLong :**

- Entrée d'un nombre entier long au clavier puis conversion.
- Equivalent Basic : aucun.
- Valeur de retour :
I : type long, valeur du nombre entré.

- **printFit :**

- Conversion puis affichage d'un nombre flottant sur le périphérique courant.
- Equivalent Basic : **PRINT D**.
- Paramètres :
X : type extended, représente la valeur à convertir et à envoyer.
L : type int, représente le nombre de caractères à justifier à droite.
I : type extended, représente la valeur minimale pour le passage en notation scientifique.
S : type extended, représente la valeur maximale pour le passage en notation scientifique.
P : type int, représente le nombre de chiffres après la virgule.

- **InputFit :**

- Entrée d'un nombre flottant au clavier puis conversion.
- Equivalent Basic : **INPUT D**.
- Valeur de retour :
X : type extended, valeur du nombre entré.

MODE D'EMPLOI

Note : L'utilisation d'APW 1.0 et d'APWC 1.0 sont obligatoires (disponibles auprès de D.D.A.). Après avoir tapé les 3 fichiers (IOFUNC.C, IOFUNC.H et TESTIO.C), compiler IOFUNC.C et TESTIO.C :

```
COMPILE -E -W IOFUNC.C KEEP=$ CC=(-P)
COMPILE -E -W TESTIO.C KEEP=$ CC=(-P)
```

linker TESTIO.ROOT :

```
LINK 2/START TESTIO IOFUNC KEEP=TESTIO
```

Pour lancer le programme TESTIO, taper : TESTIO

Copier ensuite le fichier IOFUNC.H dans le dossier 2/CINCLUDE et IOFUNC.ROOT dans 2/ :

```
COPY IOFUNC.H 2/CINCLUDE
COPY IOFUNC.ROOT 2/
```

Pour utiliser les fonctions de IOFUNC quel que soit votre programme, il suffit d'inclure dans le fichier à compiler : `#include <IOFUNC.H>`

et lors de l'édition de liens, de joindre à la liste des fichiers IOFUNC.ROOT

```
LINK 2/START THEFILE 2/IOFUNC KEEP=THEFILE
```

BIBLIOGRAPHIE

- *CLEFS POUR APPLE IIGS* 2^e édition (P.S.I. par Nicole Bréaud-Pouliquen).
- *APPLE IIGS TECHNICAL REFERENCE* (Mc Graw Hill).
- *Text Tools* 0.17 (D.D.A.).
- *Integer Math Tools* 00:30 (D.D.A.).
- *SANE Tool set* ERS 0.80 (D.D.A.).

```
1
2 /*
3
4     Nom du fichier: IOFUNC.C
5     Description: Fonctions d'entrées/sorties et de conversions.
6     (c) 1988 CHARLY et F. SIGAL, et TREMLIN MICRO.
7     Version IIGS, pour Compilateur C APWC 1.0
8
9 */
10
11
12 /*
13     Inclut les définitions, structures et fonctions
14     utilisées dans les fonctions.
15 */
16
17 #include <types.h>
18 #include <texttool.h>
19 #include <intmath.h>
20 #include <sane.h>
21
22 /* Définitions internes. */
23
24 #define _carriageReturn 0x0D
25 #define _printerSlot 1L
26 #define _screenSlot 3L
27 #define _andMask 0x7F
28 #define _orMask 0
29
30 /*
31     Variables globales:
32     - drapeau si sélection imprimante.
33     - buffer servant aux entrées/sorties.
34     - séquence du retour chariot pour l'écran.
35 */
36
37 int _printerFlag;
38 char _iobuffer[81];
39 char _crSeq[]={_carriageReturn,0x0A,0è;
40
41 /* Structures utilisées pour la conversion des nombres flottants. */
42
43 decimal _decStrConv;
44 decform _decFrmConv;
45
46
47 /*
48     Initialisation des entrées/sorties.
49     Devices Pascal pour l'écran et sélection de l'écran,
50     mise en place des masques de filtrages.
51 */
52
53 initIO()
54 {
55     SetInGlobals(_andMask,_orMask);
```

Les numéros de lignes ne doivent pas être tapés. Ils ne sont indiqués que pour information.

```
56 SetOutGlobals(_andMask,_orMask);
57 SetErrGlobals(_andMask,_orMask);
58 SetInputDevice(pascalType,_screenSlot);
59 SetOutputDevice(pascalType,_screenSlot);
60 SetErrorDevice(pascalType,_screenSlot);
61 InitTextDev(input);
62 InitTextDev(output);
63 InitTextDev(errorOutput);
64 _printerFlag=FALSE;
65 }
66
67
68 /*
69 Sélection de l'imprimante.
70     Devices Basic pour l'imprimante.
71 */
72
73 selectPrinter()
74 {
75     if(!_printerFlag) /* Teste si imprimante non sélectionnée, */
76     { /* sinon on passe. */
77         SetOutputDevice(basicType,_printerSlot);
78         InitTextDev(output);
79         _printerFlag=TRUE;
80     }
81 }
82
83
84 /*
85 Sélection de l'écran.
86     Devices Pascal pour l'écran.
87 */
88
89 selectScreen()
90 {
91     if(_printerFlag) /* Teste si imprimante sélectionnée, */
92     { /* sinon on passe. */
93         SetOutputDevice(pascalType,_screenSlot);
94         InitTextDev(output);
95         _printerFlag=FALSE;
96     }
97 }
98
99
100 /*
101 Effacement de l'écran.
102     Envoi du caractère de controle.
103 */
104
105 clearScreen()
106 {
107     if(!_printerFlag) WriteChar(0x0C); /* Teste si imprimante non */
108 } /* sélectionnée, pour écran seulement. */
109
110
```

```

111 /*
112     Positionnement du curseur sur l'écran.
113     Voir code gotoxy de la carte 80 colonnes.
114 */
115
116 gotoXY(_x, _y)
117 char _x, _y;
118 {
119     if(!_printerFlag)                /* Teste si imprimante non          */
120     {                                  /* sélectionnée, pour écran seulement. */
121         _x+=32;                        /* Position X + 32.                  */
122         _y+=32;                        /* Position Y + 32.                  */
123         *_iobuffer=0x1E;               /* Code de gotoxy.                  */
124         *(_iobuffer+1)=_x;             /* Stocke position X.                */
125         *(_iobuffer+2)=_y;             /* Stocke position Y.                */
126         *(_iobuffer+3)=0;              /* Fin de chaine C, Stocke 0.        */
127         WriteCString(_iobuffer);      /* Envoi de la séquence de commande.*/
128     }
129 }
130
131 /*
132     Envoi un retour chariot sur le périphérique courant.
133     Teste le périphérique pour connaître le(s) code(s) à envoyer.
134 */
135
136 newLine()
137 {
138     if(_printerFlag) WriteChar(_carriageReturn); /* Teste si          */
139     /* imprimante sélectionnée, envoi du code retour chariot. */
140     else WriteCString(_crSeq); /* Sinon, envoi d'une séquence */
141     /* pour l'écran. */
142 }
143
144 /* Envoi d'une chaine de caractères sur le périphérique courant. */
145
146 printWord(_string)
147 char *_string;
148 {
149     WriteCString(_string);
150 }
151
152
153 /* Utilisation interne de cette fonction. */
154
155 int inputLine(_length)
156 int _length;
157 {
158     char _code;
159     int _count;
160
161     _count=0;
162     do
163     {                                  /* Lecture d'un caractère au clavier sans l'afficher.*/
164         _code=ReadChar(noEcho);
165         if((_code>0x1F)&&(_code<0x7F)) /* Teste si touche majuscule, */

```

```

166         {                               /* minuscule ou chiffre.      */
167                                         /* Affichage et stockage du caractère.*/
168         WriteChar(_code);
169         *(_iobuffer+_count++)=_code;
170         }
171                                         /* Teste si touche BackSpace ou Delete.*/
172         else if((_code==0x08||_code==0x7F)&&(_count>0))
173         {
174                                         /* Efface le dernier caractère entré. */
175         WriteChar(0x08);
176         --_count;
177         }
178     }
179     /*
180     Continue tant que le nombre maximal de caractères n'est pas atteint
181     ou que le code n'est pas celui de la touche Return.
182     */
183     while((_count<_length)&&(_code!=_carriageReturn)); /* Teste si */
184     /* touche Return pour annulation des caractères après le curseur. */
185     if(_code==_carriageReturn) WriteChar(0x1D);
186     *(_iobuffer+_count)=0; /* Fin de chaine C, stocke 0.      */
187     /* Retourne le nombre de caractères recus.                */
188     return(_count);
189 }
190
191
192 /* Entrée d'une chaine de caractères.                          */
193
194 int inputString(_string,_length)
195 char *_string;
196 int _length;
197 {
198     int _i, _count;
199
200     if(!_printerFlag) /* Teste si imprimante non sélectionnée, */
201     {                 /* pour écran seulement.                */
202         /*
203         Appel de la fonction inputLine et récupération du nombre de
204         caractères recus.
205         */
206         _count=inputLine(_length);
207         /* Recopie de la chaine de iobuffer dans _string.      */
208         for(_i=0;_i<_count;_i++)
209             *(_string+_i)=*(_iobuffer+_i);
210             /* Fin de chaine C, stocke 0.                      */
211         *(_string+_count)=0;
212         /* Retourne le nombre de caractères recus.            */
213         return(_count);
214     }
215 }
216
217
218 /* Envoi d'une chaine de caractères suivit d'un retour chariot. */
219
220 printLine(_string)

```

```
221 char *_string;
222 {
223     WriteCString(_string);
224     newLine();
225 }
226
227
228 /*Conversion et envoi d'un nombre entier sur le périphérique courant*/
229
230 printInt(_value,_just,_flag)
231 int _value, _just, _flag;
232 {
233     int _length;
234
235     /* Appel de la fonction Int2Dec pour effectuer la conversion. */
236     Int2Dec(_value,_iobuffer,_just,_flag);
237     *(_iobuffer+_just)=0; /* Fin de chaine C, stocke 0. */
238     WriteCString(_iobuffer); /* Envoi de la chaine de caractères.*/
239 }
240
241
242 /* Entrée d'un nombre entier au clavier. */
243
244 int inputInt()
245 {
246     int _length;
247
248
249     if(!_printerFlag) /* Test si imprimante non sélectionnée, */
250     { /* pour écran seulement. */
251         _length=inputLine(20); /*Lecture d'une chaine de 20 caractères */
252         /*
253          Retourne le nombre entier,
254          après conversion grace à un appel Dec2Int.
255          */
256         return(Dec2Int(_iobuffer,_length,1));
257     }
258 }
259
260
261 /* Conversion puis envoi d'un nombre entier long
262 sur le périphérique courant. */
263 printLong(_value,_just,_flag)
264 long _value;
265 int _just, _flag;
266 {
267     int _length;
268
269     /* Appel de la fonction Long2Dec pour effectuer la conversion. */
270     Long2Dec(_value,_iobuffer,_just,_flag);
271     *(_iobuffer+_just)=0; /* Fin de chaine C, stocke 0. */
272     WriteCString(_iobuffer); /* Envoi de la chaine de caractères. */
273 }
274
275
```

```
276 /* Entrée d'un nombre entier long au clavier. */
277
278 long inputLong()
279 {
280     int _length;
281
282     if(!_printerFlag) /* Teste si imprimante non sélectionnée, */
283     { /* pour écran seulement. */
284         _length=inputLine(20); /* Lecture d'une chaîne de 20 caractères.*/
285         /*
286             Retourne le nombre entier long,
287             après conversion grâce à un appel Dec2Long.
288         */
289         return(Dec2Long(_iobuffer,_length,1));
290     }
291 }
292
293
294 /* Conversion puis envoi d'un nombre flottant */
295     sur le périphérique courant. */
296 printFlt(_value,_length,_vmin,_vmax,_virg)
297 extended _value;
298 int _length;
299 extended _vmin, _vmax;
300 int _virg;
301 {
302     extended _valabs;
303     int _index, _jmove;
304     char *_string;
305
306     /*
307         Si le nombre est égal à 0.0,
308         on stocke "0.0" dans la chaîne de caractères.
309     */
310     if(_value==0.0)
311     {
312         *_iobuffer='0';
313         *(_iobuffer+1)='.';
314         *(_iobuffer+2)='0';
315         *(_iobuffer+3)=0;
316     }
317     else
318     {
319         _valabs=fabs(_value); /* Valeur absolue. */
320
321         /* Teste les limites pour le passage en mode scientifique. */
322         if((_valabs>=_vmax)||(_valabs<_vmin))
323         {
324             /* En dehors, notation décimale. */
325             _decFrmConv.style=FLOATDECIMAL;
326             _decFrmConv.digits=_virg+1;
327         }
328         else
329         {
330             /* Entre les deux limites, notation scientifique. */
```

```

331         _decFrmConv.style=FIXEDEDECIMAL;
332         _decFrmConv.digits=_virg;
333     }
334
335     /* Appel aux fonctions num2dec et dec2str pour les conversions.*/
336     num2dec(&_decFrmConv,_value,&_decStrConv);
337     dec2str(&_decFrmConv,&_decStrConv,_iobuffer);
338 }
339
340 /* Donne la longueur de la chaine dans _iobuffer. */
341 _length=strlen(_iobuffer);
342 /* Calcule la valeur de déplacement pour la justification. */
343 _jmove=_length-_index;
344 /* Si valeur de déplacement supérieure à 0 justification du nombre */
345 if(_jmove>0)
346 {
347     /* Décalage. */
348     for(;_index>=0;--_index)
349         *(_iobuffer+_index+_jmove)=*(_iobuffer+_index);
350     /* Remplacement de la zone de justification par des espaces.*/
351     for(_index=_jmove-1;_index>=0;--_index)
352         *(_iobuffer+_index)=' ';
353     /* Fin de chaine C, stocke 0. */
354     *(_iobuffer+_length)=0;
355 }
356 /* envoi de la chaine de caractères.*/
357 WriteCString(_iobuffer);
358 }
359
360
361 /* Entrée d'un nombre flottant au clavier. */
362
363 extended inputFlt()
364 é
365     short _ix, _vp;
366
367     if(!_printerFlag) /* Teste si imprimante non sélectionnée, */
368     { /* pour écran seulement. */
369         /* Flag de position dans la chaine */
370         _ix=0;
371         /* Lecture d'une chaine de 30 caractères.*/
372         inputLine(30);
373         /* Si chaine vide, mise en forme de la chaine.*/
374         if(*_iobuffer==0)
375         {
376             *_iobuffer=1;
377             *(_iobuffer+1)='0';
378         }
379         /* Appel à str2dec pour la conversion.*/
380         str2dec(_iobuffer,&_ix,&_decStrConv,&_vp);
381         /* Teste le drapeau d'erreur et traitement.*/
382         if(_vp) return(dec2num(&_decStrConv));
383         else return(0.0);
384     }
385 }

```

```
1
2 /*
3
4     Nom du fichier: IOFUNC.H
5     Description: fichier Header des fonctions de IOFUNC.C
6     (c) 1988 CHARLY et F. SIGAL, et TREMLIN MICRO.
7     Version IIGS, pour Compilateur C APWC 1.0
8
9 */
10
11
12 extern int     inputString();
13 extern int     inputInt();
14 extern long    inputLong();
15 extern extended inputFlt();
```

```
1
2 /*
3
4     Nom du fichier: TESTIO.C
5     Description: Programme test des fonctions de IOFUNC.C
6     (c) 1988 C. Mafaure, S.N. Cehef et TREMLIN-MICRO
7     Version IIGS, pour Compilateur C APWC 1.0
8
9 */
10
11
12 #include "iofunc.h"
13
14
15 int n;
16 long l;
17 extended d=0;
18 char c[31];
19 char rp[2];
20
21
22 main()
23 {
24     initIO();
25     clearScreen();
26
27     printWord("Entrez un nombre entier : ");
28     n=inputInt();
29     newLine();
30     printWord(" 'n' est égal à : ");
31     printInt(n,5,1);
32     newLine();
33     newLine();
34
35     printWord("Entrez un nombre entier long : ");
36     l=inputLong();
```

```
37  newLine();
38  printWord(" 'l' est égal à : ");
39  printLong(l,12,1);
40  newLine();
41  newLine();
42
43  printWord("Entrez un nombre décimal : ");
44  d=inputFlt();
45  newLine();
46  printWord(" 'd' est égal à : ");
47  printFlt(d,12,1e-5,1e+5,5);
48  newLine();
49  newLine();
50
51  printWord("Entrez une chaine de caratères : ");
52  inputString(c,30);
53  newLine();
54  printWord(" 'c' est égal à : ");
55  printLine(c);
56  newLine();
57
58  printWord("Disposez-vous d'une imprimante en slot 1, prête à
59  l'utilisation ?");
60  inputString(rp,1);
61  if(rp[0]=='O' || rp[0]=='o')
62  {
63      selectPrinter();
64      printWord(" 'n' est égal à : ");
65      printInt(n,5,1);
66      newLine();
67      printWord(" 'l' est égal à : ");
68      printLong(l,12,1);
69      newLine();
70      printWord(" 'd' est égal à : ");
71      printFlt(d,12,1e-5,1e+5,5);
72      newLine();
73      printWord(" 'c' est égal à : ");
74      printLine(c);
75      newLine();
76      selectScreen();
77  }
78  else clearScreen();
79
80  gotoXY(25,10);
81  printLine(" Test terminé, à bientôt...");
81 }
```

FACTARIF

Vous le savez peut-être déjà, je suis céramiste depuis 1971. Après avoir dû refaire des factures et des lettres à cause de fautes de frappe disgracieuses, j'ai décidé que la micro informatique était un outil à essayer. Naïf, je croyais que cet instrument du progrès allait me décharger de tâches ennuyeuses, me permettre de disposer de plus de temps libre et d'avoir un bureau bien rangé. Si établir une facture n'est plus en effet une corvée, puisqu'il est facile de la refaire en cas d'erreur, côté courrier, j'ai plutôt davantage de travail car le fait de disposer d'un traitement de texte incite à prendre la plume plus souvent que par le passé.

Pour ce qui est du temps libre et du bureau dégagé, c'est complètement raté ! Entre les piles de revues, de livres, de disquettes de tous calibres, on trouve des câbles, des interfaces et, parfois... des factures de fournisseurs ou de clients ! A ce jour, rien ne semble s'être perdu, mais ça relève du miracle. D'autant que, depuis près de deux ans, je réponds à vos questions quelquefois farfelues, parfois très embarrassantes.

Après s'être drogué au langage machine pendant de longs mois, Guy-Hachette réclame du Basic. Comme, par ailleurs, on me demandait un programme de facturation, j'ai décidé de vous communiquer mon auxiliaire fidèle : **FACTARIF**.

En fait, je ne vous en donne qu'une partie car la pagination de *Tremplin Micro* n'est pas extensible. Pour utiliser le programme publié, il faudra que vous prépariez vos fichiers **CLIENTS** et **TARIF** à l'aide d'un traitement de texte comme **APPLEWRITER** ou par l'intermédiaire d'un petit programme **BASIC** créant les deux fichiers séquentiels conformément aux structures suivantes :

CLIENTS

Stocker le nombre de fiches clients, puis les

fiches proprement dites qui sont composées de 5 rubriques, séparées par des virgules. Des **RETURN** feraient l'affaire, mais lors d'un listage du fichier à l'aide de **TYPE** les virgules permettent de n'utiliser qu'une ligne par fiche (au lieu de 5 avec des **RETURN**).

Les rubriques sont : Patronyme, Rue, Complément d'adresse, Code Postal + Ville, Taux de remise éventuellement consenti (ce dernier détail est un ajout spécial *Tremplin Micro* répondant au souhait d'un correspondant ; moi, je ne fais pas de remise).

TARIF

Un en-tête regroupe : la moitié du nombre d'articles répertoriés (arrondi à l'unité supérieure) pour l'affichage sur deux colonnes du tarif. C'est un reliquat de la toute première version et je n'ai jamais pris le temps de remplacer par le nombre d'articles. Ensuite vient le taux de TVA (ben oui, je n'en traite qu'un !). A mes débuts, j'accomplissais parfois des tâches que l'ordinateur aurait fort bien pu faire. C'est pour cela que la troisième information stockée est la chaîne 'TVA xx.xx%'. Les courageux pourront effacer ces rides. Enfin, une chaîne de 8 caractères informe de la date de mise en vigueur du tarif (c'est bien utile pour moucher un grincheux qui clame devant quinze personnes que vos tarifs ont encore été augmentés pour l'arrivée des touristes...).

Viennent ensuite les fiches articles contenant ici encore 5 rubriques séparées par des virgules. Il s'agit :

1. De la désignation de l'article ;
2. De son prix H.T. en centimes ;
3. Du prix T.T.C., en francs (c'est hétérogène, mais ça permet de gagner quelques octets) ;
4. La hauteur en cm ;
5. Le diamètre en cm.

(suite page 38)

Maintenant que vous avez les fichiers de données, vous pouvez employer **FACTARIF**, composé en fait d'un programme BASIC auquel est accrochée une routine en langage machine permettant d'accéder aisément à un article sans devoir mémoriser son numéro d'identification. On utilise également la commande externe ProDOS 'FLOAD', publiée dans *Tremplin Micro n°20*.

Deux fonctions principales sont offertes d'entrée :

- Etablissement d'une FACTURE ;
- Impression d'un TARIF.

L'option TARIF n'appelle pas de commentaire particulier, mais FACTURE est plus complexe. Dans un premier temps, vous devrez saisir les quantités d'articles vendus, chacun étant accessible par son numéro d'ordre, formule simple, mais peu agréable. Il est heureusement possible de donner tout ou partie de la désignation de l'article en question. La routine en langage machine, précédemment citée, vous proposera successivement tous les articles contenant cette sous-chaîne. Il vous restera à valider en tapant O. Si vous dépassez d'un rang l'article choisi, inutile de refaire un balayage complet, tapez *flèche montante* (ou flèche à gauche) qui permettra de récupérer l'avant-dernière proposition. Comme j'aime le confort, *DEJ* vous donnera accès à *Déjeuner carrelé* (ou décoré).

En cas d'erreur sur une quantité, il est possible de la modifier en saisissant une complémentaire qui peut être négative. Il arrive parfois que ma femme vienne me signaler qu'un client ayant demandé une facture a modifié sa sélection d'articles pendant que je commençais à saisir ce qu'il avait choisi. On peut aussi saisir jusqu'à cinq articles hors *catalogue* (c'est indispensable quand, comme moi, on vend fréquemment des pièces uniques).

Une fois toutes les ventes enregistrées, tapez RETURN pour passer aux opérations complémentaires : saisie des éventuels frais de port, avoir, acompte...

Il ne restera plus alors qu'à enregistrer l'identité du client, soit par sélection dans la liste en mémoire, pour un client connu, soit par saisie *au vol* pour un client occasionnel.

Toutes les opérations d'impression sont codées

pour IMAGEWRITER2. Il est facile de passer en version compatible DMP ou IMW1 puisque la seule commande spécifique IMW2 est l'envoi de la séquence ESC "M" qui force le mode Quasi-Courrier.

L'utilisation de la commande externe FLOAD (*T.M. 20*) permet de réaliser un en-tête à l'aide de caractères personnalisés. Dans la version publiée, trois fontes sont utilisées :

- COUNT qui figurait déjà sur la disquette *T.M. 20* ;
 - CARAC.DOUBLE
 - CARAC.INC
- sur la disquette *T.M. 7*

Vous avez bien entendu toute liberté pour adapter l'en-tête à vos propres besoins. Je signale à ceux qui ne l'auraient pas remarqué que les fontes COUNT, EPSDROITE et ITALEPS de *T.M. 20* contiennent le nouveau caractère Franc en remplacement du £.

Pour tout simplifier, le programme est chargé en deux temps. Le STARTUP utilise PRO.FP pour remettre le système en configuration normale, puis installe FLOAD et copie les fontes dans le disque virtuel de la carte 80 colonnes étendue (on pourrait se servir de /RAM5 sur GS ou /RAM4 sur IIc étendu). Enfin, il lance l'exécution de **FACTARIF** lui-même. Si vous ne disposez pas de disque virtuel, il est possible de s'en passer en chargeant les fontes depuis le disque magnétique ; c'est simplement plus lent.

En examinant le listage du programme, vous remarquerez sans doute qu'aucune instruction ne charge la routine CONT2 précédemment citée. C'est normal, elle est accrochée à la fin du programme BASIC et initialisée par la ligne 8.

Si l'association BASIC — Langage Machine ne vous est pas familière, je vous explique rapidement la marche à suivre :

1. Saisir, puis sauver SANS l'exécuter, le programme BASIC sous le titre BAS.SEUL. Attention, le programme comporte de nombreuses lignes très longues. Elles peuvent toutes être saisies en utilisant le symbole ? en remplacement des multiples PRINT et en s'abstenant de taper les espaces ajoutés par la fonction LIST de l'APPLESOFT lors des listages. Un éditeur de BASIC se révèle une fois encore un accessoire utile. ProCMD,

toujours lui, contient, entre autres gâteries, un éditeur de lignes et un RENUMBER, performants et fiables.

2. Saisir et sauver la routine CONT2 (avec MERLIN ou tout autre assembleur si vous êtes outillé).

3. Pour faire du travail 'propre', mettre à 0 une zone importante de la mémoire, (disons de \$2000 à \$3000) à l'aide de la commande moniteur :

```
2000:00 N 2001 < 2000.2FFFM (pour les
GSmen 00 < 2000.3000Z)
```

Charger BAS.SEUL, passer en moniteur et faire AF.B0 (RETURN) ; l'adresse de fin du Basic s'affiche alors (octet bas d'abord). Ce devrait être \$28BD si aucune modification n'a été apportée au programme, mais comme je ne saurais vous reprocher de ne pas avoir tapé les REMarques, je préfère représenter symboliquement cette adresse par \$wxyz. Retour au mode Basic. BLOAD CONT2,A\$wxyz. Retour en moniteur et faire wxyz LLLLLLLLL (RETURN) (List 10 fois).

Les deux dernières instructions listées seront alors JMP \$DA63, JMP \$DEC9. Si c'est bien le cas, faire encore un L (RETURN). La première ligne de l'écran portera stuv- 00 00.

Il faudra alors faire AF: uv st (RETURN) et ENFIN retourner en BASIC et faire SAVE FACTARIF.

Ouf ! C'est plus difficile à décrire qu'à exécuter et, si vous le demandez fermement mais poliment, je vous offrirai peut-être un jour une commande ProDOS qui réalise ce type d'accouplement automatiquement.

Je profite de l'occasion pour signaler que, depuis des mois, RELOPRO.S contient une erreur que je viens de repérer. Dans la routine PROTECT, la ligne 313 a été amputée (ça va vite avec un éditeur plein écran) et doit se lire :

```
313 ADC R_OFFSET+1
```

L'absence du (+1) final est responsable de l'apparition désagréable d'un 'NO BUFFERS AVAILABLE' du meilleur effet.

Yvan KOENIG — 16 mai 1988.

FACTARIF

```

10 D$ = CHR$ (4)
20 PRINT D$"-PRO.FP"
30 PRINT D$"-FLOAD"
40 PRINT D$"BLOAD COUNT,A$4000": PRINT D$"BSAVE /RAM/COUNT,A$4000,L955"
50 PRINT D$"BLOAD CARAC.INC,A$4000": PRINT D$"BSAVE /RAM/CARAC.INC,A$40
00,L955"
60 PRINT D$"BLOAD CARAC.DOUBLE,A$4000": PRINT D$"BSAVE /RAM/CARAC.DOUBL
E,A$4000,L919"
70 PRINT D$"-FACTARIF"
1 REM *****
2 REM * Installe FACTARIF *
3 REM *****
                                     B3A4
                                     9248
                                     8DF9
                                     142E
                                     96E0
                                     48A2
                                     28D3
1 REM *****
2 REM * FACTARIF *
3 REM *****

6 TEXT : PRINT CHR$ (21):D$ = CHR$ (4):G$ = CHR$ (7):R$ = CHR$ (13
):K$ = - 16368
8 CALL PEEK (175) + 256 * PEEK (176) - 416
10 HOME :AZ% = 48:CE = 100:ZE = .01:SE$ = "-----":E% = 27:E$ = CHR$
(E%):W0$ = CHR$ (E%) + "0":FI$ = "TARIF":A9% = 57:AP% = 46:AV% = 44
: DEF FN A(R) = INT (R + .5) / CE
11 SLOT = 1: REM "Port Imprimante
12 PRINT D$"OPEN"FI$: PRINT D$"READ"FI$: INPUT Z1,AA,N$,DV$:AB = CE * A
A / (AA + CE): PRINT D$:Z = 2 * Z1:Z5 = Z + 5: PRINT "TARIF "en vigu
eur au "DV$: PRINT : HTAB 7: PRINT Z" articles répertoriés": DIM P(Z
5,1),N0$(Z5),T(Z5,1),Q%(Z5)
                                     0E41
                                     E5C0
                                     BA7E
                                     1243
                                     9158

```

/ FACTARIF /

```

13 NO$(0) = LEFT$(N$,6) + "," + MID$(N$,8): PRINT D$"READ"FI$: FO
R N = 1 TO 2: INPUT NO$(N),T0,P(N,1),T(N,0),T(N,1):P(N,0) = T0 / C
E: NEXT : PRINT D$:T0 = 0:CL$ = "CLIENTS": PRINT D$"OPEN"CL$: PRIN
T D$"READ"CL$: INPUT CN: PRINT D$
8CB5

17 JJ% = CN / 10: DIM R%(CN),AD$(CN,3): PRINT D$"READ"CL$: FOR I = 1
TO CN: INPUT AD$(I,0),AD$(I,1),AD$(I,2),AD$(I,3),R%(I): NEXT : PRI
NT D$"CLOSE"
780E
49DA
A963

20 VTAB 5: CALL - 958: GOSUB 770: IF NOT S1 THEN HOME : END
30 IF S1 = 1 THEN GOSUB 130:DF$ = RE$:MF% = M%: GOSUB 440
40 HOME : GOSUB 830: ON PR$ = "N" GOTO 20: GOSUB 1200: GOSUB 3000: ON
S1 GOSUB 350,700: GOSUB 3230: HOME : VTAB 22: FLASH : PRINT "ENCOR
E UNE ": GOSUB 830: HOME : ON PR$ = "O" GOTO 40: GOSUB 900: FOR I
= 1 TO 25:Q%(I) = 0: NEXT : GOTO 20
4F6F

59 REM ctJctJFORMATTAGESctJ
60 PRINT RIGHT$( " " + STR$(W),3);: RETURN
5F71
70 IF ABS(W) < ZE THEN W = 0:W$ = " "000": GOTO 80
1EDA
72 W$ = RIGHT$( " " + STR$( INT (.1 + CE * W)),7)
E8FD
80 IF S2 AND RIGHT$(W$,2) = "00" THEN W$ = LEFT$(W$,5) + " "": R
ETURN
4F01
90 W$ = LEFT$(W$,5) + "," + RIGHT$(W$,2): RETURN
8E5E
100 IF NOT W THEN W$ = ". "": RETURN
CAA4
110 IF W = INT(W) THEN W$ = RIGHT$( " " + STR$(W),2) + " "": RETU
RN
52E5
120 W$ = RIGHT$( " " + STR$(W * 10),3):W$ = LEFT$(W$,2) + "," +
MID$(W$,3): RETURN
FF9E

129 REM "ctJctJMASQUE DE SAISIEctJ
130 H% = 7:V% = 4:T% = 2:L% = 8: VTAB V%: PRINT "Date ":JJ MM AN"
E0B9
135 P% = 0
C775
140 CH$ = "":RE$ = "": VTAB V%: HTAB H%: PRINT "":
F147
150 POKE K%,0: GET X$:X% = ASC(X$): IF X% = 13 THEN CALL - 868: PR
INT X$: ON CH$ = "": GOTO 210: GOTO 200
C39A
155 IF X% = E% AND CH$ = "": THEN PRINT : RETURN
CE97
160 ON X% < > 8 GOTO 180: ON CH$ < > "": GOTO 170: ON T% < > 6 GOTO 15
0: POP : GOTO 570
C656
165 IF RIGHT$(CH$,1) = "." THEN P% = 0
436E
170 CH$ = LEFT$(CH$, LEN(CH$) - 1): PRINT X$."X$;: GOTO 150
E884
180 ON T% GOSUB 230,260,220,1000,225,220: ON Q% GOTO 150:CH$ = CH$ + X
$: PRINT X$;: ON LEN(CH$) < L% + 1 GOTO 150: PRINT R$
2A1B
200 RE$ = MID$(CH$,2)
04D3
210 RETURN
63B1

219 REM "ctJctJNB ARTICLESctJ
220 Q% = X% < > 45 AND (X% < A2% OR X% > A9%): RETURN
1D9C
224 REM ctJ
225 Q% = X% < > 63 AND X% < > 44 AND (X% < A2% OR X% > A9%): RETURN
CDEE
229 REM ctJctJSOMMEctJ
230 Q% = 1:PQ% = X% = AP% OR X% = AV%: IF (P% AND PQ%) OR X% > A9% OR
PQ% < 1 AND X% < A2% THEN RETURN
C100
240 IF PQ% THEN X$ = ".":P% = 1
69A5
250 Q% = 0: RETURN
9761
259 REM ctJctJDATEctJ
260 Q% = 1: IF X% < A2% OR X% > A9% THEN RETURN
8E9A
270 IF LEN(CH$) = 2 THEN J% = VAL(MID$(CH$ + X$,2)): IF J% < 1 0
R J% > 31 THEN RETURN
E694
280 IF LEN(CH$) = 5 THEN M% = VAL(MID$(CH$ + X$,5)): IF M% < 1 0
R M% > 12 THEN RETURN
84A2

```

```

290 IF LEN (CH$) = 8 THEN A% = VAL ( MID$ (CH$ + X$,8)):A% = 1900 +
  A% + CE * (A% < 50): GOSUB 330: IF J% > F% THEN POP : GOTO 140      45B8
300 IF LEN (CH$) = 5 OR LEN (CH$) = 2 THEN X$ = X$ + " "           8A00
310 Q% = 0: RETURN                                                  9761
319 REM "ctJctJDERNIER JOUR DU MOISctJ
320 M% = MF% + (CR = 5) * (1 - (MF% = 12) * 12)                    502B
330 F% = 31 - (M% = 4 OR M% = 6 OR M% = 9 OR M% = 11): IF M% = 2 THEN
  F% = 28:T1 = A% / 4:T1 = T1 - INT (T1):T2 = A% / CE:T2 = T2 - IN
  T (T2):T3 = A% / 400:T3 = T3 - INT (T3): IF T1 + T2 + T3 = 0 OR T
  1 = 0 AND T2 THEN F% = 29                                         863B
340 RETURN                                                           63B1
349 REM "ctJctJIMPRESSION FACTUREctJ
350 LT = 5: FOR N = 1 TO 25:LT = LT + NOT NOT Q%(N): NEXT :LT = LT +
  3 * NOT NOT R + 2 * NOT NOT (E AND AV) + NOT NOT E + NOT N
  OT AV + 2 * NOT S2 + NOT NOT AK + 2 * AS                            9A70
351 MG = 11                                                         7AC6
354 PRINT TAB( 91)"VALLAURIS le "DF$: IF LT < 40 THEN PRINT W0$:   BADE
356 PRINT : PRINT : PRINT : PRINT SPC( MG)CL$: FOR I = 1 TO 5: PRINT
  : NEXT :T0 = 0: PRINT SPC( MG)"Code ""Quantité ""Désignation" SPC
  ( 36)"Prix"B$ SPC( 13)"Prix Total": PRINT                          0782
357 FOR N = 1 TO 25: ON NOT Q%(N) GOTO 370                          062E
360 PRINT SPC( MG): IF N > 2 THEN PRINT SPC( 3): GOTO 365          82B8
362 W = N: GOSUB 60                                                 1CC5
365 PRINT SPC( 6):W = Q%(N): GOSUB 60: PRINT SPC( 6)NO$(N) SPC( 48 -
  LEN (NO$(N))):W = P(N,S2): GOSUB 70: PRINT W$ SPC( 14):W = Q%(N) *
  P(N,S2): GOSUB 70: PRINT W$:T0 = T0 + W                            B480
370 NEXT : IF R THEN PRINT SPC( 88 + MG)SE$:W = T0: GOSUB 70: PRINT
  SPC( 43 + MG)AV$ SPC( 42)W$:W = FN A(T0 * R): GOSUB 70: PRINT SP
  C( 48 + MG)"Remise "R" %" SPC( 31 - LEN ( STR$ (R)))W$:T0 = T0 -
  W                                                                    E5E1
380 ON NOT E AND NOT AV GOTO 410: PRINT SPC( 88 + MG)SE$:W = T0: GO
  SUB 70: PRINT SPC( 43 + MG)"Marchandises "AV$ SPC( 29)W$: IF E TH
  EN W = E: GOSUB 70: PRINT SPC( 43 + MG)"Port "AV$ SPC( 37)W$:T0 =
  T0 + E                                                                BA71
400 IF AV THEN W = AV: GOSUB 72: PRINT SPC( 43 + MG)"Avoir "AV$" du "
  DA$ SPC( 24)W$:T0 = T0 + AV                                         0360
410 PRINT SPC( 88 + MG)SE$:W = T0: GOSUB 70: PRINT SPC( 43 + MG)"Tot
  al "AV$ SPC( 36)W$: IF NOT S2 THEN W = FN A(AA * T0):T0 = T0 + W
  : GOSUB 70: PRINT SPC( 48 + MG)NO$(0) SPC( 30)W$: PRINT SPC( 88
  + MG)SE$:W = T0: GOSUB 70: PRINT SPC( 43 + MG)"Net TTC" SPC( 38)W
  $                                                                    493A
420 IF S2 THEN W = FN A(AB * T0): GOSUB 70: PRINT SPC( 48 + MG)"dont
  "NO$(0) SPC( 3)W$                                                  B880
424 ON NOT AK GOTO 434: PRINT SPC( 17 + MG)"Acompte de";:W = AK: IF
  W - INT (W) THEN GOSUB 70: GOTO 428                                DEEE
426 W$ = " " + STR$ ( INT (W))                                       7427
428 PRINT W$" frs. Reste à payer";:W = T0 - AK: IF W - INT (W) THEN
  GOSUB 70: GOTO 432                                                  8BA6
430 W$ = " " + STR$ ( INT (W))                                       7427
432 PRINT W$" frs."                                                  B1B2
434 PRINT : IF AS THEN PRINT SPC( MG)"Téléphoner en cas d'avarie pou
  r communication du numéro de la police d'assurance.": PRINT        408B
435 PRINT SPC( MG)"Adhérent d'un C.G.A. nous acceptons les règlements
  par chèque. "";: IF S1 < > 2 THEN PRINT DW$                       7E52
438 RETURN                                                           63B1

```

/ FACTARIF /

```

439 REM "ctJctJSAISIE FACTUREctJ
440 HOME : HTAB 14: PRINT "FACTURE" B$ du "DF$: VTAB 23: PRINT "si ter
miné tapez "; INVERSE : PRINT "RETURN": NORMAL : POKE 35,20 DF6D
450 V% = 14:H% = 14:L% = 2:T% = 5: VTAB V%: PRINT "Code article :.. "
"( ? pour consulter)": GOSUB 135:N = VAL (RE$): IF RE$ = "?" OR R
E$ = "," THEN GOSUB 2000 8DAA
455 ON NOT N GOTO 490: ON N > 25 GOTO 450: ON N > 2 AND NOT Q%(N) GO
SUB 660 F547
460 P% = 1:V% = 18:H% = 10:L% = 4:T% = 3: VTAB V% - 2: CALL - 868: PR
INT NO$(N): VTAB V%: PRINT "Quantité :...": GOSUB 140: ON X% = E%
GOTO 450:Q% = VAL (RE$):Q%(N) = Q%(N) + Q% 2B09
470 VTAB 5: CALL - 868: PRINT "ajout "; INVERSE : PRINT "ARTICLE "N"
"; NORMAL :KQ = Q% * P(N,S2): PRINT " "Q%" x "P(N,S2)" = "KQ:T0
= T0 + KQ: VTAB 7: CALL - 868: PRINT "cumul:"Q%(N)" "NO$(N) 38D6
480 VTAB 9: CALL - 868: HTAB 15: PRINT "total partiel = "T0: CALL -
958: GOTO 450 A9E7
489 REM "ctJctJEXPEDITION Etc..ctJ
490 TEXT : VTAB 2: CALL - 958 19CC
500 T% = 1:V% = 3:H% = 16:L% = 6: VTAB V%: PRINT "Expédition "AV$ " :..
....": GOSUB 135:E = VAL (RE$) 88CF
510 V% = 5:H% = 16:L% = 2: VTAB V%: PRINT " "Remise en % :..": GOSUB
135: ON X% = E% GOTO 500:R = VAL (RE$) 7722
520 T% = 1:V% = 7:L% = 7:H% = 11: VTAB V%: PRINT "Avoir "AV$ " :.....
": GOSUB 135: ON X% = E% GOTO 510:AV = - VAL (RE$): IF NOT AV T
HEN 530 D4F0
525 T% = 2:H% = 27:L% = 8: VTAB V%: HTAB 22: PRINT "Date :.. .. .": G
OSUB 135: ON X% = E% GOTO 520:DA$ = RE$ 5E75
530 T% = 1:V% = 9:H% = 16:L% = 7: VTAB V%: PRINT SPC( 6)"Acompte " :..
....": GOSUB 135: IF X% = E% THEN ON NOT AV GOTO 520: GOTO 525 3FF7
535 IF E THEN VTAB 11: HTAB 5: PRINT "Assurance": GOSUB 840:AS = PR$
= "0" B624
539 REM ctJctJCONDITIONSctJ
540 AK = VAL (RE$): VTAB 16: PRINT "1 : "sans condition": PRINT "2 : "r
églé ce jour": PRINT "3 : "à réception": PRINT "4 : "règlement fin d
e mois": PRINT "5 : "30 jours fin de mois": PRINT : INVERSE : PRINT
"VOTRE CHOIX": NORMAL : PRINT " < >": HTAB 15 C3D8
550 POKE K%,0: GET PR$:CR = VAL (PR$): ON CR < 1 OR CR > 5 GOTO 550:
PRINT CR: ON CR GOSUB 620,630,640,650,650: VTAB 16: CALL - 958: H
TAB (41 - LEN (DW$)) / 2: PRINT DW$ 5FE6
559 REM "ctJctJSELECTION CLIENTctJ
560 V% = 21: VTAB V%: HTAB 10: PRINT "nouveau client": GOSUB 835: IF
PR$ = "0" THEN C% = 0:T% = 4:H% = 15:L% = 80: VTAB V%: CALL - 958
: PRINT "Nom & adresse :...": GOSUB 140:CL$ = RE$: GOSUB 615: RET
URN CD8A
570 P% = 1:T% = 6:V% = 23:H% = 33:L% = 2: FOR J = 0 TO JJ%: HOME : FOR
I = 1 TO 10: IF I + J * 10 > CN THEN I = 10:J = JJ%: GOTO 590 98B5
580 PRINT RIGHT$ ("0" + STR$ (I + J * 10) + " ",4)AD$(I + J * 10,0)
: PRINT 9467
590 NEXT 0582
600 VTAB V%: PRINT G$"numéro client "ou RETURN ou <= " :..": GOSUB 140:
C% = VAL (RE$): ON C% > CN GOTO 600: IF C% > J * 10 AND C% < 11 +
J * 10 THEN J = JJ% 4C6A
601 NEXT :CL$ = AD$(C%,0) + ". " + AD$(C%,1) + ".": IF AD$(C%,2) > ""
THEN CL$ = CL$ + AD$(C%,2) 9254
602 IF R%(C%) THEN RK = R%(C%) 363E

```

```

611 IF RK THEN R = RK
612 CL$ = CL$ + "." + AD$(C%,3): RETURN
614 REM "ctJctJDécomposition adressedctJ
615 LA = LEN (CL$):PT = 0: FOR I = 0 TO 3:AD$(0,I) = " ": NEXT : FOR I
= 1 TO LA:X$ = MID$ (CL$,I,1): IF X$ = "." THEN PT = PT + (PT < 3
):LA = LA + 1 + (PT = 1): GOTO 617
616 AD$(0,PT) = AD$(0,PT) + X$
617 NEXT : RETURN
619 REM "ctJctJCONDITIONS DE REGLEMENTctJ
620 DW$ = " ": RETURN
630 DW$ = "Réglé ce jour": RETURN
640 DW$ = "Règlement à réception": RETURN
650 GOSUB 320:DW$ = "Règlement avant le " + STR$ (F%) + " " + RIGHT$
("0" + STR$ (M%) + " ",3) + RIGHT$ ( STR$ ((M% = 1) + A%),2): RE
TURN
659 REM "ctJctJARTICLE NON REPERTORIEctJ
660 T% = 4:V% = 16:H% = 13:L% = 24: VTAB V%: PRINT "Désignation :.....
...": GOSUB 140:NO$(N) = RE$:T% = 1:V% = 18:H% = 6:L% = 7: VTAB V%
: PRINT "Prix :.....": GOSUB 140: IF X% = E% THEN 660
670 P(N,S2) = VAL (RE$): RETURN
699 REM "ctJctJIMPRESSION TARIFctJ
700 MG = 7: GOSUB 435: PRINT TA$DV$: PRINT :S4 = 2 * S2: PRINT SPC( M
G)"Code "Désignation" SPC( 13)"haut "diam" SPC( 3 - S4)"Tarif" SPC
( 7 + S4)"Code "Désignation" SPC( 12)"haut "diam" SPC( 3 - S4)"Tar
if": PRINT W0$
710 FOR N = 1 TO Z1: PRINT SPC( MG):W = N: GOSUB 60: PRINT SPC( 2)NO
$(N) SPC( 25 - LEN (NO$(N))):W = T(N,0): GOSUB 100: PRINT W$ SPC(
2):W = T(N,1): GOSUB 100: PRINT W$;W = P(N,S2): GOSUB 70: PRINT W
$ SPC( 7)
720 W% = N + Z1:W = W%: GOSUB 60: PRINT SPC( 2): IF W% < 83 OR NOT S
2 THEN PRINT NO$(W%) SPC( 24 - LEN (NO$(W%))):W = T(W%,0): GOSUB
100: PRINT W$ SPC( 2):W = T(W%,1): GOSUB 100: PRINT W$;W = P(W%,S
2): GOSUB 70: PRINT W$;
730 PRINT : NEXT : RETURN
769 REM ctJctJMENUSctJ
770 VTAB 6: PRINT "0 :Terminé": PRINT : PRINT "1 :Facture": PRINT :
PRINT "2 :Tarif": PRINT : PRINT "votre choix "<>": HTAB 15
780 POKE K%,0: GET PR$: ON PR$ < "0" OR PR$ > "2" GOTO 780: PRINT PR$:
S1 = VAL (PR$): IF NOT S1 THEN RETURN
790 PRINT : PRINT "1 :H.T.": PRINT : PRINT "2 :T.T.C.": PRINT : PRIN
T "votre choix "<>": HTAB 15
800 POKE K%,0: GET PR$:S2 = VAL (PR$) - 1: ON S2 < 0 OR S2 > 1 GOTO 8
00: PRINT PR$:B$ = " H.T.":AV$ = "HT ": IF S2 THEN B$ = " TTC ":AV
$ = "TTC"
810 TA$ = "Tarif" + " " + AV$ + " au ": HOME : RETURN
820 REM ctJctJIMPRESSION?ctJ
830 T0 = 0: VTAB 22: HTAB 12: FLASH : PRINT " IMPRESSION ";
835 PRINT G$;
840 NORMAL : PRINT " "<0/N>" CHR$ (8) CHR$ (8) CHR$ (8) CHR$ (8);
845 POKE K%,0: GET PR$: IF PR$ < "N" OR PR$ > "0" THEN PRINT G$;: GOT
0 845
850 PRINT PR$> " ": RETURN
899 REM "ctJctJIMPRESSION ENVELOPPEctJ
900 VTAB 22: HTAB 12: FLASH : PRINT G$"ENVELOPPE";: GOSUB 835: IF PR$
= "N" THEN RETURN

```

ADAC

PRO.FP,A#0300,L#00AE

PRO.FP

0300:	20 F8 BE A9 9E 8D 07 BE A9 BE 8D 08 BE A9 03 8D F6 03 A9 BE 8D F7 03 A9 08 85 68 A2 01 86 67 CA	A141
0320:	8E 00 08 8A A2 12 9D 58 BF CA D0 FA A9 3F 8D 68 BF A9 CF 8D 58 BF A9 F0 8D 10 BE A9 FD 8D 11 BE	E8CD
0340:	8D 21 BE A9 18 8D 20 BE A9 00 8D 12 BE 8D 16 BE 8D 22 BE 8D 26 BE A9 C1 8D 13 BE 8D 23 BE A9 C3	3882
0360:	8D 17 BE 8D 27 BE A9 95 20 ED FD 20 84 FE 20 2F FB 20 58 FC A9 00 8D F2 03 A9 BE 8D F3 03 49 A5	947F
0380:	8D F4 03 AD B3 FB 30 0E AD C8 FB F0 09 C9 E0 D0 05 20 1F FE 90 0D A2 01 8E AD 03 20 A4 03 E8 E0	B146
03A0:	05 90 F5 60 78 20 00 BF 41 AC 03 60 01 01	0293

CONT2,A#0000,L#01A0

CONT2

0000:	20 58 FF BA 8D FF 00 18 69 19 8D F6 03 8D 00 01 69 00 8D F7 03 A9 4C 8D F5 03 60 20 B7 00 48 20	C4D4
0020:	81 00 68 C9 8B D0 73 20 E3 DF 85 3C 84 3D 20 6C DD 20 BE DE 20 E3 DF 20 6C DD A0 02 B1 83 99 40	C763
0040:	00 88 10 F8 C8 84 3E 84 3F 84 D1 A8 F0 39 88 B1 41 C9 5C 90 18 D0 02 A9 63 C9 60 90 13 C9 78 F0	F92B
0060:	04 C9 7D D0 02 A9 65 C9 7C D0 02 A9 75 38 E9 20 99 00 02 98 D0 D8 20 BE DE 20 67 DD 20 52 E7 E6	D6DA
0080:	50 D0 19 E6 51 D0 15 20 8E FD A2 FA 8D 16 D2 09 00 20 ED FD E8 D0 F5 4C 2A D4 D0 59 20 BE DE 20	EC08
00A0:	E3 DF 85 85 84 86 20 6A DD A0 00 B1 3C C5 40 90 4C 85 41 C8 B1 3C 85 5E C8 B1 3C 85 5F A6 40 A4	F32C
00C0:	41 88 B1 5E C9 5C 90 18 D0 02 A9 63 C9 60 90 13 C9 78 F0 04 C9 7D D0 02 A9 65 C9 7C D0 02 A9 75	02E5
00E0:	38 E9 20 D0 FF 01 F0 0F 8D FF 01 C9 3D F0 08 C6 41 D0 CA F0 08 D0 27 CA F0 28 98 D0 C4 E6 3E D0	236A
0100:	02 E6 3F A5 50 C5 3E D0 06 A5 51 C5 3F F0 11 18 A5 3C 69 03 85 3C 90 91 E6 3D D0 8D D0 DF D0 7D	3A83
0120:	F0 6F A0 00 B1 3C AA B1 5E 09 00 20 ED FD C8 CA D0 F5 A9 A0 20 ED FD 20 80 FE A9 CF 20 ED FD A9	AEAB
0140:	2F 20 ED FD A9 CE 20 ED FD 20 84 FE 2C 10 C0 AD 00 C0 10 FB 8D 10 C0 48 A2 04 A9 88 20 ED FD CA	1628
0160:	D0 FA 20 9C FC 20 8E FD 68 C9 88 F0 04 C9 88 D0 0F A5 D1 F0 88 A2 03 B5 D0 95 3C CA 10 F9 D0 9A	6E33
0180:	29 DF C9 CF F0 0B A2 03 B5 3C 95 D0 CA 10 F9 D0 8B A4 3E A5 3F 20 F2 E2 A5 12 4C 63 DA 4C C9 DE	B381

FLOAD,A#6000,L#0347

FLOAD

6000:	A9 02 20 F5 BE 90 65 A0 16 B9 95 60 20 ED FD 88 10 F7 4C 00 BE 8D 87 C5 C3 C1 CC D0 A0 C5 C4 A0	783C
60A0:	D3 C1 D0 A0 C1 A0 D9 A7 CE A0 CC C9 04 15 04 2A 05 14 00 2A 16 15 04 2A 15 15 00 2A 04 15 04 2A	6276
60C0:	15 14 00 28 04 15 04 2A 15 15 00 2A 15 15 00 2A 15 15 04 2A 05 15 00 2A 05 15 04 2A	9E98
60E0:	15 14 00 28 05 15 04 2A 15 14 00 28 85 43 AE 08 BE 8E 05 62 8D 08 BE AE 07 BE 8E 04 62 A9 00 85	AF13
6100:	3C 85 04 A9 62 85 3D 85 05 A9 8B 85 3E A9 63 85 3F A9 C6 85 06 A9 63 85 07 D8 A9 00 85 42 8D 07	7551
6120:	BE 38 E5 04 85 02 A5 43 E5 05 85 03 A0 00 B1 3C C9 20 D0 17 C8 B1 3C D0 12 C8 B1 3C C9 BF D0 0B	37CC
6140:	A0 00 A2 02 20 C3 61 A9 02 D0 16 A0 00 B1 3C 48 29 03 A8 68 4A 4A AA 8D AC 60 88 30 04 4A 4A D0	9A51
6160:	F9 29 03 AA A0 00 E0 02 90 05 C8 20 9C 61 88 20 C3 61 90 88 B0 12 A9 8B 85 3E A9 63 85 3F 20 9C	CE24
6180:	61 A2 01 20 C3 61 90 F6 90 08 A5 06 85 3E A5 07 85 3F 20 2C FE 20 CE 61 60 6C FD 00 A5 06 D1 3C	EE61
61A0:	C8 A5 07 F1 3C 90 1A 88 A5 04 D1 3C C8 A5 05 F1 3C B0 0E 88 B1 3C 65 02 91 3C C8 B1 3C 65 03 91	043D
61C0:	3C 88 60 B1 3C 91 42 20 B4 FC CA 10 F6 60 60 A5 05 18 65 03 AA 18 69 02 85 43 8A 48 4A 4A A8	0ABB
61E0:	8A 29 07 AA A9 00 38 6A CA 10 FC 19 58 BF 99 58 BF 68 AA E8 E4 43 90 E2 60 00 00 00 00 00 00	EAS8
6200:	4C 1E 62 4C 9E BE 4C 58 FF D0 FB A0 00 B9 AE 63 20 ED FD C8 C0 19 90 F5 20 87 00 F0 E9 EA A2 05	788D
6220:	BD FF 01 29 DF 5D AD 63 38 D0 D8 CA D0 F2 8E 53 6E A9 04 8D 52 BE A9 81 8D 54 BE A9 84 8D 55 BE	7C1D
6240:	A9 4D 8D 50 BE AD 47 62 8D 51 BE 18 60 AD 56 BE 4A 90 0B A5 70 E5 6E E9 09 B0 07 A9 0E 2C A9 0B	1949
6260:	38 60 A9 0A 8D B4 BE A9 C4 20 70 BE 80 4B AD 88 BE C9 06 D0 42 A2 01 AD 57 BE 10 08 AD 58 BE 29	BC6D
6280:	07 F0 01 AA 8E 8C 63 A4 74 8C CF BE A9 C8 20 70 BE 80 26 AD D0 BE 8D D6 BE 8D DE BE 8D C7 BE A9	8425
62A0:	D1 20 70 BE B0 13 AD CA BE D0 8C AD C8 BE C9 C6 AD C9 BE E9 06 90 09 A9 8D 48 20 67 63 68 38 60	FC59
62C0:	AD C8 BE 8D D9 BE AD C9 BE 8D DA BE A4 6E A5 6D F0 01 C8 8C D8 BE 84 07 A9 00 8D D7 BE 85 06 85	D715
62E0:	09 A9 CA 20 70 BE 80 D1 20 67 63 AD 8C 63 89 C0 8D 80 63 8D 8B 63 8D 11 63 9D 56 63 0A 0A 0A 0A	57F4
6300:	8D 88 63 18 69 8A 8D 5E 63 A5 36 48 A5 37 48 AD 11 C1 C9 85 D0 88 A9 09 AC 8C 63 99 F8 05 A0 8D	1F4D
6320:	20 7E 63 A0 EA B9 A3 62 20 6F 63 C8 D0 F7 A0 00 B1 06 20 6F 63 E6 06 D0 04 E6 07 E6 09 A5 06 CD	C327
6340:	C8 BE A5 09 ED C9 BE 90 E5 A0 F5 B9 AE 62 20 6F 63 C8 D0 F7 AD 11 C1 C9 85 D0 85 A9 00 8D 9A C0	402E
6360:	68 85 37 68 85 36 60 8D DE BE A9 CC 4C 70 BE 88 78 84 08 40 A0 0F 20 7E 63 68 A4 08 28 60 BE 80	031D
6380:	C1 8E 8A 63 AE 80 63 A0 10 4C 00 C1 00 8D 89 32 33 32 4E 1B 5A 00 20 1B 5A 80 00 09 5A 00 00 00	A8F2
63A0:	00 00 00 1B 44 00 20 8D 00 00 00 00 00 00 C6 CC CF C1 C4 A0 F0 E1 F4 E8 AC A0 BC AC C1 A0 F3 EC	EC33
63C0:	EF F4 E9 ED F0 BE 8D	F4F4

/ FACTARIF /

COUNT,A#4000,L#0388

COUNT

4000:	1B 2D 1B 49 20 48 00 00 00 00 00 00 00 21 48 00 00 6F 6F 00 00 00 00 22 48 00 05 03 00 05 03	9AD5
4020:	00 00 23 48 14 14 7F 15 15 01 01 00 24 48 14 26 6F 7B 32 14 00 00 25 48 46 26 10 00 64 62 00 00	96CB
4040:	26 48 30 7E 49 56 20 50 48 00 27 48 00 00 05 03 00 00 00 00 28 48 00 1C 3E 77 63 00 00 00 29 48	84FF
4060:	00 63 77 3E 1C 00 00 00 2A 48 00 2A 1C 00 1C 2A 00 00 2B 48 00 10 10 7C 10 10 00 00 2C 68 00 00	E40D
4080:	00 00 70 00 00 00 2D 48 00 18 18 18 18 18 00 00 2E 48 00 00 00 60 60 00 00 00 2F 48 40 20 10 00	8532
40A0:	04 02 01 00 30 48 7E 7F 7F 41 41 3F 00 00 31 48 00 02 7F 7F 7F 00 00 00 32 48 62 71 7F 5F 46 40	9265
40C0:	00 00 33 48 22 41 49 7F 7F 36 00 00 34 48 1C 12 7F 7F 7F 10 00 00 35 48 2F 48 7B 7B 7B 33 00 00	A82D
40E0:	36 48 3E 7F 7F 49 49 32 00 00 37 48 61 79 7F 1F 07 01 00 00 38 48 36 7F 7F 49 49 36 00 00 39 48	1E7A
4100:	26 49 49 7F 7F 3E 00 00 3A 48 00 00 00 00 08 08 00 00 00 3B 68 00 00 00 00 06 76 00 00 00 3C 48 00 00	8881
4120:	1C 3E 77 63 00 00 3D 48 00 6C 6C 6C 6C 6C 00 00 3E 48 C3 E7 7E 3C 00 00 00 00 3F 48 02 01 51 5F	9F01
4140:	5F 0E 00 00 40 48 38 7C 45 3E 7C 78 00 00 41 48 7E 13 13 7F 7F 7E 00 00 42 48 7F 7F 7F 49 49 36	FA98
4160:	00 00 43 48 3E 7F 7F 41 41 22 00 00 44 48 7F 7F 7F 41 41 3E 00 00 45 48 3E 7F 7F 49 49 41 00 00	96D0
4180:	46 48 7E 7F 7F 09 09 01 00 00 47 48 3E 7F 7F 41 51 32 00 00 48 48 7F 7F 7F 00 00 7F 00 00 49 48	2AD9
41A0:	00 00 7F 7F 7F 00 00 00 4A 48 00 20 40 7F 7F 3F 00 00 4B 48 7F 7F 7F 00 00 77 00 00 4C 48 7F 7F	FAD5
41C0:	7F 40 40 40 00 00 4D 48 7E 7F 03 7E 03 7E 00 00 4E 48 7E 7F 7F 01 01 7E 00 00 4F 48 3E 7F 7F 41	ED79
41E0:	41 3E 00 00 50 48 7F 7F 7F 09 09 06 00 00 51 48 3E 7F 7F 41 41 FE 00 00 52 48 7F 7F 7F 09 09 76	1920
4200:	00 00 53 48 26 4F 5D 5D 79 32 00 00 54 48 00 01 7F 7F 7F 01 00 00 55 48 3F 7F 7F 40 40 3F 00 00	E729
4220:	56 48 07 3F 7F 78 38 07 00 00 57 48 3F 60 3F 60 7F 3F 00 00 58 48 77 7F 7F 00 00 77 00 00 59 48	5AF2
4240:	07 0F 7F 78 78 0F 00 00 5A 48 61 71 79 4F 47 43 00 00 5B 48 00 00 07 05 07 00 00 00 5C 68 1C BE	0DAE
4260:	FE 22 22 14 00 00 5D 48 00 4A 55 55 55 28 00 00 5E 48 00 02 01 02 00 00 00 00 5F 68 00 00 00 00	49DE
4280:	00 00 00 00 60 48 00 00 00 03 05 00 00 00 61 48 38 44 44 3C 7C 78 00 00 62 48 3F 7F 7F 44 44 38	6FF0
42A0:	00 00 63 48 38 7C 7C 44 44 28 00 00 64 48 38 44 44 7F 7F 3F 00 00 65 48 38 7C 7C 54 54 48 00 00	6601
42C0:	66 48 00 7E 7F 7F 09 0A 00 00 67 68 9C A2 A2 FE FE 7C 00 00 68 48 7F 7F 7F 00 00 70 00 00 69 48	49C2
42E0:	00 00 7D 7D 7D 00 00 00 6A 48 00 40 80 FD FD 7D 00 00 6B 48 7F 7F 7F 00 00 76 00 00 6C 48 00 00	EACA
4300:	7F 7F 7F 00 00 00 6D 48 78 7C 0C 78 0C 78 00 00 6E 48 78 7C 7C 04 04 78 00 00 6F 48 38 7C 7C 44	81FF
4320:	44 38 00 00 70 68 FE FE FE 22 22 1C 00 00 71 68 1C 22 22 FE FE FE 00 00 72 48 00 70 7C 7C 04 00	2517
4340:	00 00 73 48 28 4C 54 54 64 28 00 00 74 48 00 3F 7F 7F 44 24 00 00 75 48 3C 7C 7C 40 40 3C 00 00	0F71
4360:	76 48 0C 3C 7C 40 30 0C 00 00 77 48 3C 60 3C 60 7C 3C 00 00 78 48 6C 7C 7C 10 10 6C 00 00 79 68	803E
4380:	1E A0 A0 FE FE 7E 00 00 7A 48 64 74 7C 5C 4C 44 00 00 7B 48 38 7C 7E 55 54 48 00 00 7C 48 3C 7C	113C
43A0:	79 42 40 3C 00 00 7D 48 38 7C 7D 56 54 48 00 00 7E 48 00 01 00 00 01 00 00 00 04	ACEB

CARAC.INC,A#4000,L#0388

CARAC.INC

4000:	1B 2D 1B 49 20 48 FF FF FF FF FF FF FF 21 48 FF FF FF 21 21 21 FF FF 22 48 FF FF E9 E1 FF E9	54E0
4020:	E1 FF 23 48 FF 93 01 93 93 01 93 FF 24 48 FF A3 A3 29 29 8B 8B FF 25 48 FF 63 A3 C3 0F 17 1B FF	E824
4040:	26 48 FF 8B 55 6D 5B 8F 5F FF 27 48 FF FF FF E9 E1 FF FF FF 28 48 FF FF C7 83 01 39 7D FF 29 48	E974
4060:	FF FF 7D 39 01 83 C7 FF 2A 48 FF FF AB C7 01 C7 AB FF 2B 48 FF E7 E7 81 81 E7 E7 FF 2C 48 FF FF	75D3
4080:	FF 8F 3F 3F FF FF 2D 48 FF FF E7 E7 E7 E7 E7 FF 2E 48 FF FF FF 3F 3F 3F FF FF 2F 48 FF 3F 1F 8F	F1C0
40A0:	C7 E3 F1 F9 30 48 FF 83 01 7D 7D 01 83 FF 31 48 FF F7 F3 F9 FD 01 01 FF 32 48 FF 33 11 19 09 21	2565
40C0:	23 FF 33 48 FF 8B 29 29 29 01 83 FF 34 48 FF CF C7 D3 D9 01 01 FF 35 48 FF A1 21 29 29 09 99 FF	DF44
40E0:	36 48 FF 87 03 69 6D 0C 9F FF 37 48 FF F9 F9 19 09 E1 F1 FF 38 48 FF 93 01 29 29 01 93 FF 39 48	9963
4100:	FF F3 21 29 29 01 83 FF 3A 48 FF FF FF 33 33 33 FF FF 3B 48 FF FF FF B3 33 33 FF FF 3C 48 FF FF	0317
4120:	EF C7 83 11 39 7D 3D 48 FF FF 93 93 93 93 93 FF 3E 48 FF 7D 39 11 83 C7 EF FF 3F 48 FF F3 F1 59	89A8
4140:	49 E1 F3 FF 40 48 FF 8F 17 55 53 07 0F FF 41 48 FF 03 01 C9 C9 01 03 FF 42 48 FF 01 01 29 29 01	4635
4160:	93 FF 43 48 FF 83 01 39 39 11 93 FF 44 48 FF 01 01 39 39 01 83 FF 45 48 FF 01 01 29 29 29 39 FF	C078
4180:	46 48 FF 01 01 C9 C9 C9 F9 FF 47 48 FF 83 01 39 29 09 8B FF 48 48 FF 01 01 CF CF 01 01 FF 49 48	8BA4
41A0:	FF 39 39 01 01 39 39 FF 4A 48 FF 9F 1F 3F 3F 01 01 FF 4B 48 FF 01 01 C7 83 11 39 FF 4C 48 FF 01	85E0
41C0:	01 3F 3F 3F 3F FF 4D 48 FF 01 03 F7 F7 03 01 FF 4E 48 FF 01 01 E7 CF 01 01 FF 4F 48 FF 83 01 39	A926
41E0:	39 01 83 FF 50 48 FF 01 01 C9 C9 C1 E3 FF 51 48 FF 83 01 79 59 B9 43 FF 52 48 FF 01 01 C9 89 01	8C61
4200:	23 FF 53 48 FF B3 21 29 29 09 9B FF 54 48 FF F9 F9 01 01 F9 F9 FF 55 48 FF 81 01 3F 3F 01 81 FF	9F22
4220:	56 48 FF C1 81 3F 3F 81 C1 FF 57 48 FF 01 81 DF DF 81 01 FF 58 48 FF 39 11 C7 C7 11 39 FF 59 48	505E
4240:	FF F1 E1 0F 0F E1 F1 FF 5A 48 FF 39 19 09 21 31 39 FF 5B 48 FF F1 F1 F1 FF FF FF 5C 48 FF C7	4021
4260:	83 3B 8B 93 D7 FF 5D 48 FF C7 11 55 55 11 C7 FF 5E 48 FF F7 F3 F1 F1 F3 F7 FF 5F 48 7F 7F 7F 7F	C1DC

4280:	7F 7F 7F 7F 60 48 55 AA 55 AA 55 AA 55 AA 61 48 FF BF 17 57 57 07 0F FF 62 48 FF 01 01 77 77 07	C481
42A0:	8F FF 63 48 FF 0F 07 77 77 27 AF FF 64 48 FF 0F 07 77 77 01 81 FF 65 48 FF 0F 07 57 57 47 EF FF	6966
42C0:	66 48 FF EF 03 01 ED E9 FB FF 67 48 FF CF 87 B7 B7 07 0F FF 68 48 FF 01 01 F7 F7 07 0F FF 69 48	FF5C
42E0:	FF FF FF 0B 0B FF FF FF 6A 48 FF 3F 3F 7F 7F 05 05 FF 6B 48 FF 01 01 DF 0F 27 77 FF 6C 48 FF FF	8487
4300:	7D 01 01 7F FF FF 6D 48 FF 07 07 F7 07 F7 07 FF 6E 48 FF 07 07 F7 F7 07 0F FF 6F 48 FF 0F 07 77	8A42
4320:	77 07 0F FF 70 48 FF 07 07 B7 B7 07 CF FF 71 48 FF CF 07 07 B7 07 07 FF 72 48 FF 07 07 F7 F7 E7	2D89
4340:	EF FF 73 48 FF EF 47 57 57 17 BF FF 74 48 FF F7 81 01 77 37 BF FF 75 48 FF 07 07 7F 7F 07 07 FF	4C56
4360:	76 48 FF C7 87 3F 3F 07 C7 FF 77 48 FF 07 7F 07 07 7F 07 FF 78 48 FF 77 27 0F 0F 27 77 FF 79 48	A506
4380:	FF C7 87 BF BF 07 07 FF 7A 48 FF 77 37 17 47 67 77 FF 7B 48 FF 0F 07 57 53 45 EF FF 7C 48 FF 87	E401
43A0:	07 7F 7F 07 07 FF 7D 48 FF 0F 07 55 53 47 EF FF 7E 48 FF FF FF F3 FF F3 FF FF 04	FBF3

CARAC.DOUBLE,A44000,L30397

CARAC.DOUBLE

4000:	1B 2B 1B 49 62 50 00 00 FF FF FF FF 83 83 83 83 83 83 C7 FF 7E 3C 42 50 00 00 FF FF FF FF E1 E1	843A
4020:	E1 E1 E1 E1 F3 FF 7E 3C 6F 50 00 00 FC FE FF FF 07 03 03 03 03 03 07 FF FE FC 4F 50 00 00 3F 7F	955A
4040:	FF FF F0 E0 E0 E0 E0 F0 FF 7F 3F 6E 50 00 00 FF FF FF FF 1C 38 70 E0 C0 80 00 00 FF FF 4E 50	0F35
4060:	00 00 FF FF FF FF 00 00 00 00 01 03 07 0E FF FF 61 50 00 00 F8 FC FE FF 83 83 83 83 83 83 83 86	2FD0
4080:	FC F8 41 50 00 00 FF FF FF FF 03 03 03 03 03 03 03 03 FF FF 65 50 00 00 FF FF FF FF 83 83 83 83	8854
40A0:	83 83 03 03 03 03 45 50 00 00 FF FF FF FF E1 E1 E1 E1 E1 E1 E0 E0 E0 E0 6D 50 00 00 FF FF FF FF	2122
40C0:	38 70 E0 C0 E0 70 38 FF FF FF 4D 50 00 00 FF FF FF FF 00 00 00 01 00 00 00 FF FF FF 6C 50 00 00	8120
40E0:	FF FF FF FF 00 00 00 00 00 00 00 00 00 00 4C 50 00 00 FF FF FF FF E0 E0 E0 E0 E0 E0 E0 E0 E0	F854
4100:	75 50 00 00 FF FF FF FF 00 00 00 00 00 00 00 FF FF FF 55 50 00 00 3F 7F FF FF F0 E0 E0 E0 E0	8C6F
4120:	F0 FF 7F 3F 76 50 00 00 FF FF FF FF 00 00 00 00 00 00 00 FF FF FF 56 50 00 00 03 07 0F 1F 38 70	ABF2
4140:	E0 E0 60 30 10 0F 07 03 73 50 00 00 3C 7E FF FF E7 C3 C3 C3 C3 C3 87 0E 0C 53 50 00 00 10 30	AEF9
4160:	70 F1 E1 E1 E1 E1 E1 F3 7F 3F 1E 74 50 00 00 07 07 07 07 07 FF FF FF FF 07 07 07 07 07 54 50	5120
4180:	00 00 00 00 00 00 00 FF FF FF FF 00 00 00 00 72 50 00 00 FF FF FF FF 83 83 83 83 83 C3 E7 7F	2072
41A0:	3E 1C 52 50 00 00 FF FF FF FF 03 03 07 0F 1F 39 70 E0 C0 80 78 50 00 00 03 07 0F 1F 3C F8 F0 E0	A000
41C0:	F0 B8 1C 0F 07 03 58 50 00 00 F0 F8 FC FE 0F 07 03 01 03 07 0E FC F8 F0 69 47 00 00 FF FF FF FF	1E29
41E0:	00 49 47 00 00 FF FF FF FF 00 70 50 00 00 FF FF FF FF 83 83 83 83 83 C3 E7 7F 3E 1C 50 50 00 00	0CFA
4200:	FF FF FF FF 03 03 03 03 01 00 00 00 63 50 00 00 FC FE FF FF 07 03 03 03 03 07 1F 1E 1C	222D
4220:	43 50 00 00 3F 7F FF FF F0 E0 E0 E0 E0 F0 F8 78 38 64 50 00 00 FF FF FF FF 03 03 03 03 03 03	CDF9
4240:	07 FF FE FC 44 50 00 00 FF FF FF FF E0 E0 E0 E0 E0 F0 FF 7F 3F 66 50 00 00 FF FF FF FF 83 83	4735
4260:	83 83 83 83 03 03 03 03 46 50 00 00 FF FF FF FF 01 01 01 01 01 01 00 00 00 67 50 00 00 FC FE	6961
4280:	FF FF 07 03 03 03 03 07 1F 1E 1C 47 50 00 00 3F 7F FF FF F0 E0 E0 E0 E7 E7 F7 FF 7F 3F 68 50	4391
42A0:	00 00 FF FF FF FF 80 80 80 80 80 80 FF FF FF 48 50 00 00 FF FF FF FF 03 03 03 03 03 03 03 FF	E521
42C0:	FF FF 6A 4E 00 00 00 00 00 00 00 00 FF FF FF FF 4A 4E 00 00 3E 7E FE F0 E0 E0 F0 FF FF	5182
42E0:	7F 3F 68 50 00 00 FF FF FF FF 80 80 C0 E0 F0 78 3C 1E 0F 07 4B 50 00 00 FF FF FF FF 03 03 03 07	8694
4300:	0F 1E 3C F8 F0 E0 71 50 00 00 FC FE FF FF 07 03 03 03 03 07 FF FE FC 51 50 00 00 3F 7F FF FF	025D
4320:	F0 E0 E4 EC FC F8 70 FF FF BF 77 50 00 00 FF FF FF FF 00 00 80 80 80 80 80 FF FF FF 57 50 00 00	69A8
4340:	0F 1F 3F 7F E0 70 3F 1F 3F 70 E0 7F 3F 1F 79 50 00 00 03 07 0F 1F 3C 78 F0 E0 F0 F8 3C 1F 0F 07	80E4
4360:	59 50 00 00 1E 3F 7F FF F0 E0 E1 E3 F7 FF 7E 3C 18 00 7A 50 00 00 07 07 07 07 07 07 07 C7 E7 F7	DD00
4380:	7F 3F 1F 0F 5A 50 00 00 E0 F0 F8 FC FE FF F7 F3 F1 F0 F0 F0 F0 04	0BE6

Bien sûr que l'Applesoft existe encore...

- Comment prouver à nos lecteurs que nous nous intéressons toujours à l'APPLESOFT ?
 Simplement en les invitant à trouver une autre revue consacrant autant de pages au Basic Applesoft et à l'Apple II... aussi longtemps que *TREMP-LIN MICRO* existera, l'Applesoft y aura bien entendu droit de cité, mais nous continuerons, parce que c'est aussi notre rôle, d'intéresser nos lectrices et lecteurs aux autres langages et notamment au **C**, au GS Basic... et au plus naturel de tous : le langage machine !

T.M.

VOTRE FICHER CLIENTS aura cette allure...

AUDIBERTI S.A.,16 rue Nougaro, ,11111 ANTIBES,0
 BAUDELAIRE,23 rue des Fleurs, ,12345 DU MAL,10
 CALVINO,Place voir Arioste, ,33333 ITALO,20
 DIOGENE,'LAERCE',Vie Doctrines et Sentences,44444 TONNEAU,0
 ESCHYLE,8 rue des Euménides, ,55555 THEBES,10
 FLAUBERT,5 rue de la Tentation, ,99008 Saint ANTOINE,20
 GOBINEAU,15 place de la Renaissance, ,77777 ADELAIDE,30
 HACHETTE,9 bd. des Bugs,La Petite Faute-Corrigée,86100 CHATELLERAULT,0
 IONESCO,212 rue des Chaises, ,99999 EUGENE,10
 JAMES,49 av. des Deux Visages, ,10101 FRANCIS,20
 KAFKA,Le Chateau,24 Place du Procès,20202 FRANZ,30
 LUCRECE,24 Place de la Nature, ,30303 BORGIA,0
 MARGUERITE ,55 rue de Navarre, ,40404 HEPTAMERON,10
 NIETZSCHE,14 rue des Idoles, ,50505 ZARATHOUSTRA,20
 OVIDE,32 av. des Métamorphoses, ,60606 CHAMONARD,30
 PLATON,Premiers Dialogues,Eutyphron,70707 REPUBLIQUE,0
 QUENEAU,209 Rue du Métro, ,80808 ZAZIE,10
 RABELAIS,35 rue Pantagruel, ,90909 GARGANTUA,20
 SEGALIN,12 rue des Marmottes,Le Fils du Ciel,22000 SAINT BRIEUC,30

structure: NB clients
 NOM,Adr1,Adr2,Code Ville,Taux remise

**Début et fin d'un fichier
 TARIF**

45,18.6,TVA 18.68%,01/01/87
 Assiette creuse,18971,225,2,23
 Bonbonnière 1°,13491,160,8.5,14
 Bonbonnière 2°,11804,140,7,12
 Bonbonnière 3°,9612,114,5,10
 Boule GM,9612,114,13,11
 Boule PM,6408,76,9,8
 Boule PM,6408,76,9,8
 Pique fleurs GM,9612,114,11,11
 Pique fleurs PM,6408,76,8,8
 Bol à riz GM,9612,114,7.5,11
 Bol à riz PM,6408,76,6,9
 Boule à col GM,15599,185,15,11
 Boule à col PM,9612,114,12,8
 Cachepot GM S PO,54806,650,23,27
 Cachepot GM S GO,62395,740,28,27
 Cachepot GM D PO,28246,335,17,21
 Cachepot GM S GO,28246,335,15,21
 Cachepot PM S GO,28246,335,17,21
 Cachepot PM D PO,35413,420,17,21
 Cachepot PM D GO,35413,420,15,21
 Capeline GM.S.,54806,650,8,36
 Capeline GM.D.,70826,840,8,36
 Capeline PM.S.,38786,460,8,31
 Capeline PM.D.,54806,650,8,31
 Vide poches GM,7083,84,4,14
 Vide poches PM,5649,67,3.5,12.5
 Coupelle GM,4132,49,3,11
 Coupelle PM,3288,39,2.5,9.5

Urne 1°,48861,570,28,20
 Urne 1° Déc,62395,740,28,20
 Urne 2°,31197,370,23,17
 Urne 2° Déc,38786,460,23,17
 Urne 3°,18550,220,19,13
 Vase 3 anses,51433,610,26,21
 Vase 3 anses Déc,62395,740,26,21
 Corps de 3 anses,47218,560,26,21
 Coupe de Paris GM.D,31197,370,6,26
 Coupe à glace PM.S,12648,150,6,17
 Pique fleurs 84 GM,10624,126,11,11
 *,0,0,0,0
 Cachepot GM TREMLIN,50590,600,23,27
 Cachepot PM TREMLIN,25717,305,17,21
 Galet PM TREMLIN,8685,103,0,0
 Galet GM TREMLIN,9781,116,0,0
 Pot bas TREMLIN,11804,140,0,0
 Massue TREMLIN,19393,230,0,0
 Vapo bas décoré TREMLIN,8094,96,0,0
 Poudrier décoré TREMLIN,8094,96,0,0
 *,0,0,0,0
 *,0,0,0,0
 Structure 1ère ligne:
 Nb articles/2,Taux TVA format APPLE,
 'TVA XX,XX%': date du tarif JJ/MM/AA

Notez au passage que les prix
 ci-dessous sont réellement
 ceux pratiqués par

Yvan KOENIG

Mosaïque GERBINO

4, avenue du Stade
 06220 VALLAURIS

Le meilleur accueil y attend
 nos lectrices et lecteurs !

TURBO BASIC ***\$INCLUDE****Appel de sous-programme**

Voici un exemple d'utilisation de la méta-commande **\$INCLUDE** avec **Turbo Basic**.

On a d'abord écrit la routine **CADRE.TBS**. Celle-ci permet de tracer des cadres de diverses tailles.

Le programme **EXCADRE.BAS** montre comment passer par cette routine qui, grâce à **\$INCLUDE**, sera incorporée au programme principal par le compilateur.

* **Turbo Basic** fonctionne sur PC et compatibles, mais aussi sur Apple IIGS grâce au PC-Transporter.

EXCADRE.BAS

```
' EXEMPLE D'UTILISATION DE "CADRE.TBS"
' *****

cls: key off
cadre1 :
  h%=5: v%=5: haut%=20: larg%=70: cd%=1
  color 15,0
  gosub execution

cadre2 :
  h%=8: v%=8: haut%=14: larg%=50: cd%=0
  color 7,0
  gosub execution

touche:
  r%=inkey$: if r%="" then touche
  cls: end

execution:
  $INCLUDE "Cadre.tbs"
  return
```

CADRE.TBS

```
' [CADRE.TBS] trace un cadre simple ou double filet, en utilisant les
' paramètres suivants: h% et v% pour point de départ
'
'          haut% et larg% pour hauteur et largeur
'
'          cd%=0 pour cadre simple et cd%=1 pour double
'
'          color esr transmis à partir du programme principal
```

```
cdr:
  if h%+larg%>81 or v%+haut%>26 then cdrrts
  cdr1$="┌": cdr2$="=": cdr3$="┐": cdr4$="│": cdr5$="└": cdr6$="┘"
  if cd%=0 then cdr1$="┌": cdr2$="-": cdr3$="┐": cdr4$="|": _
    cdr5$="└": cdr6$="┘"

  locate v%,h%: print cdr1$; string$(larg%-2,cdr2$);cdr3$
  for i% = v%+1 to v%+haut%-2
    locate i%,h%: print cdr4$;: locate ,h%+larg%-1:print cdr4$;
  next i%
  locate v%+haut%-1,h%: print cdr5$;string$(larg%-2,cdr2$);cdr6$;

cdrrts:
  locate 1,1
```

TURBO BASIC

DEF SEG

Cette seconde démonstration utilise aussi le sous-programme **ECRAN.TBS**, mais montre également comment modifier les attributs des caractères-écran. **DEF SEG = \$HB000** (ou **B800** avec un écran couleur) fixe l'adresse de départ à **B000** (premier caractère de la première ligne). Chaque caractère occupe deux

octets : un pour l'attribut (couleur), l'autre pour le code ASCII.

ECRNCHNG.BAS

```
'L'ECRAN CHANGEANT (Simple démo)
'*****
                                UTILISE ECRAN.TBS

cls: key off
randomize timer

ecran:
  for d% = 1 to 10
    h%=1+int(rnd(1)*8)
    v%=1+int(rnd(1)*8)
    haut%=8+int(rnd(1)*(18-v%))
    larg%=20+int(rnd(1)*(61-h%))
    color 7,0
    if 1+int(rnd(1)*2)=1_
      then color 15,0
    cd%=int(rnd(1)*2)

    gosub execution
  next d%
  locate 25,30: color 0,7: print " ECHAP pour TERMINER ";
  color 7,0: locate 1,1

def seg=&hb000 :bsave "ecrndemo",0,4000
  col1%=15:col2%=7:col3%=112

change:
  swap col1%,col2%
  for i%=1 to 4000 step 2
    poke i%,col1%
  next i%
  if inkey$=chr$(27)_
    then bload "ecrndemo",0:_
    goto termine

  swap col3%,col2%
  goto change

termine:
  r$=inkey$:if r$="" then termine
  kill "ecrndemo"
  cls : end

execution:
  $INCLUDE "cadre.tbs"
  return
```

BLOAD donnera un résultat correct sur un Apple IIGS, mais pas sur un PC (par contre, le fonctionnement sera correct si l'on mémorise sur disque dur ou disque virtuel).

Avec un écran couleur:
DEF SEG= &HB800

TURBO BASIC**IMPRESSION****Encadrez vos listages**

Si vous désirez présenter agréablement les listages de vos programmes en **TURBO BASIC** (ceux que nous vous présentons ont été obtenus de cette manière), utilisez cette routine qui fonctionne sur **EPSON LQ 2500**, mais peut être facilement adaptée. Un seul impératif : ne pas dépasser 80 caractères par ligne lors de l'écriture des programmes. Une option permet de mémoriser le listage ainsi obtenu en vue d'une éventuelle réimpression.

IMPRLIST.TBS

```
IMPRLIST: Impression
encadrée d'un programme
en TURBO.BASIC
```

Le nombre de lignes par page est de 64

 60 de programme + titre et pagination

 Respecter 80 caractères maxi par LIGNE

init:

```
cls: key off: clear
dim lg$(70)
longpage%=60

width "lpt1:",85:lprint
ex$="dir/p "
```

titre:

```
locate 24,1: input "Titre à éditer (dir=? ■ néant=fin): ",titre$
if titre$="" then end
if titre$="?"then dire

locate 24,1: print space$(78);
locate 24,1: input "Mémo sous le titre suivant (néant pour rien): ",tm$

ltete$="┌"+string$((80-len(titre$))/2,"=")+" "+titre$+" "
ltete$=ltete$+string$(83-len(ltete$),"=")+"┐"
lvid$="└"+string$(82," ")+"┘"

open "i",#1,ch$+titre$
```

page:

```
incr page%:num$=str$(page%)
lfin$="┌"+string$((81-len(num$))/2,"=")+num$+" "
lfin$=lfin$+string$(83-len(lfin$),"=")+"┘"
```

lec:

```
if eof(1) then close #1: fin%=1:goto suite
incr nl%: line input #1, lg$
lg$(nl%)=lg$
if nl%<longpage% then lec
```

```

suite:
  cls
  lprint ltete$;
  lprint lvid$ :l$=""

  if nl%<longpage% then longpage%=nl%
  for i% = 1 to longpage%

    for j%=1 to len(lg$(i%))
      c$=mid$(lg$(i%),j%,1)
      if c$=chr$(9) then c$=" "
      if c$<chr$(32) then c$=""
      l$=l$+c$
    next j%

    l%=len(l$):if l%<81 then l$=l$+string$(81-l%," ")
    l$="||"+ " "+left$(l$,81)+"||"

  impr:
    lprint l$;:l$(i%)=l$: l$=""
  next i%

  lprint lvid$;
  lprint lfin$;
  if tm$="" then nonmemo

  open "a",#2, tm$
  print #2, ltete$ : print #2, lvid$
  for i% = 1 to longpage% : print #2, l$(i%) : next
  print #2, lvid$: print #2, lfin$
  close #2

  nonmemo:
    nl%=0 : lprint chr$(12): if fin% <> 1 then page
  goto init

dire:
  cls: shell "cd "
  locate 24,1: input "Chemin éventuel: ",ch$
  if ch$="" then affichage
  ex$=ex$+ch$
  if right$(ch$,1)<>"\" then ch$=ch$+"\"

  affichage:
    cls:shell ex$

  attente:
  r$=inkey$: if len(r$)=0 then attente
  goto init

```

Prodos.NDA

Prodos.NDA est l'accessoire de bureau qui vous manquait. Il vous permettra de : renommer, effacer, copier, estimer la taille, verrouiller et déverrouiller n'importe quel fichier sans quitter votre application !

Dès que vous avez sélectionné l'accessoire par le menu Pomme, une fenêtre apparaît et vous propose six boutons correspondant aux six choix possibles. Il est à noter que :

- L'option **Effacer** n'efface pas un fichier verrouillé ;
- L'option **Unlock** autorise tous les accès au fichier ;
- L'option **Lock** les interdit tous, sauf la lecture ;
- L'option **Copier** copie un fichier en le chargeant entièrement en mémoire mais gonflera un fichier à trous (sparse file) : celui-ci ne sera pas copié à l'identique car, sur la copie, les trous seront remplacés par des blocs réellement alloués (ce type de fichier est de toutes façons très rare).

Cet accessoire de bureau est écrit en assembleur et principalement en **C** (version 1.0). Saisissez les programmes avec l'éditeur du Workshop, sans oublier de taper ASM65816,CC, LINKED avant d'entrer les sources respectivement : Assembleur, **C** et Linker. Compilez le tout par :

- ASSEMBLE PRODA.SRC
- COMPILE PRODC.CC KEEP = PRODC
- ALINK PROD.MAKE
- COMPACT PRODOS.NDA
- FILETYPE PRODOS.NDA NDA

Il ne vous reste plus qu'à copier PRODOS.NDA dans le sous-directory SYSTEM/DESK.ACDS de l'application de votre choix et de la lancer. Prodos.NDA sera désormais accessible par le menu Pomme.

Notes sur le programme :

Les sous-programmes DAInit et DAAction ont une partie en assembleur dont la fonction est de récupérer les registres du microprocesseur, chose impossible en **C** .

Comme l'accessoire est appelé par un outil, le

registre de banc de données n'est pas valide. Or, on ne peut pas forcer le **C** à utiliser l'adressage long pour accéder aux données ; il faut donc fixer correctement ce banc de données, ce qui est possible par les instructions assembleur PHK et PLB. Cela ne marche que si le programme et les données se trouvent dans le même banc mémoire, ce dont on est sûr que si le programme n'est constitué que d'un seul segment. Si on laissait le Linker travailler tout seul, il ferait trois segments : un pour le programme assembleur, un pour le programme et un pour les constantes **C** . Le fichier PROD.MAKE, appelé par ALINK PROD.MAKE, permet d'y remédier en forçant le linkage dans un seul segment.

Il est désormais possible de renommer un directory ou un sous-directory à partir de l'option *Renommer*.

Pour cela, cliquez dans le bouton *Renommer* puis faites en sorte que le sous-directory à renommer apparaisse dans la fenêtre de sélection, mais ne cliquez pas dessus, cliquez dans le bouton *Annuler*. Tapez le nom du sous-directory à renommer et son nouveau nom dans les cases adéquates puis cliquez dans *OK*.

Pour renommer un directory, cliquez directement dans le bouton *Annuler* et tapez le nom du directory à renommer, et son nouveau nom en les faisant précéder du slash /.

Précisions d'Yvan KOENIG :

Les fonctions de Prodos.NDA ne sont pas admises sous le FINDER. C'est normal. Ce qui l'est moins, c'est que GS.WRITE empêche l'affichage des boutons. Il est alors nécessaire de cliquer en aveugle, mais ça fonctionne.

/ Prodos.NDA /

```
1 *
2 *      "PRODA.SRC"          faire 'ASSEMBLE PRODA.SRC'
3 *
4 *      ProDOS utilitaires en NDA
5 *      (c) 1987 S. Hadinger & Tremplin Micro
6 *
7      absaddr on
8      65816 on
9      keep  PRODA
10     case  on
11
12 IDSection START
13     dc    i4'DAOpen'
14     dc    i4'DAClose'
15     dc    i4'DAAction2'
16     dc    i4'DAInit2'
17     dc    i2'$FFFF'          ; pas de rafraichissement
18     dc    i2'$0143'         ; Activate, Update & MouseDown events
19     dc    c'  ProDOS utilitaires\H**' ; Titre de l'accessoire
20     dc    i1'13'
21     END
22
23                                     ;Sauve le registre A dans Code
24                                     ; et passe le contrôle à DAInit
25 DAInit2  START
26     phb
27     phk
28     plb
29     sta   Code
30     jsl   DAInit
31     plb
32     rtl
33     END
34
35                                     ;Sauve les registres A, X & Y
36                                     ; et passe le contrôle à DAAction
37 DAAction2 START
38     phb
39     phk
40     plb
41     sta   Code
42     stx   Param
43     sty   Param+2
44     jsl   DAAction
45     plb
46     rtl
47     END

1 *      "PROD.MAKE"
2 *      faire ALINK PROD.MAKE
3 *
4 *      Prodos utilitaires en NDA
5 *      (c) 1987 S. Hadinger & Tremplin Micro
6 KEEP PRODOS.NDA
7 LINK/ALL PRODA
```

```
8 LINK/ALL PRODC
9 LIBRARY *
```

```

1 /*
2 *      "PRODC.CC"                faire 'COMPILE PRODC.CC KEEP=PRODC'
3 *                                  avec le compilateur C 1.0
4 *
5 *      ProDOS 16 utilitaires en NDA
6 *      (c) 1987 S. Hadinger & Tremplin Micro
7 */
8 #include <types.h>                /* Interfaces pour les outils      */
9 #include <event.h>
10 #include <quickdraw.h>
11 #include <window.h>
12 #include <memory.h>
13 #include <control.h>
14 #include <dialog.h>
15 #include <misctool.h>
16 #include <prodos.h>
17 #include <stdfile.h>
18 #include <intmath.h>
19
20 typedef struct SFReply {          /* Utilisé par Standard File      */
21     int    good;
22     int    type;
23     int    auxType;
24     char   filename[16]; /* Nom du fichier                  */
25     char   fullname[129]; /* Nom avec préfixe                */
26
27     int    _toolErr;          /* Erreur du dernier appel d'outil */
28     int    WindOpen, Code, itemHit; /* WindOpen: fenetre ouverte ?    */
29     EventRecordPtr Param;     /* Evènement à traiter            */
30     ParamList  ProdWind={sizeof(ParamList),0xC0A0,
31     "\33ProDOS16 utilitaires v1.1",
32     0L,{0,0,0,0},0L,0,0,0,0,0,0,0,0,0,0,0L,0,0L,0L,0L,
33     {27,30,74,310},(GrafPortPtr) -1L,0L};
34     /* Fenêtre du NDA                */
35     GrafPortPtr  WindPtr, NdaDlog;
36     CtlRecHndl   RenCtrl, DelCtrl, CopCtrl, PlaCtrl, LocCtrl, UnlCtrl;
37     /* les 6 Boutons                */
38
39     /* Paramètres pour ProDOS et Standard File */
40     SFReply      myReply;
41     FileIORec    Pread;
42     FileRec      Pinfo, Pcreate = {0L,0xC3,0,0L,1,0,0,0,0,0L};
43     PathNameRec  Pren;
44
45     char   Error[] = "$0000"; /* Numéro de l'erreur            */
46     char   BlockNum[] = " Ko."; /* Taille du fichier              */
47     /* Rectangles d'affichage */
48     Rect   rCentr = {27,110,41,190}; /* Bouton central                */
49     Rect   rLeft  = {27,30,41,110}; /* Bouton gauche                 */
50     Rect   rLeft2 = {41,30,55,110};
51     Rect   rRight = {27,170,41,250}; /* Bouton droit                  */
52     Rect   rRight2 = {41,170,55,250};
53     Rect   rDlog  = {80,10,128,310}; /* Dialog standard               */
54     Rect   rDlog2 = {80,10,142,310};
55     Rect   rLine1 = {4,90,19,290}; /* Ligne éditable 1              */

```

/ Prodos.NDA /

```

56 Rect   rLine2   = {20,90,35,290};          /* Ligne editable 2          */
57 Rect   rAlert   = {50,35,97,285};          /* Fenetre d'alerte          */
58 Rect   rbOK     = {28,85,42,165};          /* Bouton OK (alerte)        */
59
60 static char *ErrMsg[] = {                  /* Messages d'erreurs        */
61     "ProDOS occup\216",                    "Unit\216 introuvable",
62     "Erreur d'E/S",                        "Protection d'\216criture",
63     "Unit\216 non en ligne",              "Syntaxe incorrecte",
64     "Table FCB pleine",                   "Nom introuvable",
65     "Volume introuvable",                 "Fichier introuvable",
66     "Nom en double",                      "Volume plein",
67     "Directory plein",                    "Erreur de version",
68     "Type non support\216",              "Fin-de-fichier rencontr\216e",
69     "Fichier verrouill\216",             "Fichier d\216j\210 ouvert",
70     "Type non support\216",              "Plus de m\216moire",
71     "Fichier en double",                  "Ce n'est pas un disque",
72     "N\241 de bloc trop grand",          "Changement de nom ill\216gal",
73
74     "M\216moire insuffisante !", "ID non disponible !" };
75
76 static int  ErrCode[] = {                  /* Liste des codes d'erreurs */
77     0x07, 0x10, 0x27, 0x2B, 0x2F, 0x40, 0x42, 0x44, 0x45, 0x46, 0x47,
78     0x48, 0x49, 0x4A, 0x4B, 0x4C, 0x4E, 0x50, 0x52, 0x54, 0x57, 0x58,
79     0x5A, 0x5B,          0x0201, 0x030B, 0x0000 };
80
81
82     pascal GrafPortPtr DAOpen() /* Ouvre la fenetre          */
83 {
84     GrafPortPtr   wind;
85     Rect          r;
86
87     asm{phb                      /* fixe correctement banc de données */
88         phk
89         plb}
90
91     if (WindOpen)
92         SelectWindow(WindPtr); /* Amène la fenetre au premier plan */
93     else
94     {
95         WindPtr = NewWindow(&ProdWind); /* Ouvre la nouvelle fenetre */
96         SetSysWindow(WindPtr); /* Déclare la fenetre comme appartenant
97                                 au système */
98
99                                 /* Ouvre les 6 boutons*/
100        r.v1 = 13; r.v2 = 28; r.h1 = 10; r.h2 = 90;
101        RenCtrl = NewControl(WindPtr,&r,"\10Renommer",0,0,0,0,0L,1L,0L);
102        r.h1 = 100; r.h2 = 180;
103        CopCtrl = NewControl(WindPtr,&r,"\6Copier" ,0,0,0,0,0L,2L,0L);
104        r.h1 = 190; r.h2 = 270;
105        LocCtrl = NewControl(WindPtr,&r,"\4Lock" ,0,0,0,0,0L,3L,0L);
106        r.v1 = 30; r.v2 = 45;
107        UnlCtrl = NewControl(WindPtr,&r,"\6Unlock" ,0,0,0,0,0L,4L,0L);
108        r.h1 = 100; r.h2 = 180;
109        PlaCtrl = NewControl(WindPtr,&r,"\6Taille" ,0,0,0,0,0L,5L,0L);
110        r.h1 = 10; r.h2 = 90;

```

```

111     DelCtrl = NewControl(WindPtr,&r,"\7Effacer" ,0,0,0,0,0L,6L,0L);
112     }
113     WindOpen = true;
114     wind = WindPtr; /* Copie le pointeur dans une variable LOCALE
115                    (sur la pile) */
116     asm{ plb }      /* Rétablit le banc de données */
117     return wind;  /* On renvoie la valeur par une variable LOCALE et
118                  non globale puisque le banc de données n'est plus valable */
119 } /* DAAOpen */
120
121 pascal void DAClose()      /* Ferme la fenêtre */
122 {
123     asm{phb          /* fixe le banc de données */
124         phk
125         plb}
126     if (WindOpen)
127         CloseWindow(WindPtr); /* Ferme la fenêtre */
128     WindOpen = false;
129     asm{plb}        /* Rétablit le banc de données */
130 } /* DAClose */
131
132
133 void ErreurTool()         /* Traitement des erreurs */
134 {
135     GrafPortPtr    ErrDlog; /* Dialog d'alerte */
136     int            ToolErr, index;
137     char           *msg;
138
139     ToolErr = _toolErr;    /* Copie le code d'erreur qui ne */
140     if (ToolErr) {        /* reste que jusqu'au prochain */
141         SysBeep();        /* appel d'outil */
142         Int2Hex(ToolErr, &Error[1], 4); /* Convertit l'erreur en hexa */
143         msg = Error;
144         for (index = 0; ErrCode[index] != 0; ++index)
145             /* Récupère le message d'erreur s'il existe */
146             if (ToolErr == ErrCode[index]) msg = ErrMsg[index];
147         ErrDlog = NewModalDialog(&rAlert, true, 0L);
148         SetPort(ErrDlog); /* Ouvre le Dialog et le bouton */
149         NewDItem(ErrDlog,1,&rbOK,10,"\2OK",0,0,0L);
150         MoveTo(30,10);
151         if (ToolErr < 0x100) DrawCString("Erreur ProDOS :");
152         else DrawCString("Erreur :");
153         MoveTo(30,20); DrawCString(msg); /* Affiche l'erreur */
154         ModalDialog(0L); /* Attend le Click dans le bouton */
155         CloseDialog(ErrDlog); /* Ferme le Dialog */
156     }
157 } /* ErreurTool */
158
159
160 void FileModif(k)         /* Rename, Lock, Unlock fichier */
161
162 int    k;
163 {
164     char *chaine;
165
166     /* Choisit le message */

```

```
166     if (k==3)         chaine = "\25Verrouiller fichier :";
167     else if (k==4) chaine = "\27D\216verrouiller fichier :";
168         else         chaine = "\21Effacer fichier :";
169     SFGetFile(20,80,chaine,0L,0L,&myReply); /* Affiche le sélecteur */
170     if (myReply.good) {
171         NdaDlog = NewModalDialog(&rDlog,true,0L); /* Ouvre le Dialog */
172         SetPort(NdaDlog); /* et les boutons */
173         NewDItem(NdaDlog,1,&rLeft,10,"\20K",0,0,0L);
174         NewDItem(NdaDlog,2,&rRight,10,"\7Annuler",0,0,0L);
175         if (k==3)         chaine = "verrouiller :";
176         else if (k==4) chaine = "d\216verrouiller :";
177             else         chaine = "effacer :";
178         MoveTo(10,10); /* Affiche les messages */
179         DrawCString("Voulez-vous vraiment ");
180         DrawCString(chaine);
181         MoveTo(10,20);
182         DrawString(myReply.filename);
183         DrawCString(" ?");
184         itemHit = ModalDialog(0L); /* Attend un Click dans un bouton */
185         CloseDialog(NdaDlog); /* Ferme le Dialog */
186         if (itemHit==1) { /* si bouton = "OK" */
187             Pinfo.pathname = myReply.fullname;
188             if (k==6) {DESTROY(&Pinfo); ErreurTool();}
189             /* Efface le fichier */
190             else {GET_FILE_INFO(&Pinfo);
191                 if (_toolErr) goto fin;
192                 if (k==3) Pinfo.fAccess = 1; /* Change le champ */
193                 else Pinfo.fAccess = 0xC3; /* access du fichier */
194                 SET_FILE_INFO(&Pinfo);
195                 fin: ErreurTool(); } } }
196 } /* FileModif */
197
198 void Taille() /* Taille d'un fichier */
199 { /* Sélecteur */
200     SFGetFile(20,80,"\23Taille du fichier :",0L,0L,&myReply);
201     if (myReply.good) {
202         Pinfo.pathname = myReply.fullname;
203         GET_FILE_INFO(&Pinfo); /* Lit la taille */
204         if (_toolErr) {ErreurTool(); goto fin;}
205         NdaDlog = NewModalDialog(&rDlog,true,0L); /* Ouvre le Dialog */
206         SetPort(NdaDlog);
207         NewDItem(NdaDlog,1,&rCentr,10,"\20K",0,0,0L);
208         MoveTo(10,10); DrawCString("Taille de : ");
209         DrawString(myReply.filename); /* Affiche les messages */
210         MoveTo(10,20); /* Convertit en décimal */
211         Long2Dec((Pinfo.blocksUsed + 1) >> 1, BlockNum, 5, 0);
212         DrawCString(BlockNum);
213         itemHit = ModalDialog(0L); /* Attend le Click dans OK */
214         CloseDialog(NdaDlog); /* Ferme le Dialog */
215         fin: ; }
216 } /* Taille */
217
218
219 void Rename() /* Renommer un fichier */
220 {
```

```

221 char   newName[25];           /* Nouveau nom           */
222 char   changeName[25];
223 char   *origname;
224                               /* Selecteur             */
225   SFGetFile(20,80,"\\22Renommer fichier :",0L,0L,&myReply);
226   origname = "";
227   if (myReply.good) origname = myReply.filename;
228   NdaDlog = NewModalDialog(&rDlog2,true,0L); /* Ouvre le Dialog      */
229   SetPort(NdaDlog);
230   NewDItem(NdaDlog,1,&rLeft2,10,"\\2OK",0,0,0L); /* et les boutons     */
231   NewDItem(NdaDlog,2,&rRight2,10,"\\7Annuler",0,0,0L);
232   MoveTo(9,15);           DrawCString("Renommer :");
233   MoveTo(9,30);           DrawCString("en :");
234                               /* ligne éditable       */
235   NewDItem(NdaDlog,3,&rLine2,17,origname,24,0,0L);
236   NewDItem(NdaDlog,4,&rLine1,17,origname,24,0,0L);
237   do itemHit = ModalDialog(0L);
238       while (itemHit > 2); /* Attend un Click dans un bouton */
239   GetIText(NdaDlog,4,changeName); /* Récupère le nom à changer */
240   GetIText(NdaDlog,3,newName); /* Récupère le nouveau nom */
241   CloseDialog(NdaDlog); /* Ferme le Dialog */
242   if (itemHit == 1) { /* Si OK */
243       Pren.pathname = changeName;
244       Pren.newPathname = newName;
245       CHANGE_PATH(&Pren); /* Change le nom */
246       ErreurTool(); }
247 } /* Rename */
248
249
250 void   Copier() /* Copier un fichier */
251 {
252 Ptr    MemAdr; /* Adresse du buffer */
253 long   MemSize; /* Taille du buffer */
254 int    MemoryID; /* ID de l'accessoire */
255 Handle MemHndl; /* Handle du buffer */
256                               /* Sélecteur */
257   SFGetFile(20,80,"\\20Copier fichier :",0L,0L,&myReply);
258   if (myReply.good) {
259       Pread.dataBuffer = myReply.fullname; /* open filename */
260       Pread.requestCount = 0L; /* i/o buffer */
261       OPEN(&Pread); /* Ouvre le fichier */
262       if (_toolErr) {ErreurTool(); goto fin;}
263       GET_EOF(&Pread); /* Lit sa taille */
264       if (_toolErr) goto finClose;
265       MemoryID = GetNewID(0x5000); /* Demande un nouvel ID */
266       if (_toolErr) goto finClose;
267       MemSize = (long) Pread.dataBuffer; /*Copie taille dans MemSize*/
268                               /* Ouvre le buffer */
269       MemHndl = NewHandle(MemSize, MemoryID, 0xC000, 0L);
270       if (_toolErr) goto finID;
271       Pread.requestCount = MemSize;
272       MemAdr = *MemHndl;
273       Pread.dataBuffer = MemAdr;
274       READ(&Pread); /* Lit le fichier dans le buffer */
275       if (_toolErr) goto finClMem;

```

```

276     CLOSE(&Pread);
277     Pinfo.pathname = myReply.fullname;
278     GET_FILE_INFO(&Pinfo); /* Lit les attributs du fichier */
279     if (_toolErr) goto finMem;
280         /* Sélecteur de destination */
281     SFPutFile(20,60,"\\15Destination:",myReply.filename,15,&myReply);
282     if (~myReply.good) goto finMem; /* Si 'Cancel' alors on sort */
283     Pren.pathname = myReply.fullname;
284     DESTROY(&Pren); /* Efface le fichier s'il existe déjà */
285     if ((_toolErr) && (_toolErr != 0x46)) goto finMem;
286     Pcreate.pathname = myReply.fullname;
287     Pcreate.createDate = Pinfo.createDate;
288     Pcreate.createTime = Pinfo.createTime;
289     CREATE(&Pcreate); /* Crée le nouveau fichier */
290     if (_toolErr) goto finMem;
291     Pread.dataBuffer = myReply.fullname; /* open filename */
292     Pread.requestCount = 0L; /* i/o buffer */
293     OPEN(&Pread); /* Ouvre le nouveau fichier */
294     if (_toolErr) goto finMem;
295     Pread.dataBuffer = MemAdr;
296     Pread.requestCount = MemSize;
297     WRITE(&Pread); /* Ecrit le contenu du buffer */
298     if (_toolErr) goto finClMem;
299     CLOSE(&Pread);
300     if (_toolErr) goto finMem;
301     Pinfo.pathname = myReply.fullname;
302     SET_FILE_INFO(&Pinfo); /* Ecrit les attributs de l'original */
303     goto finMem;
304         /* Traitement des erreurs */
305     finClMem: ErreurTool(); CLOSE(&Pread);
306     finMem : ErreurTool();
307         DisposeHandle(MemHndl); /* Libère le buffer */
308         DeleteID(MemoryID); /* Libère l'ID */
309     goto fin;
310     finID : ErreurTool(); DeleteID(MemoryID);
311     finClose: ErreurTool(); CLOSE(&Pread);
312     fin : ;}
313 } /* Copier */
314
315
316 void Bouton() /* Traitement du bouton de la souris */
317 {
318 int v,h,i,j;
319 CtlRecHndl myControl; /* Bouton du Click souris */
320
321 h = Param->where.h; /* Coordonnées du Click */
322 v = Param->where.v;
323 i = FindControl(&myControl,h,v,WindPtr); /* Récupère bouton Click */
324 if (i==2) { /* Si Click dans un bouton alors.. */
325 j = TrackControl(h,v,0L,myControl); /* Gère le Click */
326 if (i==j) { /* Si Click confirmé alors.. */
327 FlushEvents(-1,0); /* Vide la chaine des évènements */
328 j = GetCtlRefCon(myControl); /* Récupère numéro du bouton */
329 switch(j) { /* Aiguille suivant le numéro du bouton */
330 case 1: Rename(); break;

```

```

331         case 2: Copier(); break;
332         case 3:
333         case 4:
334         case 6: FileModif(j); break;
335         case 5: Taille(); break;
336         } } }
337 } /* Bouton */
338
339
340 pascal void DAAction()      /* Traitement des évènements      */
341 {
342 GrafPortPtr  currPort;
343 int          what, modifiers;
344
345     if (Code == 1)          /* Si Action = Event      */
346     {
347         currPort = GetPort(); /* Sauve le port courant  */
348         SetPort(WindPtr);    /* Fixe celui de la fenêtre */
349         what = Param->what;
350         if (what == updateEvt) { /* Si évènement de mise à jour */
351             BeginUpdate(WindPtr);
352             SetForeColor(0);
353             SetBackColor(15);
354             MoveTo(5,9);      /* Affiche l'en-tête      */
355             DrawString("\50(c) 1987 S. Hadinger & Tremplin Micro.");
356             DrawControls(WindPtr); /* Redessine les boutons */
357             EndUpdate(WindPtr); }
358         else if (what == activateEvt) /* Si évènement d'activation */
359             DrawControls(WindPtr); /* Redessine les boutons */
360         else if (what == mouseDownEvt) /* Si Click souris dans fenêtre */
361             Bouton();          /* Gère le Click          */
362         SetPort(currPort); } /* Rétablit le port original */
363 }
364
365
366 pascal void DAInit()      /* Initialisation de l'accessoire */
367 {
368     if ((Code == 0) && (WindOpen)) DAClose(); /* Si ShutDown */
369     WindOpen = false;
370 }

```

Si vous désirez vous initier sérieusement au **GS Basic**, offrez-vous l'excellent bouquin d'Emile Schwarz :

**À LA DÉCOUVERTE
DU GS Basic**

Voir bulletin de commande page 64.

Il est indispensable pour tirer le meilleur parti possible de la documentation américaine et du langage lui-même.

NOMBRES PREMIERS

Les nombres premiers passionnent les mathématiciens depuis leur découverte, mais ils ne laissent pas indifférents les amateurs de micro-informatique. Vous les afficherez en utilisant une routine ressemblant à celle que je vous propose ci-contre, mais ne soyez pas trop pressé si la quantité demandée est importante ! Plus il y en aura et plus notre programme se traînera ! **Nestor**

```

10 TEXT : PRINT CHR$(21): HOME                27E9
20 INVERSE                                     3D9E
30 T$ = " NOMBRES PREMIERS "                  5A69
40 HTAB (41 - LEN (T$)) / 2: PRINT T$        192A
50 NORMAL : PRINT                              3E91
60 PRINT " Un nombre est dit PREMIER s'il n
   'admet pas d'autre division que lui-meme
   et 1. ....
   .....EXEMPLES: 1,2,3,5,7,11,13,etc."      F02A
70 VTAB 21: HTAB 1: FOR I = 1 TO 40: PRINT
   "-";: NEXT                                688C
80 VTAB 22: INPUT "Combien en cherchons-nou
   s (2 et +) ? ";N$                          AEBA
90 ND = VAL (N$): IF NOT ND THEN 200         219F
100 IF ND < 2 THEN 80                        096E
110 DIM NP(ND)                               7107
120 NT = 1: REM NT=Nombre de premiers trouvé
   s                                           66A3
130 NP(1) = 2:H = - 5: VTAB 8                 7A86
140 FOR I = 2 TO N:NP(I) = 0: NEXT            C009
150 N = 3: REM Départ                          4151
160 UN = 1: REM Vaut 1 ou 0                    D6A4
170 PT = 1: REM Pointeur                       5BA5
180 IF N / NP(PT) = INT (N / NP(PT)) THEN U
   N = 0                                       C160
190 PT = PT + 1                                D411
200 IF NP(PT - 1) < (N / NP(PT - 1)) AND UN
   THEN 180                                   03CF
210 IF NOT UN THEN 250                        D871
220 NT = NT + 1:NP(NT) = N                    21F6
230 H = H + 6: IF H > 34 THEN H = 1: PRINT   A2C4
240 HTAB H: PRINT N;                          3B5B
250 N = N + 1                                  7F65
260 IF NT < ND THEN 160                       9D0D

```

* Si votre série doit dépasser 78, il sera préférable de commencer par un HOME à la ligne 130.

Indispensables pour gagner du temps sans passer par un compilateur

LES RECUEILS DE ROUTINES LM DE TREMPLIN MICRO

(voir bulletin de commande)

Bien débiter

Une série d'ouvrages d'initiation à recommander aux néophytes...

Quand on n'a jamais écrit une ligne de Basic, maîtriser le **GW/Basic**, s'y retrouver dans un fichier **Autoexec.bat...** ou s'initier au **Turbo Pascal** de Borland présentent bon nombre de difficultés. Evidemment, toutes les machines sont maintenant livrées avec une documentation convenable et généralement traduite en français, mais il est rare qu'elle réponde réellement à l'attente des néophytes.

C'est ce qu'ont parfaitement compris les responsables de cette sympathique collection. **BIEN DÉBUTER**, tout un programme !

Le spécialiste — ou l'habitué — verra dans chacun de ces ouvrages de la série un abc un peu trop simpliste, mais il en ira autrement pour une personne ignorant tout de l'informatique. Ce qui paraît enfantin au premier est parfois difficile à concevoir pour la seconde... d'où l'utilité de cette collection.

● BIEN DÉBUTER GW/PC BASIC

Le GW Basic est le plus utilisé des Basics sur PC. Dans bien des cas, il représente la première approche de la programmation sur un micro-ordinateur. En encadrant parfaitement le néophyte dans ses premières applications, **BIEN DÉBUTER GW/PC BASIC** lui permettra d'assimiler les mécanismes de la programmation, tant au niveau de la structure et de la logique des programmes, que dans le rôle et la syntaxe des mots-clés. Parfaitement écrit et appuyé d'une multitude d'exemples, il offre une introduction complète aux commandes et apporte de nombreuses réponses aux utilisateurs en les éclairant sur les possibilités d'applications à partir de leur compatible PC.

(220 pages — 99 F)

● BIEN DÉBUTER TURBO PASCAL 4.0

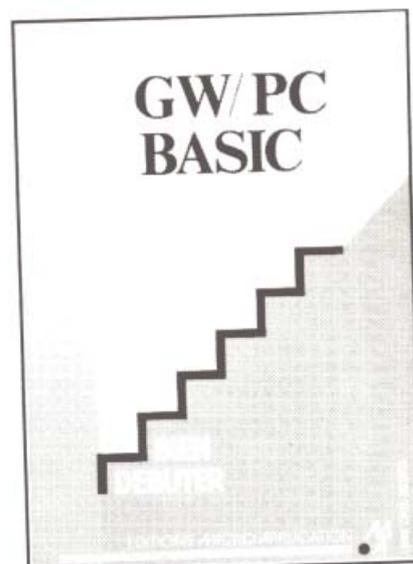
Parce que la maîtrise d'un langage évolué comme le Turbo Pascal n'est pas une affaire de quelques heures, **BIEN DÉBUTER TURBO PASCAL** se propose comme un guide souple et complet, bien adapté au niveau de connaissances des débutants en matière de langage informatique. **BIEN DÉBUTER TURBO PASCAL 4.0** traite tous les aspects du logiciel : installation, syntaxe et explication du vocabulaire, analyse et structure des programmes... Les instructions et fonctions sont présentées pas à pas et abondamment éclairées d'exemples significatifs (réalisation de petits utilitaires : banque de données, formatage d'un disque...). Pouvant s'utiliser comme manuel d'apprentissage ou guide pratique des fonctionnalités du Turbo Pascal, l'ouvrage bénéficie d'un style agréable, propre aux qualités pédagogiques de son auteur.

(240 pages — 129 F)

● BIEN DÉBUTER MULTIPLAN

Utilisé dans le monde par des milliers de personnes, le tableur Multiplan ne démonte pas de son succès. Micro Application propose un ouvrage de formation efficace permettant aux futurs utilisateurs d'aborder le plus simplement et le plus rapidement possible leur nouveau logiciel. Rédigé dans un style accessible à tous, **BIEN DÉBUTER MULTIPLAN** bénéficie du côté pratique et fonctionnel déjà mis en valeur dans les autres ouvrages de la gamme (style simple, guide des fonctions...). Grâce à de nombreux exemples choisis pour leur originalité, le lecteur pourra aborder de suite : la mise en marche du programme (installation, structure, fonctionnalités) ; le travail sur une feuille de calcul (utilisation des formules, obtention des résultats...) ; la maîtrise des fonctions mathématiques, statistiques... ; la programmation à l'aide des macro-commandes, etc. (250 pages — 129 F)

Ouvrages édités par **MICRO APPLICATION**, 13 rue Sainte Cécile, 75009 PARIS



/ BULLETIN DE COMMANDE ET D'ABONNEMENT — N°21 /

A retourner à : **TREMLIN MICRO — Senillé — 86100 CHATELLERAULT**

Nom Prénom

Adresse complète

Code postal Ville

LES DISQUETTES FONCTIONNENT SOUS DOS ET ProDOS (à condition de posséder une version Apple de ce SYSTEME D'EXPLOITATION)

Qté.	DÉSIGNATION	PRIX UNITAIRE		TOTAL
	Abonnement à 6 numéros (un an) : FRANCE à partir du numéro : _____	5, 1/4	3, 1/2	
		190 F		
	Abonnement à 6 disquettes (un an) : FRANCE à partir du numéro : _____	600 F	700 F	
	MINIE LE BUREAU DE MINIE LES SOURCES DU BUREAU] Logiciels de Claude AUBRY	180 F	200 F	
		150 F	170 F	
		50 F	70 F	
	LIVRE : APPLE II PRODOS, GUIDE DU PROGRAMMEUR APPLESOFT	120 F		
	LIVRE : LE 6502 PAS à PAS	120 F		
	LIVRE + DISQUETTE : CLINS D'OEIL AU 6502	160 F	180 F	
	LIVRE + DISQUETTE : ROUTINES LM POUR 65C02 ET 6502	160 F	180 F	
	LIVRE + DISQUETTE : NOUVELLES ROUTINES POUR LE 65C02	160 F	180 F	
	LIVRE + DISQUETTE : PROGRAMMES LM 65C02 & 65C816	160 F	180 F	
	LIVRE + DISQUETTE : A LA DÉCOUVERTE DU GS BASIC	—	300 F	
	DISQUETTE DE JEU WAR	20 F	40 F	
	DISQUETTE DE JEU "Le Monde des Sorciers"	20 F	40 F	
	DISQUETTE-INDEX (N°1 à 12)	30 F	50 F	
	DISQUETTE N°21 MS-DOS	50 F	70 F	
	SIGNATURE (CHECK-LIST)	30 F	50 F	
	RELIURE-ÉCRIN	40 F		
	DISQUETTE DE TREMLIN MICRO : * <i>(Ile, Ilc, GS)</i> <input type="checkbox"/> N°1 <input type="checkbox"/> N°4 <input type="checkbox"/> N°7 <input type="checkbox"/> N°10 <input type="checkbox"/> N°13 <input type="checkbox"/> N°16 <input type="checkbox"/> N°19 <input type="checkbox"/> N°2 <input type="checkbox"/> N°5 <input type="checkbox"/> N°8 <input type="checkbox"/> N°11 <input type="checkbox"/> N°14 <input type="checkbox"/> N°17 <input type="checkbox"/> N°20 <input type="checkbox"/> N°3 <input type="checkbox"/> N°6 <input type="checkbox"/> N°9 <input type="checkbox"/> N°12 <input type="checkbox"/> N°15 <input type="checkbox"/> N°18 <input type="checkbox"/> N°21	105 F	125 F	
	N° DE T.M. : * <input type="checkbox"/> N°1 <input type="checkbox"/> N°3 <input type="checkbox"/> N°4 <input type="checkbox"/> N°5 <input type="checkbox"/> N°6	30 F		
	COLLECTION 86 à 88 : * <input type="checkbox"/> N°7 <input type="checkbox"/> N°8 <input type="checkbox"/> N°9 <input type="checkbox"/> N°10 <input type="checkbox"/> N°11 <input type="checkbox"/> N°12 <input type="checkbox"/> N°13 <input type="checkbox"/> N°14 <input type="checkbox"/> N°15 <input type="checkbox"/> N°16 <input type="checkbox"/> N°17 <input type="checkbox"/> N°18 <input type="checkbox"/> N°19 <input type="checkbox"/> N°20 <input type="checkbox"/> N°21	33 F		

* Cochez la (ou les) case(s) de votre choix.

Merci de libeller votre règlement à l'ordre de **TREMLIN MICRO / Editions JIBENA.**

Participation aux frais d'envoi (gratuit pour les abonnés) + 10 F

Envoi en paquet-poste recommandé à partir de 400 F

Numéro d'abonné ou de client : _____ Total à payer _____ F

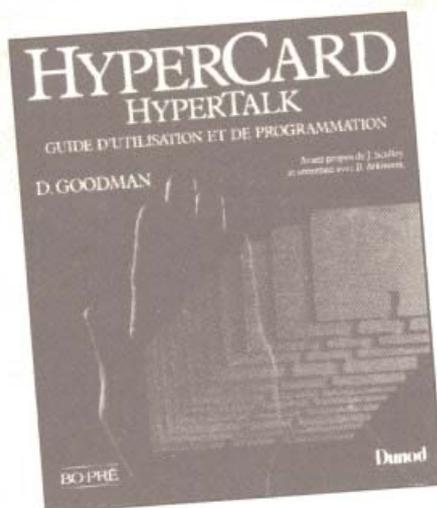
Mode de règlement choisi :

- Chèque
- Mandat-lettre
- Carte Bleue (complétez ci-contre)

N° de votre Carte Bleue

Date d'expiration Signature (obligatoire)

Montant à débiter _____ F



HyperCard inspire les auteurs

Événement majeur pour les uns, révolution pour les autres, l'HyperCard prouve au monde de l'informatique personnelle qu'Apple n'a pas fini d'innover... et d'étonner. Si le Macintosh n'avait certes pas besoin de cet extraordinaire logiciel pour démontrer sa supériorité, il est évident qu'il possède maintenant, avec HyperCard et HyperTalk, son langage associé, une bonne avance sur la grande famille des PC et compatibles.

Gageons que le nouveau produit d'Apple continuera d'inspirer les auteurs et les éditeurs. Voici déjà deux excellents manuels, bien rédigés et très complets :

- **HyperCard, HyperTalk, Guide d'utilisation et de programmation par Danny GOODMAN**

Grâce à sa parfaite connaissance du logiciel (au développement duquel il a travaillé pendant un an et demi), Danny Goodman a pu rédiger cet ouvrage remarquable, qui a eu un très grand succès aux Etats-Unis.

La traduction de ce best-seller, faite en étroite collaboration avec Apple pendant la mise au point de la version française d'HyperCard, doit être *la référence* pour tous les lecteurs francophones, car il est tout à la fois :

- un guide progressif et détaillé d'utilisation d'HyperCard ;
- une description complète de l'environnement auteur et des outils d'HyperCard ;
- un manuel clair et pédagogique du langage de programmation HyperTalk ;
- une source d'applications développées et expliquées qui mettront le lecteur sur la voie pour créer ses propres piles.

Six grands axes d'intérêt :

- Utiliser HyperCard.
- L'environnement auteur.
- L'environnement programmation.
- Utiliser HyperCard et HyperTalk.
- Importation et exportation de données.
- Références.

L'auteur : Depuis le début des années 70, Danny Goodman fait de la vulgarisation en matière d'utilisation des micro-ordinateurs. Il collabore à MacWorld et à PCWorld. Il est auteur de nombreux ouvrages sur la micro-informatique. L'un de ces livres, *Going places with the New Apple IIC*, fut un best-seller aux Etats-Unis. Il a été traduit en français, italien, allemand et hollandais.

Dunod/Bo-Pré, 18 x 21, 690 pages, broché, 290 F.

DUNOD, 17, rue Remy-Dumoncel B.P. 50, 75661 PARIS CEDEX 14

- **HyperCard : la programmation en HyperTalk par Frédéric RINALDI.**

L'ouvrage de Frédéric Rinaldi complète parfaitement celui de Danny Goodman et ne fait absolument pas double emploi avec lui. On notera qu'il est préfacé par Jean-Louis Gassée... une bonne référence !

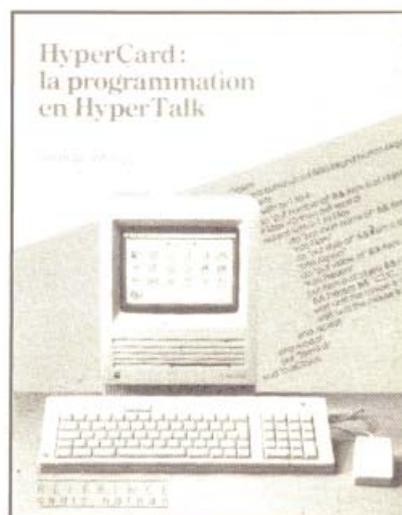
Frédéric Rinaldi commence par une description détaillée des caractéristiques du langage et s'étend notamment sur la conception de l'idée de langage-objets.

Toutes les commandes, fonctions et propriétés sont passées en revue, avec une multitude d'exemples.

L'auteur passe ensuite aux *Routines externes*, l'une des parties maîtresses de son manuel, puis à la *Maîtrise d'HyperTalk*, avec toutes les astuces et ficelles connues, permettant de devenir expert en la matière.

Une bonne méthode. De quoi vous occuper pendant de longues soirées !

Cedic/Nathan,
440 pages,
broché,
285 F TTC.



PCV. Diffusion
BP 86
77402 LAGNY/
SUR/MARNE.

Chasseur d'Images

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



**Chez votre
marchand de journaux**