

Micro
Informatique
Standards AMSTRAD

CPC

infos

JEU :

CHARLIE
BOND
SOUND

DISQUES :

EXPLOITEZ
LES FORMATS
SPECIAUX



M 2604 - 34 - 23,00 F



Mensuel . N°34
Juillet 1991

FANZ'

I N F O S

Enfin, j'ai reçu assez de fanzines pour alimenter la rubrique de ce mois-ci. Alors continuez à envoyer vos fanzines à la rédaction sur disques ou sur papier, c'est très bien et en plus cela vous fait de la pub. Attention, vérifiez bien la qualité de vos disques avant de les envoyer, car certains sont illisibles : c'est le cas de Return for Escape.

Conclusion, il ne passe pas dans ce numéro. Le mois dernier, Ali Gator Dundee n'avait pas d'adresse mais est-elle vraiment nécessaire ? Enfin, pour ceux qui ne connaîtraient pas Claude Le Moullec : 83, rue Joliot Curie 22420 Plouaret.

Dernière petite chose, j'ai reçu la préview de Génération CPC 4 et je n'ai pas eu le temps de prendre des photos.

Apparemment, il y a de l'overscan, des rasters, du multi mode et de la musique. En fait, cela ressemble plus à une démo avec beaucoup de textes qu'à un fanzine. Je suppose que ce numéro sera déjà diffusé lorsque CPC Infos sera paru.



ARKADIA

II

Le numéro deux d'Arkadia est particulièrement intéressant : tout d'abord c'est le plus important en quantité puisqu'il occupe 4 faces de disquettes et ensuite, la réalisation est très bonne. Bien sûr, il n'y a pas d'écrans en overscan, de rasters, de scrollings différentiels sur 5 plans (on se calme !) mais des images numérisées bien placées, et une débauche de textes (2 disquettes quand même). C'est clair, c'est agréable à regarder et à écouter. Donc précipitez vous sur Arkadia :

Grégory Cann
1, rue A Le Bras
29200 Brest



THE OTHER WORLD

Le premier numéro de T.O.W. est présenté sur une seule face de disque, ce qui peut sembler un peu léger par rapport aux autres productions, mais ne vaut-il pas mieux privilégier une certaine simplicité pour mieux respecter des délais de parution. Enfin, le numéro 2 comprendra 2 faces.

I.F.C.
Les Pataudes Boisseuil
87220 FEYTIAT





SOS FANZINE

Effectivement, le numéro 4 est plus complet que le numéro 3 et la copie intégrale n'est plus nécessaire pour diffuser le fanzine. En revanche, je ne sais pas si mon exemplaire est buggé ou quoi, mais il y a des problèmes d'accès à certains fichiers (Read Fail). Enfin les rubriques habituelles sont là : test de démos, interviews, pokes, etc... Pour se procurer le fanzine : 1 timbre à 3.80 + 1 timbre à 2.30 + 1 disquette (pleine de préférence) + une enveloppe self-adressée.

Au fait SOS recherche un graphiste et un musicien alors, si vous êtes intéressé, n'hésitez pas :

SOS Fanzine (Moussa)
12, rue de Corse
93600 Aulnay-sous-bois

ou

SOS Fanzine (Steve)
5, rue J. Prévert
93430 Villetaneuse



MICROSOFT

Ne cherchez plus ce titre dans la presse des fanzines, puisqu'il change dès le mois de Juin. Désormais vous trouverez Roolstrad. Cela ne changera pas le nombre de pages qui est supérieur à 20. Le numéro 7 est prévu en Septembre et il y aura même un numéro hors-série spécial Help

Carole Duguy
Bel-égo
44850 St Mars du désert

CROCO DECHAINED

Le numéro 15 du Croco est toujours en freeware. Il comprend 20 pages et beaucoup de textes. Les rubriques sont diverses et comprennent même une BD plus des pages de délire qui tournent vite aux «private jokes» pas très intéressantes pour ceux qui n'ont pas tout suivi. Pendant que j'y suis, l'édito de ce numéro me semble quelque peu inspiré d'un édito de Mad Movies par le célèbre JPP, non ?

Le Croco Déchainé
5, rue A. Tournade
17000 La Rochelle

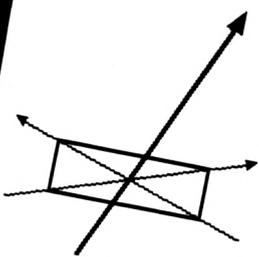
LES FANZINES C'EST BON POUR TOI ! SURTOUT CELUI LA...
5, RUE A. TOURNADE-17000 LA ROCHELLE-46 67 63 46 (SAYM)

LE CROCO DECHAINED

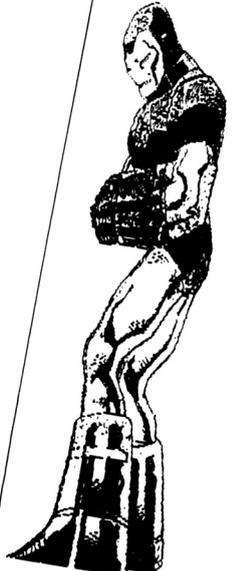
FREWARE 0,00FF

No15:FEVRIER-MARS-AVRIL

1 TIMBRE A 3,00FF



ADHERANT A L'AFIC



Liberty, they began

Liberty est un fanzine traitant de tous les micro-ordinateurs mais bien sûr plus particulièrement des CPC. L'on y trouve listings, pokes, news, nouvelles (d'amateurs), blagues, petites annonces, dossiers (connaître OCP Art Studio, etc...), bidouilles pour les vies infinies et autres. Plus du délire condensé (N'imporTnawaq). Son prix ? 13,50 F (port compris) pour plus de 25 pages à ce jour.

J'ai repris le texte de la lettre, car en fait je n'ai pas vu les 25 pages, je n'en ai reçu que deux. Pourquoi n'ais-je eu droit qu'à un échantillon ?

Liberty
70, chemin des bretoux
95320 St Leu la forêt

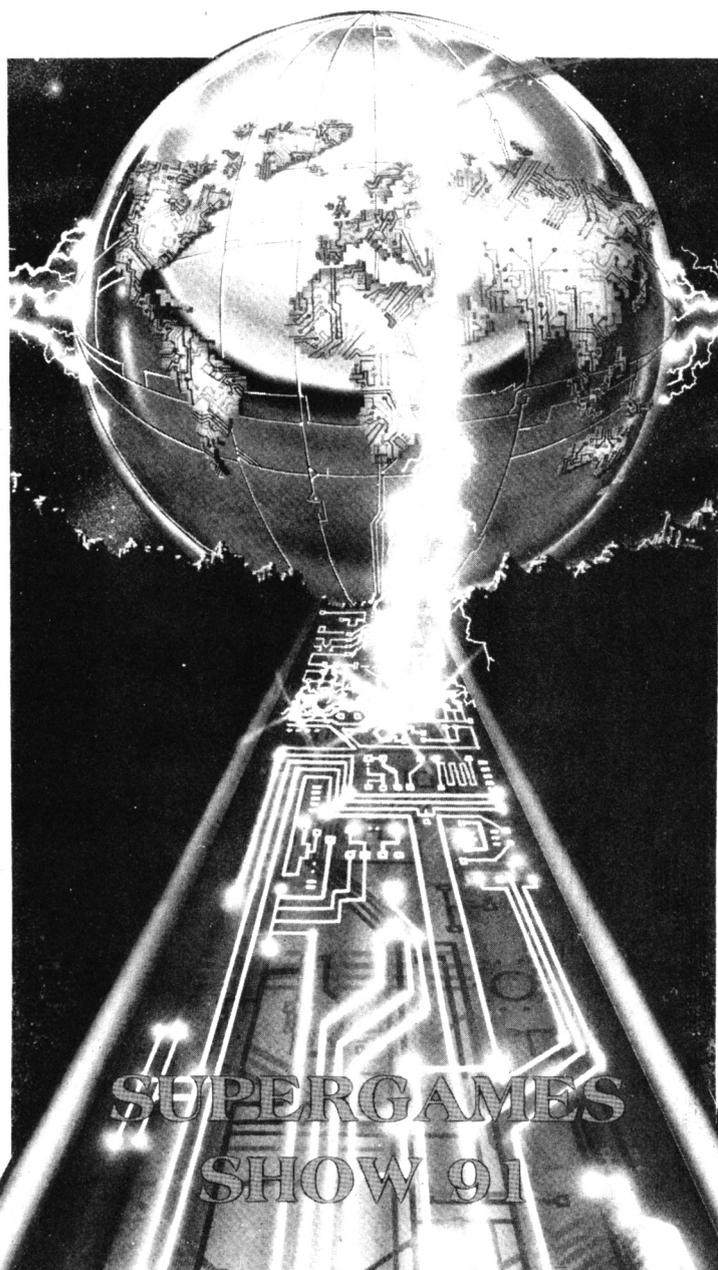
LIBERTY

Liberty est un fanzine traitant de tous les micro-ordinateurs mais bien sûr plus particulièrement des CPC. L'on y trouve listings, pokes, news, nouvelles (d'amateurs), blagues, petites annonces, dossiers (connaître OCP Art Studio, etc...), bidouilles pour les vies infinies et autres. Plus du délire condensé (N'imporTnawaq). Son prix ? 13,50 F (port compris) pour plus de 25 pages à ce jour.

J'ai repris le texte de la lettre, car en fait je n'ai pas vu les 25 pages, je n'en ai reçu que deux. Pourquoi n'ais-je eu droit qu'à un échantillon ?

Liberty
70, chemin des bretoux
95320 St Leu la forêt

ACTUALITÉ



EXPOS

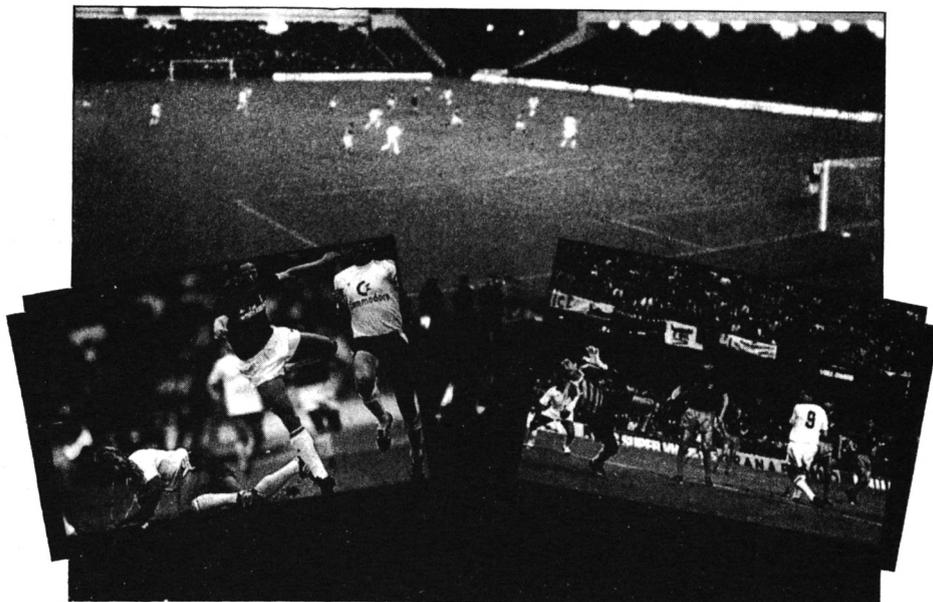
Vous ne connaissez pas le «Supergames show»? Ce n'est pas étonnant, puisqu'il s'agit de la première édition de cette exposition (organisée par notre confrère Micronews). Pour découvrir les merveilles des jeux électroniques, il suffira de se présenter le 6 Décembre (enfin peut-être pas le 6 : ce sera une journée professionnelle) porte de Champerret pour entrer dans les lieux dédiés à l'électronique ludique.

Normalement, ce sera l'occasion de rencontrer les éditeurs et tous les acteurs du secteur micro-informatique à une période de l'année où les achats des fêtes ne sont pas encore effectués.

Dans un registre moins ludique, voici le salon Micro & Co. Sa vocation est de rassembler divers passionnés de micro. Ainsi, durant les 4 journées du salon, un thème différent sera abordé : Technique et bureau, arcade, l'espace créativité et la foire aux affaires. Ce salon est organisé avec le concours de la société Pressimage et aura lieu au Parc des expositions de Paris, Porte de Versailles, les 18, 19, 20 et 21 octobre 1991.

CDS SOFTWARE

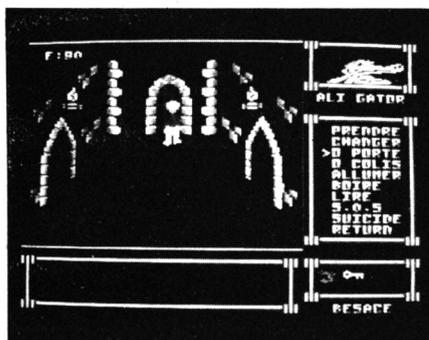
Vous qui aimez le foot, vous allez pouvoir vous défouler sur les écrans verts de vos CPC avec European Superleague. Mais attention, il n'est pas question de s'amuser à courir après une baballe. Il faut plutôt s'occuper du côté financier de la chose et se mettre dans la peau du manager d'une grande équipe européenne. Le choix des joueurs, les transferts, l'entraînement, les tactiques et toutes les opérations financières sont à votre charge, donc bon courage.





LANKHOR

Lankhor devient le spécialiste des jeux d'aventures sur CPC puisqu'après *Alive* et *Le Fugitif*, deux nouvelles histoires vont bientôt trouver place dans les douillots circuits de votre machine préférée. Le premier est intitulé : le trésor d'Ali Gator.



Les habitués des listings de CPC auront certainement reconnu ce nom et deviné alors l'auteur : il s'agit bien sûr de Claude Le Moullec. Dans un château mystérieux pétris de labyrinthes aux portes aléatoirement closes, vous devez retrouver un trésor en vous aidant de divers éléments, bouteilles, livres, clefs et autres objets.

Autre aventure bientôt disponible : *Infernal House* qui promet de vous donner des frissons dans une étrange demeure, à l'origine de nombreuses disparitions, dont celle de votre petite amie. Cela vous encouragera à mener vos recherches tambour battant parmi les diverses

TITUS

On retourne à l'ère préhistorique avec un personnage fruste et plein de poils (voir la couverture). La principale préoccupation de cette époque, et la mienne, est bien entendue la recherche de nourriture. Dans un décor luxuriant, le sympathique Grawagars, massue à la main, quête le moindre bout de grailon et bastonne accessoirement, mais d'une main décidée la faune pas câline. La version CPC de *Préhistorik* n'est pas encore prête mais elle promet beaucoup avec des graphismes colorés et une action «cartoonesque».

Puisque nous sommes dans le dessin animé, restons-y avec la bande à Picsou, célèbre personnage Disneyen. La préoccupation actuelle de notre héros est de devenir le canard de l'année. Ce défi, lancé par son pire ennemi, Flairsou est à l'origine des aventures épiques de Picsou et de ses neveux. Exploration de la jungle, safari photo et pilotage de biplan sont au programme.



pièces et surtout les passages secrets que recèlent la maison.





DOMARK

Décidemment, le foot revient en force avec Euro Football Champ de chez Taito. La compagnie japonaise a signé avec Domark pour l'adaptation du célèbre jeu d'arcade et vous devriez bientôt pouvoir vous exercer sur les pelouses électronique.

La vitesse toujours et les Joies nautiques en prime, tel est le menu alléchant de Hydra. Dans votre petit bateau, vous devez échapper à différents éléments hostiles. Pourquoi tant de haine ? Eh bien, voyez-vous, vous êtes une sorte de transporteur pour services secrets et un tas de

villains ne demandent pas mieux que de s'emparer de votre précieux chargement. Vous vous doutez donc que vos traversées seront loin d'être paisibles.



OCEAN

Autre adaptation de jeu d'arcade à succès : Narc. On se doute bien qu'avec ce titre, il ne s'agit pas de traiter de la cueillette des olives. Une fois de plus, les méfaits d'un caïd de la drogue vous forcent à sortir de votre tanière et à faire jouer vos arguments percutants : mitrailleuse lourde et muscles d'acier. Le tout possède des images digitalisées en présentation et une certaine dose d'action. Tous ces éléments seront indispensables pour venir à bout du chef de la pègre : Mr Big.

TRUCS POUR L'EXPLOITATION DE DISQUETTES A FORMATAGE SPECIAL EN LECTEUR A OU B

Valable pour CPC 464+DDI - 664 - 6128

Les possesseurs de CPC se sont vus de multiples fois proposer dans le commerce ou dans des revues des utilitaires de formatage pour augmenter la capacité des disquettes.

Le truc le plus simple est déjà d'utiliser les pistes 40 et 41. Un autre est d'avoir 10 secteurs par piste, ce qui donne pour 42 pistes (00-41) 210 KO de capacité. Le problème est que bien souvent, dès qu'il s'agit de faire tourner des programmes sur de telles disquettes, rien ne va plus. A coup sûr on obtient un «read fail», quand ce n'est pas un renvoi pour le lecteur A alors qu'on travaillait sur le B: «Drive A: disc missing». L'objet de cet article est d'aider à résoudre de tels problèmes.

AVERTISSEMENT

On aura d'autant plus de chances de résoudre tous ses problèmes, et même avant de comprendre l'ensemble de l'article (rapidement), que l'on est familier du langage d'assemblage Z80 et du système d'exploitation des CPCs. Néanmoins le maximum a été fait pour que même un néophyte, sinon un ignorant de ces choses, puisse obtenir de bons résultats. Pour cela, les programmes proposés sont là.

PROBLEME OU PAS ?

Examinons les différents cas de figures, pour déterminer lesquels permettent une exploitation sans problème d'une disquette au format spécial et lesquels nécessitent du bidouillage (logiciel). Au départ on a une disquette au format spécial dans le lecteur A ou B. On a dû lancer un petit programme BASIC ou choisir une option dans un quelconque utilitaire, qui permet au CPC de reconnaître le nouveau format. Jusqu'ici tout va bien. Ensuite on lance :

- un programme BASIC : si ce programme ne réinitialise pas le système de disque du CPC, alors il ne doit y avoir aucune difficulté à exploiter la disquette comme une disquette normale ;
- un programme en langage machine : si on a chargé ce programme par un LOAD et qu'on l'exécute ensuite (évidemment) par un CALL et que ce programme n'a pas à accéder ultérieurement à la disquette, alors pas de problème non plus. De même si on a lancé ce programme par un RUN et qu'il n'effectue pas ensuite d'accès au disque. Par contre s'il doit effectuer des opérations sur disque (chargement, sauvegarde), alors on est assuré d'avoir des problèmes, car il doit auparavant réinitialiser le système disque.

Le lecteur attentif aura deviné la source de tous les maux : la réinitialisation du système disque. Celle-ci ne s'opère qu'en langage machine. Elle est facultative quand on travaille sous BASIC et indispensable quand on a lancé un programme en Langage Machine (qu'on désignera désormais par LM) avec RUN et qu'il doit accéder au disque.

QUAND ET POURQUOI REINITIALISER LE SYSTEME DISQUE ?

Quand on lance un programme en LM par RUN, le système d'exploitation du CPC fait un appel à une routine qui remet en quelque sorte l'ordinateur à zéro, si bien qu'il en «oublie» même qu'il pouvait y avoir un lecteur de disquette connecté. Pour qu'un lecteur puisse être à nouveau employé, il faut réinitialiser le système disque, c'est-à-dire en faire effectuer la reconnaissance de présence par le CPC. On parlera désormais pour système disque de DOS (de l'anglais Disk Operating System = «Système Opérateur de Disque», d'où le nom du DOS du CPC : AMSDOS pour AMStrad DOS), histoire d'utiliser un terme consacré.

Un programme en LM chargé par un LOAD et exécuté par un CALL peut aussi nécessiter de réinstaller le DOS, car certains programmes procèdent à une remise à zéro du CPC. Le résultat est le même qu'avec un RUN. Un programme en BASIC peut avoir besoin de loger ailleurs la partie de la mémoire vive réservée au DOS (un possesseur de CPC 464 avec lecteur un peu curieux aura remarqué que le HIMEM passe de 43903 sans lecteur à 42619 avec. Les 1284 octets manquants sont presque entièrement utilisés par le DOS). Dans ce cas il y a encore réinitialisation du DOS.

LES ROUTINES DU CPC EN CAUSE

On va ici examiner les routines du système d'exploitation du CPC, qui nous concernent, à savoir la remise à zéro du CPC et la réinitialisation du DOS. La terminologie employée est celle de la BIBLE du CPC, qui est celle standard. Ces routines ont appelées via des vecteurs de saut, à adresse fixe pour tous les CPCs, ce qui permet la compatibilité entre 464, 664 et 6128. C'est à l'adresse de ces vecteurs qu'il est fait référence dans la suite.

MC BOOT PROGRAM (&BD13) : remet à zéro le CPC. Cette routine est employée lors du lancement d'un



programme. Pour un programme en BASIC, la main est transmise à l'interpréteur BASIC et le DOS n'est pas touché (sinon, même pour les formatages normaux, il faudrait réinitialiser le DOS après chaque chargement). Par contre pour un programme en LM, la main est transmise à la routine suivante, avec les conséquences que l'on va voir.

MC START PROGRAM (&BD16) : réinitialise le CPC et déconnecte toute périphérie (Attention : dans la terminologie propre au CPC, une initialisation est plus poussée qu'une remise à zéro. Celle-ci laisse certains paramètres Intacts tandis que celle-là ramène comme à l'allumage de l'ordinateur). Il y a finalement saut à une adresse. Le fait important ici est que, la périphérie étant déconnectée et le lecteur de disquette en faisant partie, pour accéder de nouveau au disque, il faudra auparavant réinitialiser le DOS. Le fait gênant est que tous les vecteurs originaux sont restaurés (cf JUMP RESTORE).

KL INIT BACK (&BCCE) : initialise une ROM (de l'anglais Read Only Memory = Mémoire Seulement Lisible, soit en français Mémoire Morte, que l'on peut trouver encore parfois dans la littérature sous l'abréviation MEM, opposée à MEV pour Mémoire Vive) de second plan. A l'origine le CPC n'a qu'une ROM de premier plan, la ROM BASIC. Les autres ROMs rajoutées par des extensions sont (en général) des ROMs de second plan, comme celle du lecteur de disquette. Les ROMs de second plan font partie de la périphérie. C'est à cette routine qu'il est le plus couramment fait appel pour réinitialiser le DOS.

KL ROM WALK (&BCCD) : trouve et initialise toutes les ROMs de second plan. A la différence de la précédente routine, celle-ci n'a pas besoin qu'on lui indique un numéro de ROM. Elle commence cependant sa recherche par la ROM du lecteur de disque. Comme très souvent elle est la seule ROM de second plan, le résultat est le même qu'avec un INIT BACK.

JUMP RESTORE (&BD37) : initialise tous les vecteurs de sauts (dont font partie les 4 précédents), ce qui signifie que ce JUMP RESTORE restaure les valeurs originales des vecteurs de saut. Donc si on veut en détourner quelques-uns, et que l'on veut être à coup sûr prémuni contre ce JUMP RESTORE, il faut veiller à le détourner lui-même.

SYNTAXE DES ROUTINES

Il s'agit ici de la description d'utilisation des routines évoquées, quand c'est nécessaire, en langage d'assemblage Z80. A ceux qui n'y connaissent rien, cela restera abstrus, quant aux autres, cela fera une petite révision. Les nombres sont indiqués directement en hexadécimal, la seule base, avec le binaire, qui se respecte en LM.

MC BOOT PROGRAM : je ne l'ai jamais vu employé par quelqu'un d'autre que le système d'exploitation ou l'interpréteur BASIC. Sa description est sans intérêt dans le cadre de cet article.

```
MC START PROGRAM: LD C,n
                  LD HL,dd
                  CALL BD16 (ou JP BD16)
```

- n est un code de configuration mémoire, qui détermine quelles RAMs ou ROMs sont sélectionnées. Presque toujours on utilise FF, ce qui fait que la totalité de la mémoire vive (que l'on désignera dorénavant par le terme anglais RAM - Random Access Memory = Mémoire à Accès Aléatoire ou Read And Modify Memory = Mémoire Lisible et Modifiable) est adressable de 0000 à FFFF.

- dd est l'adresse de saut après l'exécution de la routine.

```
KL INIT BACK: LD HL,dd1
              LD DE,dd2
              LD C,n
              CALL BCCE
```

- dd1 est le haut de la RAM disponible dont la ROM à installer peut disposer. Inversement dd2 est le bas de la RAM disponible.

- n est le numéro de la ROM. Pour le lecteur de disquette c'est 07.

Au retour de la routine, HL et DE contiennent les nouvelles valeurs pour la RAM disponible. En ce qui concerne le système disque, seuls les 1284 octets supérieurs sont utilisés, si bien que seul dd1 est modifié. En général, on indique 0040 pour dd2. On reviendra au chapitre suivant en détail sur les conséquences de l'appel de cette routine ou de la suivante.

```
KL ROM WALK: LD HL,dd1
             LD DE,dd2
             CALL BCCB
```

- dd1 et dd2 ont la même signification que pour KL INIT BACK et sont semblablement modifiés. Cette routine se contente en fait d'appeler la précédente, en ajoutant évidemment le numéro de la ROM. Il n'y a donc pas de surprise sur son fonctionnement dès qu'on a compris celui INIT BACK.

```
JUMP RESTORE : CALL BD37
```

- On ne peut pas rêver plus simple !

LA RAM AFFECTEE AU DOS

Le DOS a besoin d'un peu de place en RAM pour travailler. On ne va pas décrire ici l'ensemble de cette RAM dédiée au DOS mais seulement les parties nécessaires aux bidouillages pour exploiter des disquettes en formats spéciaux.

- Le DOS s'est vu attribuer par les concepteurs du logiciel une zone fixe en mémoire, qui est aussi la même sous CP/M (en fait l'AMSDOS a été conçu pour une compatibilité maximale avec le CP/M). Cette zone va de BE40 à BE7F. L'information qui nous intéresse ici est située en BE7D. Là est stockée l'adresse (donc sur 2 octets) de départ de la RAM dédiée au DOS (les 1284 octets dont on a déjà parlé). Pour abrégé par la suite, on parlera de «RAMDOS», qu'on se le dise ! Pour résumer, en BE7D se trouve un pointeur sur le départ de la RAMDOS.



- La RAMDOS débute par 3 octets extrêmement intéressants, qui sont dans l'ordre :

- 1) le lecteur appelé (0 pour A et 1 pour B)
- 2) le user appelé
- 3) le lecteur actif

Lorsqu'une tentative de mise sur un lecteur a réussi, le lecteur actif est le même que l'appelé. Quand tout marche, c'est le cas et on fera donc comme si c'était le cas. Si on veut continuer à travailler avec un formatage spécial avec les lecteurs et users de son choix malgré une réinitialisation de DOS, il faudra veiller à rétablir ces 3 octets dans leur état avant l'appel de ROM WALK ou INIT BACK. En effet, en cas de réinitialisation normale du DOS, on se retrouve d'une part d'office en lecteur A et sous USER 0, et d'autre part, le DOS ne reconnaît plus que les formats habituels. Cette dernière constatation permet de faire la transition avec la suite.

- Pour l'exploitation d'un formatage quelconque de disquette, l'AMSDOS ne requiert que 25 octets seulement. Ces 25 octets, qui correspondent à 19 paramètres et que nous allons bientôt détailler, indiquent la capacité disc, la taille des blocs, celle du directory, le nombre d'entrées dans celui-ci, les numéros de secteurs, etc... Ils constituent le Bloc de Paramètres Disc, Disk Parameter Block en anglais, que l'on désignera par DPB. Comme l'AMSDOS peut travailler avec 2 lecteurs, il y a 2 DPBs, le DPB A et le DPB B. L'AMSDOS accède aux DPBs par l'intermédiaire de 2 pointeurs, qui indiquent donc les adresses des DPBs. Si la RAMDOS est située à l'adresse dd, alors on trouve le :

- pointeur de DPB A en dd+021A
- pointeur de DPB B en dd+022A

Lors d'une initialisation de DOS, les 2 DPBs sont remis aux paramètres standards du format System (recopiés depuis la ROM DISC). Pour exploiter une disquette en format spécial, il faut donc installer le DPB propre à ce format.

On a déjà tous les éléments en main pour imaginer une méthode d'exploitation de disquette en format spécial, malgré une réinitialisation de DOS. Rappelons d'abord les zones vitales de la RAM :

BE7D : pointeur sur le départ de RAMDOS ou son adresse, ad.

ad+0000 : lecteur appelé

ad+0001 : user appelé

ad+0002 : lecteur actif

ad+021A : pointeur sur le DPB de lecteur A

ad+022A : pointeur sur le DPB de lecteur B

La méthode choisie et qui sera mise en œuvre de diverses manières dans les programmes de démonstration, surtout selon les capacités de programmation de chacun, sera la suivante :

- 1) déterminer l'adresse de la RAMDOS
- 2) stocker le lecteur et le user courants
- 3) en fonction du lecteur actif, déterminer l'adresse de son DPB
- 4) stocker le DPB du lecteur actif
- 5) réinitialiser le DOS
- 6) déterminer la nouvelle adresse de la RAMDOS
- 7) restaurer à cette adresse les lecteur et user actifs
- 8) en fonction du lecteur actif, trouver le pointeur de son DPB
- 9) remplacer ce pointeur par l'adresse de stockage du DPB original

Il est plus rapide de modifier simplement un pointeur de DPB que de reloger un DPB. Cela réduit le nombre d'opérations. Autant s'épargner du travail inutile.

On a maintenant les connaissances nécessaires et suffisantes à la réalisation de notre entreprise, ainsi que la méthode à employer. Avant de se lancer dans la description plus ou moins approfondie des différents programmes qui mettent tout ceci en application, on va s'offrir une petite récréation avec la description détaillée d'un DPB, ce qui ouvrira peut-être à certains de nouveaux horizons (pour concevoir des formatages personnalisés par exemple).

DESCRIPTION D'UN DPB

On a déjà évoqué la compatibilité maximale entre l'AMSDOS et le CP/M. C'est ainsi que les DPBs sont directement issus du monde du CP/M. Les DPBs de l'AMSDOS ne sont rien d'autre que des DPBs CP/M, pour les 15 premiers octets, qui correspondent à 9 paramètres. On peut donc rechercher leur description dans un livre dédié au CP/M ; c'est valable pour le CPC comme pour n'importe quel ordinateur dont le système d'exploitation est le CP/M. Leur fonction est purement logique et décrit l'organisation des données sur la disquette. Complémentairement, les 10 octets suivants, spécifiques des CPCs, décrivent la structure physique de la disquette ; ils font partie des paramètres FDC (Floppy Disc Controller = Contrôleur de Disque Souple, c'est-à-dire disquette).

Voyons d'abord un tableau de ce DPB, avec la terminologie en vigueur :

adresse	NOM	taille	description très sommaire
0000	SPT	0002	nb de secteurs logiques par piste
0002	BSH	0001	décalage de bloc
0003	BLM	0001	masque de bloc
0004	EXM	0001	masque d'incrément
0005	DSM	0002	nb maxi de blocs
0007	DRM	0002	nb maxi d'entrées au catalogue
0009	ALO, AL1	0002	taille du catalogue en binaire
000B	CKS	0002	nb d'entrées catalogue à vérifier
000D	OFF	0002	décalage de piste
000F	FSC	0001	numéro physique du 1er secteur
0010	PST	0001	nb de secteurs par pistes
0011	GLS	0001	GAP 3 pour lecture/écriture
0012	GLT	0001	GAP 3 pour formatage
0013	FLB	0001	octet de remplissage pour formatage
0014	BPS	0001	taille de secteur codée
0015	RPS	0001	nb d'enregistrements par secteur
0016	BCT	0001	buffer piste actuelle (DOS)
0017	FT0	0001	indicateur d'alignement (DOS)
0018	FLG	0001	indicateur de sélection de format

Voyons maintenant plus en détail la signification de ces valeurs :

- SPT (Sectors Per Track = secteurs par piste) : c'est le nombre de secteurs logiques de 128 octets contenus sur une piste. Pourquoi 128 octets ? Pour des raisons historiques ! A l'origine le CP/M travaillait avec des secteurs physiques de 128 octets. Les modèles ultérieurs de lecteurs de disquette permettant d'utiliser des secteurs d'une autre taille, on a cependant con-



servé ce découpage virtuel en secteurs de 128 octets (qu'on appelle encore parfois enregistrements) d'une piste pour des raisons compréhensibles de compatibilité. Ceci explique la dénomination «secteur logique», par opposition à physique. Ainsi en format Data, on a 9 secteurs de 512 octets par piste, ce qui donne un SPT de $36 = 9 \cdot 512 / 128$. Après cette description des valeurs, on en donnera des tableaux pour les formats standards et d'autres plus exotiques. Les curieux de la disquette, qui se sont un jour intéressés à la structure du catalogue (ou directory en anglais, ou répertoire dans une traduction fidèle), ont pu remarquer en 16e position d'une entrée une valeur plafonnant à &80. C'est la taille du fichier de l'entrée en blocs de 128 octets. En standard une entrée correspond à 16 blocs de 1 KO au maximum. D'où on retrouve bien : $16 \cdot 1024 / 128 = 128 = \&80$. Ceci est très important, car même en travaillant avec des blocs de 4 KO, on peut connaître la taille d'un fichier à moins de 128 octets près. Mathématiquement on a : $SPT = PST \cdot 2^{\wedge} RP S$

- BSH (Block Shift = décalage de bloc) : c'est (attention il faut être à première vue matheux) le logarithme en base 2 du nombre de secteurs logiques, donc de 128 octets, par bloc. Ceci sera beaucoup plus clair dans quelques instants. Le bloc, dont il a été question pour le SPT, est la taille minimale occupée par un fichier sur la disquette. Sous AMSDOS c'est 1 KO. Même pour sauver un seul octet, 2 secteurs de 512 octets sont employés. Soit donc TB la taille en octets d'un bloc. On a alors : $TB = 128 \cdot 2^{\wedge} BSH$. En format standard on a : $1024 = 128 \cdot 2^{\wedge} 3$, donc le BSH vaut 3. Mathématiquement on a : $BSH = \text{LOG}(TB/128) / \text{LOG}(2)$

- BLM (Block Mask = masque de bloc) : c'est le nombre de secteurs logiques par bloc, moins 1. En standard, on a des blocs de 1 KO, soit 8 secteurs logiques, d'où le BLM est $7 = 8 - 1$. Mathématiquement on a : $BLM = 2^{\wedge} BSH - 1$

- EXM (Extend Mask = masque supplémentaire) : c'est une extension au masque fonction de la taille d'un bloc et du nombre de blocs sur la disquette. C'est égal au nombre maximum d'incrémentations sur un FCB moins un. Explicitons : un FCB est un File Control Block, c'est-à-dire un Bloc de Contrôle de Fichier. Il s'agit d'une entrée au catalogue, déjà évoquée pour le SPT. Une entrée, ou donc un FCB, est composée de 32 octets, les 16 premiers pour le user, le nom, un numéro d'extension pour les fichiers nécessitant plus d'une entrée, et la taille en blocs de 128 octets. Les 16 restant sont les numéros des blocs du disque occupés par le fichier. L'incrément est la taille maximale d'un fichier tenant sur un FCB, dans le cas le plus défavorable de formatage. Or le CP/M, ou l'AMSDOS, peut travailler avec des secteurs de 128 octets. Le 16e octet, qui donne la taille du fichier, nous limite donc à 16 KO, 32 KO n'étant pas exprimable sur un octet : $32 \cdot 1024 / 128 = 256 = \&100$. L'incrément vaut donc 16 KO (N.B. : ceci n'est vrai en fait que pour les «petits» disques, pour lesquels on ne peut gérer que moins de 256 blocs. Au-delà on parle de «grands» disques et l'incrément est de 32 KO). En standard on a 16 KO par entrée au maximum, soit 1 incrément, donc le masque d'extension vaut $0 = 1 - 1$. Mathématiquement on a : $EXM = \text{nb max incréments par FCB} - 1$.

- DSM (Disk Storage Mask = masque de stockage du disque) : C'est le numéro du dernier bloc du disque,

ou, la numérotation commençant à 0, la capacité de la disquette moins 1. En format SYSTEM, cette valeur est $170 = \&AA$. Le bloc 0 est le premier bloc réservé au répertoire. Tous les blocs potentiellement en amont de ce bloc sont ignorés, comme pour les 2 pistes dites «système», réservées au CP/M. Voir à ce propos la description de OFF. Donc DSM est le numéro du dernier bloc ou la capacité totale de la disquette, directory y compris, moins 1.

- DRM (Disk Repertory Mask = masque de répertoire du disque) : c'est le nombre d'entrées au catalogue, moins 1, ou le numéro de la dernière entrée, en débutant à 0. On a déjà vu qu'une entrée est longue de 32 octets. Dans les formats standards on a 2 KO de directory, d'où un nombre d'entrées de $64 = 2 \cdot 1024 / 32$. On obtient donc 63.

- AL0, AL1 (ALlocation 0, ALlocation 1) : ces deux valeurs expriment les blocs occupés par le catalogue. Il faut exprimer AL0 et AL1, dans cet ordre, en binaire sur 8 bits : 1 bit correspond alors à 1 bloc et la lecture se fait naturellement de gauche à droite ; le bit 7, soit de plus fort poids ou MSB en anglais (Most Significant Bit = Bit le Plus Significatif) de AL0 correspond au premier bloc et le bit 0, soit de plus faible poids ou LSB en anglais (Less Significant Bit = Bit le Moins Significatif), de AL1 au dernier bloc. Un bit mis (à 1) signifie que le bloc correspondant appartient au directory. En format standard on a 2 blocs de réservés pour le directory. Ecrivons le en binaire sur 8 bits :

1100 0000, 0000 0000 soit en hexadécimal C0,00.

AL0 , AL1

La taille TC du catalogue, exprimée évidemment en blocs, se déduit du nombre d'entrées et de la taille d'un bloc. En standard on a 64 entrées et des blocs de 1 KO, d'où $32 \cdot 64 / 1024 = 2$ KO de répertoire. Mathématiquement on a : $TC = (DRM + 1) / 4 / 2^{\wedge} BSH$

- CKS (Checked Sectors = secteurs vérifiés) : c'est le nombre de secteurs logiques du répertoire à vérifier avant toute opération d'écriture. Ainsi avant d'écrire un nouveau fichier, il faut vérifier qu'il reste assez de place sur le disque et qu'il n'existe pas de fichier portant déjà le même nom, auquel cas il faut se livrer à des opérations supplémentaires. En standard on a 64 entrées (de 32 octets), donc CKS vaut $16 = 64 \cdot 32 / 128$. Mathématiquement on a : $CKS = (DRM + 1) / 4$

- OFF (OFFset = décalage) : il s'agit simplement du nombre de pistes réservées au système, ou bien à ignorer avant que ne débute la partie de la disquette exploitée par le DOS pour le stockage de données. En format System OFF vaut 2, les 2 premières pistes étant employées par le CP/M. On peut aussi utiliser ce paramètre pour diviser une disquette «physique» en plusieurs disquettes «logiques». Par exemple il existe des lecteurs 80 pistes, qui avec un formatage de 10 secteurs de 512 octets offrent par face physique 400 KO de capacité de stockage. Pour exploiter un tel volume on recourt à des blocs de 2 KO. Or l'AMSDOS ne gère pas correctement de tels blocs (alors que le CP/M sait le faire). Il en résulte que beaucoup de chargements aboutissent à des plantages. Il faut donc revenir à des blocs de 1 KO. Seulement le DOS ne peut travailler qu'avec au plus 255 blocs. L'astuce est de diviser une face physique de 400 KO en 2 faces logiques



de 200 KO. Pour la 1ère le décalage OFF vaut 00 tandis que pour la seconde il vaut 40. On commute aisément d'une face à l'autre en mettant OFF à la valeur convenable, ce par un simple POKE sous BASIC. Ce décalage est également très utile pour une disquette à format spécial, totalement incompatible avec les formats standards. Si toutes les pistes sont dans ce format, il faut lancer un utilitaire d'implantation du DBP depuis une autre disquette, ce qui est contraignant. L'astuce est de formater la première piste en Data et les autres comme on le veut. Sur les 2 KO disponibles sur cette première piste, on a largement la place pour installer un utilitaire d'implantation de DPB. Le décalage pour le reste des pistes en format spécial est de 1. Il en sera donné un exemple dans les tableaux de DPBs à la fin de ce chapitre.

Jusqu'ici les paramètres relevaient du monde du CP/M et les explications étaient valables pour toute machine CP/M. Désormais on a affaire à des paramètres du FDC, spécifiques au CPC.

- FSC (First Sector = premier secteur) : c'est le numéro physique du premier secteur de chaque piste. On use du qualificatif «physique» car il s'agit du numéro effectivement écrit sur la disquette pour identifier le premier secteur. En System c'est &41 et en Data &C1.
- PST (Physical Sectors per Track = secteurs physiques par piste) : c'est le nombre de secteurs physiques par piste, soit 9 en System ou Data.
- GLS (Gap Length for Sector = longueur de Gap pour secteur) : c'est la longueur du Gap pour les opérations de lecture et écriture. Qu'est-ce qu'un Gap sur une disquette ? C'est tout simplement un blanc (gap veut dire brèche en anglais). Il ne faut pas oublier que les lecteurs de disquettes sont avant toutes choses des machines mécaniques, donc soumises à des tolérances. Ainsi, à partir d'une certaine décimale, il n'y a pas deux lecteurs de disquette qui aient la même vitesse de rotation de disque. Un même lecteur peut voir sa vitesse varier en fonction des conditions ambiantes, sinon tout simplement de l'âge. Les Gaps sont là pour que ces variations, quoi qu'il en soit relativement infimes en cas de bon fonctionnement, ne créent aucun problème. Ils établissent des zones de sécurité, espèces de «no man's lands» entre les zones de données, en séparant les secteurs et les pistes. Le GLS est le Gap pris en compte en lecture ou écriture de secteur. Il est aussi appelé GAP 3 dans la littérature spécialisée. Il est situé après les secteurs. Pour les formats standards il vaut &2A.
- GLT (Gap Length for Track = longueur de Gap pour piste) : c'est la longueur du Gap employée lors du formatage d'une piste. Ce Gap est le même physiquement que le précédent. Seulement cette valeur n'est utilisée que lors du formatage d'une piste. Normalement le GLS doit être plus petit que ce GLT, afin que les nouveaux Gaps 3 dus aux opérations d'écriture (de longueur GLS), ne «débordent» pas au delà du Gap 3 résultant du formatage. On a d'autant moins de risques d'avoir des problèmes de recouvrement de données par un Gap 3, que la différence entre GLT et GLS est grande. Pour les formats standards le GLT vaut &52. Cela représente presque le double de GLS ; la marge de sécurité est excellente.

- FLB (FILL Byte = octet de remplissage) : c'est l'octet utilisé pour remplir les secteurs lors d'un formatage de piste. Pour les formats standards, il s'agit de &E5.
- BPS (Bytes Per Sectors = octets par secteurs) : c'est de façon «codée», afin que l'information tienne sur un octet, la longueur en octets d'un secteur. Pour connaître la longueur d'un secteur, il suffit d'employer cette formule, $l = 128 \cdot 2^{\wedge} \text{BPS}$. C'est cette valeur qui est en fait directement envoyée au FDC lors du formatage de piste pour indiquer la longueur d'un secteur. Pour les formats standards on a 512 octets par secteurs, or $512 = 128 \cdot 2^{\wedge} 2$, donc BPS vaut 2.
- RPS (Records Per Sector = enregistrements par secteur) : c'est le nombre de secteurs logiques par secteur physique ; plus prosaïquement c'est la taille d'un secteur en octets divisée par 128. Mathématiquement on a : $\text{RPS} = 2^{\wedge} \text{BPS}$.
- BCT (Buffer for Current Track = mémoire pour piste courante) : c'est une sorte de mémoire pour stocker le numéro de la piste actuellement en service. C'est une des 2 valeurs variables du DPB sans intervention de l'utilisateur. Elle est utilisée exclusivement par le DOS et donc n'a que le mérite d'être signalée.
- FT0 (Flag for Track 0 = indicateur pour piste 0) : c'est un indicateur pour faire savoir au DOS où se trouve la tête de lecture. Le FDC possède une fonction qui lui permet d'aller automatiquement à la recherche de la piste 0. Il interrompt sa recherche au bout d'un certain temps, et indique s'il a ou non atteint la piste 0. La piste 0 a été atteinte si FT0 vaut &FF, sinon il vaut 0. C'est la deuxième valeur variable du DPB du fait du FDC et comme BCT elle ne présente d'intérêt que pour le DOS.
- FLG (Flag for Login = indicateur d'accès) : c'est un indicateur destiné au DOS, pour lui faire savoir si, avant toute opération de lecture ou écriture, il doit accéder au disque afin d'en reconnaître le format. On parle couramment pour désigner cette opération de login. Si FLG vaut 0, il y a automatiquement login, s'il vaut &FF, il n'y a pas de login. Naturellement le DOS ne reconnaît que les formats standards, à savoir System, Data et IBM, qui sont implantés dans la ROM Disc. FLG est un paramètre important car si le login est autorisé, alors qu'on est sous un format spécial, il y a recherche du format de disquette. Cela débouche sur un «read fail», ou, quand le formatage peut tromper le DOS, comme en rajoutant un 10ème secteur, à un plantage. Il faut donc mettre cet indicateur à &FF pour éviter une telle déconvenue (en fait le problème du login ne se pose apparemment que si le DPB est situé dans la RAMDOS ; c'est une bizarrerie de l'AMSDOS).

Voilà ! On a enfin passé en revue le contenu d'un DPB, qui ne doit plus maintenant avoir de secret pour personne. On va voir maintenant quelques valeurs de paramètres pour ceux à qui cette récréation informatique aurait donné envie d'expérimenter des formats exotiques.

Commençons par un tableau de BSH, BLM et EXM, qui sont certainement les paramètres les plus étonnantes de l'ensemble. Ils sont donnés en fonction de la taille du bloc. On rappelle qu'on parle de «petit disque» pour moins de 256 blocs et de «grand disque» pour plus de 255 blocs.



Taille de bloc en KO	BSH	BLM	EXM	
			petit disk	grand disk
1	3	7	0	-
2	4	15	1	0
4	5	31	3	1
8	6	63	7	3
16	7	127	15	7

Finissons par un tableau de DPBs entiers. On va y retrouver pour la forme les formats System et Data. J'ai appelé Data+ le format déduit du Data, avec 10 secteurs par piste (on a rajouté un secteur &CA), et en utilisant les pistes 40 et 41, ce qui offre 208 KO de stockage de données. Comme les 9 neuf premiers secteurs sont identiques à ceux en Data, un programme situé en piste 0, qui installera le DPB du Data+, peut être chargé dès l'allumage du système sans aucune modification de DPB. Le format 202K est un format débutant en piste 1, la piste 0 étant en Data, afin d'y mettre l'utilitaire installant le DPB pour le 202K. D1-198K et D2-198K sont les 2 DPBs pour partager une face de disquette 80 pistes en deux. Ils ne se distinguent que par la valeur de OFF (0 pour l'un et 40 pour l'autre). La disquette est formatée comme en Data+. Toutes les valeurs sont en hexadécimal.

	System	Data	Data+	202K	D1-198K	D2-198K
SPT	24	24	28	28	28	28
BSH	03	03	03	03	03	03
BLM	07	07	07	07	07	07
EXM	00	00	00	00	00	00
DSM	AA	B3	D1	C9	C7	C7
DRM	3F	3F	3F	3F	3F	3F
ALO, AL1	C000	C000	C000	C000	C000	C000
CKS	0010	0010	0010	0010	0010	0010
OFF	02	00	00	01	00	28
FSC	41	C1	C1	C1	C1	C1
PST	09	09	0A	0A	0A	0A
GLS	2A	2A	1C	1C	1C	1C
GLT	52	52	2A	2A	2A	2A
FLB	E5	E5	E5	E5	E5	E5
BPS	02	02	02	02	02	02
RPS	04	04	04	04	04	04
BCT	—	—	—	—	—	—
FT0	—	—	—	—	—	—
FLG	00	00	FF	FF	FF	FF

Voilà qui conclut définitivement cette longue dégression et, je l'espère, récréation sur les DPBs. On va enfin aborder les programmes pour utiliser sans (trop de) problèmes les formats spéciaux.

PROGRAMMES POUR EXPLOITER DES FORMATS SPECIAUX

Il va s'agir ici de commenter les différents programmes proposés, bien que leurs listings le soient déjà grandement. Mais mieux vaut deux fois qu'une. Des explications parfois importantes sont données ici, qui ne figurent pas dans les commentaires des program-

mes. Lorsqu'il est fait explicitement appel à un format spécial, on a choisi le Data+, qui est le plus intéressant pour tout possesseur de système disque.

1) BOOT-DPB : est souvent appelé BOOT un petit programme qui installe un système d'exploitation ou qui lance un programme en faisant un peu de ménage avant, pour lui préparer la place. C'est la fonction de la routine du système d'exploitation MC BOOT PROGRAM.

Ce programme se contente d'installer un DPB et a surtout un intérêt pédagogique. Le DPB est mis en &BE00. C'est en effet une des rares zones de la mémoire qui soient inemployées tant par le système d'exploitation que par le BASIC, et par même beaucoup de programmes. Un rappel de l'emploi de la mémoire peut être utile, pour que chacun puisse loger un DPB, les RSXs (Resident System eXtension = extension au système résident) créées par le programme qui suivra, ou d'autres routines sensibles, à des endroits relativement sûrs :

```
0000I
    saut dans le système d'exploitation pour
    remettre à zéro le CPC.
0007  Peut être employé sans risque.

X 0008
X      série de restarts et de sauts ; strictement réservé
X 0040  au système d'exploitation. ZONE SENSIBLE

0041
    RAM disponible sous BASIC
A6FB

X A6FC
X      RAM réservée à l'AMSDOS (RAMDOS)
X ABFF IZONE SENSIBLE

X AC00I
X      RAM réservée à la ROM BASIC. ZONE SENSIBLE
X B0FF I

X B100I
X      RAM réservée au système d'exploitation. ZONE SENSIBLE
X B8FF I

X B900I
X      ensemble de routines et de vecteurs de saut. ZONE SENSIBLE
X BDF3I

BDF4I
    ESPACE LIBRE
BE3FI

X BE40I
X      strictement réservé au DOS pour la gestion physique du
X BE7F  lecteur de disquette. ZONE SENSIBLE
BE80I
    ESPACE LIBRE, mais attention à la pile descendante du Z80
BFFF  débutant en C000

X C000I
X      RAM VIDEO
X FFFF I
```



Il s'agit de la configuration sous BASIC. En LM, on libère la RAM réservée à la gestion du BASIC, on peut placer ailleurs la RAMDOS, ainsi que la RAM VIDEO... On constate donc que les zones libres et protégées sont rares : de BDF5 à BE3F et de BE40 à BF60 (la pile du Z80 s'étend très rarement en dessous de BF60).

Revenons sur le programme BOOT-DPB : l'adresse de placement du DPB peut être aisément modifiée. Il opère ainsi :

- 1) Installation du DPB
- 2) Lecture du pointeur de la RAMDOS
- 3) Lecture du lecteur actif (A ou B)
- 4) Remplacement du pointeur de DPB du lecteur actif par l'adresse d'implantation du nouveau DPB

On rappelle que, le format utilisé étant le Data+, ce programme peut se trouver sur la piste 0. A la mise en route, le DOS identifie le format comme du Data et BOOT-DPB peut être chargé. Ce dernier «réusscite» alors en quelque le 10ème secteur, en installant le Data+. On se rend compte ici du danger à manipuler sans précautions des disquettes en Data et en Data+. Le DOS peut travailler avec une disquette en Data+ comme si elle était en Data, ce qui peut amener une altération de données en cas de sauvegarde. Prudence ! L'installation systématique sur toute disquette d'un premier fichier installant le DPB (et pourquoi pas faisant office aussi de menu - cf le programme MENU) est le meilleur moyen de se garder de tels désagréments.

2) LOADRUN : ce programme implante 2 RSXs pour charger des programmes en binaire. Ces RSXs sont ILOAD et IRUN et remplissent les mêmes fonctions que les commandes du même nom du BASIC. Elles sont logeables n'importe où dans la mémoire disponible, LOADRUN demandant une adresse de départ (une sauvegarde est effectuée automatiquement à la fin). Il est préférable d'indiquer la valeur en hexadécimal. Voyons d'abord les particularités de ces RSXs, sans quoi il n'y aurait pas de raison de remplacer les commandes originales :

- ILOAD : à la différence de la commande BASIC qui le nécessite, il n'y a pas besoin qu'il y ait 4 KO de mémoire disponible pour un buffer de transfert de données. La syntaxe est identique à celle en BASIC :

- ILOAD,»XXXXXX» pour charger à son adresse de chargement (!) le fichier XXXXXX. Les possesseurs de 464, à cause d'un bug de la ROM BASIC, doivent naturellement mettre le nom dans une variable et indiquer le pointeur de cette variable. Cela donne par exemple: a\$=»XXXXXX»:ILOAD, àa\$. Ce sera cette dernière forme qui sera employée lorsqu'on aura affaire à cette instruction, ou à l'autre, pour des raisons évidentes de compatibilité. Si on oublie d'indiquer le nom de fichier, il y a retour immédiat au BASIC, de même si le nom est illégal, auquel cas on a droit au message d'erreur habituel de l'AMSDOS.

- ILOAD,»XXXXXX»,adr pour charger le fichier XXXXXX à l'adresse adr. La commande gère les erreurs qui peuvent survenir dans les routines de chargement: fichier absent ou bloc altéré. La main est rendue au

BASIC sans aucun problème. Le seul risque de plantage est un chargement dans une des zones sensibles, signalées dans le plan de la mémoire, auxquelles il faut ajouter le programme BASIC lui-même ainsi que les 2 RSXs.

- IRUN : cette instruction fonctionne comme celle du BASIC. Il en résulte qu'elle ne peut être employée qu'une seule fois, la main étant passée à la fin de son exécution au programme qui a été chargé. En cas de problème de chargement (pas de nom fourni, nom illégal, fichier introuvable ou bloc altéré) il y a cependant retour au BASIC. La syntaxe est bien naturellement: IRUN,»XXXXXX» ou a\$=»XXXXXX»:IRUN,àa\$. La différence majeure entre cette RSX et le RUN du BASIC est que cette routine ne fait pas appel à la routine du système d'exploitation MC BOOT PROGRAM, qui passe elle-même la main au MC START PROGRAM, avec les conséquences que l'on a déjà vu. Ainsi, grâce à cette RSX, il n'y a pas de JUMP RESTORE, c'est-à-dire que les vecteurs éventuellement détournés sont laissés intacts, et le système disque est toujours reconnu. C'est l'absence de JUMP RESTORE qui est toutefois important.

On dispose donc maintenant de 2 RSXs très utiles, tant pour l'objet de cet article que pour un usage personnel : avec ILOAD on peut se passer de buffer de 4 KO pour des chargements simples et avec IRUN on peut lancer des programmes binaires sans que le système disque soit «oublié» et sans que les vecteurs soient réinitialisés. On va donc pouvoir en détourner quelques-uns à nos fins. Ce sera mis en application dans le programme MENU. Mais auparavant on va enfin mettre en application la méthode décrite dans le chapitre LA RAM AFFECTEE AU DOS pour procéder à une réinitialisation de DOS sans perte des lecteur, user et DPB.

3) INITDOS1 : ce programme est encore un chargeur de routine, qui fonctionne comme le précédent. Il est destiné à être utilisé sousBASIC et opère un détournement de INIT BACK, ainsi d'ailleurs que de ROM WALK. La routine implantée est sauvegardée sous forme binaire en fin de programme. En queue de routine se trouve le véritable vecteur pour INIT BACK. Il est utile de rappeler que ce vecteur est spécifique à chaque type de CPC (464, 664 ou 6128). On ne pourra pas se servir donc directement de la routine en binaire écrite





sur un type de CPC avec un autre type. Il faut la resauver sur ce dernier type. Un listing en langage d'assemblage suffisamment commenté est fourni en annexe, pour les curieux. Ceux qui n'y connaissent rien pourront cependant en suivre les étapes.

Le chargeur fonctionne ainsi :

- demande de l'adresse d'implantation de la routine
- écriture de la routine
- écriture en fin de routine du vecteur original INIT BACK
- remplacement du vecteur INIT BACK par un saut à la routine
- remplacement du vecteur ROM WALK par un saut à la routine
- sauvegarde de la routine

On constate que pour rendre opérante cette routine, il faut détourner les vecteurs INIT BACK et ROM WALK. Cette manœuvre de détournement sera évidemment à répéter ultérieurement après tout chargement de la routine binaire. Quoi qu'il en soit, le chargeur est très simple et peut être aisément modifié selon sa convenance personnelle. Ce détournement de vecteur sera mis en œuvre très bientôt dans le programme MENU. Ajouté aux RSXs ILOAD et IRUN on obtiendra d'excellents résultats. Mais voyons auparavant une routine quasiment identique, à la différence près qu'elle ne requiert pas de détournement de vecteur ; son usage reste en contrepartie réservé aux connaisseurs de l'assembleur ou autres triffouilleurs d'octets.

4) INITDOS2 : encore une fois un chargeur BASIC, identique aux précédents, pour une routine à utiliser exclusivement sous assembleur pour des programmes en LM. Comme indiqué en fin de paragraphe précédent, cette routine est uniquement destinée aux connaisseurs de langage machine. L'adresse d'implantation demandée par le chargeur BASIC n'a d'intérêt que pour la sauvegarde de la routine en binaire. Comme il apparaît à la lecture du listing en langage d'assemblage, elle est logeable telle quelle à n'importe quelle adresse, puisqu'elle n'effectue de saut à aucune adresse absolue, ni ne se modifie elle-même, comme le fait la routine de INITDOS1, pour que soient injectées dans HL et DE les valeurs désirées pour l'appel de INIT BACK. Il s'agit en fait de la routine proposée dans INITDOS1, mais un peu simplifiée. Le DPB est placé en BE00, qui est une zone très rarement utilisée. Les 3 octets précédents sont utilisés comme mémoire temporaire de lecteur et user. La routine a été ainsi conçue que l'adresse de ce buffer (mémoire tampon+DPB) n'est à indiquer qu'une fois, au début. On peut donc très rapidement le placer ailleurs.

Cette routine est à utiliser en sus et place du simple appel à INIT BACK. Si on dispose d'assez de place, soit 63 octets sans "RET", on n'a qu'à remplacer le INIT BACK par elle. Sinon il faut remplacer l'appel original à INIT BACK par un appel à la routine. Il faut veiller alors à injecter les valeurs correctes dans HL et DE avant l'appel du INIT BACK à l'intérieur de la routine.

L'extraordinaire intérêt de la routine proposée est qu'on peut utiliser alors les programmes binaires qui en sont pourvus avec des disquettes de tous formats, standards ou non, et qu'en plus du INIT BACK les

lecteur et user actifs sont préservés. Il n'y a nul besoin de détourner un seul vecteur ou d'utiliser la RSX I RUN. On se sert du RUN>XXXXXX du BASIC.

Pour résumer, et en conclusion de ce paragraphe, grâce à cette routine, le lecteur, le user et le DPB du lecteur actifs avant la réinitialisation du système disque sont restaurés après.

5) MENU : enfin le programme attendu au moins par ceux qui ne jurent que par le BASIC, faute de mieux. Il met en application toutes les explications données : installation de DPB, RSXs, écriture d'une routine de remplacement pour INIT BACK, d'ailleurs détournée, et finalement modification de quelques vecteurs pour que le détournement effectué ne soit pas réduit à néant. Ce programme n'est pas exactement un menu, mais il prépare le terrain pour un véritable menu, comme il en sera donné un exemple avec MENU2, lancé par MENU. Je le redis encore une fois, mais mieux vaut trop que pas assez : MENU doit être le premier fichier sur la disquette, au sens physique du terme, c'est-à-dire résider en piste 0. On a 4,5 KO par piste en format Data. Les 2 premiers servant au directory, il en reste 2 pour un programme lisible indifféremment en Data ou Data+. MENU est très succinctement commenté pour rester dans la limite des 2 KO (il y a cependant encore de la marge pour des ajouts personnels). Hormis des initialisations de blocs et la modification de 2 vecteurs, tout a déjà été vu dans les programmes BOOT-DPB, LOADRUN et INITDOS1. Voyons de plus près le fonctionnement de MENU :

- Initialisations de blocs : c'est facultatif si on vient d'allumer l'ordinateur, sinon cela effectue une partie du travail du MC START PROGRAM, que l'on désactivera. On commence par restaurer les vecteurs originaux avec JUMP RESTORE (BD37). Cela verrouille sur le lecteur de cassette, ce qui explique ensuite l'instruction I DISC pour retourner au lecteur de disquette. On initialise les blocs :

- son : BCA7 - SOUND RESET
- gestion d'écran : BBFF - SCR INITIALISE
- écran graphique : BBBA - GRA INITIALISE
- écran de texte : BB4E - TXT INITIALISE
- clavier : BB00 - KM INITIALISE

- Installation du DPB : absolument rien de nouveau ici par rapport à ce qui est fait dans BOOT-DPB. On a seulement supprimé les commentaires et fait en sorte que cela prenne le moins de place possible. Le souci d'économie de place a été constant dans l'écriture de MENU, pour que l'utilisateur ait le maximum de liberté pour des apports dans la limite des 2 KO.

- Installation des RSX ILOAD et IRUN : se reporter à LOADRUN pour plus de détails. A noter l'utilisation d'un sous-programme commun à cette phase et à la suivante, toujours pour un gain d'espace. Comme on peut le constater à la lecture du listing, on peut très facilement loger à l'adresse que l'on désire les RSXs. Il n'y a qu'une variable dont il faut modifier la valeur, qui est celle d'implantation des RSXs.

- Ecriture du INITBACK détourné : se reporter à INITDOS1 pour plus de détails. Les remarques faites pour installer ILOAD et IRSX sont valables ici. On installe la routine remplaçant le INITBACK, dont on détourne le vecteur vers elle, ainsi que celui de ROM WALK.



- Modification de 2 vecteurs : Il s'agit de MC START PROGRAM et de JUMP RESTORE. La nécessité de détourner ces routines a été expliquée dans le chapitre LES ROUTINES DU CPC EN CAUSE. Pour résumer, leur appel réinitialise les vecteurs, ce qui invalide tout détournement de vecteur. Examinons comment on opère :

- JUMP RESTORE : on annule tout simplement l'effet de cette routine en remplaçant le vecteur de saut par un retour immédiat. On fait par prudence un peu plus. En effet la plupart des vecteurs se décomposent ainsi : 1) 1 Restart (1 octet), soit un saut automatique à une des 7 premières adresses multiples de 8.

- 2) l'adresse sur 2 octets en ROM ou RAM de la routine. Aussi certains programmes, plutôt que d'appeler une routine via son vecteur, vont lire son adresse et exécutent directement eux-mêmes la routine en question. Ceci est très fâcheux dans notre cas. C'est pourquoi on remplace les 2 octets par une fausse adresse, sans conséquence. J'ai choisi l'adresse 000F, où se trouve l'instruction RET (RETurn) : un CALL y est donc sans effet. A noter que c'est une des rares adresses de même valeur pour les 3 types de CPC. Est-il nécessaire de préciser pour conclure qu'on a remplacé le restart par un RET ?

- MC START PROGRAM : comme vu au chapitre SYNTAXE DES ROUTINES, cette routine, après avoir effectué son travail, saute à l'adresse contenue dans HL. On remplace donc le Restart par un saut direct à cette adresse, grâce à l'instruction JP (HL). Aux 2 octets suivants ont été substitués l'adresse 001E, où l'instruction est encore commune aux 3 types de CPC; il s'agit d'un JP (HL), comme avaient pu le deviner les plus perspicaces.

- Lancement du menu : par un simple RUN est lancé le véritable programme de MENU, MENU2, que l'on va voir maintenant.

6) MENU2: ce programme de menu n'est là qu'à titre indicatif et se propose simplement de donner l'exemple pour que chacun puisse utiliser ses propres menus avec l'acquis dû à MENU. MENU2 est donc très simple. Sa partie la plus intéressante est la ligne de DATA. Examinons la :

```
DATA BASIC,BINAIRE,MIXTE,0,basic,1,binair,0,mixte
```

On y trouve d'abord en majuscules les noms des 3 programmes au menu, tels qu'on veut les porter à la connaissance de l'utilisateur. Puis suivent en minuscules les noms des 3 fichiers correspondant dans l'ordre aux 3 programmes. Chaque nom de fichier est précédé d'un indicateur : 0 ou 1. Un 0 indique un fichier qu'on peut lancer avec l'ordre BASIC RUN tandis qu'un 1 signale un fichier qui nécessite

l'emploi de la RSX I RUN. Cet indicateur donne en quelque sorte le type du fichier. D'où la ligne BASIC de lancement :

```
READ t,a$:IF t=0 THEN RUN a$ ELSE I RUN,àa$.
```

La procédure à suivre s'avère donc, pour ceux qui en auraient douté jusque là, enfantine. Pour exemple encore, on peut consulter le listing de MIXTE, qui se contente de lancer avec I RUN le programme en LM «BINAIRE», lequel charge un écran. On va enfin pouvoir conclure.

CONCLUSION

Tout d'abord bravo au lecteur qui aura tout lu et tout compris du premier coup. Je me suis efforcé de rendre le maximum d'information accessible au plus grand nombre et espère y avoir réussi. C'est pourquoi je n'ai pas hésité parfois à me répéter, au risque d'agacer les connaisseurs, pour que les notions fondamentales soient bien assimilées. Mon souhait le plus cher est que cet article et particulièrement les routines et trucs présentés puissent rendre de grands services, que les formatages spéciaux de disquette ne soient plus un problème. A une prochaine fois !

Yannick GOUR

BIBLIOGRAPHIE :

- LA BIBLE DU PROGRAMMEUR DE L'AMSTRAD CPC - Micro Application - 1985

- LE LIVRE DU LECTEUR DE DISQUETTE AMSTRAD - Micro Application - 1985

- CLEFS POUR AMSTRAD - Système de base - Editions du P.S.I. - 1986

- CLEFS POUR AMSTRAD - Système disque - Editions du P.S.I. - 1986

- CP/M APPROFONDI - Sybex - 1980

N'oubliez pas !!

36 • 15

ARCADES

Des centaines de logiciels pour CPC à télécharger avec le kit ARCADES.

Voir bon de commande page 24

TRUCS POUR L'EXPLOITATION DE DISQUETTES EN FORMAT SPÉCIAL POUR LECTEUR A OU B

BOOT-DPB.BAS

```

100 '***** >RB
110 '* Installation de DBP * >RC
120 '***** >RD
130 DEFINT A-Z >WA
140 ' >RF
150 '--- IMPLANTATION DU DPB EN &BE00 --- >RG
160 AD=&BE00:FOR i=0 TO 24:READ i$:POKE AD+i,VAL("&" +i$ >FF
):NEXT
170 ' >RJ
180 '--- DETERMINATION DE L'ADRESSE --- >RK
190 '--- DE LA RANDOS --- >TA
200 randos=UNT(PEEK(&BE7D)+256*PEEK(&BE7E)) >NQ
210 ' >RD
220 '--- DETERMINATION DU LECTEUR ACTIF --- >RE
230 '--- PAR LECTURE EN DEBUT DE RANDOS --- >RF
240 lecteur=PEEK(randos) >VY
250 ' >RH
260 '--- REMPLACEMENT DU POINTEUR DE DPB --- >RJ
270 '--- DU LECTEUR ACTIF PAR &BE00 --- >RK
280 adptr=randos+21A+lecteur*10 >DC
290 POKE adptr,ad AND &FF:POKE adptr+1,INT(ad/256) AND >CW
&FF
300 ' >RD
310 '--- DPB FORMAT "Data+" --- >RE
320 DATA 28,00,03,07,00,D1,00,3F,00,C0,00,10,00,00,00,C >GG
1,0A,1C,2A,E5,02,04,00,00,FF

```

INITDOS 1.BAS

```

100 '***** >RB
110 '* INIT BACK (& ROM WALK) DETOURNE * >RC
120 '* copyright Y. GOUR (04/1991) * >RD
130 '***** >RE
140 ' >RF
150 DEFINT a-z:MODE 1:CALL &BC02 >ZA
160 INPUT"adresse de INIT BACK detourne";a:ad=a >BH
170 ' >RJ
180 ' --- installe la nlle routine --- >RK
190 b=a-1:MEMORY MIN((b AND 255)+(INT(b/256) AND 255)*2 >HD

```

```

56,HIMEM)
200 WHILE a$("<" "FIN" >NH
210 READ a$ >NF
220 IF a$="*" THEN READ a$:POKE a,(ad+VAL("&" +a$)) AND >HN
255:POKE a+1,INT((ad+VAL("&" +a$))/256) AND 255:a=a+2:GO
TO 240
230 IF a$("<" "FIN" THEN POKE a,VAL("&" +a$):a=a+1 >MZ
240 WEND >CE
250 IF a("<"(ad+48) THEN PRINT"Erreur":END >LR
260 ' >RJ
270 '--- reloge le vrai INIT BACK --- >RK
280 POKE a,PEEK(&BCCF):POKE a+1,PEEK(&BCD0) >LH
290 ' >TB
300 '--- detourne INIT BACK ---- >RD
310 POKE &BCCF,&C3:POKE &BCCF,ad AND 255:POKE &BCD0,INT >RJ
(ad/256) AND 255
320 ' >RF
330 '--- detourne aussi ROM WALK --- >RG
340 POKE &BCCC,&C3:POKE &BCCC,ad AND 255:POKE &BCCD,INT >RT
(ad/25.) AND 255
350 ' >RJ
360 '--- sauve INIT BACK detourne --- >RK
370 SAVE"initdos1",b,ad,&4A >YE
380 END >TE
390 ' >TC
400 DATA 22,$,2B,ED,53,$,2E,2A,7D,BE,11,FD,BD,1,3,0,7E, >VX
D5,ED,B0,1,17,2,B7
410 DATA 28,2,E,27,C5,9,4E,23,66,69,1,18,0,ED,B0,3E,FF, >WJ
12,21,FB,AB,11,40,0
420 DATA E,7,CD,$,47,E1,E3,ED,5B,7D,BE,1,3,0,ED,B0,EB,C >WX
1,9,73,23,72,C9,CF
430 DATA FIN >CK

```

INITDOS 2.BAS

```

100 '***** >RB
110 '* INIT BACK AMELIORE * >RC
120 '* copyright Y. GOUR (04/1991) * >RD
130 '***** >RE
140 ' >RF
150 DEFINT a-z:MODE 1:CALL &BC02 >ZA
160 INPUT"ad de INIT BACK ameliorer";a:ad=a >TN
170 ' >RJ

```

```

180 ' --- installe la nlle routine --- >RK
190 b=a-1:MEMORY MIN((b AND 255)+(INT(b/256) AND 255)*2 >HD
56,HIMEM)
200 WHILE a$<>"FIN" >NM
210 READ a$ >NF
220 IF a$=" $" THEN READ a$:POKE a,(ad+VAL("&"a$)) AND >HN
255:POKE a+1,INT((ad+VAL("&"a$))/256) AND 255:a=a+2:GO
TO 240
230 IF a$<>"FIN" THEN POKE a,VAL("&"a$):a=a+1 >MZ
240 WEND >CE
250 IF a<>(ad+&40) THEN PRINT"Erreur":END >LH
260 ' >RJ
270 '--- sauve INIT BACK ameliore --- >RK
280 SAVE"initdos2",b,ad,&40 >YM
290 END >TE
300 ' >RD
310 DATA 2A,7D,BE,11,FD,BD,1,3,0,7E,D5,ED,B0,1,17,2,B7, >YU
28,2,E,27,C5,9,4E,23
320 DATA 66,69,1,18,0,ED,B0,3E,FF,12,21,FB,AB,11,40,0,E >ZD
,7,CD,CE,BC,E1,E3,ED
330 DATA 5B,7D,BE,1,3,0,ED,B0,EB,C1,9,73,23,72,C9 >QM
340 DATA FIN >CK

```

LOADRUN.BAS

```

10 '***** >LA
15 '* RSX LOAD ET RUN: * >LF
20 '* _LOAD sans Buffer * >LB
25 '* _RUN sans MC BOOT PROGRAM * >LG
30 '* copyright Y. GOUR (04/1991) * >LC
35 '***** >LH
40 ' >LD
45 DEFINT a-z:MODE 1:CALL &BC02 >YE
50 INPUT"adresse des RSXs";a:ad=a >HP
55 b=a-1:MEMORY MIN((b AND 255)+(INT(b/256) AND 255)*25 >GX
6,HIMEM)
60 WHILE a$<>"FIN" >ME
65 READ a$ >HG
70 IF a$=" $" THEN READ a$:POKE a,(ad+VAL("&"a$)) AND 2 >EC
55:POKE a+1,INT((ad+VAL("&"a$))/256) AND 255:a=a+2:GOT
0 80
75 IF a$<>"FIN" THEN POKE a,VAL("&"a$):a=a+1 >LU
80 WEND >XB
85 IF a<>(ad+&75) THEN PRINT"Erreur":END >KP
90 SAVE"loadrun",b,ad,&75 >XK
95 CALL ad >MB
100 END >RE
105 ' >RG
110 DATA 21,$,9,1,$,D,C3,D1,BC,0,0,0,0,$,15,C3,$,1D,C3, >QR
$,54,4C,4F,41,C4,52,55,CE,0,B7,C8,3D,28,B,DD,66,1,DD,6E,0,E5,DD >ZN
,23,DD,23,DD,66,1,DD
120 DATA 6E,0,46,23,5E,23,56,21,0,C0,EB,F5,CD,77,BC,38, >XD
5,F1,B7,C8,E1,C9,F1
125 DATA B7,28,1,D1,EB,CD,83,BC,DC,7A,BC,C9,3D,C0,DD,66 >YW
,1,DD,6E,0,46,23,5E
130 DATA 23,56,21,0,C0,EB,CD,77,BC,D0,EB,CD,83,BC,D0,E5 >XX
,CD,7A,BC,D8,E1,C9
135 DATA FIN >DB

```

MENU.BAS

```

100 'RAZ et DPB >RB
110 DEFINT a-z:RESTORE 210:READ a$:WHILE a$<>"FIN":CALL >YJ
VAL("&"a$):READ a$:WEND:ùDISC:d=&BE00:FOR i=0 TO 24:R
EAD a$:POKE d+i,VAL("&"a$):NEXT:a=UNT(PEEK(&BE7E)*256+
PEEK(&BE7D)):a=PEEK(a)*16+&21A+a:POKE a,d AND 255:POKE
a+1,INT(d/256) AND 255
120 'RSXs et INIT BACK detourne >RD
130 a=&A600:MEMORY a-1:GOSUB 170:CALL ad:a=&BF00:RESTOR >BH
E 310:GOSUB 170:POKE a,PEEK(&BCCF):POKE a+1,PEEK(&BCD0)
:POKE &BCCE,&C3:POKE &BCCB,&C3:b=ad AND 255:POKE &BCCF,
b:POKE &BCCC,b:b=INT(ad/256) AND 255:POKE &BCD0,b:POKE
&BCCD,b
140 'modifie START PROGRAM et JUMP RESTORE >RF
150 a=&BD16:POKE a,&E9:POKE a+1,&1E:POKE a+2,0:a=&BD37: >TB
POKE a,&C9:POKE a+1,&F:POKE a+2,0
160 RUN"menu2 >LE
170 ad=a:a$="":WHILE a$<>"FIN":READ a$:IF a$=" $" THEN R >FQ
EAD a$:b=ad+VAL("&"a$):POKE a,b AND 255:POKE a+1,INT(b
/256) AND 255:a=a+2 ELSE IF a$<>"FIN" THEN POKE a,VAL("
&"a$):a=a+1
180 WEND:RETURN >MK
190 ' >TA
200 'remise a zero >RC
210 DATA BD37,BCA7,BBFF,BB4E,BB00,FIN >FG
220 ' >RE
230 'DPB >RF
240 DATA 28,0,3,7,0,D1,0,3F,0,C0,0,10,0,0,0,C1,A,1C,2A, >MT
E5,2,4,0,0,FF
250 ' >RH
260 'LOAD et RUN >RJ
270 DATA 21,$,9,1,$,D,C3,D1,BC,0,0,0,0,$,15,C3,$,1D,C3, >YA
$,54,4C,4F,41,C4,52,55,CE,0,B7,C8,3D,28,B,DD,66,1,DD,6E
,0,E5,DD,23,DD,23,DD,66,1,DD,6E,0,46,23,5E,23,56,21,0,C
0,EB,F5,CD,77,BC,38,5,F1,B7,C8,E1,C9,F1,B7,28,1,D1,EB,C
D,83,BC,DC,7A,BC,C9
280 DATA 3D,C0,DD,66,1,DD,6E,0,46,23,5E,23,56,21,0,C0,E >CA
B,CD,77,BC,D0,EB,CD,83,BC,D0,E5,CD,7A,BC,D8,E1,C9,FIN
290 ' >TB
300 ' INIT BACK detourne >RD
310 DATA 22,$,2B,ED,53,$,2E,2A,7D,BE,11,FD,BD,1,3,0,7E, >FH
D5,ED,B0,1,17,2,B7,28,2,E,27,C5,9,4E,23,66,69,1,18,0,ED
,B0,3E,FF,12,21,FB,AB,11,40,0,E,7,CD,$,47,E1,E3,ED,5B,7
D,BE,1,3,0,ED,B0,EB,C1,9,73,23,72,C9,CF,FIN

```

MENU 2.BAS

```

10 MODE 1:LOCATE 19,6:PRINT"MENU":FOR i=1 TO 3:LOCATE 1 >RU
5,7+2*i:READ a$:PRINT i;a$:NEXT
20 i=VAL(INKEY$):IF i>4 OR i=0 THEN 20 ELSE k=i:FOR i=1 >VV
TO i:READ t,a$:NEXT:IF t=0 THEN RUN a$ ELSE ùRUN,aa$
30 DATA BASIC,BINAIRE,MIXTE >XV

```

MIXTE.BAS

```

10 CLS:PRINT"Je suis un programme en BASIC":PRINT"Je me >UW
contente de lancer le programme binaire avec ùRUN":FOR
i=0 TO 3000:NEXT:a$="BINAIRE":ùRUN,aa$

```

chronique du TELECHARGEMENT

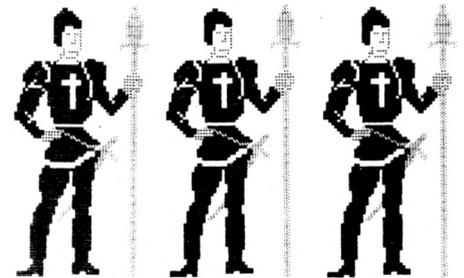
FRANCOIS Ier	1515
QUESTION : Quelle victoire permet à la France de récupérer le Milanais ?	
1.. Turin	
2.. Rocroi	
3.. Milan	
4.. Marignan	
5.. Rome	
..TAPEZ LE CHIFFRE CHOISI..	

HISTOIRE

Tout d'abord, il faut souligner l'utilité des remarques sur le serveur puisqu'ainsi, vous pouvez signaler les problèmes rencontrés lors du téléchargement. Pour Desktop, il manque effectivement le fichier Easy.doc. Ce dernier sera ajouté et vous pourrez le télécharger séparément. Pour Omicron Démo, il n'y a pas de fichier «Menu», celui-ci étant un reliquat du fanzine duquel il a été extirpé. Il suffit de supprimer cette référence dans le premier fichier (Omicron.bin).

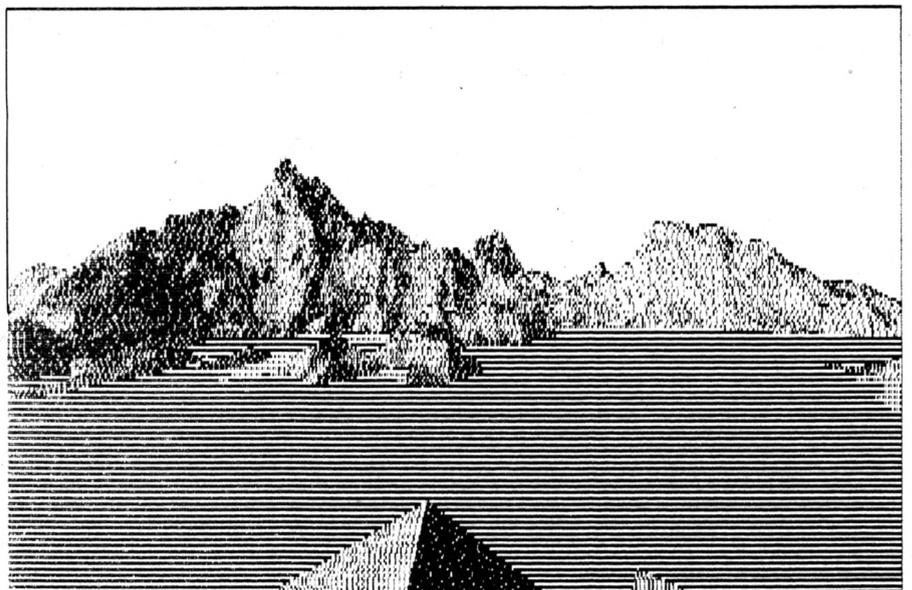
Au programme aujourd'hui, voici Histoire, un logiciel éducatif envoyé spontanément par son auteur pour venir grossir la quantité de programmes disponibles. Le programme est constitué de séries de questions sur plusieurs périodes de notre histoire. En répondant correctement à toutes les séries, vous obtiendrez un code donnant accès à un jeu. Mais il faudra auparavant répondre et ce n'est pas facile, car, à chaque erreur, il faut reprendre à partir du début. Mais normalement, à la fin de votre «apprentissage» vous aurez acquis pas mal de connaissances historiques.

Un programme déjà publié dans CPC et qui n'avait pas encore connu les joies du téléchargement : Fractal Landscapes. Ce logiciel permet de concevoir des scènes fractales d'une manière relativement simple. A noter tout de même



qu'en modifiant tous les paramètres un peu au hasard, il est rare d'obtenir un bon résultat. De toutes façons, il est nécessaire d'être patient lorsque l'on a demandé un calcul avec ombrage et facettes pleines puisque cela peut aller jusqu'à plusieurs heures. Pour ceux qui voudront avoir des exemples, il suffira de télécharger la démo ainsi que les matrices utilisées pour la démo. Fractal Landscapes ne fonctionne que sur 6128 puisqu'il utilise les 64 Ko supplémentaires. Il faudra d'abord lancer Bankman, puis taper Symbol After 256 et ensuite lancer Fracrun.

Pour rester dans le domaine (merveilleux) des programmes de CPC Infos, voici la journée des pro-



FRACTAL LANDSCAPES

grammes du mois correspondant à 2 numéros (31 et 32). Easy Screen est un utilitaire permettant de composer des pages avec des menus et des sous-menus. En fait, Easy Screen est un générateur de programme. Une fois définies les caractéristiques de vos pages et les liens entre les menus, un programme basic est créé que vous pourrez utiliser ensuite dans vos propres créations. Ce programme est lui-même constitué de menus et possède toutes les explications nécessaires.

Oxysudis est un jeu de plate-formes se déroulant sur des planètes inconnues, avec un temps très limité et des tas de pièges. Sur l'ensemble des niveaux, ce jeu vous donnera du fil à retordre.

Un programme graphique, à paramètres multiples et qui permet de réaliser des motifs géométriques et multicolores, c'est Automate cellulaire. Comme son nom l'indique, il travaille au niveau du pixel et d'une «semence» aléatoire qui lui permet de réaliser des écrans très rapidement (routine en assembleur).

Le jeu que beaucoup de personnes attendent : Axys est enfin sur le téléchargement. Au moment où j'écris, je ne sais pas si ce program-

CREATION MATRICE AFFICHAGE CHARGEMENT SAUVEGARDE QUITTER

AFFICHAGE

```

OK
Lancement
▶ Observateur
Eclairage
Altitude zones planes = 0
Facteur d'échelle = 1
Angle cone de vision = 50
ierPlaine
PaysageCarte
SqueletteFacettesOmbrages
DensitéFill
                
```

NOTICE

```

X : entre -4000 et 4000
Y : entre -4000 et 4000
Dir : entre 0 et 359
                
```

OBSERVATEUR

```

▶ OK
X = 0
Y = 0
Dir = 0
                
```

ECLAIRAGE

```

OK
X = 0
Y = 0
Z = 0
                
```

FRACTAL LANDSCAPES

me sera compacté avec CRUNCH20, étant donné sa taille mais vous le saurez facilement puisqu'il y aura alors un Z dans l'extension du nom de fichier.

Claude Le Moulec vous présente un autre jeu de son crû : Charlie Bond Sound. Il s'agit d'un jeu musical, où vous devez recomposer un air de musique sur 3 voies.

Pour les démos vous trouverez les démos de AST System : Joke demo, Short demo II, Demo3, Joke demo II, Equinox, Demo 6

allo CPC infos

Attention, notez bien ce qui suit :
 Dès le mois de septembre, tous les mercredis de 14 h à 17 h, la rédaction se tiendra à votre disposition au 99 52 98 11 pour tous renseignements techniques. En dehors de ce créneau les réponses techniques se feront par courrier (n'oubliez pas de mentionner votre adresse et joignez un timbre pour la réponse) ou bien par Minitel sur le 3615 code : MHZ.

23

BON DE COMMANDE

A expédier à : Editions SORACOM
La Haie de Pan - 35170 BRUZ

DESIGNATION	Prix unitaire	Quantité	Port	Montant
Pour chaque article, entourez le ou les n° choisis.				
ANCIENS Nos AMSTAR & CPC N° 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 35 - 36 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47	25 F (unité) 26 F (unité)		Franco Franco	
DISQUETTES AMSTAR & CPC N° 20 - 21 - 22 - 23 - 24 - 25 - 26 - 27 - 28 - 29 - 30 (une disquette réunit 2 numéros consécutifs d'Amstar & CPC)	110 F (unité) 140 F (unité)		Franco Franco	
ANCIEN No CPC Infos N° 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33	26 F (unité)		Franco	
DISQUETTES CPC Infos - MEGA SOUND - N° 31 (Am'Star 48 & CPC Infos 24) - N° 32 (CPC Infos 25 & 26) - N° 33 (CPC Infos 27 & 28) - N° 34 (CPC Infos 29 & 30) - N° 35 (CPC Infos 31 & 32) - N° 36 (CPC Infos 33 & 34)	Abonné : 110 F (unité) Non abonné : 140 F (unité)		Franco Franco	
ANCIENS Nos CPC N° 11 - 13 - 16 - 18 - 23 - 24 - 25 - 26 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 36 - 37	25 F (unité)		Franco	
ANCIENS Nos HORS-SERIE CPC N° 7 - 8 - 9 - 10 - 12 - 13 - 14 - 15 - 16 N° 17 - 18 - 19 - 20 - 21 - 22 - 23	21 F (unité) 23 F (unité)		Franco Franco	
DISQUETTES CPC ET DISQUETTES HORS-SERIE CPC N° 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 (une disquette réunit 2 numéros consécutifs de CPC) HS1 - HS2 - HS3 - HS4 - HS5 - HS6 - HS7 - HS8 - HS9 - HS10 - HS11 HS12 - HS13 - HS14 - HS15 - HS16 - HS17 - HS18 - HS19 - HS20 - HS21 - HS22 - HS23	Abonné : 110 F (unité) Non abonné : 140 F (unité)		Franco Franco	
ABONNEMENT 6 DISQUETTES (non rétroactif)	600 F		Franco	
DISQUETTES "ARCADES" Téléchargement AMSTRAD CPC	60 F		5 F	
CABLE pour téléchargement (Téléchargement non compatible avec 6128+ et 464+)	89 F		5 F	
LOGICIELS EDUCATIFS - LIVRES - DIVERS				
DESIGNATION	REFERENCE			
Forfait port + 10 F par logiciel et 10 % pour livres				
Pour tout envoi par avion : prendre contact avec le service commercial			<input type="checkbox"/> Facultatif : recommandé + 10 F par commande	10 F
MONTANT GLOBAL				

Je joins mon règlement : chèque bancaire chèque postal mandat

CPC Infos 34

PAYEZ PAR CARTE BANCAIRE

Date d'expiration _____ Signature _____

(inscrire les numéros de la carte, la date et signer)

Nom : _____ Prénom : _____

Adresse : _____

Code Postal : _____ Ville : _____

ECRIRE EN MAJUSCULES

Date et signature _____ Signature _____

Afin de faciliter le traitement des commandes, nous remercions notre aimable clientèle de ne pasagrafer les chèques, et de ne rien inscrire au dos.

Commande : La commande doit comporter tous les renseignements demandés sur le bon de commande (désignation ou référence si celle-ci existe). Toute absence de précisions est sous la responsabilité de l'acheteur. La vente est conclue dès acceptation du bon de commande par notre société, sur les articles disponibles uniquement.

Les prix : Les prix indiqués sont valables du jour de la parution du catalogue jusqu'au mois suivant ou au jour de parution du nouveau catalogue, sauf erreur dans le libellé de nos tarifs au moment de la fabrication du catalogue, et de variation de prix importante des fournisseurs.

Livraison : La livraison intervient après le règlement. Les délais de livraison étant de 10 à 15 jours environ, SORACOM ne pourra être tenue pour responsable des retards dus aux transporteurs ou aux grèves des services postaux.

Transport : La marchandise voyage aux risques et périls du destinataire. La livraison se faisant par colis postal ou par transporteur. Les prix indiqués sur le bon de commande sont valables sur toute la France métropolitaine, + 20 F par article pour Outre-Mer par avion et au-dessus de 5 kg nous nous réservons la possibilité d'ajuster le prix de transport en fonction du coût réel de celui-ci. Pour bénéficier de recours possible nous invitons notre aimable clientèle à opter pour l'envoi en recommandé. A réception des paquets, toute détérioration doit être signalée.

Réclamation : Toute réclamation doit intervenir dans les dix jours suivants la réception des marchandises.

SORACOM SOLDE A SES LECTEURS

Lots de jeux divers cassettes ou disquettes

Par 5 jeux **40 F** le lot de cassettes

Par 10 jeux **60 F** le lot de cassettes

Par 5 jeux **100 F** le lot de 5 disquettes

Par 10 jeux **150 F** le lot de 10 jeux



ATTENTION:

Il s'agit de lots en solde. Les jeux ne sont ni repris ni échangés.
Merci de ne pas demander au téléphone la liste de ces titres.

JUSQU'À EPUISEMENT DES STOCKS

Bon de commande

Je commande

- 5 jeux K7 pour 40 F
- 10 jeux K7 pour 60 F
- 5 jeux disc pour 100 F
- 10 jeux disc pour 150 F

Nom : _____ Prénom : _____

Adresse : _____

Code postal : _____ Ville : _____

Date : _____ Signature _____



Je joins mon règlement chèque bancaire
chèque postal mandat



JE REGLE PAR CARTE BANCAIRE

Date d'expiration

Signature

Retournez ce coupon, accompagné de votre chèque libellé à l'ordre des Editions SORACOM - La Haie de Pan - 35170 BRUZ

CPC

utilitaire

EASY SCREEN



Guy DUBUS

Valable pour CPC 664 - 6128

Avec ce générateur de menus, vous allez pouvoir créer des pages écran ayant une présentation professionnelle. Reportez-vous au numéro précédent pour avoir toutes les explications et les débuts des listings.

S U I T E

```

210 GRAPHICS PEN 1:MOVE 64,368:PRINT a$;:MOVE 160,176:P >HZ
RINT b$;:MOVE 80,144:PRINT c$;:MOVE 96,112:PRINT d$;:MO
VE 0,16:PRINT f$
220 GRAPHICS PEN 14:MOVE 68,372:PRINT a$;:MOVE 164,180: >MG
PRINT b$;:MOVE 84,148:PRINT c$;:MOVE 100,116:PRINT d$;:
MOVE 4,20:PRINT f$
230 SYMBOL AFTER 33:g=112:GRAPHICS PEN 1 >HB
240 FOR n=1 TO LEN(e$):f$=MID$(e$,n,1):IF f$=" " THEN g >NG
=g+32:GOTO 280
250 b=ASC(f$)-32:c=40956+(b*8) >XC
260 FOR m=42732 TO 42746 STEP 2:POKE m,PEEK(c):POKE m+1 >AZ
,PEEK(c):c=c+1:NEXT m
270 g=g+32:MOVE g,288:PRINT CHR$(254);:MOVER -32,-16:PR >TP
INT CHR$(255);
280 NEXT >EF
290 MOVE 112,288:FOR larg=1 TO 8:FOR stylo=2 TO 12 STEP >ZD
2:GRAPHICS PEN stylo:DRAWR 0,-32:MOVER 4,32:DRAWR 0,-3
2:MOVER 4,32:NEXT stylo, larg:TAGOFF
300 col(1)=7:col(2)=6:col(3)=15:col(4)=9:col(5)=2:col(6 >EH
)=4
310 INK 1,17:INK 15,8:INK 14,3:i=1:k=0 >EG
320 FOR stylo=3 TO 13 STEP 2:INK stylo,col(i):i=i+1:IF >NF
i>6 THEN i=1
330 NEXT stylo:i=i+1:IF i>6 THEN i=1 >DN
340 FOR j=1 TO 40:NEXT j:k=k+1 >XY
350 WHILE k<40:GOTO 320:WEND >XT
360 PRINT CHR$(23);CHR$(0);:CLEAR >BM
370 '----- >TA
380 MODE 2:INK 0,15:INK 1,1:LOCATE 35,1:PRINT"*** AIDE >BR

```

```

***:PRINT:PRINT"CREER/COMPLETER UNE SERIE DE PAGES (Ed
iteur de pages)":PRINT:PRINT" * OPTION TEXTE";" - M
ENUS : l'emplacement de la dernière option est réservé
au programme, qui";
390 PRINT" y inscrit ";CHR$(34);"QUITTE";CHR$(34);" >FN
pour le menu principal (le premier) ou ";CHR$(34);"RET
OUR";CHR$(34);" pour les";" suivants.":PRINT" Dans
s l'option ";CHR$(34);"Pointe sur S/Programme Code...";
CHR$(34);", vous pouvez entrer :";
400 PRINT" - un label mn(monique pour la routine @ >XC
appeler (dans VOTRE programme)," - le numéro de
la première ligne de cette routine (précéd de";CHR$(3
4);"##";CHR$(34);"),";
410 PRINT" - un numéro de page (type Message avec >FR
réponse O/N attendue, ou Saisie).";" Dans les 3 cas,
NE PAS OUBLIER D'AJOUTER UN RETURN EN FIN DE ROUTINE
OU DE";" PAGE !"
420 PRINT" - Utilisez la commande ";CHR$(34);"Fin de >GQ
texte";CHR$(34);" (CTRL+F) dans le type Message, si vo
us";" voulez une fenêtre vide (avec ou sans titre)
: fenêtre de travail de votre";
430 PRINT" programme, ou pour incrustation d'une aut >LD
re fenêtre. L'écriture du programme";" généré en ser
a allégée, mais cette commande est irréversible, sauf
si vous";
440 PRINT" décidez de refaire la page.":PRINT:PRINT" >RR
* OPTION COULEURS";" Pour chaque page, définition
n de 4 INK (0 @ 3) ; INK 0 est affecté au fond";
450 PRINT" d'écran et au BORDER. Pour la fenêtre c >GM
ourante (F1 @ F4), affectation d'un";" PAPER (FEN : f
ond de fenêtre) et de 2 PEN (CAD : cadre et CAR : cara

```



```

ct)res)."
```

460 ' *****	>TA
470 ' MENU PRINCIPAL	>TB
480 ' *****	>TC
490 SYMBOL AFTER 219	>PJ
500 SYMBOL 219,38,38,38,38,38,38,38,38	>EK
510 SYMBOL 220,0,0,255,0,0,255,255,0	>BL
520 SYMBOL 221,0,55,72,72,55,72,72,72	>CD
530 SYMBOL 222,0,219,36,36,219,0,0,0	>BX
540 SYMBOL 223,0,236,18,18,236,18,18,18	>EF
550 SYMBOL 224,72,72,48,72,72,48,72,72	>EA
560 SYMBOL 225,18,18,12,18,18,12,18,18	>DR
570 SYMBOL 226,72,72,72,55,72,72,55,0	>DF
580 SYMBOL 227,0,0,0,219,36,36,219,0	>BH
590 SYMBOL 228,18,18,18,236,18,18,236,0	>ER
600 SYMBOL 229,0,127,64,73,86,72,72,80	>DB
610 SYMBOL 234,0,255,0,153,102,0,0,0	>BE
620 SYMBOL 235,0,254,2,146,106,18,18,10	>EP
630 SYMBOL 236,80,72,72,80,80,72,72,80	>DA
640 SYMBOL 237,10,18,18,10,10,18,18,10	>DX
650 SYMBOL 239,80,72,72,86,73,64,127,0	>DH
660 SYMBOL 248,0,0,0,102,153,0,255,0	>BQ
670 SYMBOL 249,10,18,18,106,146,2,254,0	>EA
680 SYMBOL 250,0,63,127,96,96,103,100,100	>GY
690 SYMBOL 251,0,255,255,0,0,255,0,0	>BA
700 SYMBOL 252,0,252,254,6,6,230,38,38	>DL
710 SYMBOL 253,100,100,100,100,100,100,100	>LJ
720 SYMBOL 254,100,100,103,96,96,127,63,0	>GX
730 SYMBOL 255,38,38,230,6,6,254,252,0	>DT
740 OPENOUT "toto":MEMORY HIMEM-1:CLOSEOUT:DEFINT a-z	>DX
750 DIM com(18),m1\$(5),m2\$(6),com\$(5),m\$(3),help\$(15,4)	>DH
, index(99,1),hpos(6),box(15,8),fen(25,16),fen\$(25,6):fo	
rm\$="###"+ " "+CHR\$(149)+" "+CHR\$(149)+" "+CHR\$(149)+"	
"+"#"	
760 l=1:FOR i=1 TO 18:READ com(i):NEXT i	>GK
770 FOR i=1 TO 5:READ m1\$(i):NEXT i	>BJ
780 FOR i=1 TO 6:READ m2\$(i):NEXT i	>BM
790 FOR i=1 TO 5:READ com\$(i):NEXT i	>CW
800 FOR i=1 TO 15:FOR j=1 TO 8:READ box(i,j):NEXT j,i	>VY
810 FOR i=0 TO 25:FOR j=0 TO 16:READ fen(i,j):NEXT j:FO	>CX
R k=1 TO fen(i,1):READ fen\$(i,k):NEXT k,i	
820 GOSUB 2550:CLS:PRINT " * OPTION RETOUR";	>UM
Pour ajouter des fenetres, vous pouvez demander ici @ refa	
ire la page, ou";	
choisir ultrieurement ";CHR\$(34);	
"Modifier/Mettre @ jour";CHR\$(34);	
dans le menu princ	
ipal, puis";	
830 PRINT " ";CHR\$(34);"Modifier page";CHR\$(34);" et ";	>AH
CHR\$(34);"Ajouter une ou des fenetres";CHR\$(34);".";	
L'effacement d'cran au changement de page concer	
ne l'cran PRECEDENT,";	
840 PRINT " lors de l'affichage de la page courante dan	>YV
s votre application.":PRINT:PRINT"MODIFIER/METTRE A JOU	
R UNE SERIE":PRINT:PRINT" Dans une srie de pages D	
EJA ENREGISTREE, vous pourrez ajouter, supprimer,";	
850 PRINT"modifier ou refaire des pages. Vous pourrez	>VE
aussi les d{placer, en changeant";	
leur numro de ra	
ng dans la srie ; le numro de page, lui, NE CHANGE JA	
MAIS. ";	
860 PRINT" ATTENTION : c'est le numro de page, et n	>DU

```

on de rang, que vous devez indiquer";dans l'(diteur de
pages lorsqu'il vous est demand( (Pointe sur S/Menu Pa
ge...):PRINT:PRINT"COMMANDES DANS LE PROGRAMME D'APPL
ICATION":PRINT
870 PRINT" Les memes que dans EASY SCREEN.":PRINT" >CA
Messages n'attendant pas de rponse : taper une t
ouche quelconque pour";"passer @ la suite.;" Fene
tres de saisie : entr(e alphanum(rique ou num(rique, en
mode insertion.;"
880 PRINT"Corrections :";CHR$(242);"/";CHR$(243);"/CLR/ >BK
DEL. Passage au champ pr(cident/suivant et validation :
";CHR$(240);"/";CHR$(241);"Fin de saisie : ENTER ; imp
ossible de sortir si tous les champs ne sont pas";"
saisis."
890 '----- >TH
900 help$(2,1)=" S{lection:"+CHR$(241)+"/"+CHR$(243)+"/
>QD
"+CHR$(240)+"/"+CHR$(242)+ " Validation:ENTER":m$(0)="
Tapez un nom de fichier, puis ENTER":m$(1)=" ou ENTER
pour choisir sur r{pertoire":m$(2)=" Annulation de la
commande : COPY"
910 help$(3,0)=" R{ponse : o/0/n/N":help$(1,1)=" Corr >DG
rections:DEL Validation:ENTER":m$(3)="Choix du num(ro:
"+CHR$(240)+"/"+CHR$(241)+ " Validation:ENTER"
920 help$(3,1)=help$(1,1):help$(4,0)=help$(2,1):help$(4 >ND
,1)=" Sortie d'option : COPY":help$(5,0)="Changer de fe
netre:tapez un No valide":help$(5,1)="de 1 @ 4. Correct
ion:retapez un num(ro"
930 help$(5,2)=help$(4,1):help$(6,0)="Curseur=coin haut >GW
/gauche de la fenetre":help$(6,1)="D{placements du curs
eur:"+CHR$(241)+"/"+CHR$(243)+"/"+CHR$(240)+"/"+CHR$(24
2):help$(6,2)="Apparition de la fenetre minimum:ENTER"
940 help$(7,0)="D{placer la fenetre:"+CHR$(241)+"/"+CHR >XM
$(243)+"/"+CHR$(240)+"/"+CHR$(242):help$(7,1)="Changer
les dimensions:SHIFT"+CHR$(241)+"/"+CHR$(243)+"/"+CHR$(
240)+"/"+CHR$(242):help$(7,2)="Centrer en largeur/en h
auteur:C/CTRL+C"
950 help$(7,3)="Modifier marges:G/D"+CHR$(242)+"/"+CHR >WU
$(243)+ " Sortie:COPY":help$(8,0)=" Tapez un No en he
rad(cimal (1 @ F)":help$(8,1)=" Changer de cadre : tape
z un autre No":help$(8,2)=help$(4,1)
960 help$(9,0)=" Caract(re pr(cident/suivant:"+CHR$(242 >TD
)+"/"+CHR$(243):help$(9,1)="Corrections:CLR/DEL Valide
r ligne:ENTER":help$(9,2)="Ligne pr(c(d./suiv.:"+CHR$(2
40)+"/"+CHR$(241)+ " Effacer:CTRL+E":help$(9,3)=" Fin
de fenetre:CTRL+F"
970 help$(9,4)=help$(4,1):help$(10,0)=" Options s{parfe >VL
s par un seul espace":help$(10,1)=help$(9,0):help$(10,2
)=help$(9,1):help$(10,3)=help$(4,1):help$(11,0)=" 2 ch
iffres maximum":help$(11,1)=help$(1,1)
980 help$(12,0)="10 caract(eres alphanum(riques maxi :": >YB
help$(12,1)=" * Label, d{butant par une lettre":help$(1
2,2)=" * No de ligne appel(e, pr(c(d{ de #":help$(12,3)
=" * No de page (2 chiffres maxi)":help$(12,4)=help$(1,
1)
990 help$(13,0)=" (A)lphanum(rique ou (N)um(rique":hel >HN
p$(13,1)=help$(1,1):help$(14,0)="6 caract(eres alphanum(
riques maxi":help$(14,1)="commentant par une lettre":he
lp$(14,2)="Code de type ($,!,% facultatif"
1000 help$(14,3)="Variable num(rique = r{elle par d{fau >ME

```



```
t":help$(14,4)=help$(1,1):help$(15,0)=" S(lection et
validation:"+CHR$(241)+"/"+CHR$(243)+"/"+CHR$(240)+"/"+
CHR$(242):help$(15,1)="Corrections:DEL Sortie d'optio
n:COPY":GOSUB 2550:CLS:LOCATE 23,1
1010 PRINT"QUELQUES CONSEILS EN VRAC POUR FINIR":PRINT >MR
* Un menu doit toujours etre appele par un autre
menu, directement ou par";"une page intermediaire te
rminee par un GOTO, pour que le retour automatique";
1020 PRINT"fonctionne correctement.":PRINT * Si v >FA
ous faites pointer une option d'un menu sur un sous
-programme de";"votre cru par un label alphanumerique,
n'oubliez pas, dans l'application, de";
1030 PRINT"remplacer ce label par le No de ligne correc >KM
t.>";" * La sauvegarde n'est effective que lors
que vous retournez au Menu";"Principal (travail ter
min(, ou pause). Si vous avez lieu de craindre une coup
ure";
1040 PRINT"de courant, demandez p(riodiquement @ inte >XG
rompre le travail (dans ";CHR$(34);"RETOUR";CHR$(34);"
):";"vous serez sauve, et vos donnees aussi...";"
* Une astuce pour examiner une page dijon cree (";C
HR$(34);"Modifier Page...";
1050 PRINT CHR$(34);" sans";"ajouter de fenetres, et al >CH
ler tout de suite @ ";CHR$(34);"COULEURS";CHR$(34);" (q
ui affiche l'ensemble";"de la page), puis ";CHR$(34);"R
ETOUR";CHR$(34);". NOTA : les No de page et de rang n
e sont pas alors";
1060 PRINT"precisez dans l'editeur.":PRINT * Fene >TT
tres ";CHR$(34);"Message";CHR$(34);" sans cadre : s
i vous remplissez completement la";"derniere ligne, l
e texte remonte. Validez par ENTER, sortez par COPY et
revenez";
1070 PRINT"dans ";CHR$(34);"TEXTE";CHR$(34);" : vous >DN
constaterez que tout est en ordre. Si alors vous vou
lez";"corriger un caractere, effacez-le d'abord par C
LEAR ou DEL (vous etes en mode";"insertion).";
1080 PRINT * Si une fenetre incrustee doit, @ un >HB
moment donne, laisser la place @ une";"autre plus petit
e ou decalce, vous devez ajouter, au bon endroit du
programme";
1090 PRINT"gn(raf : CLS#f si le fond de la fenetre >KN
environnante est de la meme couleur";"sinon PAPER#f,x
:CLS#f, x (tant le No de PAPER de la fenetre environna
nte."
1100 GOSUB 2550:fes=1:ON BREAK CONT >DL
1110 '----- >XF
1120 nom$="":ON ERROR GOTO 2620:MODE 1:BORDER 0:INK 0,0 >LR
:INK 1,4:INK 2,2:INK 3,15:PAPER#1,1:PEN#1,2:PAPER#3,0:w
f=1:cg=2:cd=39:ch=1:cb=13:cad=5:GOSUB 2570:WINDOW#1,3,3
8,2,12
1130 LOCATE#1,1,2:PEN#1,3:PRINT#1, " ";m1$(1):PRINT#1:PR >WJ
INT#1, " ";m1$(2):PRINT#1:PRINT#1, " ";m1$(3):PRINT#1:PRI
NT#1, " ";m1$(4):PRINT#1:PRINT#1, " ";m1$(5)
1140 PEN#3,3:GOSUB 2510:PEN#3,2:LOCATE#3,1,2:PRINT#3,he >FW
lp$(2,1);:GOSUB 2290:r=0:WHILE r<13:GOSUB 2530:GOSUB 2
290:FOR i=5 TO 9:IF r<com(i) THEN NEXT i
1150 ON i-5 GOSUB 2300,2310,2300,2310:GOSUB 2290:WEND:1 >RG
F I=5 THEN CALL 0
1160 PAPER#2,3:PEN#2,1:wf=2:cg=4:cd=37:ch=15:cb=19:cad= >RQ
```

```
3:GOSUB 2570
1170 PEN#2,2:WINDOW#2,5,36,16,18:CLS#2:PRINT#2, " Nom d >HB
u fichier de donnees ?":GOSUB 1250
1180 CLS#3:IF I>1 THEN PRINT#3,help$(2,1);m$(2);:GOSUB >NB
2360:GOTO 1260 ELSE PRINT#3,m$(0);m$(1);m$(2);
1190 GOSUB 2530:IF r=127 THEN IF POS(#2)=13 THEN PRINT >LJ
CHR$(7); ELSE PRINT#2,CHR$(24);CHR$(8);CHR$(16);CHR$(24
);:r$="":GOTO 1240
1200 IF r=13 THEN IF nom$="" THEN CLS#3:PRINT#3,help$(2 >DH
,1);m$(2);:GOSUB 2360:GOTO 1260 ELSE GOSUB 1250:GOTO 12
60
1210 IF r=224 THEN I=1:GOSUB 2490:GOSUB 2290:GOTO 1140 >XJ
1220 IF r<45 AND (r<48 OR r>57) AND (r<65 OR r>90) THE >BG
N PRINT CHR$(7);:GOTO 1190
1230 IF POS(#2)=21 THEN PRINT CHR$(7);:GOTO 1190 >PN
1240 nom$=LEFT$(nom$,POS(#2)-13)+r$:PRINT#2,CHR$(24);r$ >BN
;CHR$(24);:GOTO 1190
1250 LOCATE#2,13,3:PRINT#2,CHR$(24);SPACE$(8);STRING$(8 >XL
,8);nom$;STRING$(LEN(nom$),8);CHR$(24);:RETURN
1260 IF nom$="" THEN 1130 ELSE PRINT CHR$(21);:file$=no >FX
m$+".NDX":OPENIN file$:INPUT#9,pp,pmx,rmax:PRINT CHR$(
6);:CLOSEIN
1270 ON I GOTO 1310,1370,1420,2260 >ZR
1280 ' >YD
1290 ' Cr(er)/Compl(ter une s(rie >YE
1300 ' >XG
1310 IF pp THEN IF pp<pmx THEN fidx=1:fsav=1:CHAIN"CRE >EU
ECRAN.BAS" ELSE CLS#2:LOCATE#2,9,1:PRINT#2,"S(rie compl
(ete !":LOCATE#2,6,3:PRINT#2,"Appuyez sur une touche";:
CALL &BB06:l=3:GOSUB 2490:IF fdir=1 THEN fdir=0:CLS#1:G
OTO 1130 ELSE GOSUB 2290:GOTO 1140
1320 CLS#2:LOCATE#2,4,1:PRINT#2,nom$;".DAT n'existe pas >HY
";:LOCATE#2,4,2:PRINT#2,"Voulez-vous le cr(er) O/N ?":C
LS#3:PRINT#3,help$(3,0);help$(1,1);m$(2);:LOCATE#2,15,3
:fmp=2:GOSUB 2330
1330 IF rr THEN fsav=0:CHAIN"CREECRAN.BAS" ELSE fan=0:G >QR
OSUB 2490:GOSUB 2290:GOTO 1140
1340 ' >YA
1350 ' Cr(er) le programme d'application >YB
1360 ' >YC
1370 IF pp=pmx THEN CHAIN"CREBASIC.BAS" >KF
1380 GOSUB 2480:IF rr THEN fidx=1:fsav=1:CHAIN"CREECRAN >FW
.BAS" ELSE fan=0:l=5:GOSUB 2490:IF fdir=1 THEN fdir=0:C
LS#1:GOTO 1130 ELSE GOSUB 2290:GOTO 1140
1390 ' >YF
1400 ' Modifier/Mettre @ jour une s(rie >XH
1410 ' >XJ
1420 MODE 1:BORDER 0:INK 0,0:INK 1,15:INK 2,9:INK 3,2:W >HF
INDOW#1,1,13,1,11:PAPER#1,1:WINDOW#2,14,40,1,15:PAPER#2
,3:WINDOW#4,1,40,17,19:PAPER#4,2:PEN#4,0:WINDOW#5,1,12,
14,14:PAPER#5,1:PEN#5,0
1430 CLS#1:PEN#1,2:PRINT#1,CHR$(150);STRING$(11,154);CH >JM
R$(156);:FOR i=1 TO 2:PRINT#1,CHR$(149);STRING$(11,9);C
HR$(149);:NEXT i:PRINT#1,CHR$(151);STRING$(11,154);CHR$(
157);:FOR i=1 TO 6:PRINT#1,CHR$(149);STRING$(11,9);CHR
$(149);:NEXT i
1440 PRINT#1,CHR$(147);STRING$(11,154);CHR$(153);:PEN#1 >XQ
,3:LOCATE#1,2,2:PRINT#1,"MISE A JOUR";:LOCATE#1,5,3:PRI
NT#1,"PAGES";:WINDOW#1,3,11,5,10:CLS#1:FOR i=1 TO 6:PRI
```



```
NT#1,m2$(i);:NEXT i:CLS#5:PRINT#5,nom$;.DAT";
1450 CLS#2:PEN#2,0:PRINT#2,CHR$(150);STRING$(7,154);CHR >XZ
$(158);STRING$(7,154);CHR$(158);STRING$(9,154);CHR$(156
);:FOR i=1 TO 2:PRINT#2,CHR$(149);STRING$(7,9);CHR$(149
);STRING$(7,9);CHR$(149);STRING$(9,9);CHR$(149);:NEXT i
1460 PRINT#2,CHR$(151);STRING$(7,154);CHR$(159);STRING$ >JT
(7,154);CHR$(159);STRING$(9,154);CHR$(157);:FOR i=1 TO
10:PRINT#2,CHR$(149);STRING$(7,9);CHR$(149);STRING$(7,9
);CHR$(149);STRING$(9,9);CHR$(149);:NEXT i
1470 PRINT#2,CHR$(147);STRING$(7,154);CHR$(155);STRING$ >EK
(7,154);CHR$(155);STRING$(9,154);CHR$(153);:PEN#2,1:LOC
ATE#2,2,2:PRINT#2,"NUMERO";STRING$(2,9);"NUMERO";STRING
$(2,9);"NOMBRE DE";
1480 LOCATE#2,2,3:PRINT#2,"DE RANG";CHR$(9);"DE PAGE";C >PY
HR$(9);"FENETRES";:CLS#4:PEN#3,1:GOSUB 2510
1490 PEN#3,3:WINDOW#2,17,35,5,14 >ZY
1500 OPENIN file$:INPUT#9,pp,pmax,rmax:FOR i=1 TO pmax: >MX
INPUT#9,index(i,0),index(i,1):NEXT i:CLOSEIN:DIM tri(99
):FOR i=1 TO pmax:tri(index(i,0))=i:NEXT i:it=1:PEN#2,0
:LOCATE#2,1,it:FOR ir=1 TO MIN(10,rmax):GOSUB 2000:NEXT
ir:ir=1:iup=1
1510 PRINT#3,help$(2,1):PRINT#3,m$(3);m$(2); >MQ
1520 '----- >YA
1530 PAPER#1,2:PEN#1,0:GOSUB 2020:r=0:WHILE r<>13:GOSUB >HW
2530:PAPER#1,1:PEN#1,3:GOSUB 2020:FOR i=5 TO 9:IF r<>c
om(i) THEN NEXT i
1540 ON i-5 GOSUB 2030,2040,2030,2040:PAPER#1,2:PEN#1,0 >UN
:GOSUB 2020:WEND
1550 IF iup=6 THEN GOSUB 1960:GOTO 1120 >FN
1560 PAPER#1,1:PEN#1,3:GOSUB 2020:PAPER#2,2:LOCATE#2,1, >HZ
it:GOSUB 2000:PAPER#2,3:PRINT#4," ";com$(iup);:GOSUB 20
60:GOSUB 2170
1570 IF fan THEN fan=0:LOCATE#2,1,it:GOSUB 2000:CLS#4:G >KQ
OTO 1530
1580 IF iup>2 AND index(tri(ir),1)=0 THEN LOCATE#4,2,2: >EU
PEN#4,3:PRINT#4,"Page vide ! ";:PEN#4,0:PRINT#4,"Appuy
ez sur une touche";:CLEAR INPUT:CALL &BB06:CLS#4:LOCATE
#2,1,it:GOSUB 2000:GOTO 1530
1590 ON iup GOTO 1630,1680,1750,1810,1940 >GK
1600 ' >XK
1610 ' Ajouter >YA
1620 ' >YB
1630 pmax=pmax+1:rmax=rmax+1:IF num=rmax-1 THEN tri(rma >YA
x)=pmax ELSE FOR i=rmax TO num+2 STEP -1:tri(i)=tri(i-1
):NEXT i:tri(i)=pmax
1640 GOSUB 2150:CLS#4:GOTO 1530 >YQ
1650 ' >YE
1660 ' D(placer >YF
1670 ' >YG
1680 r1=num:LOCATE#4,5,2:PRINT#4,"APRES LA PAGE DE RANG >VR
No ";:GOSUB 2060:GOSUB 2170:IF fan THEN GOTO 1570 ELSE
r2=num:stri=tri(r1)
1690 IF r2>r1 THEN FOR i=r1 TO r2-1:tri(i)=tri(i+1):NEX >VZ
T i:tri(i)=stri
1700 IF r1>r2 THEN FOR i=r1 TO r2+2 STEP -1:tri(i)=tri( >EP
i-1):NEXT i:tri(i)=stri
1710 IF r1=r2 THEN PRINT CHR$(7);:LOCATE#2,1,it:GOSUB 2 >XL
000:CLS#4:GOTO 1530 ELSE GOSUB 2150:CLS#4:GOTO 1530
1720 ' >YC
1730 ' Supprimer >YD
1740 ' >YE
1750 LOCATE#4,2,2:PRINT#4,"Confirmez par ";CHR$(34);"o" >RA
;CHR$(34);" ou ";CHR$(34);"0";CHR$(34);:GOSUB 2530:IF r
$<>"0" THEN CLS#4:LOCATE#2,1,it:GOSUB 2000:GOTO 1530
1760 rmax=rmax-1:index(num,0)=0:FOR i=ir TO rmax:tri(i) >FU
=tri(i+1):NEXT i:ir=ir+(ir>1):it=it+(it>1):GOSUB 2150:I
F rmax<10 THEN GOSUB 1980
1770 CLS#4:GOTO 1530 >NN
1780 ' >YJ
1790 ' Modifier >YK
1800 ' >YB
1810 fmax=index(tri(ir),1):IF fmax<4 THEN fmp=4:LOCATE# >AT
4,2,2:PRINT#4,"Ajouter une ou des fenetres O/N ? ";:GOS
UB 2330 ELSE fmax=fmax:fupd=1:GOTO 1890
1820 IF rr THEN LOCATE#4,2,3:PRINT#4,"Combien ? ";:PAP >BD
ER#4,1:rr$=" ":GOSUB 1900 ELSE IF fan THEN 1570 ELSE ff
max=fmax:fupd=1:GOTO 1890
1830 r=0:WHILE r<>13 >PB
1840 GOSUB 2530:IF r=127 THEN IF rr$=" " THEN PRINT CHR >PF
$(7);:GOTO 1880 ELSE rr$=" ":GOSUB 1900:GOTO 1880
1850 IF r=224 THEN fan=1:PAPER#4,2:GOTO 1570 >LW
1860 IF r<>13 THEN rr$=r$:nfen=INSTR("123",r$):IF nfen= >XF
0 OR fmax+nfen>4 THEN PRINT CHR$(7); ELSE GOSUB 1900
1870 IF rr$=" " THEN PRINT CHR$(7);:r=0 >FT
1880 WEND:fmax=fmax+nfen:fupd=2:fidx=1 >LG
1890 pupd=tri(ir):GOSUB 1960:fsav=1:CHAIN"CREECRAN.BAS" >EN
1900 PRINT#4,CHR$(8);rr$;:RETURN >CD
1910 ' >YD
1920 ' Refaire >YE
1930 ' >YF
1940 fupd=3:pupd=tri(ir):GOSUB 1960:fsav=1:fidx=1:CHAIN >ZT
"CREECRAN.BAS"
1950 '----- >YH
1960 OPENOUT file$:WRITE#9,pp,pmax,rmax:FOR i=1 TO pmax >NM
:WRITE#9,index(i,0),index(i,1):NEXT i:CLOSEOUT:old$=nom
$+".BAK";:ERA,old$:ERASE tri:RETURN
1970 '----- >YK
1980 PRINT#2," ";CHR$(149);" ";CHR$(149);" >KR
";:RETURN
1990 '----- >ZB
2000 PRINT#2,USING form$;ir;tri(ir);index(tri(ir),1);:R >LX
ETURN
2010 '----- >XF
2020 LOCATE#1,1,iup:PRINT#1,m2$(iup);:RETURN >QW
2030 iup=iup-6*(iup=1)-1:RETURN >BE
2040 iup=iup+iup*(iup=6)+1:RETURN >DK
2050 '----- >XK
2060 num=tri(ir)*ABS(iup>2)+ir*ABS(iup<3):PAPER#4,1:PRI >YF
NT#4,USING"###";num;:PRINT#4,STRING$(2,8);:PAPER#4,2:RET
URN
2070 '----- >YB
2080 IF ir=1 THEN RETURN >TD
2090 IF it>1 THEN LOCATE#2,1,it:GOSUB 2000:it=it-1:ir=i >GW
r-1:PAPER#2,2:LOCATE#2,1,it:GOSUB 2000:PAPER#2,3:GOSUB
2060:RETURN
2100 LOCATE#2,1,1:ir=ir-1:sir=ir:PAPER#2,2:GOSUB 2000:P >WC
APER#2,3:FOR ir=sir+1 TO sir+9:GOSUB 2000:NEXT ir:ir=si
r:GOSUB 2060:RETURN
```



```
2110 IF ir>=rmax THEN RETURN >XX
2120 IF it<10 THEN LOCATE#2,1,it:GOSUB 2000:it=it+1:ir= >HD
ir+1:PAPER#2,2:LOCATE#2,1,it:GOSUB 2000:PAPER#2,3:GOSUB
2060:RETURN
2130 LOCATE#2,1,1:sir=ir+1:FOR ir=sir-9 TO sir-1:GOSUB >KY
2000:NEXT ir:PAPER#2,2:GOSUB 2000:PAPER#2,3:ir=sir:GOSU
B 2060:RETURN
2140 '----- >XK
2150 sir=ir:i=ir-it+1:FOR j=1 TO rmax:index(tri(j),0)=j >HH
:NEXT j:LOCATE#2,1,1:FOR ir=i TO MIN(i+9,rmax):GOSUB 20
00:NEXT ir:ir=sir:RETURN
2160 '----- >YB
2170 CLEAR INPUT:CALL &BB06:WHILE INKEY(18)=-1 AND INKE >JD
Y(6)=-1
2180 IF INKEY(0)>-1 THEN GOSUB 2080 >BZ
2190 IF INKEY(2)>-1 THEN GOSUB 2110 >BW
2200 IF INKEY(9)>-1 THEN fan=1:RETURN >FD
2210 WEND:RETURN >MZ
2220 '----- >XJ
2230 ' >XK
2240 ' Supprimer un fichier de donn(e)s >YA
2250 ' >YB
2260 CLS#2:LOCATE#2,5,2:PRINT#2,"Confirmez par ";CHR$(3 >UJ
4);"o";CHR$(34);" ou ";CHR$(34);"0";CHR$(34);:CALL &BB0
6:IF INKEY(34)>-1 THEN file$=nom$+".NDX":!ERA,file$:fil
e$=nom$+".DAT":!ERA,file$
2270 l=4:GOSUB 2490:fdir=0:CLS#1:GOTO 1130 >KP
2280 '----- >YE
2290 MOVE 40,404-(32*1),3,1:DRAWR LEN(m1$(1))*16+16,0:D >HD
RAWR 0,-26:DRAWR -(LEN(m1$(1))*16+16),0:DRAWR 0,26:RETU
RN
2300 l=1-5*(l=1)-1:RETURN >UA
2310 l=l+1*(l=5)+1:RETURN >UB
2320 '----- >XK
2330 GOSUB 2530:IF r$="0" THEN rr=1:PRINT#fmp,"OUI"; EL >CD
SE IF r$="N" THEN rr=0:PRINT#fmp,"NON"; ELSE IF r=224 T
HEN rr=0:fan=1:RETURN ELSE PRINT CHR$(7);:GOTO 2330
2340 GOSUB 2530:IF r=13 THEN RETURN ELSE IF r=224 THEN >FE
rr=0:fan=1:RETURN ELSE IF r=127 THEN r$="":FOR i=1 TO 3
:PRINT#fmp,CHR$(8);CHR$(16);:NEXT i:GOTO 2330 ELSE PRIN
T CHR$(7);:GOTO 2340
2350 '----- >YC
2360 fdir=1:WINDOW SWAP 0,1:CLS:PEN 3:LOCATE 1,1:DIR," >NY
*.DAT":xd=1:yd=4:GOSUB 2450
2370 IF nom$=" " THEN nom$="":CLS#2:PRINT#2," >ZY
Aucun fichier de donn(e)s ! ";:LOCATE#2,6,3:PRINT#2,"App
uyez sur une touche";:CALL &BB06:WINDOW SWAP 0,1:fdir=0
:CLS#1:GOSUB 2490:RETURN
2380 LOCATE xd,yd:IF INKEY(8)=0 AND xd=16 THEN xd=1:PRI >WP
NT n$;:GOSUB 2450
2390 IF INKEY(1)=0 AND xd=1 THEN LOCATE 16,yd:IF COPYCH >BJ
R$(#0)>" " THEN xd=16:LOCATE 1,yd:PRINT n$;:GOSUB 2450
2400 IF INKEY(0)=0 AND yd>4 THEN yd=yd-1:PRINT n$;:GOSU >FR
B 2450
2410 IF INKEY(2)=0 THEN LOCATE xd,yd+1:IF COPYCHR$(#0)> >XP
" " THEN LOCATE xd,yd:PRINT n$;:yd=yd+1:GOSUB 2450
2420 IF INKEY(18)=0 OR INKEY(6)=0 THEN WINDOW SWAP 0,1: >GZ
RETURN
2430 IF INKEY(9)>-1 THEN WINDOW SWAP 0,1:fdir=0:CLS#1:G >XQ
OSUB 2490:RETURN
2440 CALL &BB06:GOTO 2380 >TA
2450 LOCATE xd,yd:n$="":FOR i=xd TO xd+11:LOCATE i,yd:r >WD
$=COPYCHR$(#0):n$=n$+r$:NEXT i:LOCATE xd,yd:PAPER 3:PEN
0:PRINT n$;STRING$(12,0);:PAPER 1:PEN 3:nom$=LEFT$(n$,
8):GOSUB 1250:RETURN
2460 '----- >YE
2470 CLS#2:LOCATE#2,5,1:PRINT#2,nom$;".DAT n'existe pas >PF
";:LOCATE#2,6,3:PRINT#2,"Appuyez sur une touche";:CALL
&BB06:nom$="":RETURN
2480 CLS#2:PRINT#2," S(rie incompl)te !";:LOCATE# >PZ
2,2,2:PRINT#2,"Voulez-vous la compl)ter O/N ?";:CLS#3:P
RINT#3,help$(3,0);help$(1,1);m$(2);:LOCATE#2,15,3:fmp=2
:GOSUB 2330:RETURN
2490 WINDOW#2,4,37,15,19:PAPER#2,0:CLS#2:nom$="":RETURN >CT
2500 '----- >XK
2510 wf=3:cg=1:cd=40:ch=21:cb=25:cad=2:GOSUB 2570:WINDO >PC
W#3,2,39,22,24:CLS#3:RETURN
2520 '----- >YB
2530 CLEAR INPUT:r$="":WHILE r$="":r$=UPPER$(INKEY$):WE >ER
ND:r=ASC(r$):RETURN
2540 '----- >YD
2550 LOCATE 30,25:PRINT CHR$(24);"APPUYEZ SUR UNE TOUCH >VK
E";CHR$(24);:CALL &BB06:RETURN
2560 '----- >YF
2570 hb=cb-ch-1:gd=cd-cg-1:WINDOW#wf, cg, cd, ch, cb:CLS#wf >DU
2580 PRINT#wf,CHR$(box(cad,1));STRING$(gd,box(cad,2));C >ZT
HR$(box(cad,3));
2590 FOR i=1 TO hb:PRINT#wf,CHR$(box(cad,4));STRING$(gd >QE
,9);CHR$(box(cad,5));:NEXT i
2600 PRINT#wf,CHR$(box(cad,6));STRING$(gd,box(cad,7));C >LL
HR$(box(cad,8));:RETURN
2610 '----- >YB
2620 PRINT CHR$(6);:IF DERR=146 THEN CLOSEIN:pp=0:pmax= >EA
0:rmax=0:RESUME NEXT
2630 ON ERROR GOTO 0 >NL
2640 '----- >YE
2650 DATA 5,6,16,127,13,240,241,242,243,224,3,67,71,68, >PA
244,245,246,247
2660 DATA "CREER/COMPL)TER UNE SERIE DE PAGES","CREER L >MV
E PROGRAMME D'APPLICATION","MODIFIER/METTRE A JOUR UNE
SERIE","SUPPRIMER UN FICHIER DE DONNEES","QUITTER LE PR
OGRAMME"
2670 DATA "AJOUTER ","DEPLACER ","SUPPRIMER","MODIFIER >NV
","REFAIRE ","RETOUR "
2680 DATA "AJOUTER 1 PAGE APRES LE RANG No ","DEPLACER >KB
LA PAGE DE RANG No ","SUPPRIMER LA PAGE No ","MODIFIER
LA PAGE No ","REFAIRE LA PAGE No "
2690 DATA 135,131,139,133,138,141,140,142 >FR
2700 DATA 150,154,156,149,149,147,154,153 >FH
2710 DATA 250,251,252,253,219,254,220,255 >FU
2720 DATA 229,234,235,236,237,239,248,249 >FX
2730 DATA 221,222,223,224,225,226,227,228 >FQ
2740 DATA 42,42,42,42,42,42,42,42 >YM
2750 DATA 238,238,238,238,238,238,238,238 >FG
2760 DATA 203,203,203,203,203,203,203,203 >FN
2770 DATA 230,230,230,230,230,230,230,230 >FP
```



2780 DATA 231,231,231,231,231,231,231,231 >FZ
 2790 DATA 232,232,232,232,232,232,232,232 >FJ
 2800 DATA 233,233,233,233,233,233,233,233 >FJ
 2810 DATA 128,244,128,247,246,128,245,128 >FN
 2820 DATA 207,207,207,207,207,207,207,207 >FV
 2830 DATA 127,127,127,127,127,127,127,127 >FE
 2840 DATA 2,1,2,4,0,7,7,33,10,15,3,1,2,3,14,4,2,"NOMBRE >WG
 DE PAGES-ECRAN ?"
 2850 DATA 1,1,2,1,0,2,4,37,1,23,8,1,3,2,0,0,0,"PAGE No >KH
 RANG No"
 2860 DATA 1,1,0,4,0,0,5,36,3,5,6,1,3,2,27,2,1,"NOMBRE D >PG
 E FENETRES ?"
 2870 DATA 1,4,2,1,0,3,5,36,5,22,2,2,1,0,0,0,0," >FB
 FENETRE No","","","TYPE :"
 2880 DATA 2,4,2,3,0,5,12,30,7,17,3,3,0,2,0,0,0,"MESSAGE >QK
 ","MENU HORIZONTAL","MENU VERTICAL","SAISIE"
 2890 DATA 3,1,0,1,1,0,6,35,11,11,1,2,2,1,0,0,0,"VOULEZ- >AG
 VOUS UN CADRE O/N ?"
 2900 DATA 3,2,0,1,1,0,6,35,13,14,1,2,2,1,0,0,0,"TITRE O >AJ
 U EN-TETE EN","PREMIERE LIGNE O/N ?"
 2910 DATA 3,2,0,1,1,0,6,35,16,17,1,2,2,1,0,0,0,"REPONSE >VN
 PAR OUI OU NON","ATTENDUE O/N ?"
 2920 DATA 3,1,0,1,1,0,6,35,19,19,1,2,2,1,0,0,0,"LIGNES >RB
 CENTREES O/N ?"
 2930 DATA 3,1,0,1,1,0,6,35,21,21,1,2,2,1,0,0,0,"LIGNES >VT
 JUSTIFIEES O/N ?"
 2940 DATA 5,6,0,2,0,0,1,40,25,25,1,0,2,3,0,0,0,"No ","F >LB
 ENETRE","CADRE","TEXTE","COULEURS","RETOUR"
 2950 DATA 5,1,0,1,0,0,1,40,25,25,1,1,2,3,0,0,0,"POINTE >BH

SUR"
 2960 DATA 7,2,0,2,0,0,12,40,25,25,1,1,2,3,0,0,0,"S/MENU >RV
 ","S/PROGRAMME"
 2970 DATA 7,1,0,4,0,0,12,40,25,25,1,1,2,3,16,1,2,"S/MEN >LW
 U PAGE No "
 2980 DATA 7,1,0,4,0,0,12,40,25,25,1,1,2,3,18,1,10,"S/PR >QA
 OGRAMME CODE "
 2990 DATA 7,1,0,4,0,0,1,12,25,25,1,1,2,3,10,1,2,"LONGUE >BF
 UR:"
 3000 DATA 7,1,0,4,0,0,13,19,25,25,1,1,2,3,6,1,1,"TYPE:" >WR
 3010 DATA 7,1,0,4,0,0,20,40,25,25,1,1,2,3,10,1,6,"VARIA >CG
 BLE:"
 3020 DATA 7,2,2,1,1,1,13,28,11,14,2,1,2,3,0,0,0,"PAGE T >XA
 ERMINEE?"," O/N ?"
 3030 DATA 7,4,2,1,0,1,9,31,10,16,3,1,2,3,0,0,0,"Vous >KR
 devez d'finir","toutes les fenetres","Pressez une tou
 che"
 3040 DATA 7,2,2,1,0,7,10,31,9,17,6,1,2,3,0,0,0,"VOUL >GU
 EZ-VOUS"
 3050 DATA 7,2,0,3,0,0,11,30,12,16,2,1,2,3,0,0,0,"MODIFI >UC
 ER LA PAGE ?","REFAIRE LA PAGE ?"
 3060 DATA 7,3,0,1,1,0,11,30,12,16,2,1,2,3,0,0,0,"EFFACE >QR
 R L'ECRAN AU","CHANGEMENT DE PAGE"," O/N ?"
 3070 DATA 7,3,0,1,1,0,11,30,12,16,2,1,2,3,0,0,0," INTE >AU
 RROMPRE LE"," TRAVAIL O/N?"," "
 3080 DATA 7,2,2,1,0,2,11,30,12,16,6,1,2,3,0,0,0,"SAU >GG
 VEGARDE"
 3090 DATA 1,3,2,1,0,3,4,37,10,15,3,2,3,1,0,0,0,"" >PM
 CHARGEMENT DU FICHIER","Ah si j'avais un acc's direct!"

GESTION BANCAIRE 6128

LA GESTION DE COMPTE BANCAIRE INDISPENSABLE POUR VOTRE CPC

Quelques caractéristiques :

- Gère jusqu'à 10 comptes (banque, épargne, caisse...).
- Codes secrets possibles pour chacun des comptes.
- Saisie des opérations très simple, avec aide en ligne.
- Fonction archivage, vous permettant de stocker année par année vos opérations.
- Fonction TRIER, pour obtenir des listes d'opérations par dates croissantes.
- A l'aide de POINTER, vous pouvez effectuer la liaison avec votre relevé de banque 'officiel'.

- NOMBREUSES POSSIBILITES DE SORTIES :

- Recherche particulière répondant à 1 ou plusieurs critères parmi les suivants :
 - DATES DE DEBUT ET DE FIN
 - MONTANTS MINIMUM ET MAXIMUM
 - UN LIBELLE PARTICULIER
- Liste de chèques pouvant répondre aux critères suivants:
 - DATES DE DEBUT ET DE FIN
 - NUMEROS DE DEBUT ET DE FIN.
- Relevé complet -ENTRE DATES
 - AFFICHAGE SOLDE REEL OU SOLDE OPERATIONS POINTEES.
- Sorties sur ECRAN, IMPRIMANTE ou DISQUETTE.
- UTILITAIRES IMPRIMANTE, ECRAN, COPIE D'ECRAN...
- UTILISE LES 128 Ko DE VOTRE CPC 6128.

MICROLOGIC		SELECTION	MODIFIER	RELEVES	BILAN		
MICROLOGIC		SUPPORT/SORTIE		CRITERE			
SUPPORT / SORTIE							
C	DATE	TYPE	No	ORDRE/OBJET	DEBIT	CREDIT	SOLDE
x	26/05/87	REK. CHEQ		VERSEMENT INITIAL		10000.00	10000.00
x	26/05/87	VIREMENT		SALAIRE MOIS DE MAI		12500.00	22500.00
x	26/05/87	CHEQUE	10000	AMSTRAD CPC 6128	2390.00		19510.00
x	27/05/87	CHEQUE	10001	CONSULTATION MEDECIN	110.00		19400.00
x	28/05/87	PATEN.CAR		PHARMACIE	250.28		19149.74
x	30/05/87	VIREMENT		REMBOURSEMENT S.S.		107.00	19149.74
x	31/05/87	PRELEVEM		TELEPHONE	854.25		18112.29
x	01/06/87	RET. COUC			1000.00		16112.29
x	02/06/87	PRELEVEM		ELECTRICITE	2023.45		14088.84
x	02/06/87	REK. LIQU		RETRAIT EPARGNE		500.00	13588.84
x	02/06/87	CHEQUE	10002	SUPERMARCHÉ	562.47		13026.37
x	03/06/87	RET. CARTE		POINT ARGENT	400.00		12626.37
x	04/06/87	CHEQUE	10003	ASSURANCE AUTO	350.00		12276.37
x	05/06/87	PATEN.CAR		ESSENCE	220.12		12056.25
15 N 06				BANQUE TEST	No 12345678901	Occup. 1 X	Solde: 11237.44

BON DE COMMANDE à retourner à
MICROLOGIC - B.P. 18 - 91211 DRAVEIL CEDEX
 par téléphone:(1) 69.21.61.65 / par minitel (1) 69.24.49.08

Nom : Prénom:

Adresse:

Code postal:..... Ville :

Je commande **GESTION BANCAIRE 6128**, au prix de **265,00 Frs.** (port compris)

Je désire recevoir votre **CATALOGUE GRATUIT** présentant **vos autres produits pour CPC.**

Je choisis de régler par :

CHEQUE C.B. No Expire
 MANDAT CONTRE RBT (+35 Frs) Signature FIN



MODBASE.BAS

```

1 ' >FA
2 'MODBASE.BAS >FB
3 ' >FC
10 SYMBOL AFTER 219 >NJ
20 SYMBOL 219,38,38,38,38,38,38,38,38 >DZ
30 SYMBOL 220,0,0,255,0,0,255,255,0 >AC
40 SYMBOL 221,0,55,72,72,55,72,72,72 >CK
50 SYMBOL 222,0,219,36,36,219,0,0,0 >BE
60 SYMBOL 223,0,236,18,18,236,18,18,18 >DU
70 SYMBOL 224,72,72,48,72,72,48,72,72 >DP
80 SYMBOL 225,18,18,12,18,18,12,18,18 >CF
90 SYMBOL 226,72,72,72,55,72,72,55,0 >CW
100 SYMBOL 227,0,0,0,219,36,36,219,0 >BV
110 SYMBOL 228,18,18,18,236,18,18,236,0 >ED
120 SYMBOL 229,0,127,64,73,86,72,72,80 >DY
130 SYMBOL 234,0,255,0,153,102,0,0,0 >BB
140 SYMBOL 235,0,254,2,146,106,18,18,10 >EL
150 SYMBOL 236,80,72,72,80,80,72,72,80 >DX
160 SYMBOL 237,10,18,18,10,10,18,18,10 >DU
170 SYMBOL 239,80,72,72,86,73,64,127,0 >DE
180 SYMBOL 248,0,0,0,102,153,0,255,0 >BM
190 SYMBOL 249,10,18,18,106,146,2,254,0 >EX
200 SYMBOL 250,0,63,127,96,96,103,100,100 >GK
210 SYMBOL 251,0,255,255,0,0,255,0,0 >BM
220 SYMBOL 252,0,252,254,6,6,230,38,38 >DH
230 SYMBOL 253,100,100,100,100,100,100,100,100 >LF
240 SYMBOL 254,100,100,103,96,96,127,63,0 >GU
250 SYMBOL 255,38,38,230,6,6,254,252,0 >DP
260 DEFINT a-z:del%=CHR$(8)+CHR$(16):DEF FNx=-((1+fcad/2 >AH
)*(i=0)-mg*(i>0):DEF FNy=-2*(i=0)-(1+fcad/2-(ftype<>1)+
ftitre+(i-1)*(1-(ftype<>1)))*(i>0)
270 DIM l$(12),com(8),hpos(9),rep$(10),cx(10),cy(10),lc >MH
(10),fc(10)
280 DATA 13,240,241,242,243,16,127,5 >BP
290 FOR i=1 TO 8:READ com(i):NEXT i:MODE 1:GOTO 2010 >VU
300 DATA 135,131,139,133,138,141,140,142 >EQ
310 DATA 150,154,156,149,149,147,154,153 >ER
320 DATA 250,251,252,253,219,254,220,255 >EC
330 DATA 229,234,235,236,237,239,248,249 >EF
340 DATA 221,222,223,224,225,226,227,228 >EZ
350 DATA 42,42,42,42,42,42,42,42 >XE
360 DATA 238,238,238,238,238,238,238,238 >FG
370 DATA 203,203,203,203,203,203,203,203 >EX
380 DATA 230,230,230,230,230,230,230,230 >EY
390 DATA 231,231,231,231,231,231,231,231 >EH
400 DATA 232,232,232,232,232,232,232,232 >EH
410 DATA 233,233,233,233,233,233,233,233 >ET
420 DATA 128,244,128,247,246,128,245,128 >EX
430 DATA 207,207,207,207,207,207,207,207 >ED
440 DATA 127,127,127,127,127,127,127,127 >EN
450 '**** AFFICHAGE DES FENETRES **** >RK
460 ERASE l$:DIM l$(12):READ f,11,ftype,fcad,ftitre,fre >WB
p,fcentre,cad,cg,cd,ch,cb,mg,a,b,c:IF ftitre=3 THEN REA
D l$(0)
470 IF ftype=2 THEN READ sp:ERASE hpos:DIM hpos(9) >VA
480 IF ftype=4 THEN ERASE rep$,cx,cy,lc,fc:DIM rep$(11) >FK
,cx(11),cy(11),lc(11),fc(11):FOR i=1 TO 11:READ cx(i),c
y(i),lc(i),fc(i),l$(i):NEXT i:GOTO 500
490 IF 11>0 THEN FOR i=1 TO 11:READ l$(i):NEXT i >PJ
500 PAPER #f,a:PEN #f,b:WINDOW #f,cg,cd,ch,cb:CLS #f:IF >BL
fcad=2 THEN GOSUB 530
510 GOSUB 760:RETURN >QC
520 '.... DESSIN DU CADRE .... >RH
530 bh=cb-ch-1:gd=cd-cg-1 >UH
540 IF ftitre=0 THEN hb=bh:LOCATE #f,1,1:GOSUB 560:RETU >DK
RN
550 IF ftitre=3 THEN hb=1:LOCATE #f,1,1:GOSUB 560:hb=bh >BV
-3:GOSUB 560:RETURN
560 ON cad GOSUB 600,610,620,630,640,650,660,670,680,69 >TD
0,700,710,720,730,740:READ c1,c2,c3,c4,c5,c6,c7,c8
570 PRINT #f,CHR$(c1);STRING$(gd,c2);CHR$(c3); >QY
580 FOR i=1 TO hb:PRINT #f,CHR$(c4);STRING$(gd,CHR$(9)) >XK
;CHR$(c5);:NEXT i
590 PRINT #f,CHR$(c6);STRING$(gd,c7);CHR$(c8);:RETURN >AD
600 RESTORE 300:RETURN >TX
610 RESTORE 310:RETURN >TZ
620 RESTORE 320:RETURN >TB
630 RESTORE 330:RETURN >TD
640 RESTORE 340:RETURN >TF
650 RESTORE 350:RETURN >TH
660 RESTORE 360:RETURN >TK
670 RESTORE 370:RETURN >TM
680 RESTORE 380:RETURN >TP
690 RESTORE 390:RETURN >TR
700 RESTORE 400:RETURN >TZ
710 RESTORE 410:RETURN >TB
720 RESTORE 420:RETURN >TD
730 RESTORE 430:RETURN >TF
740 RESTORE 440:RETURN >TH
750 '.... ECRITURE DU TEXTE .... >TC
760 PEN #f,c:ON ftype GOSUB 800,910,800,800:RETURN >UZ
770 ' >TE
780 'message,menu vertical,saisie >TF
790 ' >TG
800 IF ftitre=3 THEN i=0:GOSUB 870 >BC
810 IF 11=0 THEN RETURN >RH
820 IF fcentre=1 THEN FOR i=1 TO 11:GOSUB 870:NEXT i:GO >FQ
TO 840
830 FOR i=1 TO 11:LOCATE #f,FNx,FNy:PRINT #f,l$(i);:NEX >DG
T i
840 IF frep=1 THEN GOSUB 1020 >WN
850 IF ftype=3 THEN GOSUB 1090 >XU
860 RETURN >ZK
870 xx=1+(cd-cg+1-LEN(l$(i)))\2:LOCATE #f,xx,FNy:PRINT >YW
#f,l$(i);:RETURN
910 RETURN >ZF
1020 RETURN >EG
1090 RETURN >FD
1180 RETURN >FD
1440 '----- >YB
1450 iop=iop-11*(iop=1)-1:RETURN >CY
1460 iop=iop+iop*(iop=11)+1:RETURN >EZ
1470 PRINT CHR$(7); >NE
1480 RETURN >FG

```



```

1490 ' >YG
1500 ' Interrogation Clavier >XJ
1510 ' >XK
1520 CLEAR INPUT:r$="":WHILE r$="":r$=UPPER$(INKEY$):WE >EP
ND:r=ASC(r$):RETURN
1530 ' ----- >YB

```

MODMENUH.BAS

```

880 ' >TG
890 ' MODMENUH. BAS >TH
900 ' >RK
910 iop=0:GOSUB 1460:hpos(1)=mg:lign#=1$(1) >MD
920 FOR j=2 TO 11:lign#=lign#+SPACE$(sp):hpos(j)=LEN(li >RP
gn$)+mg:lign#=lign#+1$(j):NEXT j:PEN#f,c:LOCATE#f,mg,1:
PRINT#f,lign$;
930 PEN#f,b:GOSUB 980:PEN#f,c >YR
940 r=0:WHILE r<>13:GOSUB 1520:GOSUB 980 >HC
950 FOR j=1 TO 5:IF r<>com(j) THEN NEXT j >GW
960 ON j GOSUB 1480,1450,1460,1450,1460,1470 >JZ
970 PEN#f,b:GOSUB 980:PEN#f,c:WEND:choix=iop:RETURN >AR
980 LOCATE#f,hpos(iop),1:PRINT#f,l$(iop);:RETURN >WZ

```

MODMENUV.BAS

```

1060 ' >XK
1070 ' MODMENUV. BAS >YA
1080 ' >YB
1090 iop=1:gxv=16*(cg+mg)-38:gyv=16*(25-ch-fcad/2-ftitr >HT
e)+36
1100 GOSUB 1140:r=0:WHILE r<>13:GOSUB 1520:GOSUB 1140 >XC
1110 FOR j=1 TO 5:IF r<>com(j) THEN NEXT j >HP
1120 ON j GOSUB 1480,1450,1460,1450,1460,1470 >KA
1130 GOSUB 1140:WEND:choix=iop:RETURN >HC
1140 long=12+LEN(l$(iop))*16:MOVE gxv,gyv-32*iop,c,1:DR >JH
AWR long,0:DRAWR 0,-26:DRAWR -long,0:DRAWR 0,26:RETURN■

```

MODREP.BAS

```

990 ' >TJ
1000 ' MODREP. BAS >XD
1010 ' >XE
1020 GOSUB 1520:IF r=79 THEN PRINT#f,"OUI";:rr=r ELSE I >PR
F r=78 THEN PRINT#f,"NON";:rr=r ELSE PRINT CHR$(7);:GOT
O 1020
1030 GOSUB 1520:IF r=13 THEN RETURN ELSE IF r=127 THEN >QK
r$="":FOR i=1 TO 3:PRINT#f,del$;:NEXT i:GOTO 1020
1040 PRINT CHR$(7);:GOTO 1030 >XH
1050 ' rr=79 : OUI rr=78 : NON >XJ

```

MODSAISI.BAS



```

1150 ' >XK
1160 ' MODSAISI. BAS >YA
1170 ' >YB
1180 WINDOW SWAP 0,f:CURSOR 1,1:i=1 >DR
1190 PEN b:LOCATE cx(i),cy(i):PRINT STRING$(lc(i),CHR$( >BM
210));:IF rep$(i)=" 0" THEN rep$(i)=" " ELSE PEN c:LOCAT
E cx(i),cy(i):PRINT MID$(rep$(i),1-(LEFT$(rep$(i),1)="
"));
1200 PEN b:LOCATE cx(i),cy(i):flag=0 >EK
1210 r$=INKEY$:IF r$="" THEN 1210 ELSE r=ASC(r$) >PA
1220 FOR j=1 TO 8:IF r<>com(j) THEN NEXT j >HV
1230 ON j GOSUB 1330,1360,1370,1380,1390,1400,1420,1430 >EU
:IF j<9 THEN ON flag GOTO 1250,1260:GOTO 1210
1240 fc=fc(i)+1:ON fc GOSUB 1270,1290:GOTO 1210 >NY
1250 GOTO 1190 >MG
1260 CURSOR 0:WINDOW SWAP 0,f:RETURN >FF
1270 IF r>31 AND r<126 THEN GOSUB 1310 >CA
1280 RETURN >FE
1290 IF r>47 AND r<58 THEN GOSUB 1310 >CZ
1300 RETURN >EH
1310 IF LEN(rep$(i))=lc(i) THEN PRINT CHR$(7);:RETURN >YM
1320 x=POS(#0):rep$(i)=LEFT$(rep$(i),x-cx(i))+r$+MID$(r >KD
ep$(i),x-cx(i)+1):PEN b:LOCATE cx(i),cy(i):PRINT rep$(i
);:LOCATE x+1,cy(i):RETURN
1330 IF 11=1 THEN IF rep$(i)=" " THEN PRINT CHR$(7);:LOC >GD
ATE cx(i),cy(i):RETURN ELSE PEN c:LOCATE cx(i),cy(i):PR
INT rep$(i);:flag=2:RETURN
1340 PEN c:LOCATE cx(i),cy(i):PRINT MID$(rep$(i),1-(LEF >VA
T$(rep$(i),1)=" "));:flag=1:FOR i=1 TO 11:IF rep$(i)=" "
THEN PRINT CHR$(7);:LOCATE cx(i),cy(i):RETURN ELSE NEX
T i
1350 flag=2:RETURN >NG
1360 IF 11=1 THEN RETURN ELSE PEN c:LOCATE cx(i),cy(i): >NP
PRINT MID$(rep$(i),1-(LEFT$(rep$(i),1)=" "));:i=-11*(i=
1)-(i-1)*(i>1):flag=1:RETURN
1370 IF 11=1 THEN RETURN ELSE PEN c:LOCATE cx(i),cy(i): >MH
PRINT MID$(rep$(i),1-(LEFT$(rep$(i),1)=" "));:i=-(i=11)
-(i+1)*(i<11):flag=1:RETURN
1380 IF POS(#0)=cx(i) THEN PRINT CHR$(7);:RETURN ELSE P >BY
RINT CHR$(8);:RETURN
1390 IF POS(#0)=LEN(rep$(i))+cx(i) OR VPOS(#0)=cy(i)+1 >CV
THEN PRINT CHR$(7);:RETURN ELSE PRINT CHR$(9);:RETURN
1400 IF POS(#0)=cx(i)+lc(i) OR VPOS(#0)=cy(i)+1 THEN PR >ZF
INT CHR$(7);:RETURN
1410 PRINT CHR$(16);:xx=POS(#0):rep$(i)=LEFT$(rep$(i),x >TM
x-cx(i))+MID$(rep$(i),xx-cx(i)+2):PEN b:LOCATE cx(i),cy
(i):PRINT rep$(i);CHR$(210);:LOCATE xx,cy(i):RETURN
1420 IF POS(#0)=cx(i) THEN PRINT CHR$(7);:RETURN ELSE P >TV
RINT CHR$(8);:GOSUB 1410:RETURN
1430 rep$(i)="":LOCATE cx(i),cy(i):PRINT STRING$(lc(i), >EE
CHR$(210));:LOCATE cx(i),cy(i):RETURN■

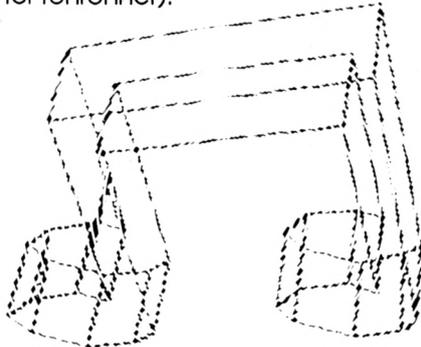
```

MUSIC MACHINE

Ayant rencontré de grandes difficultés pour faire fonctionner la Music Machine (impossibilité de saisir un échantillon (pardon de "sampler" un son), impossibilité de faire exécuter le morceau pré-programmé sur mon clavier, impossibilité de jouer à partir de mon clavier les "samples" préenregistrés ???), ceci n'étant pas faute d'avoir compris comment fonctionnait l'interface, ni d'avoir tout essayé, je vous livre quand même mes impressions de musicien amateur sur la Music Machine.

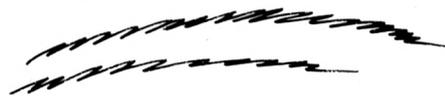
N'ayant travaillé qu'au casque, je dois avouer qu'il manque quelque chose de très utile : un potentiomètre de réglage de volume d'écoute. Je crois avoir perdu environ 10 dB à chaque oreille. Ou alors il aurait fallu fixer d'usine un niveau sonore un peu moins... DESTROY ! Deuxième point fort désagréable : le problème de ronflement résiduel. Il faut s'amuser à déplacer la Music Machine par rapport à l'écran, aux câbles d'alimentation pour trouver un emplacement plus ou moins acrobatique pour l'interface afin que celle-ci ne capte trop de parasites

(le seul fait d'approcher la main provoque un ronflement terrible. Une vraie chatte, on peut passer des heures à la caresser et l'écouter ronronner).



Bon, le ton est donné, vous vous en doutez : critique sévère. Un bon point quand même pour la présentation. Les écrans sont bien faits, le passage d'un écran à l'autre est psychédélique. Mais gros problème : Peut-on vraiment parler de Music Machine quand on voit les limitations données par le manuel ? Tout d'abord en éditeur de mesure : On ne peut avoir que deux notes alignées verticalement tout en sachant que la Music Machine ne pourra en jouer qu'une.

Si vous voulez entendre l'autre, il vous faudra connecter un instrument MIDI qui, lui, pourra jouer les deux notes simultanément (même avec un instrument MIDI connecté, il faut avouer qu'un maximum de deux notes simultanées c'est un peu léger pour jouer une partition). La plage de programmation des notes sur la portée est de quatre octaves...



La partie éditeur de rythmes vous permettra quant à elle de programmer des arrangements de vos "samples" en mémoire. Là, vous pourrez superposer au maximum trois "samples" parmi un choix de huit mémorisables. C'est un peu mieux que l'éditeur de mesures. Grâce à cet éditeur, vous pourrez mémoriser votre partition rythmique. Il faut savoir que vous pouvez enregistrer un maximum de 1.22 secondes de son "simplé" en tout. Comme vous pouvez mémoriser huit "samples" au maximum, cela donne une moyenne de 0.15 seconde par "sample". Quant à la qualité des sons de percussion préenregistrés, on peut dire qu'elle est assez bonne. Un bon point aussi pour l'éditeur de "sample" qui possède en plus une fonction REVERSE qui peut s'avérer intéressante.

Il faut savoir aussi que la Music Machine ne pourra pas restituer à la fois la musique et les percussions simultanément. Il faudra choisir l'une ou l'autre. Par contre, il vous sera possible d'envoyer la partition musicale vers un synthétiseur MIDI, de faire jouer la partie percussion en synchronisation par la Music Machine. On pourra même, si on le désire, superposer à cela un accompagnement en le jouant directement sur les touches du synthétiseur.

Voici un petit résumé des choses que vous pourrez faire en association avec un instrument MIDI :



- Jouer un des "sample" en mémoire dans la Music Machine à partir d'un clavier ou tout autre instrument MIDI.

- Envoyer au synthétiseur MIDI la partition musicale programmée et superposer en le jouant sur votre clavier un des "sample" en mémoire ou bien une des voies que votre clavier MIDI peut générer.

- En utilisant le microphone, vous pourrez "sampler" n'importe lequel des sons générés par votre synthétiseur MIDI et l'éditer en utilisant l'éditeur de "sample".

- Si vous connectez un clavier MIDI sur votre Music Machine, votre clavier sera en mesure de restituer jusqu'à deux notes simultanément (alors que la Music Machine seule ne peut en reproduire qu'une en mode exécution d'une partition musicale).

Il est clairement spécifié dans la notice que la Music Machine est tout ce qu'on veut sauf un séquenceur du fait que le logiciel est trop gourmand de mémoire, et que si l'on est assez riche pour posséder un sé-



quenceur MIDI, on peut l'utiliser comme matériel principal pour piloter à la fois un clavier externe, un synthétiseur de sons de percussion externe et la Music Machine pour la génération de sons "sampler".

En résumé, je pense que le principal intérêt de la Music Machine est de pouvoir capturer un son - voire le "SAMPLER" comme ils disent

autre Manche - et de le restituer en association avec un clavier MIDI sur toute la gamme accessible par celui-ci. Donc grâce à la Music Machine, vous pourrez faire ce que ne peut faire votre synthétiseur c'est-à-dire jouer des sons "SAMPLES".

Thierry Mangion

Participez à CPC Infos

LE PROGRAMMEUR

Nom : _____ Prénom : _____

Adresse complète : _____

Tél. : _____ Age : _____

Attestation sur l'honneur

Je soussigné _____

déclare être l'auteur du programme ci-joint et ne l'avoir jamais proposé à une autre revue.

Le _____ A _____

* Ne pouvant être tenu pour responsable de l'acheminement des paquets postaux, nous conseillons à notre aimable clientèle de choisir l'envoi en recommandé.

LE PROGRAMME

Nom : _____

Catégorie Jeu Utilitaire Educatif

Taille : _____

Périphériques utilisées : _____

Support Cassette Disquette

Comptabilité (testée) avec : _____

464 664 6128 PCW 8256

Signature _____



Remplissez soigneusement ce coupon et joignez-le à votre programme, sur cassette ou disquette.

Envoyez le tout à CPC Infos - La Haie de Pan - 35170 BRUZ

RECTIFICATIF

JAQUETTE POUR 464

Aussi incroyable que cela puisse paraître, le célèbre All Gator est un Inconditionnel du CPC 464. Cela est peut-être dû au fait qu'il a été l'un des tout premier en

France à posséder un CPC mais surtout parce que derrière sa machine, il y a un drive 5,25 pouces qui ne peut s'adapter sur un 6128.

Ceci étant dit, vous comprendrez pourquoi sa déception fut grande lorsque parut l'utilitaire Jaquette de Mr Pierre Vacher (CPC Infos de février 91). Quoi ! Un tel outil réservé aux 6128 cela n'était pas normal.

Le grand All, s'est penché sur le problème (sans tomber), et en trois coups de cuillère à pot, il l'a résolu.

Trêve de plaisanterie, voici les lignes du programme qu'il faut modifier pour que ce fameux utilitaire devienne compatible tous CPC.

```
63 RESTORE 67:FOR H=150 TO 165:READ A#:POKE H,VAL("&"A#):NEXT
67 DATA DD,6E,00,DD,66,02,CD,75,BB,CD,60,BB,32,C8,00,C9

650 CALL 150,X,Y:LM=PEEK(200):IF CHR$(LM)<>"." THEN 690
660 NM=NM+1:FOR H=0 TO 15:Z=(X-B)+H:CALL 150,Z,Y:LM=PEEK(200):NUM$(NM,F)=NUM$(
,F)+CHR$(LM):NEXT:LOCATE X-B,Y:PRINT CHR$(24)+NM$(N
M,F)+CHR$(24)
665 nm$(nm,f)=LEFT$(nm$(nm,f),8)+" . "+MID$(nm$(nm,f),10,3)+" = "+RIGHT$(n
$(nm,f),2)+" Ko"
670 y=y+1:IF rf<y THEN rf=y
680 GOTO 650
690 Y=4:X=X+20:IF X>71 THEN 700 ELSE 650
700 FOR H=1 TO 3:CALL 150,H,rf+1:LM=PEEK(200):RST$(F)=RST$(F)+CHR$(LM):NEXT:RE
RN
```

Routine assembleur remplaçant la fonction COPYCHR\$ inexistante sur 464.

THE ENERGY OF WOK

Dans ce listing, une erreur rendait impossible l'accès aux niveaux 2, 3 et 4. Voici les quelques lignes à modifier (les instructions à rajouter sont en gras).

```
270 ... :POKE &9048,0:GOTO 250
280 ... :POKE &904B,0:GOTO 250
290 ... :POKE &904D,40:GOTO 250
```

Il faut rajouter une ligne :

```
55 POKE &9048,&C8:POKE &904B,&C8:POKE
&904E,&C8:FOR A=&6180 TO &6180+255:POKE
A+9945,PEEK(A):NEXT
```

Si le personnage devient invisible à un niveau, il faut éteindre l'ordinateur, recharger le jeu et passer au tableau suivant en modifiant la ligne 240:POKE &901A,tableau:CALL &9023:GOTO 260.

OXYSUDIS

La présentation d'Oxysudis est certes, facultative mais encore faudrait-il avoir le listing correspondant si l'on choisit l'option «écran de présentation». Ce listing était absent du numéro précédent, le voici donc :

```

10 MODE 0:PAPER 0:BORDER 0:INK 0,0:INK 1,26:INK 2,24:IN >FH
K 3,15:INK 4,6:INK 5,11:INK 6,10:INK 7,3:INK 8,23:INK 9
,13:INK 10,18:INK 11,9:INK 12,2:INK 13,1:INK 14,24,6:IN
K 15,9:CLS:DEFINT a-z
20 MODE 2:PEN 1:LOCATE 17,1:PRINT "/CREATION DE LA PAGE >X
DE PRESENTATION D'OXYSUDIS/"
30 FOR t=1 TO 5500:NEXT t:MODE 0 >YV
40 FOR i=1 TO 800:x=1+INT(RND*639):y=1+INT(RND*399):PLO >UA
T x,y,1:NEXT i
50 RESTORE 60:FOR w=1 TO 4:READ x,y,r,cpl,cp2,c1(1),c1( >BH
2),c1(3):st=1:GOSUB 70:NEXT w:GOTO 190
60 DATA 370,150,110,290,350,1,2,3,170,300,70,130,160,2, >ZG
3,4,470,350,40,440,460,8,5,12,306,350,20,296,306,13,6,9

70 DEG:ORIGIN 0,0:ORIGIN x,y:FOR i=0 TO 360:PLOT COS(i) >UZ
*r,SIN(i)*r,15:NEXT i:ORIGIN 0,0
80 a=(x-r)+4:b=y:h=a:g=b >UE
90 IF TEST(h,g)<>15 THEN PLOT h,g,c1(st) >HP
100 g=g-2:IF TEST(h,g)<>15 THEN PLOT h,g,c1(st):GOTO 10 >AW
0
110 g=b >PJ
120 g=g+2:IF TEST(h,g)<>15 THEN PLOT h,g,c1(st):GOTO 12 >AY
0
130 g=b >QA
140 h=h+2 >CJ
150 IF h=cpl THEN cpl=0:st=st+1:GOTO 170 >HT
160 IF h=cp2 THEN cp2=0:st=st+1 >ZK
170 IF TEST(h,g)<>15 THEN GOTO 90 >AA
180 RETURN >ZE
190 INK 15,0 >VA
200 SAVE"OXYSUDIS.PRE",b,&C000,&4000 >ET
210 END >RG

```

**BONNES VACANCES
DE LA PART DE
TOUTE
L'ÉQUIPE DE
GIGA CPC INFOS !**

TRUCS EN VRAC



Christian CHAINTREAU

Voici un mini listing permettant (encore UN) d'obtenir sur l'imprimante, le catalogue ou le Directory complet de tous les users (y compris ceux situés entre 16 et 255). Et ce d'une façon très simple, sans pokes et en moins de 10 lignes.

```

10 " Impression du CATALOGUE ou Directory COMPLET d'une disquette de tous les
USERS ( meme de 16 à 255 )
20 MODE 2:a=0:INPUT " Nom ou numéro de la disquette ";nomdisc$:INPUT"Face ";face$

30 a=a+1:INPUT"Entrer le numéro d'user à editer ";u$(a):u(a)=VAL(u$(a)):IF u$(a)
=" " THEN 40 ELSE 30
40 nu=a-1:FOR a=1 TO nu:CLS:POKE &A701,u(a):CAT:GOSUB 60:PRINT"Remplacer le papier
r pour imprimer le prochain CAT à la suite ou changer de page PUIS UNE TOUCHÉ
":CALL MBB16:NEXT
50 CLS:PRINT"Impression terminée. Un autre traitement <DN>":CALL MBB06:IF INKEY(
34)=0 THEN 20 ELSE 60
60 PRINT#B,CHR$(15):PRINT#B," DISQUE ";nomdisc$;" FACE ";face$:ph=0:pv=1
70 ph=ph+1:IF ph=80 THEN 80 ELSE 90
80 PRINT#B,CHR$(13):pv=pv+1:IF pv=255 THEN RETURN ELSE ph=0:GOTO 70
90 LOCATE ph,pv:a$=COPYCHR$(#0):PRINT#B,a$;:GOTO 70

```

DISQUE WORDSTAR FACE 1

Drive A: user 0

```

ADR .DAT* 1K MAILMRGE.OVR* 8K TOT .MAC* 1K WSM56S .OVR* 31K
ASCTAT .MAC* 9K MTELFILF.BAK* 5K WS .COM* 16K WSDVLY1 .OVR* 34K
LT . * 2K MTELFILF.MAC* 5K WS .INS* 59K

```

2K free

DISQUE FACE

Drive A: user229

```

ZAP .BAS 11K ZAP .BIN 6K

```

2K free

TRUCS EN VRAC



Bernard LECERF

Prenez un CPC 464 (le mien est pourvu d'un lecteur de disquette, ce qui explique que je ne mentionne pas certains résultats affichés), et armez-vous de patience.

Essayez les manipulations suivantes :

A/ en MODE DIRECT :

- CALL 0
- PRINT FRE(" ")
- DIM AS(2000)
- PRINT FRE(" ")
- FOR I=1 TO 2000:AS(I)="0":NEXT
- PRINT FRE(" ")...

cela peut durer un certain temps !

Notez la place mémoire affichée par les PRINT FRE.

Recommencez A/ en chargeant la valeur de AS(I): "01", puis "12",... Arrêtez à "0123456789".

Les 2 premiers FRE doivent afficher à chaque fois le même nombre. Par contre, le 3ème diminue de 2000 octets chaque fois que la chaîne s'allonge d'un caractère. A 20 ou 22 caractères, c'est le message «STRING SPACE FULL» et le plantage.

B/ en MODE PROGRAMME:

- ```
10 PRINT FRE(" ")
20 DIM AS(2000)
30 PRINT FRE(" ")
40 FOR I=1 TO 2000:AS(I)="0":NEXT
50 PRINT FRE(" ")
```

Sauvegardez, lancez et notez les résultats. Réinitialisez, rechargez, et modifiez dans la ligne 40 la valeur de AS(I), comme dans la partie A/.

Surprise, quand vous allongez la chaîne d'un caractère, tous les résultats augmentent d'un octet. C'est normal pour les 2 premiers, puisque le programme est plus long d'un octet, mais pour le 3ème... un octet de plus au lieu de 2000 !

Voici l'explication :

Recommencez la partie A/ et la partie B/, mais à chaque fois, tapez EN MODE DIRECT après affichage des résultats.

Attention, AD, L et AC ne sont pas des variables : il faut taper en clair les nombres. Toute introduction de variables perturbe les résultats.

- PRINT AS(1) (recherche de l'adresse de la variable à l'aide du pointeur de variables)

- PRINT PEEK(AD)

J'appelle la réponse L. (Longueur de la chaîne)

- PRINT PEEK(AD+1)+256\*PEEK(AD+2)

J'appelle la réponse AC (Adresse de la chaîne)

- FOR I=AC TO AC+L:PRINT CHR\$(PEEK(I));:NEXT

La réponse est la chaîne.

Recommencez avec PRINT AS(2000). Avec la procédure A/ les adresses obtenues (AC) sont différentes, cela semble évident.

Refaites ces deux opérations avec la procédure B/. Pour AS(1) et AS(2000), vous devez obtenir la même valeur pour AC !

Eh bien, en MODE DIRECT, tapez à chaque essai de la procédure B/ :

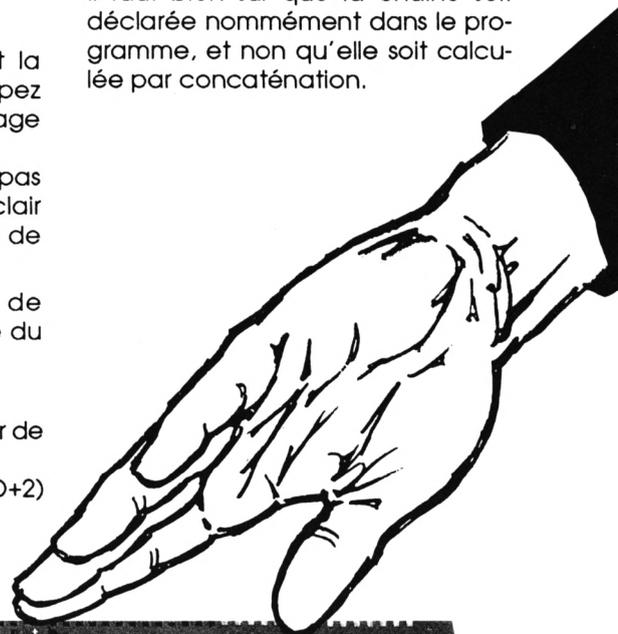
- FOR I=AC-13 TO AC+L:PRINT I,PEEK(I):NEXT

Vous devez obtenir les codes suivants :

1 3 7 0 225 40 13 5 0 233 41 239 34 48... et enfin 34. A l'adresse AC se trouve le début de la chaîne.

Pour toute valeur de I dans PRINT AS(I), et quelle que soit la longueur de la chaîne, l'adresse est la même : c'est celle où se trouve la chaîne dans le programme.

Et oui, le CPC va tout simplement chercher la chaîne à sa place DANS LE PROGRAMME, évitant d'encombrer l'espace réservé aux variables. Il faut bien sûr que la chaîne soit déclarée nommément dans le programme, et non qu'elle soit calculée par concaténation.







# CHARLIE BONDSOUND

Claude LE MOULLEC

Valable pour CPC 464 - 664 - 6128

Une balle sautillante doit capturer des couleurs (cubes colorés) afin d'allumer des projecteurs (cubes grillagés).

Lorsque tous les projecteurs d'une même couleur sont allumés, un disque apparait dans la partie contraire de celle où se trouve la balle à ce moment. Il faut prendre ce disque et le déposer sur une note... la musique commence sur une voie. Le morceau musical utilisant 3 voies, il y a donc, pour chaque tableau, 3 disques à capturer. Plus quelques surprises (passages secrets, monstres,

etc ...) que le joueur découvrira au fur et à mesure de sa progression.

Ce jeu comprend 2 fichiers.

**CHARLIE** : fichier basic, corps principal du jeu.

**CHALIBIN** : fichier binaire (sprites, tableaux et routines LM) ;

**DATACHAR** est un fichier basic de data qui, une fois lancé, donne CHALIBIN.

## CHARLIE.BAS

```

10 REM :
20 REM :
30 REM : CHARLIE BOND SON :
40 REM : by :
50 REM :
60 REM : CLAUDE LE MOULLEC :
70 REM : 83 RUE CURIE :
80 REM : 22420 PLOUARET :
90 REM : TEL 96 38 94 24 :
100 REM :
110 MEMORY &8800:LOAD "!CHALIBIN",&8820
120 REM :
130 REM :
140 REM : VARIABLES DE BASE :
150 REM :
160 REM :
170 DEFINT a-z:MODE 0:BORDER 1
180 RESTORE 190:FOR h=0 TO 15:READ a:INK h,a:NEXT
190 DATA 0,10,26,9,24,12,18,7,8,16,1,12,3,15,3,6

```

```

>WQ
>DJ
>UE
>AE
>EB
>VA
>QX
>TG
>PB
>XQ
>GU
>XT
>KE
>WX
>KG
>XX
>YB
>TB
>MZ
200 DIM sp(30):FOR h=0 TO 25:sp(h+1)=&8820+(100*h):NEXT >YR
210 DIM PY(20):RESTORE 230:FOR h=1 TO 9:READ a$ >NA
220 PY(h)=VAL("&"a$):PY(h+8)=VAL("&"a$):NEXT >PV
230 DATA BFFB,E09B,C18B,E22B,C31B,E3BB,C4AB,E54B,C63B >XH
240 DEF FN PO(X,Y)=PY(Y)+(X*5) >YR
250 DEF FN PK(X,Y)=PEEK(&9E5F+(Y*16)+X) >GJ
260 DEF FN LT(X,Y)=&BFFE+(Y-1)*80+(X-1)*2 >HK
270 ENT 4,10,-1,3:ENV 4,10,-1,3:ENT 3,20,3,5:ENV 14,1,5 >PE
,1,10,1,4,5,-3,1
280 ENV 1,15,-1,1:ENT 2,10,-1,0,10,1,1:ENV 2,1,-1,4:ENV >HR
10,15,-1,1
290 ENV 15,1,5,1,10,1,4,5,-3,1:ENT 14,40,-10,1:ENT 15,5 >MM
,-20,1,60,10,1
300 GOSUB 280 >QC
310 VIE=4:SC=0:REC=0:TA=1 >VG
320 A$=" SCORE LIFES LEVEL HIGHT":ZL=1:YL >QZ
=24:GOSUB 350
330 A$=" 00000 000 000 00000":ZL=1:YL >JY
=25:GOSUB 350
340 GOTO 430 >ZA
350 A$=UPPER$(A$):FOR T=1 TO LEN(A$):PS=(ASC(MID$(A$,T, >HX
1)))-48
360 IF PS<0 THEN PS=43 >QL

```

```

370 CALL &8ECC, FN Lt(ZL, YL)+(T*2), &8EEC+(PS*16):NEXT:RE >GD
TURN
380 REM :::::::::::::::::::: >XB
390 REM : : >LC
400 REM : BRANCHEMENT TAB : >VR
410 REM : : >KF
420 REM :::::::::::::::::::: >XW
430 ON ta GOTO 440, 450, 460, 470, 480, 490, 500, 510, 520, 530, >JV
540, 550, 560
440 RESTORE 2710:CALL &A2A8, &91B0:GOTO 570 >JE
450 RESTORE 2720:CALL &A2A8, &92C0:GOTO 570 >JJ
460 RESTORE 2730:CALL &A2A8, &93D0:GOTO 570 >JN
470 RESTORE 2740:CALL &A2A8, &94E0:GOTO 570 >JT
480 RESTORE 2750:CALL &A2A8, &95F0:GOTO 570 >JX
490 RESTORE 2760:CALL &A2A8, &9700:GOTO 570 >JC
500 RESTORE 2770:CALL &A2A8, &9810:GOTO 570 >JX
510 RESTORE 2780:CALL &A2A8, &9920:GOTO 570 >JB
520 RESTORE 2790:CALL &A2A8, &9A30:GOTO 570 >JN
530 RESTORE 2800:CALL &A2A8, &9B40:GOTO 570 >JH
540 RESTORE 2810:CALL &A2A8, &9C50:GOTO 570 >JM
550 RESTORE 2820:CALL &A2A8, &9D60:GOTO 570 >JR
560 TA=1:GOTO 430 >LC
570 READ GJ, GM, GR, X, Y:POR=0:EC=1:CALL &A1EC:mu=0 >UN
580 fin=0:gan=0:GOSUB 1840:GOSUB 1910:GOSUB 1980:GOSUB >ER
2010
590 CUB=0:tt=100:READ M, N:IF M=1 THEN GOSUB 2430 >QU
600 CALL &A1EC:CALL &A230, FN PO(X, Y), SP(15):D7=0:INK 12 >AW
, 3
610 REM :::::::::::::::::::: >XX
620 REM : : >KJ
630 REM : ROUTINE PRINCIPALE: >YT
640 REM : : >LA
650 REM :::::::::::::::::::: >XB
660 IF FIN=1 THEN 2100 >PT
670 IF INKEY(74)*INKEY(8)=0 THEN 1000 >DC
680 IF INKEY(75)*INKEY(1)=0 THEN 1110 >DZ
690 IF INKEY(72)*INKEY(0)=0 THEN 740 >CW
700 IF INKEY(73)*INKEY(2)=0 THEN 870 >CV
710 PA=PA+1:IF PA>50 THEN 1220 >XD
720 GOTO 660 >ZH
730 REM :: VERS LE HAUT :: >VY
740 IF Y=9 AND EC=2 THEN 830 >UZ
750 ADD=FN PO(X, Y):CALL &A25A, ADD:CALL &A08B, ADD >RX
760 CALL &A264, SP(16):FOR T=1 TO tt:NEXT:CALL &A28C >UK

770 CALL &A264, SP(17):FOR T=1 TO tt:NEXT:CALL &A28C >UM
780 CALL &A264, SP(17):FOR T=1 TO tt:NEXT:CALL &A28C >UN

790 IF MU=0 THEN SOUND 4, 150, 35, 10, 4, 4 >DG
800 CALL &A264, SP(16):FOR T=1 TO tt:NEXT:CALL &A28C >UE
810 CALL &A264, SP(15):FOR T=1 TO 50:NEXT:Y=Y-1 >NM
820 IF Y=9 AND EC=2 THEN 830 ELSE 1370 >CT
830 IF M=1 THEN GOSUB 2420 >TU
840 CALL &A1EC:PY(9)=&C63B:CALL &A230, FN PO(X, Y), SP(15) >ZY

850 EC=1:GOTO 1370 >MH
860 REM :: VERS LE BAS :: >UG
870 IF Y=9 AND EC=1 THEN 960 >UG
880 ADD=FN PO(X, Y):CALL &A25A, ADD:CALL &A08B, ADD >RB

890 CALL &A295, SP(16):FOR T=1 TO tt:NEXT:CALL &A28C >UU
900 CALL &A295, SP(17):FOR T=1 TO tt:NEXT:CALL &A28C >UL

910 CALL &A295, SP(17):FOR T=1 TO tt:NEXT:CALL &A28C >UM

920 IF MU=0 THEN SOUND 4, 150, 35, 10, 4, 4 >DB
930 CALL &A295, SP(16):FOR T=1 TO tt:NEXT:CALL &A28C >UN

940 CALL &A295, SP(15):FOR T=1 TO 50:NEXT:Y=Y+1 >NU
950 IF Y=9 AND EC=1 THEN 960 ELSE 1370 >CA
960 IF N=1 THEN GOSUB 2510 >TZ
970 CALL &A21E:PY(9)=&BFFB:CALL &A230, FN PO(X, Y), SP(15) >ZW

980 EC=2:GOTO 660 >LB
990 REM :: A GAUCHE :: >QE
1000 ADD=FN PO(X, Y):CALL &A08B, ADD:CALL &A230, ADD-1, SP(>CJ
16)
1010 FOR T=1 TO tt:NEXT >RQ
1020 CALL &A08B, ADD-1:CALL &A230, ADD-2, SP(17) >ML
1030 FOR T=1 TO tt:NEXT >RT
1040 CALL &A08B, ADD-2:CALL &A230, ADD-3, SP(17) >MQ
1050 FOR T=1 TO tt:NEXT:IF MU=0 THEN SOUND 4, 150, 35, 10, >BK
4, 4
1060 CALL &A08B, ADD-3:CALL &A230, ADD-4, SP(16) >MU
1070 FOR T=1 TO tt:NEXT >RX
1080 CALL &A08B, ADD-4:CALL &A230, ADD-5, SP(15) >MX
1090 FOR T=1 TO 50:NEXT:X=X-1:GOTO 1370 >GL
1100 REM :: A DROITE :: >RF
1110 ADD=FN PO(X, Y):CALL &A08B, ADD:CALL &A230, ADD+1, SP(>CJ
16)
1120 FOR T=1 TO tt:NEXT >RT
1130 CALL &A08B, ADD+1:CALL &A230, ADD+2, SP(17) >MJ
1140 FOR T=1 TO tt:NEXT >RV
1150 CALL &A08B, ADD+2:CALL &A230, ADD+3, SP(17) >MN
1160 FOR T=1 TO tt:NEXT:IF MU=0 THEN SOUND 4, 150, 35, 10, >BM
4, 4
1170 CALL &A08B, ADD+3:CALL &A230, ADD+4, SP(16) >MR
1180 FOR T=1 TO tt:NEXT >RZ
1190 CALL &A08B, ADD+4:CALL &A230, ADD+5, SP(15) >MV
1200 FOR T=1 TO 50:NEXT:X=X+1:GOTO 1370 >GB
1210 REM :: SUR PLACE :: >TU
1220 ADD=FN PO(X, Y):CALL &A08B, ADD:CALL &A230, ADD, SP(16) >AD
)
1230 FOR T=1 TO tt*2:NEXT >UF
1240 CALL &A08B, ADD:CALL &A230, ADD, SP(17) >HH
1250 FOR T=1 TO tt*2:NEXT >UH
1260 CALL &A08B, ADD:CALL &A230, ADD, SP(17) >HK
1270 FOR T=1 TO tt*2:NEXT:IF MU=0 THEN SOUND 4, 150, 35, 1 >DT
0, 4, 4
1280 CALL &A08B, ADD:CALL &A230, ADD, SP(16) >HL
1290 FOR T=1 TO tt:NEXT >RB
1300 CALL &A08B, ADD:CALL &A230, ADD, SP(15) >HC
1310 PA=0:GOTO 1380 >ME
1320 REM :::::::::::::::::::: >XD
1330 REM : : >RE
1340 REM : TESTS DES CASES : >VW
1350 REM : : >RG
1360 REM :::::::::::::::::::: >XH
1370 IF CUB=1 THEN 1430 >QH

```

```

1380 A=FN PK(X,Y)+1:IF a>14 THEN 660 >BP 1820 REM :::::::::::::::::::: >YA
1390 ON A GOTO 1400,1410,1420,1420,1420,1460,1600,1670, >HK 1830 REM :: SCORE :: >PY
1690,1730,1730,1730,1400,1400 1840 IF sc=0 THEN a$="00000":ZL=3:YL=25:GOSUB 350:RETUR >BV
1400 FIN=1 >MJ N
1410 GOTO 660 >FD 1850 a$=STR$(sc):A$=RIGHT$(A$,LEN(A$)-1) >HT
1420 IF POR<>0 THEN 660 ELSE POR=A-1:X1=X:Y1=Y:CUB=1:EC >UP 1860 IF sc<10 THEN ZL=7:YL=25:GOSUB 350:RETURN >QQ
RA=EC:GOTO 660 1870 IF sc<100 THEN ZL=6:YL=25:GOSUB 350:RETURN >RG
1430 IF EC=ECRA THEN CALL &A00D, FN PO(X1,Y1), SP(1) >TG 1880 IF sc<1000 THEN ZL=5:YL=25:GOSUB 350:RETURN >TZ
1440 POKE &9E5F+(Y1*16)+X1,0:SC=SC+5:GOSUB 1840 >QC 1890 ZL=4:YL=24:GOSUB 350:RETURN >CJ
1450 CALL &A00D,&E706, SP(POR+1):CUB=0:GOTO 660 >PU 1900 REM :: RECORD :: >QJ
1460 IF POR<2 OR POR>4 THEN 660 >XE 1910 IF REC=0 THEN a$="00000":ZL=34:YL=25:GOSUB 350:RET >DB
1470 CALL &A08B, FN PO(X,Y):CALL &A00D, FN PO(X,Y), SP(POR >DB
+1) URN
1480 CALL &A030, FN PO(X,Y), SP(6):CALL &A230, FN PO(X,Y), >FX 1920 a$=STR$(REC):A$=RIGHT$(A$,LEN(A$)-1) >JA
SP(15) 1930 IF REC<10 THEN ZL=38:YL=25:GOSUB 350:RETURN >TA
1490 POKE &9E5F+(Y*16)+X,13+POR:PP=POR-1:POR=0 >RB 1940 IF REC<100 THEN ZL=37:YL=25:GOSUB 350:RETURN >UU
1500 POKE &A01F,&AF:CALL &A00D,&E706, SP(1):POKE &A01F,& >AM 1950 IF REC<1000 THEN ZL=36:YL=25:GOSUB 350:RETURN >VN
1A
1510 ON PP GOTO 1520,1530,1540 >WZ 1960 ZL=35:YL=25:GOSUB 350:RETURN >CZ
1520 GJ=GJ-1:IF GJ=0 THEN 1550 ELSE 660 >EE 1970 REM :: VIE :: >ML
1530 GM=GM-1:IF GM=0 THEN 1550 ELSE 660 >EQ 1980 a$=STR$(vie):A$=RIGHT$(A$,LEN(A$)-1) >JT
1540 GR=GR-1:IF GR=0 THEN 1550 ELSE 660 >EH 1990 ZL=14:YL=25:GOSUB 350:RETURN >CZ
1550 IF ec=1 THEN 1580 >PL 2000 REM :: TAB :: >MF
1560 XD=INT(RND*16)+1:YD=INT(RND*8)+1:IF FN PK(XD,YD)<> >NL 2010 a$=STR$(TA):A$=RIGHT$(A$,LEN(A$)-1) >HE
1 THEN 1550 2020 IF TA<10 THEN ZL=29:YL=25:GOSUB 350:RETURN >RA
1570 POKE &9E5F+(YD*16)+XD,8+PP:SOUND 3,20,100,15,0,3:G >GM 2030 ZL=28:YL=25:GOSUB 350:RETURN >CP
OTO 660 2040 REM :::::::::::::::::::: >XD
1580 XD=INT(RND*16)+1:YD=INT(RND*9)+9:IF FN PK(XD,YD)<> >NB 2050 REM : : >RE
1 THEN 1580 2060 REM : FIN DE PARTIE : >TD
1590 GOTO 1570 >NF 2070 REM : : >RG
1600 IF ec=1 THEN 1640 >PD 2080 REM :::::::::::::::::::: >XH
1610 IF M=1 THEN GOSUB 2420 >UH 2090 ::: GAGNE ::: >LB
1620 X=INT(RND*16)+1:Y=INT(RND*8)+1:IF FN PK(X,Y)<>1 TH >HL 2100 FOR H=0 TO 3:NU=REMAIN(H):NEXT >DU
EN 1620 2110 WHILE INKEY$<>"":WEND:IF gan=0 THEN 2230 >MN
1630 EC=1:py(9)=&C63B:CALL &A1EC:CALL &A230, FN PO(X,Y), >TZ 2120 SOUND 1,0,15,15,10,,15:FOR h=1 TO 25:sc=sc+10:GOSU >EJ
SP(15):GOTO 660 B 1840
1640 IF N=1 THEN GOSUB 2510 >UM 2130 SOUND 1,0,15,15,10,,15:FOR t=1 TO 50:NEXT t,h >QR
1650 X=INT(RND*16)+1:Y=INT(RND*9)+9:IF FN PK(X,Y)<>1 TH >HC 2140 ADD=FN PO(X,Y):CALL &A08B,ADD:CALL &A230,ADD,SP(16 >AF
EN 1650)
1660 EC=2:py(9)=&BFFB:CALL &A21E:CALL &A230, FN PO(X,Y), >TX 2150 FOR T=1 TO 200:NEXT:CALL &A08B,ADD:CALL &A230,ADD, >FG
SP(15):GOTO 660 SP(17)
1670 IF POR<>100 THEN 660 >RX 2160 FOR T=1 TO 200:NEXT:CALL &A08B,ADD:CALL &A230,ADD, >FH
SP(17)
1680 gan=1:fin=1:GOTO 660 >UZ 2170 FOR T=1 TO 200:NEXT:CALL &A08B,ADD >FM
1690 IF POR<10 OR POR>12 THEN 660 >ZJ 2180 POKE &A01F,&AF:CALL &A00D,&E706, SP(1):POKE &A01F,& >AT
1700 INK 12,3,24:D7=D7+1:POR=0 >XP 1A
1710 POKE &A01F,&AF:CALL &A00D,&E706, SP(1) >JJ 2190 FOR t=1 TO 3000:NEXT t >UM
1720 POKE &A01F,&A1A:GOTO 2560 >WW 2200 TA=TA+1:IF SC>REC THEN REC=SC:GOSUB 1910 >NJ
1730 IF POR<>0 THEN 660 ELSE SOUND 1,50,21,13,3,4 >PY 2210 GOTO 430 >EH
1740 POR=A:CALL &A08B, FN PO(X,Y):CALL &A00D, FN PO(X,Y), >FR 2220 REM :: PERDU :: >PW
SP(2) 2230 BORDER 26:INK 0,26:SOUND 4,1500,50,7,0,0,10 >PH
1750 CALL &A230, FN PO(X,Y), SP(15):POKE &9E5F+(Y*16)+X,1 >YN 2240 SOUND 4,1200,0,0,14,14:SOUND 4,0,20,2,,10 >ME
2250 FOR h=1000 TO 1500 STEP 100:SOUND 4,h,30,0,15,15:N >AE
EXT
1760 IF POR=10 THEN SS=10 ELSE IF POR=11 THEN SS=12 ELS >ER 2260 ADD=FN PO(X,Y):CALL &A08B,ADD:CALL &A230,ADD,SP(16 >AJ
E SS=11)
1770 CALL &A00D,&E706, SP(SS):GOTO 660 >EB 2270 FOR T=1 TO 200:NEXT:BORDER 0:INK 0,0 >HR
1780 REM :::::::::::::::::::: >YF 2280 CALL &A08B,ADD:CALL &A230,ADD,SP(17) >HN
1790 REM : : >TE 2290 FOR T=1 TO 200:NEXT >RE
1800 REM : GESTION COMPTEURS : >YM 2300 CALL &A08B,ADD:CALL &A230,ADD,SP(17) >HF
1810 REM : : >RH

```

```

2310 FOR T=1 TO 200:NEXT >RX 2740 DATA 4,4,2,6,6,1,1 >PD
2320 CALL &A08B,ADD:POKE &A01F,&AF:CALL &A00D,&E706,SP(>VR 2750 DATA 6,6,2,6,6,0,1 >PH
1):POKE &A01F,&1A >VR 2760 DATA 4,4,3,2,2,0,0 >PW
2330 vie=vie-1:GOSUB 1980:IF vie=0 THEN 2350 >LX 2770 DATA 3,3,2,2,2,0,1 >PV
2340 FOR T=1 TO 2000:NEXT T:ta=ta-1:GOTO 2200 >MB 2780 DATA 3,3,2,5,6,1,1 >PE
2350 FOR T=1 TO 2000:NEXT T:TA=0:vie=4:sc=0:GOSUB 1980:>KF 2790 DATA 3,3,2,4,6,1,0 >PD
GOTO 2200 >KF 2800 DATA 2,2,2,0,2,1,1 >PU
2360 REM :::::::::::::::::::: >YA 2810 DATA 3,2,3,7,2,0,0 >PU
2370 REM : : >RK 2820 DATA 2,1,3,2,7,1,1 >PV
2380 REM : S/PROG DIVERS : >UU 2830 REM :::::::::::::::::::: >YC
2390 REM : : >TB 2840 REM : : >TB
2400 REM :::::::::::::::::::: >XD 2850 REM : PRESENTATION : >UM
2410 REM :: MONSTRE TAB 1 :: >WL 2860 REM : : >TD
2420 POKE ADX3,1:POKE ADX4,1:POKE ADX5,1 >HD 2870 REM :::::::::::::::::::: >YG
2430 X3=INT(RND*16)+1:Y3=INT(RND*8)+1:IF FN PK(X3,Y3)<> >MP 2880 INK 3,0,16:INK 10,16,0:SPEED INK 25,25:RESTORE 264 >BN
1 THEN 2420 >MP 0:EVERY 15,2 GOSUB 2620
2440 ADX3=&9E5F+(Y3*16)+X3:POKE ADX3,12 >GK 2890 EVERY 15,1 GOSUB 2610:EVERY 60,3 GOSUB 2600 >PE
2450 X4=INT(RND*16)+1:Y4=INT(RND*8)+1:IF FN PK(X4,Y4)<> >MZ 2900 ZX=0:ZY=0:E8=10:E9=3:GOSUB 2910:ZX=188:ZY=0:E8=3:E >TB
1 THEN 2450 >MZ 9=10:GOSUB 2910:ZX=376:ZY=0:E8=10:E9=3:GOSUB 2910:GOTO
2460 ADX4=&9E5F+(Y4*16)+X4:POKE ADX4,12 >GR 3010
2470 X5=INT(RND*16)+1:Y5=INT(RND*8)+1:IF FN PK(X5,Y5)<> >MH 2910 ORIGIN ZX,ZY:FOR h=60 TO 76 STEP 2:PLOT 100,h,4:DR >QV
1 THEN 2470 >MH AW 164,h:NEXT
2480 ADX5=&9E5F+(Y5*16)+X5:POKE ADX5,12 >GY 2920 FOR h=84 TO 100 STEP 2:PLOT h,156:DRAW 100,60:PLOT >GM
2490 RETURN >FJ h+80,156:DRAW 164,60:NEXT
2500 REM :: MONSTRE TAB 2 :: >WM 2930 FOR h=100 TO 164 STEP 2:PLOT h,156:DRAW h,76:NEXT: >QU
2510 COUP=0 >ZB PLOT 126,156,14:DRAW 100,76:DRAW 164,76:DRAW 138,156:PL
2520 X8=INT(RND*16)+1:Y8=INT(RND*9)+9:COUP=COUP+1:IF CO >AB 0T 96,156:DRAW 132,76:DRAW 168,156
UP>10 THEN RETURN >AB 2940 FOR h=80 TO 184 STEP 4:x=INT(RND*8):PLOT h,166,1:D >LM
2530 IF FN PK(X8,Y8)<>1 THEN 2510 >ZX RAW h,146+x:NEXT:PLOT 80,160,14:DRAW 184,160
2540 POKE &9E5F+(Y8*16)+X8,13:RETURN >EE 2950 PLOT 80,166:PLOT 184,166:PLOT 80,158:PLOT 184,158 >XG
2550 REM :: MUSIQUE :: >RT 2960 FOR G=146 TO 178 STEP 16:XT=G:YT=176:P=8:EN=E8:GOS >TH
2560 IF D7=1 THEN EVERY 15,1 GOSUB 2610:tt=tt-10:GOTO 6 >AE 00:XT=G:YT=200:P=8:EN=E9:GOSUB 3000:NEXT
60 >AE 2970 XT=198:YT=170:P=8:EN=E8:GOSUB 3000:XT=198:YT=194:P >CR
2570 IF D7=2 THEN EVERY 60,3 GOSUB 2600:tt=tt-15:GOTO 6 >AN =8:EN=E9:GOSUB 3000
60 >AN 2980 FOR G=88 TO 120 STEP 16:XT=G:YT=176:P=8:EN=E9:GOSU >QP
2580 IF D7=3 THEN RESTORE 2640:EVERY 15,2 GOSUB 2620:PO >MC B 3000:XT=G:YT=200:P=8:EN=E8:GOSUB 3000:NEXT
R=100:MU=1:tt=tt-15:GOTO 660 >MC 2990 YT=68:YT=170:P=8:EN=E9:GOSUB 3000:XT=68:YT=194:P=8 >AR
2590 GOTO 660 >GD :EN=E8:GOSUB 3000
2600 DI:SOUND 2,0,15,15,1,0,1:SOUND 2,0,15,15,1,0,5:EI: >GZ 3000 PLOT 700,700,EN:P2=P*P:ORIGIN XT+ZX,YT+ZY:FOR H=0 >AJ
RETURN >GZ TO P*0.71 STEP 2:J=SQR(P2-H*H):MOVE H,J:DRAW H,-J:MOVE
2610 DI:SOUND 2,0,15,15,1,0,5:EI:RETURN >HJ J,H:DRAW J,-H:MOVE -H,J:DRAW -H,-J:MOVE -J,H:DRAW -J,-H
2620 DI:READ s:IF s=0 THEN EI:RETURN ELSE IF s=-1 THEN >NY :NEXT:ORIGIN 0,0:RETURN
RESTORE 2640:GOTO 2620 ELSE 2630 >NY 3010 B$="CHARLIE":FOR H=1 TO 7:A$=MID$(B$,H,1):INK 8,0 >WB
2630 SOUND 1,s,30,14,0,2:SOUND 4,s*2,30,15,0,2:EI:RETUR >BH 3020 PEN 8:LOCATE 1,23:PRINT A$:X=H*48+30:Y=1:E1=2:E2=8 >YN
N >BH :GOSUB 3060:NEXT H
2640 DATA 451,0,379,0,301,0,0,451,379,0,301,301,451,0,3 >BK 3030 B$="BONDSOUND":FOR H=1 TO 9:A$=MID$(B$,H,1) >QN
79,0,301,0,0,451,379,0,301,301,338,0,284,0,225,0,0,338, >YV 3040 PEN 8:LOCATE 1,23:PRINT A$:X=H*48+90:Y=2:E1=15:E2=
284,0,225,225,338,0,284,0,225,0,0,338,284,0,225,225 >YV 4:GOSUB 3060:NEXT H
2650 DATA 451,0,379,0,301,0,0,451,379,0,301,301,451,0,3 >KJ 3050 LOCATE 1,23:PRINT " ":INK 8,8:GOTO 3100 >KG
79,0,301,0,0,451,379,0,301,301,338,0,301,338,338,379,0, >KN 3060 FOR I=32 TO 48 STEP 2:FOR L=1 TO 32 STEP 4
451,-1 >KJ 3070 A=TEST(L,1):IF A=0 THEN 3090 >ZV
2660 REM :::::::::::::::::::: >YD 3080 PLOT X-4+L,310-(Y*48-(I*2)),E1:PLOT X-4+L,312-(Y*4 >LT
2670 REM : : >TC 8-(I*2)),E2
2680 REM :PARAMETRES TABLEUX: >AP 3090 NEXT L,I:RETURN >QD
2690 REM : : >TE 3100 FOR H=1 TO 5:MOVE 30,H:DRAW 610,H,4:NEXT:FOR H=6 T >QT
2700 REM :::::::::::::::::::: >XG 0 20:MOVE 30,H:DRAW 610,H,8:NEXT
2710 DATA 3,2,2,6,6,0,0 >PV 3110 FOR H=20 TO 22:MOVE 30,H:DRAW 610,H,1:NEXT:FOR H=1 >RZ
2720 DATA 3,4,3,6,3,0,0 >PW TO 8:MOVE H,26:DRAW H,374,4:NEXT
2730 DATA 2,2,2,2,2,0,0 >PM 3120 FOR H=8 TO 20:MOVE H,26:DRAW H,374,8:NEXT:FOR H=20 >UD

```

```

TO 22:MOVE H,26:DRAW H,374,1:NEXT
3130 FOR H=616 TO 620:MOVE H,26:DRAW H,374,1:NEXT:FOR H >BX
=620 TO 632:MOVE H,26:DRAW H,374,8:NEXT
3140 FOR H=632 TO 636:MOVE H,26:DRAW H,374,4:NEXT:FOR H >BQ
=394 TO 400:MOVE 30,H:DRAW 610,H,4:NEXT
3150 FOR H=382 TO 392:MOVE 30,H:DRAW 610,H,8:NEXT:FOR H >BQ
=378 TO 380:MOVE 30,H:DRAW 610,H,1:NEXT
3160 X=20:FOR T=1 TO 22:MOVE T,X:DRAW T,20,15:X=X-1:NEX >CZ
T T
3170 X=380:FOR T=1 TO 22:MOVE T,X:DRAW T,380:X=X+1:NEXT >BE
T
3180 X=380:FOR T=618 TO 636:MOVE T,380:DRAW 618,X:X=X+1 >HB
:NEXT T
3190 X=1:FOR T=618 TO 636:MOVE T,20:DRAW T,X:X=X+1:NEXT >CD
T
3200 A$="L<M<C SOFTWARE":ZL=4:YL=23:GOSUB 350 >QH
3210 WHILE INKEY$="" :WEND:FOR H=0 TO 3:MU=REMAIN(H):NEX >DC
T
3220 CLS:CALL &BCA7:INK 3,9:INK 10,1:RETURN >LJ

```

# DATACHAR.BAS

```

5 MEMORY &861F:MODE 1
7 LOCATE 1,10:PRINT " DATAS DE LA LIGNE CORRECTS":PEN 2

```

```

10 A=&8820:F=&A2C0:L=100:WHILE A<=F:FOR A=A TO A+15:READ C$:K=V
AL("&"+C$):S=S+K+65536*(S+K>32767):IF A<=F THEN POKE A,K
20 NEXT:READ D$:T=VAL("&"+D$):IF T<>S THEN PRINT CHR$(7);"Erreu
r ligne";L:END ELSE L=L+5
25 LOCATE 20,10:PRINT L-5:WEND
100 DATA 2D,F0,3C,5A,B4,B4,1E,F0,2D,78,F0,2D,F0,B4,5A,B4,089D
105 DATA B4,5A,78,2D,78,F0,3C,F0,B4,2D,F0,3C,5A,B4,B4,1E,10D1
110 DATA F0,2D,78,F0,2D,F0,B4,5A,B4,B4,5A,78,2D,78,F0,3C,198C
115 DATA F0,B4,2D,F0,3C,5A,B4,B4,1E,F0,2D,78,F0,2D,F0,B4,22BF
120 DATA 5A,B4,B4,5A,78,2D,78,F0,3C,F0,B4,2D,F0,3C,5A,B4,2B2F
125 DATA B4,1E,F0,2D,78,F0,2D,F0,B4,5A,B4,B4,5A,78,2D,78,3390
130 DATA F0,3C,F0,B4,4A,C0,C0,C0,C0,4A,C0,C0,C0,C0,4A,C0,3DFE
135 DATA C0,C0,C0,4A,C0,C0,C0,C0,4A,C0,C0,C0,C0,4A,C0,C0,489C
140 DATA C0,C0,4A,C0,C0,C0,C0,4A,C0,C0,C0,C0,4A,C0,C0,C0,533A
145 DATA C0,4A,C0,C0,C0,C0,4A,C0,C0,C0,C0,4A,C0,C0,C0,C0,5DD8
150 DATA 4A,C0,C0,C0,C0,4A,C0,C0,C0,C0,4A,C0,C0,C0,C0,4A,6800
155 DATA C0,C0,C0,C0,4A,C0,C0,C0,C0,4A,C0,C0,C0,C0,4A,C0,729E
160 DATA C0,C0,C0,0F,0F,0F,0F,0F,0F,1A,30,30,30,30,1A,30,30,767D
165 DATA 30,30,1A,30,30,30,30,1A,30,30,30,30,1A,30,30,30,793B
170 DATA 30,1A,30,30,30,30,1A,30,30,30,30,1A,30,30,30,30,7BF9
175 DATA 1A,30,30,30,30,1A,30,30,30,30,1A,30,30,30,30,1A,7EA1
180 DATA 30,30,30,30,1A,30,30,30,30,1A,30,30,30,30,1A,30,815F
185 DATA 30,30,30,1A,30,30,30,30,1A,30,30,30,30,1A,30,30,841D
190 DATA 30,30,1A,30,30,30,30,0F,0F,0F,0F,0F,0F,0B,03,03,03,85B6
195 DATA 03,0B,03,03,03,03,0B,03,03,03,03,03,0B,03,03,03,85FE
200 DATA 0B,03,03,03,03,0B,03,03,03,03,0B,03,03,03,03,0B,864E
205 DATA 03,03,03,03,0B,03,03,03,03,0B,03,03,03,03,0B,03,8696
210 DATA 03,03,03,0B,03,03,03,03,0B,03,03,03,03,0B,03,03,86DE
215 DATA 03,03,0B,03,03,03,03,0B,03,03,03,03,0B,03,03,03,8726
220 DATA 03,0B,03,03,03,03,0B,03,03,03,03,0F,0F,0F,0F,0F,87A2
225 DATA 5F,FF,FF,FF,FF,5F,FF,FF,FF,FF,5F,FF,FF,FF,5F,9512

```

```

230 DATA FF,FF,FF,FF,5F,FF,FF,FF,FF,5F,FF,FF,FF,5F,FF,A322
235 DATA FF,FF,FF,5F,FF,FF,FF,FF,5F,FF,FF,FF,5F,FF,FF,B132
240 DATA FF,FF,5F,FF,FF,FF,FF,5F,FF,FF,FF,5F,FF,FF,FF,BF42
245 DATA FF,5F,FF,FF,FF,FF,5F,FF,FF,FF,5F,FF,FF,FF,FF,CD52
250 DATA 5F,FF,FF,FF,FF,5F,FF,FF,FF,FF,5F,FF,FF,FF,FF,0F,DA72
255 DATA 0F,0F,0F,0F,4A,C0,C0,C0,C0,4A,C0,C0,C0,C0,4A,40,E1CC
260 DATA 40,40,40,4A,40,40,40,40,4A,40,40,40,40,4A,40,40,E5EA
265 DATA 40,40,4A,84,84,84,C0,4A,40,40,40,40,4A,40,40,40,EB54
270 DATA 40,4A,40,40,40,40,4A,40,40,40,40,4A,40,40,40,40,EF72
275 DATA 4A,84,84,84,C0,4A,40,40,40,40,4A,40,40,40,40,4A,F4E6
280 DATA 40,40,40,40,4A,40,40,40,4A,C0,C0,C0,C0,4A,C0,FB84
285 DATA C0,C0,C0,0F,0F,0F,0F,0F,0E,48,C0,C0,0C,0E,C0,C0,017F
290 DATA C0,84,4A,C0,C0,C0,C0,4A,6C,CC,CC,68,4A,9C,3C,3C,0A21
295 DATA C8,4A,98,0B,5F,C8,4A,98,0B,5F,C8,4A,98,0B,5F,C8,1125
300 DATA 4A,98,0B,5F,C8,4A,98,0B,5F,C8,4A,98,0B,5F,C8,4A,17AB
305 DATA 98,0B,5F,C8,4A,98,0B,5F,C8,4A,98,0B,5F,C8,4A,9C,1E83
310 DATA 3C,3C,C8,4A,6C,CC,CC,68,4A,C0,C0,C0,C0,0E,C0,C0,2751
315 DATA C0,84,0E,48,C0,C0,0C,0F,0F,0F,0F,0F,00,00,00,00,2AC2
320 DATA 22,00,00,00,11,22,00,00,00,33,22,00,00,11,22,22,2BC1
325 DATA 00,00,33,00,22,00,00,22,00,22,00,00,22,00,22,04,2CA2
330 DATA 0C,26,0C,26,00,00,22,00,22,00,00,22,11,22,04,0C,2DAF
335 DATA 26,19,26,00,00,22,11,22,00,11,22,00,00,04,19,26,2EDF
340 DATA 0C,0C,00,11,22,00,00,00,00,00,00,00,04,0C,0C,2F52
345 DATA 0C,00,00,00,00,00,00,00,00,00,00,00,00,00,00,2F5E
350 DATA 4A,C0,C0,C0,5F,C0,D5,C0,D5,5F,C0,D5,C0,D5,4A,3A04
355 DATA C0,C0,C0,C0,4A,C0,C0,C0,C0,4A,85,0D,0D,C0,4A,85,42C6
360 DATA 0D,0D,C0,4A,84,0E,0E,C0,4A,84,0E,0E,C0,5F,85,0D,47E5
365 DATA 0D,D5,5F,85,0D,0D,D5,4A,84,0E,0E,C0,4A,84,0E,0E,4D2E
370 DATA C0,4A,85,0D,0D,C0,4A,85,0D,0D,C0,4A,C0,C0,C0,C0,548A
375 DATA 4A,C0,C0,C0,C0,5F,C0,D5,C0,D5,5F,C0,D5,C0,D5,0F,5EF5
380 DATA 0F,0F,0F,0F,00,00,00,00,00,00,00,00,00,00,10,30,5F71
385 DATA 30,30,00,10,B8,34,30,00,10,74,2D,0F,00,10,34,0F,6210
390 DATA 0F,0A,10,34,0F,0D,0A,10,2D,0F,0E,0F,10,2D,0F,25,636D
395 DATA 0F,10,2D,1A,30,0F,10,2D,1A,10,0F,10,2D,1A,10,0F,64FE
400 DATA 10,2D,1A,30,0F,10,2D,0F,25,0F,10,2D,0F,0F,0F,10,668E
405 DATA 3C,0E,0F,0A,00,00,0F,0D,0A,00,00,05,0F,00,00,00,672B
410 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,672B
415 DATA 00,00,FF,FF,FF,AA,00,BE,FF,FD,AA,00,FF,7C,0F,0A,6FCA
420 DATA 00,FF,AD,0F,0F,00,FF,AD,0E,0F,00,FE,0F,0F,0D,0A,7490
425 DATA FE,0F,5F,0F,0A,FE,0F,FF,AF,0A,FE,0F,AA,AF,0A,FE,7C48
430 DATA 0F,AA,AF,0A,FE,0F,FF,AF,0A,FE,0F,5F,0F,0A,FE,0F,8311
435 DATA 0F,0F,0A,FF,AD,0D,0F,00,00,05,0E,0F,00,00,00,0F,8532
440 DATA 0A,00,00,00,00,00,00,00,00,00,00,00,00,00,00,853C
445 DATA 00,00,00,00,00,00,03,03,03,02,00,16,03,FD,02,00,865F
450 DATA 03,7C,0F,0A,00,03,AD,0F,0F,00,03,AD,0E,0F,00,03,8895
455 DATA 0F,0F,0D,0A,03,0F,0B,0F,0A,03,0F,03,07,0A,03,0F,8938
460 DATA 02,07,0A,03,0F,02,07,0A,03,0F,03,07,0A,03,0F,0B,89B3
465 DATA 0F,0A,03,0F,0F,0F,0A,03,AD,0D,0F,00,00,05,0E,0F,8AF4
470 DATA 00,00,00,0F,0A,00,00,00,00,00,00,00,00,00,00,8B0D
475 DATA 00,00,00,00,00,00,00,00,00,00,00,00,40,C0,A2,00,00,8CAF
480 DATA F3,90,D1,00,00,90,3F,30,00,40,3F,35,3F,A2,40,7F,9256
485 DATA 3F,BF,A2,10,7F,35,BF,A2,10,95,D1,7B,2A,51,30,90,9947
490 DATA 7B,2A,00,90,71,71,00,00,71,3F,35,00,00,10,3F,A2,9D34
495 DATA 00,00,40,71,A2,00,00,10,F3,A2,00,00,51,3F,2A,00,A0E6
500 DATA 00,00,71,00,00,00,00,00,00,00,00,00,00,00,00,A157
505 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,A157
510 DATA C3,82,00,00,41,C3,C3,00,00,41,EB,EB,82,00,D7,5D,A830
515 DATA 5D,82,00,D7,57,57,C3,41,D7,5D,5D,C3,41,C3,EB,EB,B0C6
520 DATA C3,D7,EB,C3,C3,EB,D7,5D,EB,FF,49,D7,08,08,08,49,BA5B

```

525 DATA D7,00,08,08,EB,41,08,08,5D,82,41,AA,08,49,00,00,BEA1  
530 DATA D7,08,EB,00,00,C3,FF,82,00,00,00,C3,00,00,00,00,C372  
535 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,C372  
540 DATA 00,00,00,00,00,00,00,00,00,FF,2A,00,00,10,FF,EE,C698  
545 DATA 00,00,55,75,9D,00,00,FF,75,DD,2A,00,F7,75,DD,2A,CCED  
550 DATA 00,B2,FF,EE,2A,00,B2,FF,EE,2A,00,B2,FF,EE,2A,00,D548  
555 DATA B2,FF,EE,2A,00,F7,30,DD,2A,00,D1,FF,FF,00,00,51,DD5F  
560 DATA FF,BF,00,00,00,F7,AA,00,00,00,00,00,00,00,00,00,E0BE  
565 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,E0BE  
570 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,E0BE  
575 DATA 00,00,00,00,00,00,00,FF,AA,00,00,10,FF,EE,00,00,E464  
580 DATA FF,75,DD,2A,00,FF,75,DD,2A,00,F7,75,DD,2A,00,F7,ECC4  
585 DATA 75,DD,2A,00,F7,75,DD,2A,00,F7,75,DD,2A,00,10,FF,F435  
590 DATA EE,00,00,00,F7,AA,00,00,00,00,00,00,00,00,00,00,F6C4  
595 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,F6C4  
600 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,F6C4  
605 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,F6C4  
610 DATA FF,2A,00,00,10,FF,6E,00,00,51,75,9D,00,00,51,75,FB93  
615 DATA 9D,00,00,51,75,9D,00,00,51,75,9D,00,00,10,F7,EE,00EB  
620 DATA 00,00,00,F7,AA,00,00,00,00,00,00,00,00,00,00,00,00,028C  
625 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,028C  
630 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,DD,5E,00,DD,04A4  
635 DATA 56,01,DD,6E,02,DD,66,03,06,08,C5,E5,06,02,1A,77,09DF  
640 DATA 23,13,10,FA,E1,CD,26,BC,C1,10,EF,C9,04,0C,04,0C,1058  
645 DATA 10,10,10,10,51,51,51,51,55,FF,00,00,00,08,04,08,1344  
650 DATA 00,20,00,20,00,A2,00,A2,55,FF,00,00,04,0C,04,04,1634  
655 DATA 00,10,00,10,00,A2,51,00,55,FF,00,00,04,0C,04,04,18B3  
660 DATA 00,10,00,10,00,F3,00,51,55,FF,00,00,00,04,00,0C,1B7B  
665 DATA 00,30,10,10,51,F3,00,51,00,55,00,00,04,0C,04,00,1DC9  
670 DATA 10,00,00,30,00,51,51,51,55,FF,00,00,04,0C,04,04,2068  
675 DATA 10,00,10,30,51,51,51,51,55,FF,00,00,04,0C,00,04,2364  
680 DATA 00,10,00,20,00,A2,51,00,55,00,00,00,04,0C,04,04,24F4  
685 DATA 10,10,10,30,51,51,51,51,55,FF,00,00,04,0C,04,04,2804  
690 DATA 10,10,10,30,00,51,00,51,55,FF,00,00,00,00,00,00,2A5A  
695 DATA 00,00,00,00,00,A2,00,00,00,AA,00,00,00,00,00,00,2BA6  
700 DATA 00,00,00,00,00,A2,00,00,00,AA,55,00,00,00,00,00,2D47  
705 DATA 00,00,00,00,00,00,00,A2,00,AA,00,00,00,00,00,00,2E93  
710 DATA 00,00,00,00,00,00,00,A2,00,AA,55,00,00,08,00,08,3044  
715 DATA 10,00,00,00,00,00,00,00,00,00,00,00,00,00,00,04,04,3064  
720 DATA 10,10,00,10,00,A2,00,00,00,AA,00,00,00,08,00,08,31F0  
725 DATA 00,20,00,20,00,A2,00,00,00,AA,00,00,04,0C,04,0C,339C  
730 DATA 10,10,10,10,51,F3,51,51,55,55,00,00,04,0C,04,0C,368C  
735 DATA 10,10,10,10,51,A2,51,51,55,FF,00,00,04,0C,04,0C,39D5  
740 DATA 10,00,10,00,51,00,51,00,55,FF,00,00,04,08,04,0C,3C07  
745 DATA 10,10,10,10,51,51,51,51,55,FF,00,00,04,0C,04,0C,3EFF  
750 DATA 10,00,10,00,51,A2,51,00,55,FF,00,00,04,0C,04,0C,41D7  
755 DATA 10,00,10,00,51,A2,51,00,55,00,00,00,04,0C,04,0C,43B0  
760 DATA 10,10,10,00,51,00,51,51,55,FF,00,00,04,04,04,04,4637  
765 DATA 10,10,10,10,51,F3,51,51,55,55,00,00,04,0C,00,08,491F  
770 DATA 00,20,00,20,00,A2,00,A2,55,FF,00,00,04,0C,00,08,4C0F  
775 DATA 00,20,00,20,00,A2,00,A2,55,AA,00,00,04,04,04,04,4EA2  
780 DATA 10,10,10,20,51,A2,51,51,55,55,00,00,04,00,04,00,5139  
785 DATA 10,00,10,00,51,00,51,00,55,FF,00,00,04,04,04,0C,5367  
790 DATA 10,30,10,10,51,51,51,51,55,55,00,00,04,04,04,04,55C5  
795 DATA 10,30,10,30,51,F3,51,51,55,55,00,00,04,0C,04,04,58ED  
800 DATA 10,10,10,10,51,51,51,51,55,FF,00,00,04,0C,04,0C,5BE5  
805 DATA 10,10,10,10,51,F3,51,00,55,00,00,00,04,0C,04,0C,5E2F  
810 DATA 10,10,10,10,51,51,51,51,55,FF,00,AA,04,0C,04,0C,61D1  
815 DATA 10,10,10,10,51,A2,51,51,55,55,00,00,04,0C,04,0C,6470

820 DATA 10,00,10,00,51,F3,00,51,55,FF,00,00,04,0C,04,0C,6799  
825 DATA 00,20,00,20,00,A2,00,A2,00,AA,00,00,04,04,04,04,69D7  
830 DATA 10,10,10,10,51,51,51,51,55,FF,00,00,04,04,04,04,6CBF  
835 DATA 10,10,10,10,51,51,51,51,00,AA,00,00,04,04,04,04,6EFD  
840 DATA 10,10,10,10,51,F3,51,F3,55,55,00,00,04,04,04,04,727F  
845 DATA 10,10,00,20,00,A2,51,51,55,55,00,00,04,04,04,04,74BD  
850 DATA 10,10,00,20,00,A2,00,A2,00,AA,00,00,04,0C,00,04,76FF  
855 DATA 00,10,00,20,00,A2,51,00,55,FF,00,00,00,00,00,00,7976  
860 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,7976  
865 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,7976  
870 DATA 00,00,00,00,00,00,00,01,01,05,01,01,00,00,00,00,797F  
875 DATA 00,00,00,00,00,00,00,01,01,05,01,01,00,00,00,00,7988  
880 DATA 00,00,00,01,08,01,06,01,05,01,01,00,00,00,00,00,79A0  
885 DATA 00,00,00,01,01,01,01,01,01,01,01,00,00,00,00,00,79A8  
890 DATA 00,00,00,01,01,07,01,01,01,01,01,00,00,00,00,00,79B6  
895 DATA 00,00,00,00,00,00,00,01,01,01,01,01,00,00,00,00,79BB  
900 DATA 00,00,00,00,00,00,00,01,01,02,02,02,00,00,00,00,79C3  
905 DATA 00,00,00,00,00,00,00,01,01,01,01,01,00,00,00,00,79C8  
910 DATA 00,00,00,00,00,00,00,01,01,01,01,01,00,00,00,00,79CD  
915 DATA 00,00,00,00,00,00,00,04,04,01,01,01,00,00,00,00,79D8  
920 DATA 00,00,00,00,00,00,00,01,01,01,01,01,00,00,00,00,79DD  
925 DATA 00,00,00,00,00,00,00,01,01,01,01,01,00,00,00,00,79E2  
930 DATA 00,08,01,01,01,01,03,03,01,01,01,01,01,00,00,79FA  
935 DATA 00,01,01,01,01,01,01,01,01,01,01,01,06,01,01,00,7A0C  
940 DATA 00,01,01,01,05,05,01,01,01,01,01,01,05,05,00,00,7A29  
945 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,7A29  
950 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,7A29  
955 DATA 00,00,00,00,00,00,00,00,00,01,01,01,04,01,01,01,00,7A33  
960 DATA 00,04,01,01,01,07,00,00,01,00,01,01,06,01,01,00,7A4C  
965 DATA 00,01,06,01,01,01,01,01,04,00,01,01,01,05,01,00,7A65  
970 DATA 00,01,01,01,01,01,00,01,01,00,00,00,00,00,00,00,7A6C  
975 DATA 00,05,05,05,05,05,00,00,01,01,01,01,01,01,03,00,7A8E  
980 DATA 00,00,00,00,00,00,00,00,00,01,01,01,01,01,08,01,00,7A9C  
985 DATA 00,00,00,00,00,00,00,00,00,00,00,00,01,01,00,00,00,7A9E  
990 DATA 00,00,00,00,00,00,00,00,00,00,02,01,00,00,00,00,7AA1  
995 DATA 00,00,00,00,00,00,00,00,00,00,00,08,00,01,01,00,00,7AAB  
1000 DATA 00,00,03,01,01,01,00,00,00,01,01,01,01,01,01,00,7AB7  
1005 DATA 00,00,01,01,03,01,00,00,00,01,01,01,06,00,02,00,7AC8  
1010 DATA 00,00,01,01,01,01,00,00,00,05,05,05,05,00,01,00,7AE1  
1015 DATA 00,00,06,01,01,01,00,00,00,00,00,00,00,00,01,00,7AEB  
1020 DATA 00,00,01,01,01,01,03,01,01,06,01,02,01,01,01,00,7B00  
1025 DATA 00,00,08,01,01,01,06,01,01,01,01,01,01,00,00,00,7B17  
1030 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,7B17  
1035 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,7B17  
1040 DATA 00,07,00,00,01,01,01,01,08,01,00,01,01,01,01,00,7B2F  
1045 DATA 00,01,05,01,06,00,00,00,00,00,01,00,01,00,00,01,00,7B3F  
1050 DATA 00,00,00,01,00,01,01,01,00,01,00,01,00,01,01,00,7B47  
1055 DATA 00,01,01,01,00,01,00,01,00,05,00,01,00,00,00,00,7B52  
1060 DATA 00,01,08,00,01,01,00,01,00,04,01,01,01,01,01,00,7B67  
1065 DATA 00,01,01,00,01,00,00,01,00,00,00,00,00,00,00,01,00,7B6C  
1070 DATA 00,00,01,00,01,00,01,01,02,01,01,01,01,02,01,00,7B79  
1075 DATA 00,01,05,01,01,00,01,00,01,01,00,00,01,01,01,00,7B87  
1080 DATA 00,00,00,00,01,00,01,00,01,01,00,00,01,00,00,00,7B8C  
1085 DATA 00,01,01,01,01,00,01,01,00,01,00,01,01,01,01,00,7B97  
1090 DATA 00,01,00,03,00,00,00,01,00,01,00,03,01,00,00,00,7BA1  
1095 DATA 00,01,00,01,00,01,06,01,00,01,05,05,01,01,01,00,7BB9  
1100 DATA 00,01,00,05,00,01,00,01,01,00,00,00,00,00,01,00,7BC3  
1105 DATA 00,01,00,00,00,01,00,00,01,00,01,08,01,00,05,00,7BD5  
1110 DATA 00,01,01,01,01,04,00,01,01,01,01,01,01,01,01,00,7BE5



1705 DATA 00,00,00,00,01,01,01,08,08,01,01,01,00,00,00,00,82C3  
1710 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,82C3  
1715 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,82C3  
1720 DATA 00,01,01,02,01,01,07,00,00,06,01,01,00,00,04,00,82DC  
1725 DATA 00,01,00,00,00,00,01,06,00,00,01,00,00,00,01,00,82E6  
1730 DATA 00,06,00,01,01,00,01,01,01,00,03,00,00,06,01,00,82FB  
1735 DATA 00,00,00,05,06,00,00,00,00,00,00,05,00,00,00,00,830B  
1740 DATA 00,00,00,00,00,00,06,00,01,00,00,01,01,06,00,00,831A  
1745 DATA 00,01,01,01,00,00,03,00,01,05,00,01,00,00,05,00,832C  
1750 DATA 00,01,01,01,06,00,01,00,00,01,00,04,00,01,01,00,833D  
1755 DATA 00,00,00,00,00,00,01,00,00,06,00,00,00,00,01,00,8345  
1760 DATA 00,01,05,00,00,00,00,00,00,00,00,00,00,06,02,00,8353  
1765 DATA 00,01,00,00,00,01,01,01,00,01,01,01,00,00,00,00,835A  
1770 DATA 00,01,01,06,00,01,00,01,00,00,00,06,00,08,01,00,837B  
1775 DATA 00,05,00,00,00,06,00,05,00,01,01,05,00,01,00,00,8393  
1780 DATA 00,00,00,00,00,00,00,00,00,00,05,00,00,00,01,06,00,839F  
1785 DATA 00,06,00,01,01,05,00,04,06,00,00,05,00,01,00,00,83BC  
1790 DATA 00,01,01,08,00,00,01,01,00,00,06,05,00,01,02,00,83D6  
1795 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,83D6  
1800 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,83D6  
1805 DATA 00,00,05,05,05,00,00,05,05,05,00,00,05,00,05,00,83FE  
1810 DATA 00,00,05,00,00,00,00,00,00,05,00,00,00,05,00,05,00,8412  
1815 DATA 00,00,05,05,00,00,00,00,00,05,00,00,00,05,05,05,00,8430  
1820 DATA 00,00,05,00,00,00,00,00,05,05,05,00,00,05,00,05,00,844E  
1825 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,844E  
1830 DATA 00,07,01,01,01,01,01,01,01,01,01,01,01,01,01,01,00,8462  
1835 DATA 00,01,02,02,01,01,01,03,03,01,01,01,04,04,01,00,847C  
1840 DATA 00,01,01,02,01,01,01,01,03,01,01,01,01,04,01,00,8490  
1845 DATA 00,01,01,01,01,01,01,01,01,01,01,01,01,01,01,00,849E  
1850 DATA 00,01,01,01,01,01,01,00,00,00,00,01,01,01,01,00,84A8  
1855 DATA 00,00,01,01,01,01,01,00,00,00,00,01,01,01,08,00,84C0  
1860 DATA 00,01,01,01,01,01,01,01,01,01,01,01,01,01,01,00,84CE  
1865 DATA 00,01,01,01,01,01,01,01,01,01,01,01,01,01,00,84DC  
1870 DATA 00,01,05,05,01,01,01,05,05,01,01,01,05,05,01,00,8502  
1875 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,8502  
1880 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,8502  
1885 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,8502  
1890 DATA 00,00,00,00,00,00,01,01,05,01,01,00,00,00,00,00,850B  
1895 DATA 00,00,00,00,00,00,01,01,05,01,01,00,00,00,00,00,8514  
1900 DATA 00,00,00,01,00,01,06,01,05,01,01,00,00,00,00,00,852C  
1905 DATA 00,00,00,01,01,01,01,01,01,01,01,00,00,00,00,00,8534  
1910 DATA 00,00,00,01,01,07,01,01,01,01,01,00,00,00,00,00,8542  
1915 DATA 00,00,00,00,00,00,01,01,01,01,01,00,00,00,00,00,8547  
1920 DATA 00,00,00,00,00,00,01,01,02,02,02,00,00,00,00,00,854F  
1925 DATA 00,00,00,00,00,00,01,01,01,01,01,00,00,00,00,00,8554  
1930 DATA 00,00,00,00,00,00,01,01,01,01,01,00,00,00,00,00,8559  
1935 DATA 00,00,00,00,00,00,04,04,01,01,01,00,00,00,00,00,8564  
1940 DATA 00,00,00,00,00,00,01,01,01,01,01,00,00,00,00,00,8569  
1945 DATA 00,00,00,00,00,00,01,01,01,01,01,00,00,00,00,00,856E  
1950 DATA 00,08,01,01,01,01,03,03,01,01,01,01,01,01,00,00,8586  
1955 DATA 00,01,01,01,01,01,01,01,01,01,01,06,01,01,00,00,8598  
1960 DATA 00,01,01,01,05,05,01,01,01,01,01,01,05,05,00,00,85B5  
1965 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,85B5  
1970 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,85B5  
1975 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,85B5  
1980 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,85B5  
1985 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,85B5  
1990 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,85B5  
1995 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,85B5

2000 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,85B5  
2005 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,85B5  
2010 DATA 7C,C6,08,67,D0,D5,11,50,C0,19,D1,C9,00,DD,5E,00,8D1A  
2015 DATA DD,56,01,DD,6E,02,DD,66,03,06,14,C5,E5,06,05,1A,92CA  
2020 DATA 77,13,23,10,FA,E1,CD,00,A0,C1,10,EF,C9,00,00,00,9958  
2025 DATA DD,5E,00,DD,56,01,DD,6E,02,DD,66,03,06,14,CD,19,9F5A  
2030 DATA BD,C5,E5,06,05,1A,E6,AA,28,06,4F,7E,E6,55,B1,77,A6D4  
2035 DATA 1A,E6,55,28,06,4F,7E,E6,AA,B1,77,23,13,10,E6,E1,ADE9  
2040 DATA CD,00,A0,C1,10,DB,C9,00,00,00,00,11,00,A4,DD,6E,B3CB  
2045 DATA 00,DD,66,01,06,14,C5,E5,06,05,7E,12,13,23,10,FA,B8AE  
2050 DATA E1,CD,00,A0,C1,10,EF,C9,00,00,00,11,00,A4,DD,6E,BF85  
2055 DATA 00,DD,66,01,06,14,C5,E5,06,05,1A,77,23,13,10,FA,C469  
2060 DATA E1,CD,00,A0,C1,10,EF,C9,00,00,00,9F,E0,E6,06,10,CBBB  
2065 DATA 2A,AC,A0,E5,2A,AA,A0,E5,C5,CD,C8,A0,C1,10,F9,E1,D614  
2070 DATA 22,AA,A0,E1,22,AC,A0,C9,2A,AA,A0,7E,FE,00,C2,D7,DF21  
2075 DATA A0,11,20,88,C3,C9,A1,FE,01,C2,E2,A0,11,84,88,C3,E7CA  
2080 DATA C9,A1,FE,02,C2,ED,A0,11,E8,88,C3,C9,A1,FE,03,C2,F1F4  
2085 DATA F8,A0,11,4C,89,C3,C9,A1,FE,04,C2,03,A1,11,B0,89,FA51  
2090 DATA C3,C9,A1,FE,05,C2,0E,A1,11,14,8A,C3,C9,A1,FE,06,02D2  
2095 DATA C2,19,A1,11,78,8A,C3,C9,A1,FE,07,C2,24,A1,11,40,0A6B  
2100 DATA 8B,C3,C9,A1,FE,08,C2,38,A1,CD,E2,A1,11,DC,8A,2A,13B5  
2105 DATA AC,A0,CD,3C,A0,C3,CF,A1,FE,09,C2,46,A1,CD,E2,A1,1DDD  
2110 DATA 11,A4,8B,C3,2F,A1,FE,0A,C2,54,A1,CD,E2,A1,11,6C,263C  
2115 DATA 8C,C3,2F,A1,FE,0B,C2,62,A1,CD,E2,A1,11,08,8C,C3,2EE1  
2120 DATA 2F,A1,FE,0C,C2,70,A1,CD,E2,A1,11,D0,8C,C3,2F,A1,37DE  
2125 DATA FE,0D,C2,7E,A1,CD,E2,A1,11,34,8D,C3,2F,A1,FE,0E,408B  
2130 DATA C2,8C,A1,CD,E2,A1,11,8C,8F,C3,2F,A1,FE,0F,C2,A6,49FE  
2135 DATA A1,11,E8,88,2A,AC,A0,CD,19,A0,11,14,8A,2A,AC,A0,5141  
2140 DATA CD,3C,A0,C3,CF,A1,FE,10,C2,B7,A1,11,4C,89,2A,AC,5A01  
2145 DATA A0,CD,19,A0,C3,9A,A1,FE,11,C0,11,B0,89,2A,AC,A0,62B4  
2150 DATA CD,19,A0,C3,9A,A1,C3,C9,A1,2A,AC,A0,CD,19,A0,2A,6B8B  
2155 DATA AA,A0,23,22,AA,A0,2A,AC,A0,23,23,23,23,22,AC,7157  
2160 DATA A0,C9,11,84,88,2A,AC,A0,CD,19,A0,C9,21,00,C0,22,78A5  
2165 DATA AC,A0,21,70,9E,22,AA,A0,06,09,C5,2A,AC,A0,E5,CD,8088  
2170 DATA AE,A0,2A,AA,A0,11,10,00,ED,5A,22,AA,A0,E1,06,14,8719  
2175 DATA C5,CD,00,A0,C1,10,F9,22,AC,A0,C1,10,DD,C9,21,00,8F1B  
2180 DATA C0,22,AC,A0,21,F0,9E,22,AA,A0,C3,F8,A1,00,00,00,96C0  
2185 DATA DD,6E,00,DD,66,01,E5,DD,6E,02,DD,66,03,E5,11,00,9DBD  
2190 DATA A4,CD,74,A0,E1,D1,CD,3C,A0,C9,7C,D6,08,67,CB,74,A766  
2195 DATA C0,D5,11,B0,3F,19,D1,C9,00,00,DD,6E,00,DD,66,01,AE3D  
2200 DATA 22,58,A2,C9,2A,58,A2,CD,4A,A2,CD,4A,A2,CD,4A,A2,B671  
2205 DATA CD,4A,A2,22,58,A2,2A,58,A2,11,00,A4,CD,74,A0,2A,BD2A  
2210 DATA 58,A2,DD,5E,00,DD,56,01,CD,3C,A0,C9,2A,58,A2,11,C43A  
2215 DATA 00,A4,C3,9A,A0,2A,58,A2,CD,00,A0,CD,00,A0,CD,00,CBA0  
2220 DATA A0,CD,00,A0,C3,73,A2,C9,DD,6E,00,DD,66,01,11,70,D35E  
2225 DATA 9E,01,10,01,ED,B0,C9,00,00,00,00,00,00,00,00,00,D674  
2230 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,D674  
3000 CLS:PRINT " FICHER LANGAGE MACHINE CORRECT..."  
3010 PRINT:PRINT  
3020 PRINT "UNE TOUCHE POUR LA SAUVEGARDE..."  
3030 SAVE "CHALIBIN",B,88820,11AA0  
3040 REM :: POUR LES CPC 464 SANS D7 LE FICHER \*CHALIBIN\* DEV  
RA SE TROUVER A LA SUITE DU FICHER \*CHARLIE\* SUR VOTRE K7  
3050 REM :: EN CAS DE MSG D'ERREUR DU GENRE \*TYPE MISMATCH IN  
10 \* L'ERREUR EST DANS LA LIGNE DE DATA SUIVANT CELLE QUI A ET  
E DECLAREE CORRECTE  
3060 REM :: EXEMPLE .....LA VERIFICATION MONTE JUSQU'A LA LI  
GNE 200 PUIS VOUS AVEZ \*TYPE MISMATCH IN 10 \* VOTRE ERREUR EST  
DANS LA LIGNE 205 ■

# ECRITURE ARABE

Michel la ROSA

Valable pour CPC 464 - 664 - 6128

Voici un petit programme utilitaire qui pourrait intéresser tous nos amis de langue et de culture arabe, mais aussi tous ceux qui souhaitent comme moi s'y initier.

J e l'ai appelé Gerbert, en hommage au premier lettré français qui est allé au Xème siècle renouveler notre science balbutiante en Catalogne, haut lieu de culture arabe à l'époque (et dans le Midi, c'étaient souvent les juifs, polyglottes de tradition, qui apprenaient l'arabe aux français !). Il faut savoir que Gerbert, après avoir contribué à mettre un prince illettré (comme les autres !) sur le trône de France (Hugues Capet), a fait de son fils Robert (et de plusieurs prince de sa génération) un lettré accompli. Enfin c'est lui le pape de l'An Mil. Voilà pour l'évocation.

Ce programme utilise le clavier français pour transcrire la calligraphie arabe, en suivant un code représenté dans la figure 1. En l'utilisant, vous découvrirez quelques signes supplémentaires, par exemple les parenthèses, transformables en guillemets de début et de fin de citation, en tapant COPY avant le signe parenthèse.

Le curseur se déplace au moyen des 4 flèches de direction, de ESPA-

CE et de DEL. Attention ! Il efface tout sur son passage !

CLR permet de recommencer la page à 0.

TAB avance le curseur au milieu de la page, s'il était en deça.

On peut noter la façon d'obtenir les 4 formes de chaque lettre arabe suivant sa place dans le mot :

- Deux fois COPY puis la lettre si elle est en position isolée (1)
- COPY puis la lettre en position finale (2)
- La lettre seule si elle est en position (3)
- SHIFT puis la lettre en position initiale (4)

On peut aussi noter :

- Les 4 positions du hamza et le moyen de les obtenir, quelques variantes calligraphiques courantes dans l'arabe écrit, les signes en surimpression (hamza, shadda, fatha, damma, kasra), l'utilisation de la barre de soulignement pour prolonger la ligne d'écriture dans le cours d'un mot si on le désire.

Pour ceux que cela intéresse, il est possible de modifier l'allure des

lettres, que j'ai choisies en pleins et en déliés, parce qu'elle passe mieux à l'écran et qu'on la trouve souvent ainsi dans la presse arabe (les titres notamment). Il est également possible de modifier les datas pour obtenir des caractères complémentaires.

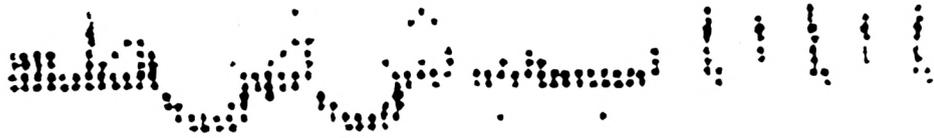
J'ai passé plus de 3 mois à ce travail, mais j'ai conscience qu'il peut être amélioré ; le choix du MODE 2 donne une plus belle lisibilité à l'écran, mais diminue la clarté à l'impression ( en tapant CONTROL puis COPY).

Enfin ce programme est bilingue : on peut passer du français à l'arabe et réciproquement en tapant COPY puis RETURN. Vous verrez que l'usage du tabulateur TAB permet, au choix, de placer les textes arabes à droite et français à gauche (le plus pratique, surtout si le texte est long) ou l'inverse (vocabulaire ou conjugaisons et leur traduction, par exemple).

Taper et sauver les fichiers ARABE1.PRG et ARABE2.PRG, qui génèrent les fichiers ARABE1.BIN et ARABE2.BIN, puis taper et lancer le programme GERBERT.

(1=copy\*2, 2=copy, 4=shift)

fig- 1

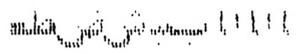


```
10 ' #####
15 ' #
20 ' # ECRITURE ARABE #
25 ' #
30 ' # par #
35 ' #
40 ' # Michel LA ROSA #
45 ' #
50 ' # ->55.74.26.12 #
55 ' #
60 ' # Juin 1991 #
65 ' #
70 ' #####
80 '
90 ' Préparation Imprimante
100 '
110 MODE 2:MEMORY &9AE2
120 ad=&9AE3:nl=170:RESTORE 170
130 FOR a=1 TO 22:b=0:FOR c=1 TO 16:READ a$
140 POKE ad,VAL("&"a$):b=b+PEEK(ad):ad=ad+1
150 NEXT:READ e:IF e<>b THEN PRINT"Erreur DATA ligne"nl >MR
:END
160 nl=nl+10:NEXT:CALL HIMEM+1 >ZJ
170 DATA 21,E1,E9,22,30,00,F7,EB,21,49,01,19,4E,23,46,7 >DD
9,1491
180 DATA B0,28,15,E5,60,69,19,E5,4E,23,46,60,69,19,44,4 >DP
D,1475
190 DATA E1,71,23,70,E1,23,18,E4,21,49,00,06,81,0E,00,1 >DY
1,1269
200 DATA 2F,00,CD,D7,BC,C9,C5,D5,E5,F5,3E,09,CD,1E,BB,2 >FD
8,2273
210 DATA 0A,3E,17,CD,1E,BB,28,03,CD,59,00,F1,E1,D1,C1,C >EM
9,1923
220 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00 >YT
0,0
230 DATA DD,21,52,00,DD,36,01,00,DD,36,02,00,DD,36,03,9 >DP
0,1311
240 DATA DD,36,04,01,DD,36,05,00,DD,36,00,00,3E,1B,CD,C >EG
C,1333
250 DATA BB,D5,E5,CD,2B,BD,3E,41,CD,2B,BD,3E,06,CD,2B,B >FZ
D,2135
260 DATA CD,2E,BD,38,FB,3E,1B,CD,2B,BD,3E,4B,CD,2B,BD,3 >GY
E,1909
270 DATA 40,CD,2B,BD,3E,01,CD,2B,BD,DD,36,00,00,DD,36,0 >EY
6,1557
280 DATA 00,DD,CB,00,26,DD,5E,01,DD,56,02,DD,6E,03,DD,6 >EX
6,1744
>LA
290 DATA 04,CD,F0,BB,B7,28,04,DD,CB,00,C6,DD,34,06,DD,7 >FR
E,2111
>LF
300 DATA 06,FE,07,28,12,DD,6E,03,DD,66,04,2B,2B,DD,75,0 >ER
3,1413
>LB
310 DATA DD,74,04,18,CC,18,A9,CD,2E,BD,38,FB,DD,7E,00,C >FY
D,2061
>LG
320 DATA 2B,BD,DD,34,01,DD,34,01,20,03,DD,34,02,DD,7E,0 >EM
1,1438
>LC
330 DATA D6,02,20,07,DD,7E,02,FE,02,28,12,DD,6E,03,DD,6 >EC
6,1703
>LH
340 DATA 04,11,0C,00,19,DD,75,03,DD,74,04,18,8C,DD,34,0 >DW
5,1182
>LD
350 DATA 3E,0A,CD,2B,BD,3E,0D,CD,2B,BD,3E,15,CD,1E,BB,2 >FC
0,1558
>LE
360 DATA 0F,DD,36,01,00,DD,36,02,00,DD,7E,05,FE,22,20,A >EZ
5,1405
>LF
370 DATA 3E,1B,CD,2B,BD,3E,40,CD,2B,BD,E1,D1,CD,C9,BB,C >FG
9,2317
>LG
380 DATA 22,00,29,00,42,00,5B,00,00,00,00,00,00,00,0 >AX
0,232
>LH
390 ' >TC
400 ' Routine de Dessin des Lettres >RE
410 ' >RF
420 MEMORY &2FFF >LW
430 ad=&4700:nl=480:RESTORE 480 >ZV
440 FOR a=1 TO 22:b=0:FOR c=1 TO 16:READ a$ >GU
450 POKE ad,VAL("&"a$):b=b+PEEK(ad):ad=ad+1 >MA
460 NEXT:READ e:IF e<>b THEN PRINT"Erreur DATA ligne"nl >MW
:END
470 nl=nl+10:NEXT >MY
480 DATA ED,5B,A0,45,2A,9C,45,3A,A7,45,47,C5,D5,3A,A6,4 >EL
5,1892
490 DATA 4F,06,00,ED,B0,EB,E1,CD,26,BC,EB,C1,10,ED,C9,0 >FD
0,2271
500 ' >RF
510 ' Lancement Programme >RG
520 ' >RH
530 INK 0,13:INK 1,0:BORDER 10 >XE
540 LOAD"arabe1.bin",&3000:LOAD"arabe2.bin",&3A10 >ZB
550 ecr=49232:lign=1 >PW
560 ecr=ecr-1:GOSUB 1150 >TU
570 ' Dessin du Curseur >TC
580 POKE &45A6,1:POKE &45A7,16 >XT
590 POKE &459C,0:POKE &459D,52 >XD
600 POKE &45A0,ecr-256*INT(ecr/256):POKE &45A1,INT(ecr/ >EA
```

```

256)
610 CALL &4700 >HJ
620 a$=INKEY$:IF a$="" THEN 620 >XY
630 ' Effacement du Curseur >RK
640 POKE &459C,16 >LM
650 POKE &45A0,ecr-256*INT(ecr/256):POKE &45A1,INT(ecr/ >EF
256)
660 CALL &4700 >JD
670 a=ASC(a$):e=ecr-49072:x=(e MOD 80)*8+7:y=414-INT(e/ >FP
80)*16
680 IF a=9 THEN IF ecr>49028+lign*160 THEN ecr=49028+li >EC
gn*160:GOTO 580 ELSE a=13
690 IF a=13 THEN IF fl=0 THEN lign=lign+1:ecr=49072+lig >EY
n*160:GOTO 1000 ELSE 1280
700 IF a=16 THEN CLS:GOTO 550 >WA
710 IF a=32 THEN 1000 >NK
720 IF a=34 THEN PLOT x,y:GOSUB 1200:GOTO 580 >MK
730 IF a>34 AND a<48 THEN IF fl>0 THEN a=a+32 >JN
740 IF a=60 THEN IF fl=0 THEN PLOT x+5,y-2:GOSUB 1220:G >QG
OTO 580 ELSE 880
750 IF a=62 THEN IF fl=0 THEN PLOT x,y-30:GOSUB 1230:GO >PW
TO 580 ELSE 880
760 IF a=64 AND fl=0 THEN PLOT x+6,y:GOSUB 1210:GOTO 58 >YV
0
770 IF a=92 AND fl=0 THEN PLOT x+6,y-24:GOSUB 1210:GOTO >CE
580
780 IF a=94 THEN PLOT x,y-2:GOSUB 1230:IF fl=0 THEN 580 >DG
ELSE GOSUB 1240:GOTO 1000
790 IF a=127 THEN ecr=ecr+2:GOSUB 1150:GOTO 560 >PF
800 IF a=224 THEN fl=fl+1:GOTO 620 >AX
810 IF a<240 THEN 880 >NA
820 a=a+fl*4 >BC
830 IF a=240 AND ecr>49231 THEN ecr=ecr-159:lign=lign-1 >KP
:GOTO 560
840 IF a=241 AND ecr<50912 THEN ecr=ecr+161:lign=lign+1 >KA
:GOTO 560
850 IF a=242 THEN 560 >NC
860 IF a=243 THEN ecr=ecr+2:GOSUB 1150:GOTO 560 >PC
870 GOTO 920 >AC
880 IF fl=1 THEN IF a>96 THEN a=a-96 ELSE a=a+128 >PL
890 IF fl>1 THEN a=a+64 >QU
900 IF a=199 THEN PLOT x+6,y-6:GOSUB 1210:GOTO 1000 >UD
910 ' Dessin du Caractere >TA
920 RESTORE 1040:buf=12280 >VP
930 FOR i=1 TO a-1:READ lar:buf=buf+lar*16:NEXT:READ lg >WK
:IF lg=0 THEN 580
940 POKE &45A6,lg >LH
950 ecr=ecr-lg+1:GOSUB 1150:IF ecr=49071+lign*160 THEN >NG
ecr=ecr-lg
960 POKE &459C,buf-256*INT(buf/256):POKE &459D,INT(buf/ >FK
256)
970 POKE &45A0,ecr-256*INT(ecr/256):POKE &45A1,INT(ecr/ >EL
256)
980 CALL &4700 >JJ
990 IF a=7 OR a=9 OR a=10 OR a=15 OR a=24 OR a=167 OR a >WQ
=169 OR a=170 OR a=175 OR a=184 THEN PLOT x,y-30:DRAWR
+7,0
1000 fl=0:GOTO 560 >MC
1010 ' >XE
1020 ' Largeur des Lettres à dessiner >XF
1030 ' >XG
1040 DATA 1,3,3,2,3,3,2,2,2,2,3,2,2,2,2,3,2,2,3,2,2,2,2 >RJ
,2,2,2,3,0,2,0,1,1
1050 DATA 1,0,2,2,1,2,1,2,2,1,2,1,2,1,2,2,2,2,2,2,2,2 >RQ
,2,2,1,1,0,2,0,2,0
1060 DATA 1,1,2,2,2,1,2,2,2,2,2,1,2,1,2,1,1,2,2,1,2,2,2 >RQ
,2,1,2,1,0,2,0,1,0
1070 DATA 1,1,2,2,2,2,2,2,2,2,2,1,2,1,2,1,2,2,2,1,2,2,2 >RX
,2,1,2,2,1,2,0,0,2
1080 DATA 0,0 >QC
,0,0,0,0,2,0,0,0,0
1090 DATA 1,3,3,2,3,3,2,2,2,2,2,2,2,2,2,3,2,2,3,3,2,2,2 >RQ
,2,2,2,3,1,2,1,0,2
1100 DATA 0,0,2,2,0,0,0,2,2,1,0,0,0,0,2,0,0,0,0,0,0,0,0 >QH
,0,0,0,1,2,0,0,0,0
1110 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,2,2 >NW
,2,2,2,2,0,0,0,0
1120 ' >XG
1130 ' Dessins de Caractères en surimpression >XH
1140 ' >XJ
1150 IF ecr<48992+lign*160 THEN lign=lign+1:ecr=49071+l >JZ
ign*160
1160 IF ecr>49071+lign*160 THEN lign=lign-1:ecr=48992+l >JE
ign*160
1170 IF lign=1 AND ecr>49231 THEN PRINT CHR$(7):GOTO 55 >ZP
0
1180 IF ecr>50992 OR lign>12 THEN ecr=50912:PRINT CHR$(>MJ
7):GOTO 580
1190 RETURN >FE
1200 DRAWR +1,0:FOR i=1 TO 2:DRAWR 0,-2:DRAWR +3,0:DRAW >NE
R 0,+2:DRAWR +1,0:NEXT:RETURN
1210 DRAWR -2,0:DRAWR 0,-4:DRAWR -1,0:DRAWR +3,0:RETURN >AU
1220 DRAWR -2,0:DRAWR 0,+2:DRAWR +3,0:DRAWR 0,-4:DRAWR >UQ
-1,0:DRAWR 0,-2:DRAWR -4,0:RETURN
1230 DRAWR +7,+2:RETURN >TR
1240 DRAWR 0,-4:DRAWR -7,-2:RETURN >CC
1250 ' >YA
1260 ' Retour aux Caractères Clavier >YB
1270 ' >YC
1280 x=1:y=INT((ecr-49072)/80) >YW
1290 LOCATE x,y:PRINT CHR$(140) >ZV
1300 a$=INKEY$:IF a$="" THEN 1300 >YB
1310 a=ASC(a$):LOCATE x,y:PRINT CHR$(32) >HC
1320 IF a=9 THEN IF x<42 THEN x=42:GOTO 1290 ELSE x=1:y >PG
=y+1:GOTO 1290
1330 IF a=13 THEN IF fl=0 THEN x=1:y=y+1:GOTO 1290 ELSE >BL
1000
1340 IF a=16 THEN CLS:x=1:y=1:GOTO 1280 >FX
1350 IF a=224 THEN fl=fl+1:GOTO 1300 >CC
1360 IF a=240 AND y>1 THEN y=y-1:GOTO 1290 >HQ
1370 IF a=241 AND y<40 THEN y=y+1:GOTO 1290 >JD
1380 IF a=242 OR a=127 THEN 1420 >XB
1390 IF a=243 OR a=32 THEN 1410 >WW
1400 IF a<240 THEN LOCATE x,y:PRINT CHR$(a) >KG
1410 IF x<80 THEN x=x+1:GOTO 1430 ELSE x=1:y=y+1:GOTO 1 >BT
430
1420 IF x>1 THEN x=x-1 ELSE x=80:y=y-1 >EV
1430 fl=0:GOTO 1290 >NA

```



```

10 ' #####
20 ' #
30 ' # ECRITURE ARABE #
40 ' #
50 ' #####
60 '
70 MODE 2:ad=&3000:nl=110
75 FOR a=1 TO 161:b=0:FOR c=1 TO 16:READ a$
80 POKE ad,VAL("&"a$):b=b+PEEK(ad):ad=ad+1
85 NEXT:READ e:IF e<>b THEN 95
90 nl=nl+10:NEXT:SAVE"arabe1.bin",b,&3000,&A10:CLS:END
95 CLS:PRINT"ligne"nl;" incorrecte":END
100 '
105 ' LETTRES ARABES (1ere partie)
110 DATA 00,00,00,18,18,18,18,18,18,1F,07,00,00,00,00,00,182
120 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
130 DATA 00,00,40,00,00,00,C0,00,00,F0,00,0C,FF,FF,FF,3F,FF,1591
140 DATA F3,00,00,00,00,00,00,00,18,00,00,00,00,00,00,267
150 DATA 00,00,00,00,00,00,00,00,00,00,00,02,20,00,162
160 DATA 00,00,00,00,00,00,00,00,30,CC,00,1F,FF,40,0F,617
170 DATA 33,C0,0C,00,E0,3C,00,7F,F8,00,1F,E0,00,00,00,00,1169
180 DATA 00,00,00,00,00,00,00,00,00,00,18,00,18,00,0C,60
190 DATA 00,0C,0F,FF,0F,F3,00,00,00,00,00,00,00,00,00,540
200 DATA 00,00,00,00,00,00,00,00,00,00,18,00,00,00,00,24
210 DATA 00,00,00,00,1E,00,00,37,00,30,C7,00,1F,FF,40,0F,697
220 DATA FF,C0,0D,80,E0,3C,00,7F,F8,00,1F,E0,00,00,00,00,1502
230 DATA 00,00,00,00,00,00,00,00,00,00,00,C0,00,00,00,00,192
240 DATA 00,00,40,00,78,C0,00,CC,F0,00,EC,FF,FF,FF,3F,FF,2139
250 DATA DF,00,00,00,00,00,00,00,00,00,00,00,00,00,00,223
260 DATA 00,00,00,00,00,00,03,00,00,00,00,00,07,E0,06,60,336
270 DATA 03,C0,1E,7F,70,1F,C0,00,E0,00,F8,00,7F,C0,0F,FF,1748
280 DATA 00,00,00,00,00,00,00,00,00,00,18,00,F8,03,98,427
290 DATA 0E,3C,0F,FF,03,E7,00,00,00,00,00,00,00,00,00,578
300 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,7F,E0,03,FE,608
310 DATA 0F,98,7C,1F,E0,07,C0,00,E0,00,F8,00,7F,C0,0F,FF,1806
320 DATA 00,00,00,00,00,00,00,00,00,00,00,00,7F,E0,03,FE,608
330 DATA 0F,98,7C,1F,E0,07,C0,00,E0,00,F8,00,7F,C0,1F,FF,2014
340 DATA 00,00,00,00,00,00,00,00,00,00,00,00,0C,00,00,0C,00,24
350 DATA 1C,0C,40,10,0C,C0,3C,0C,F0,00,0C,FF,FF,FF,3F,FF,1731
360 DATA F3,00,00,00,00,00,00,00,00,00,00,00,00,00,00,243
370 DATA 00,00,00,00,00,00,00,06,00,06,00,06,00,06,00,30
380 DATA 00,06,00,07,20,07,60,06,70,1E,3F,FC,0F,F0,00,00,866
390 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,01,F0,241
400 DATA 03,3C,1F,FF,3F,F3,30,00,30,00,30,00,30,00,895
410 DATA 00,00,00,00,00,00,00,00,00,00,01,80,00,00,00,129
420 DATA 00,18,00,0F,20,07,60,06,70,1E,3F,FC,0F,F0,00,00,892
430 DATA 00,00,00,00,00,00,00,00,00,00,00,00,07,E0,06,60,333
440 DATA 03,C0,1E,7F,70,1F,C0,00,E0,00,F8,00,7F,C0,0F,FF,1748
450 DATA 00,00,00,00,00,00,00,00,00,00,08,00,00,22,00,00,42
460 DATA 00,00,40,00,00,C0,00,00,F0,00,0C,FF,FF,FF,3F,FF,1591
470 DATA F3,00,00,00,00,00,00,00,00,00,00,00,00,00,00,243
480 DATA 00,00,00,00,00,00,01,98,00,00,00,00,00,38,00,6C,317
490 DATA 00,7C,40,3F,C0,0F,E0,3C,7F,F8,1F,E0,00,00,00,00,1372
500 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
510 DATA 00,18,00,0F,00,0F,00,38,3F,F0,0F,C0,00,00,00,00,620
520 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
530 DATA 00,00,00,00,00,00,00,00,00,00,30,CC,00,1F,FF,40,0F,617
540 DATA 33,C0,0C,00,E0,3C,00,7F,F8,00,1F,E0,00,00,00,00,1169

```

```

550 DATA 00,00,00,00,00,00,00,00,00,00,66,00,00,00,00,102
560 DATA 00,00,40,00,00,C0,00,00,F0,00,0C,FF,FF,FF,3F,FF,1591
570 DATA F3,00,00,00,00,00,00,00,00,00,00,00,00,00,243
580 DATA 00,00,00,00,00,00,18,00,18,00,18,00,18,78,19,CE,447
590 DATA 1F,0E,FF,FF,FF,FF,00,00,00,00,00,00,00,00,1065
600 DATA 00,00,00,00,00,00,00,00,30,00,00,00,18,00,18,00,0C,108
610 DATA 00,0C,0F,FF,0F,F3,00,00,00,00,00,00,00,00,00,540
620 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,38,56
630 DATA 00,6C,00,7F,00,0F,00,38,3F,F0,0F,C0,00,00,00,00,816
640 DATA 00,00,00,00,00,00,00,03,00,00,00,00,00,7F,E0,03,FE,611
650 DATA 0F,98,7C,1F,E0,07,C0,00,E0,00,F8,00,7F,C0,0F,FF,1806
660 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
670 DATA 00,00,10,7F,71,F1,F8,FC,FE,1E,3F,F8,00,00,06,60,1694
680 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,30,00,00,00,48
690 DATA 00,18,00,0F,00,0F,00,38,3F,F0,0F,C0,00,00,00,00,620
700 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
710 DATA 00,00,00,00,1E,00,00,37,00,30,C7,00,1F,FF,40,0F,697
720 DATA FF,C0,0D,80,E0,3C,00,7F,F8,00,1F,E0,00,00,00,00,1502
730 DATA 00,00,00,00,00,00,18,60,18,00,18,00,18,78,19,CE,543
740 DATA 1F,0E,FF,FF,FF,FF,00,00,00,00,00,00,00,00,1065
750 DATA 00,00,00,00,00,00,00,00,FF,FF,FF,FF,00,00,00,00,1020
760 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
770 DATA 00,00,00,3C,3C,3C,3C,18,18,00,3C,3C,00,00,00,00,408
780 DATA 00,00,00,00,03,00,03,00,03,00,03,00,7F,00,07,E0,370
790 DATA 01,FE,FF,80,F0,00,00,00,00,00,00,00,00,00,00,878
800 DATA 00,00,00,0C,00,00,00,30,00,D8,0F,0C,7F,00,07,E0,661
810 DATA 01,FE,FF,80,F0,00,00,00,00,00,00,00,00,00,00,878
820 DATA 00,00,00,00,00,00,00,00,18,FB,DF,1C,18,18,00,00,574
830 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,1E,00,77,149
840 DATA 30,C7,FF,FF,CF,FC,03,00,00,00,00,00,00,00,00,1219
850 DATA 00,00,00,00,00,00,00,00,1C,10,3C,00,00,00,00,00,104
860 DATA 00,00,00,00,00,00,00,00,F0,03,C0,07,80,0F,00,0F,600
870 DATA 07,80,03,C0,00,F0,00,00,00,00,00,00,00,00,00,570
880 DATA 00,00,00,00,00,00,0F,00,03,C0,01,E0,00,F0,00,F0,915
890 DATA 01,E0,03,C0,0F,00,00,00,00,00,00,00,00,00,00,435
900 DATA 00,00,00,18,18,18,18,18,18,FC,CE,00,00,00,00,00,602
910 DATA 00,00,00,00,00,00,00,00,00,00,00,00,03,C0,03,C0,390
920 DATA 3F,FC,03,C0,03,C0,00,00,00,00,00,00,00,00,00,705
930 DATA 00,00,00,00,00,00,00,00,1E,78,78,00,00,00,00,270
940 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
950 DATA 3F,FC,00,00,00,00,00,00,00,00,00,00,00,00,00,315
960 DATA 00,00,00,00,00,00,00,00,00,00,78,78,00,00,00,240
970 DATA 00,00,00,00,00,60,00,60,00,60,60,00,60,01,F0,721
980 DATA 03,3C,1F,FC,3F,F0,30,00,30,00,30,00,30,00,889
990 DATA 00,00,00,00,00,00,00,00,00,00,00,00,03,C0,03,C0,390
1000 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
1010 DATA 00,00,00,00,00,00,07,00,07,00,03,80,01,80,01,C0,467
1020 DATA 00,C0,00,E0,00,60,00,00,00,00,00,00,00,00,00,512
1030 DATA 00,00,00,00,00,00,18,18,0F,F0,07,E0,03,00,01,80,666
1040 DATA 00,C0,00,60,00,60,00,00,00,00,00,00,00,00,00,384
1050 DATA 00,00,00,00,00,00,18,CC,0F,BC,07,98,03,00,01,80,722
1060 DATA 00,C0,00,60,00,60,00,00,00,00,00,00,00,00,00,384
1070 DATA 00,00,00,00,00,00,00,70,01,C0,07,00,03,F0,00,E0,779
1080 DATA 03,80,0F,F8,07,F8,00,00,00,00,00,00,00,00,649
1090 DATA 00,00,00,00,00,00,00,00,07,E0,0C,F0,18,38,1C,18,615
1100 DATA 0F,30,07,E0,00,00,00,00,00,00,00,00,00,00,294
1110 DATA 00,00,00,00,00,00,18,60,0F,E0,07,E0,00,60,00,60,782
1120 DATA 00,30,00,30,00,18,00,00,00,00,00,00,00,00,120
1130 DATA 00,00,00,00,00,00,00,00,70,06,38,0C,1C,18,0E,30,300

```

```

1140 DATA 07,60,03,C0,01,80,00,00,00,00,00,00,00,00,00,427
1150 DATA 00,00,00,00,00,00,00,00,01,80,03,C0,06,E0,0C,70,678
1160 DATA 18,38,30,1C,60,0E,00,00,00,00,00,00,00,00,00,266
1170 DATA 00,00,00,00,00,00,07,C0,0C,C0,0F,E0,07,E0,00,30,921
1180 DATA 00,30,00,18,00,18,00,00,00,00,00,00,00,00,00,96
1190 DATA 00,00,00,00,00,00,3C,3C,00,00,3C,3C,00,00,00,00,240
1200 DATA 00,00,00,00,00,1E,78,78,00,00,78,78,00,00,00,00,510
1210 DATA 00,00,00,00,00,00,00,00,00,00,00,00,3F,FC,00,00,315
1220 DATA 00,00,3F,FC,00,00,00,00,00,00,00,00,00,00,00,00,315
1230 DATA 00,00,00,00,00,00,07,E0,1C,78,1E,38,0F,00,07,80,615
1240 DATA 01,80,00,80,03,C0,03,C0,00,00,00,00,00,00,00,00,647
1250 DATA 00,00,00,10,18,18,18,18,18,10,00,00,00,00,00,168
1260 DATA 00,00,00,00,00,00,00,18,18,F8,E0,00,00,18,00,00,544
1270 DATA 00,00,00,00,00,00,00,80,02,20,00,00,00,00,00,0C,174
1280 DATA 30,CC,FF,FC,CF,30,00,00,00,00,00,00,00,00,00,00,1014
1290 DATA 00,00,00,00,00,00,00,00,00,00,00,00,01,C0,00,78,313
1300 DATA 00,0C,0F,FC,0F,F0,00,00,00,00,00,00,00,00,00,00,534
1310 DATA 00,00,00,00,00,00,00,18,00,00,00,00,00,1E,00,77,173
1320 DATA 30,C7,FF,FF,CF,FC,03,00,00,00,00,00,00,00,00,00,1219
1330 DATA 00,00,00,60,00,00,0E,1B,1F,FE,FC,00,00,00,00,00,674
1340 DATA 00,00,00,00,00,00,01,80,00,00,00,00,07,80,0C,00,276
1350 DATA 0F,F0,FF,00,F0,00,00,00,00,00,00,00,00,00,00,00,750
1360 DATA 00,00,00,00,00,00,00,00,00,00,00,00,0F,80,19,F0,408
1370 DATA 3F,9C,FF,FC,F1,F8,00,00,00,00,00,00,00,00,00,00,1215
1380 DATA 00,00,00,00,00,00,00,00,00,00,00,00,7F,00,07,E0,358
1390 DATA 01,FE,FF,80,F0,00,00,00,00,00,00,00,00,00,00,00,878
1400 DATA 00,00,00,00,00,00,00,00,00,00,00,00,7F,00,07,E0,358
1410 DATA 01,FE,FF,80,F0,00,00,00,00,00,00,00,C0,00,00,00,00,1070
1420 DATA 00,00,00,00,00,00,3F,03,E0,3E,00,F0,00,3C,00,07,80,787
1430 DATA 00,C0,FF,80,FE,00,00,00,00,00,00,00,00,00,00,00,829
1440 DATA 00,00,00,1C,18,18,18,18,18,F8,E0,00,00,00,00,00,620
1450 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,01,F0,241
1460 DATA 07,98,FF,F8,F9,E0,00,00,00,00,00,00,00,00,00,00,1135
1470 DATA 00,00,00,18,00,00,00,18,18,F8,E0,00,00,00,00,00,544
1480 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,07,80,0C,00,147
1490 DATA 0F,F0,FF,00,F0,00,00,00,00,00,00,00,00,00,00,00,750
1500 DATA 00,00,00,20,88,00,00,18,18,F8,E0,00,00,00,00,00,688
1510 DATA 00,00,00,CC,00,00,0E,1B,1F,FE,FC,00,00,00,00,00,782
1520 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,18,24
1530 DATA 00,0C,00,0C,00,38,3F,F0,0F,C0,00,00,00,00,00,00,590
1540 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0C,12
1550 DATA 30,CC,FF,FC,CF,30,00,00,00,00,00,00,00,00,00,00,1014
1560 DATA 00,00,00,CC,00,00,00,18,18,F8,E0,00,00,00,00,00,724
1570 DATA 00,00,00,00,00,00,18,00,18,00,18,00,18,78,19,CE,447
1580 DATA 1F,0E,FF,FC,FF,F0,00,00,00,00,00,00,00,00,00,00,1047
1590 DATA 00,00,00,00,00,00,03,00,00,00,00,00,00,01,C0,00,78,316
1600 DATA 00,0C,0F,FC,0F,F0,00,00,00,00,00,00,00,00,00,00,534
1610 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,38,00,6C,164
1620 DATA 00,7C,00,0C,00,38,3F,F0,0F,C0,00,00,00,00,00,00,702
1630 DATA 00,00,00,00,00,00,03,00,00,00,00,00,7F,00,07,E0,361
1640 DATA 01,FE,FF,80,F0,00,00,00,00,00,00,00,00,00,00,00,878
1650 DATA 00,00,00,00,00,00,00,18,18,F8,E0,00,00,CC,00,00,724
1660 DATA 00,00,00,00,00,00,00,30,00,00,00,00,00,00,00,18,72
1670 DATA 00,0C,00,0C,00,38,3F,F0,0F,C0,00,00,00,00,00,00,590
1680 DATA 00,00,00,00,00,00,00,00,00,00,FC,F8,00,00,18,00,524
1690 DATA 00,00,00,00,00,00,18,60,18,00,18,00,18,78,19,CE,543
1700 DATA 1F,0E,FF,FC,FF,F0,00,00,00,00,00,00,00,00,00,00,1047
1710 DATA 00,00,00,00,00,00,00,00,00,FF,FF,00,00,00,00,00,510
1720 DATA 00,00,00,00,00,00,00,00,00,FF,FF,00,00,00,00,00,510

```

```

10 ' #####
20 ' # #
30 ' # ECRITURE ARABE #
40 ' # #
50 ' #####
60 '
70 MODE 2:ad=&3A10:nl=110
75 FOR a=1 TO 151:b=0:FOR c=1 TO 16:READ a$
80 POKE ad,VAL("&"a$):b=b+PEEK(ad):ad=ad+1
85 NEXT:READ e:IF e<>b THEN 95
90 nl=nl+10:NEXT:SAVE"arabe2.bin",b,&3A10,&970:CLS:END
95 CLS:PRINT"ligne"nl;" incorrecte":END
100 '
105 ' LETTRES ARABES (2de partie)
110 DATA 00,00,00,18,18,18,18,18,1F,07,00,00,00,00,00,182
120 DATA 00,00,00,00,00,00,00,00,18,FF,E7,00,00,18,00,00,534
130 DATA 00,00,00,00,00,00,00,00,80,02,20,00,00,00,00,00,162
140 DATA 30,CC,FF,FF,CF,33,00,00,00,00,00,00,00,00,00,00,1020
150 DATA 00,00,00,00,00,00,00,00,00,00,18,00,18,00,0C,60
160 DATA 00,0C,0F,FF,0F,F3,00,00,00,00,00,00,00,00,00,00,540
170 DATA 00,00,00,00,00,00,00,18,00,00,00,00,00,1E,00,77,173
180 DATA 30,C7,FF,FF,CF,FF,03,00,00,00,00,00,00,00,00,00,1222
190 DATA 00,00,00,00,00,00,06,00,00,00,00,00,03,C0,06,60,303
200 DATA 07,60,FF,FF,FE,FF,00,00,00,00,00,00,00,00,00,00,1122
210 DATA 00,00,00,00,00,00,03,00,00,00,00,00,07,E0,06,60,336
220 DATA 03,C0,FE,7F,F8,1F,00,00,00,00,00,00,00,00,00,00,855
230 DATA 00,00,00,00,00,00,00,00,00,00,00,00,01,E0,03,30,276
240 DATA 07,60,FF,CF,FF,7F,03,30,01,E0,00,00,00,00,00,00,1223
250 DATA 00,00,00,00,00,00,00,00,00,00,00,00,7F,00,07,E0,358
260 DATA 01,FE,FF,98,F0,0F,00,00,00,00,00,00,00,00,00,00,917
270 DATA 00,00,00,00,00,00,00,00,00,00,00,00,7F,00,07,E0,358
280 DATA 01,FE,FF,98,F0,0F,00,00,00,00,00,00,C0,00,00,00,00,1109
290 DATA 00,00,00,00,00,00,3C,03,E0,3E,00,F0,00,3C,00,07,80,784
300 DATA 00,F0,FF,9F,FE,07,00,00,00,00,00,00,00,00,00,00,915
310 DATA 00,00,00,18,18,18,18,18,18,FF,F1,00,00,00,00,00,640
320 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,03,C0,195
330 DATA 0E,70,FF,FF,F3,CF,00,00,00,00,00,00,00,00,00,00,1086
340 DATA 00,00,00,00,00,18,00,00,18,FF,E7,00,00,00,00,00,534
350 DATA 00,00,00,00,00,00,00,00,00,00,00,00,07,E0,06,60,333
360 DATA 03,C0,FE,7F,F8,1F,00,00,00,00,00,00,00,00,00,00,855
370 DATA 00,00,00,10,44,00,00,00,18,FF,E7,00,00,00,00,00,594
380 DATA 00,00,00,00,00,00,0C,C0,00,00,00,00,03,C0,06,60,501
390 DATA 07,60,FF,FF,FE,FF,00,00,00,00,00,00,00,00,00,00,1122
400 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
410 DATA 00,18,00,0F,00,0F,00,38,3F,F0,0F,C0,00,00,00,00,620
420 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
430 DATA 30,CC,FF,FF,CF,33,00,00,00,00,00,00,00,00,00,00,1020

```

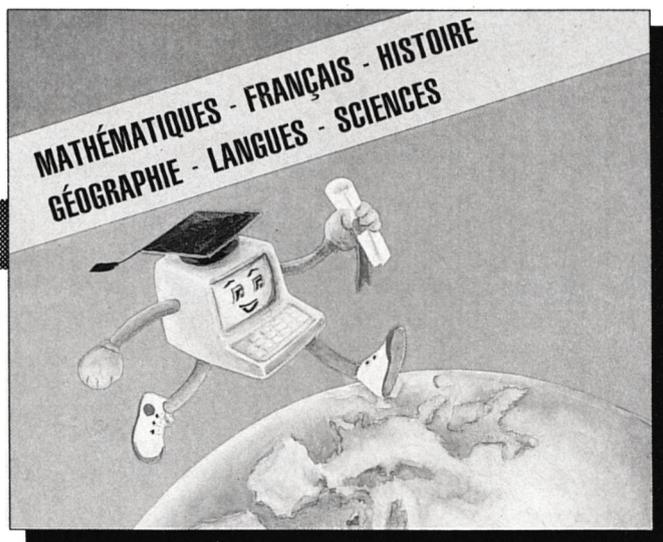
440 DATA 00,00,00,66,00,00,00,00,18,FF,E7,00,00,00,00,612  
450 DATA 00,00,00,00,00,00,18,00,18,00,18,00,18,78,19,CE,447  
460 DATA 1F,0E,FF,FF,FF,FF,00,00,00,00,00,00,00,00,00,1065  
470 DATA 00,00,00,00,00,00,00,00,30,00,00,00,18,00,18,00,0C,108  
480 DATA 00,0C,0F,FF,0F,F3,00,00,00,00,00,00,00,00,00,00,540  
490 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,38,56  
500 DATA 00,6C,00,7F,00,0F,00,38,3F,F0,0F,C0,00,00,00,00,816  
510 DATA 00,00,00,00,00,00,03,00,00,00,00,00,00,7F,00,07,E0,361  
520 DATA 01,FE,FF,98,F0,0F,00,00,00,00,00,00,00,00,00,00,917  
530 DATA 00,00,00,00,00,00,00,00,18,FF,E7,00,00,00,66,00,00,612  
540 DATA 00,00,00,00,00,00,00,00,00,00,00,00,30,00,00,00,00,48  
550 DATA 00,18,00,0F,00,0F,00,38,3F,F0,0F,C0,00,00,00,00,620  
560 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,1E,00,77,149  
570 DATA 30,C7,FF,FF,CF,FF,03,00,00,00,00,00,00,00,00,00,1222  
580 DATA 03,7E,C0,10,18,18,18,18,10,10,00,00,00,00,00,489  
590 DATA 00,00,00,00,00,00,18,60,18,00,18,00,18,78,19,CE,543  
600 DATA 1F,0E,FF,FF,FF,FF,00,00,00,00,00,00,00,00,00,00,1065  
610 DATA 00,00,00,00,00,00,03,30,00,00,00,00,03,E0,06,F0,524  
620 DATA 0E,38,07,F0,03,E0,00,00,00,00,00,00,00,00,00,00,544  
630 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0E,00,78,134  
640 DATA 10,C0,70,F0,F8,FC,FE,1E,3F,F8,00,00,00,00,00,00,1655  
650 DATA 00,00,00,10,18,18,18,18,10,10,00,00,00,00,00,168  
660 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0  
670 DATA 00,00,40,00,00,C0,00,0C,F0,00,0C,FF,FF,FC,3F,FF,1600  
680 DATA F0,00,00,00,00,00,00,00,18,00,00,00,00,00,00,00,264  
690 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,80,00,03,20,163  
700 DATA 00,00,00,00,00,00,00,00,0C,00,30,CC,00,1F,FC,40,0F,626  
710 DATA 30,C0,0C,00,E0,3C,00,7F,F8,00,1F,E0,00,00,00,1166  
720 DATA 00,00,00,00,00,00,00,00,00,00,00,00,01,C0,00,78,313  
730 DATA 00,0C,0F,FC,0F,F0,00,00,00,00,00,00,00,00,00,00,534  
740 DATA 00,00,00,00,00,00,00,00,00,00,00,00,18,00,00,00,00,24  
750 DATA 00,00,00,00,1E,00,00,77,00,30,C7,00,1F,FF,40,0F,761  
760 DATA FC,C0,0D,80,E0,3C,00,7F,F8,00,1F,E0,00,00,00,00,1499  
770 DATA 00,00,00,00,00,00,00,00,00,00,00,00,60,00,00,00,96  
780 DATA 00,0E,40,00,1B,C0,00,1F,F0,00,0F,FF,FF,FE,3F,FF,1665  
790 DATA FC,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,252  
800 DATA 00,00,00,00,00,00,18,00,00,00,00,00,78,00,C0,00,336  
810 DATA FF,00,38,00,60,00,C0,00,E0,00,F8,00,7F,C0,0F,FF,1660  
820 DATA 00,00,00,00,00,00,00,00,00,00,00,00,03,E0,06,F0,473  
830 DATA 0E,38,07,F0,03,E0,00,00,00,00,00,00,00,00,00,00,544  
840 DATA 00,00,00,00,00,00,00,00,00,00,00,00,3F,E0,03,FE,544  
850 DATA 0F,80,7C,00,E0,00,C0,00,E0,00,F8,00,7F,C0,0F,FF,1744  
860 DATA 00,00,00,00,00,00,00,00,00,00,00,00,3F,E0,03,FE,544  
870 DATA 0F,80,7C,00,E0,00,C0,C0,E0,00,F8,00,7F,C0,0F,FF,1936  
880 DATA 00,00,00,00,00,00,00,00,0C,00,0C,1C,0C,10,0C,3C,0C,164  
890 DATA 00,0C,FF,FC,FF,F0,00,00,00,00,00,00,00,00,00,00,1014  
900 DATA 00,00,00,00,00,00,00,00,06,00,06,00,06,00,06,20,06,62  
910 DATA 60,06,70,1E,3F,FC,0F,F0,00,00,00,00,00,00,00,00,814  
920 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,01,F0,241  
930 DATA 03,3C,1F,FC,7F,F0,30,00,30,00,30,00,30,00,30,00,953  
940 DATA 00,00,00,00,00,00,01,80,00,00,00,00,00,18,00,0C,165  
950 DATA 20,06,60,06,70,1E,3F,FC,0F,F0,00,00,00,00,00,00,852  
960 DATA 00,00,00,00,00,00,00,00,00,00,00,00,78,00,C0,00,312  
970 DATA FF,00,38,00,60,00,C0,00,E0,00,F8,00,7F,C0,0F,FF,1660  
980 DATA 00,00,00,00,00,00,00,00,00,00,00,00,22,00,00,42  
990 DATA 00,00,40,00,00,C0,00,0C,F0,00,0C,FF,FF,FC,3F,FF,1600  
1000 DATA F0,00,00,00,00,00,00,00,00,00,00,00,00,00,00,240  
1010 DATA 00,00,00,00,00,00,01,98,00,00,00,38,00,6C,00,7C,441  
1020 DATA 20,3C,60,0C,70,3C,3F,F8,0F,F0,00,00,00,00,00,00,938

1030 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,18,24  
1040 DATA 00,0C,00,0C,00,38,3F,F0,0F,C0,00,00,00,00,00,590  
1050 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0  
1060 DATA 00,00,00,00,00,00,00,00,0C,00,30,CC,00,1F,FC,40,0F,626  
1070 DATA 30,C0,0C,00,E0,3C,00,7F,F8,00,1F,E0,00,00,00,1166  
1080 DATA 00,00,00,00,00,00,00,00,00,00,00,00,66,00,00,00,102  
1090 DATA 00,00,40,00,00,C0,00,0C,F0,00,0C,FF,FF,FC,3F,FF,1600  
1100 DATA F0,00,00,00,00,00,00,00,00,00,00,00,00,00,00,240  
1110 DATA 00,00,00,00,00,00,18,00,18,00,18,00,18,78,19,CE,447  
1120 DATA 1F,0E,FF,FC,FF,F0,00,00,00,00,00,00,00,00,00,1047  
1130 DATA 00,00,00,00,00,00,03,00,00,00,00,00,01,C0,00,78,316  
1140 DATA 00,0C,0F,FC,0F,F0,00,00,00,00,00,00,00,00,00,534  
1150 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,38,00,6C,164  
1160 DATA 00,7C,00,0C,00,38,3F,F0,0F,C0,00,00,00,00,00,702  
1170 DATA 00,00,00,00,00,00,03,00,00,00,00,00,3F,E0,03,FE,547  
1180 DATA 0F,80,7C,00,E0,00,C0,00,E0,00,F8,00,7F,C0,0F,FF,1744  
1190 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,134  
1200 DATA 10,C0,71,F0,F8,FC,FE,1E,3F,F8,00,00,06,60,00,00,1758  
1210 DATA 00,00,00,00,00,00,00,00,30,00,00,00,00,00,00,18,72  
1220 DATA 00,0C,00,0C,00,38,3F,F0,0F,C0,00,00,00,00,00,00,590  
1230 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0  
1240 DATA 00,00,00,00,1E,00,00,77,00,30,C7,00,1F,FF,40,0F,761  
1250 DATA FC,C0,0D,80,E0,3C,00,7F,F8,00,1F,E0,00,00,00,00,1499  
1260 DATA 07,77,73,3E,E0,00,00,00,00,00,00,00,00,00,00,527  
1270 DATA 00,00,00,00,00,00,18,60,18,00,18,00,18,78,19,CE,543  
1280 DATA 1F,0E,FF,FC,FF,F0,00,00,00,00,00,00,00,00,00,1047  
1290 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,510  
1300 DATA 00,00,00,00,00,00,01,98,00,00,00,18,00,F8,03,98,580  
1310 DATA 0E,3C,0F,FF,03,E7,00,00,00,00,00,00,00,00,00,578  
1320 DATA 00,00,00,00,03,0C,03,00,03,00,03,00,03,00,07,E0,382  
1330 DATA 01,FE,FF,80,F0,00,00,00,00,00,00,00,00,00,00,878  
1340 DATA 00,00,00,33,00,00,00,30,00,00,00,00,00,00,00,700  
1350 DATA 01,FE,FF,80,F0,00,00,00,00,00,00,00,00,00,00,878  
1360 DATA 00,00,00,00,00,00,00,00,00,00,00,00,03,0C,06,18,45  
1370 DATA 0C,30,06,18,03,0C,00,00,00,00,00,00,00,00,00,105  
1380 DATA 00,00,00,00,00,00,00,00,00,00,00,00,30,C0,18,60,360  
1390 DATA 0C,30,18,60,30,C0,00,00,00,00,00,00,00,00,00,420  
1400 DATA 00,00,00,00,00,1F,31,60,30,FC,CE,00,00,03,00,00,685  
1410 DATA 00,00,00,00,00,04,00,11,00,00,00,FE,01,83,07,C0,606  
1420 DATA 0C,F0,7F,F0,FF,C0,C0,00,C0,00,C0,00,C0,00,2026  
1430 DATA 00,00,00,66,00,00,00,00,00,00,FC,F8,00,00,00,00,602  
1440 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0  
1450 DATA 00,00,10,7F,71,F1,F8,FC,FE,1E,3F,F8,00,00,00,1592  
1460 DATA 00,00,00,00,00,00,00,00,0C,00,0C,00,0C,00,0C,60  
1470 DATA 00,0C,10,17,70,21,F8,F8,FE,1C,3F,F0,00,00,00,1277  
1480 DATA 00,00,00,00,00,00,00,00,0C,00,0C,00,0C,00,0C,60  
1490 DATA 00,0C,10,17,70,21,F8,F8,FE,1C,3F,F0,00,00,06,60,1379  
1500 DATA 00,00,00,00,00,00,00,00,0C,3C,0C,0F,0C,01,DC,00,7C,456  
1510 DATA 00,EC,3F,CF,3E,03,00,00,00,00,00,00,00,00,00,571  
1520 DATA 00,00,00,0C,00,00,00,00,00,00,3C,00,6E,00,7E,00,7E,434  
1530 DATA 00,06,10,FC,70,7E,F8,06,FE,1C,3F,F0,00,00,06,60,1453  
1540 DATA 00,00,00,00,00,00,00,00,0E,00,0C,00,0C,00,0C,62  
1550 DATA 00,18,10,FC,70,7E,F8,06,FE,1C,3F,F0,00,00,00,1369  
1560 DATA 00,00,00,00,00,00,00,00,0E,00,0C,00,0C,00,0C,62  
1570 DATA 00,18,10,FC,70,7E,F8,06,FE,1C,3F,F0,00,00,06,60,1471  
1580 DATA 00,00,00,00,00,00,00,00,0C,3C,0C,0F,0C,01,CC,00,EC,552  
1590 DATA 00,78,03,F8,03,F8,00,00,00,00,00,00,00,00,00,622  
1600 DATA 00,00,00,00,00,00,0C,00,00,00,00,00,1C,00,1E,00,06,76  
1610 DATA 00,0E,10,FC,70,7E,F8,06,FE,1C,3F,F0,00,00,06,60,1461



# LE SPECIALISTE DE L'EDUCATIF

## notre CATALOGUE des éducatifs



## Les détails des logiciels les tarifs

catalogue gratuit

**J**e désire recevoir le catalogue des éducatifs, possédant  
un ordinateur(s) de  
marque(s) \_\_\_\_\_

de type(s) \_\_\_\_\_

à l'adresse ci-contre Nom: \_\_\_\_\_ Prénom \_\_\_\_\_

Adresse: \_\_\_\_\_

Code postal: \_\_\_\_\_ Ville: \_\_\_\_\_

**A envoyer à SORACOM Editions - BP 88 - 35170 BRUZ**  
**Joindre deux timbres à 2,30 F**

**SORACOM**  
éditions

**Media Box  
Cassettes  
Vidéo (L)**

222x135  
x348 mm  
Réf. 310.540.0

**155 F + port**

pour 9 cassettes  
VHS, V2000, Betamax



**Media Box Mini Cassettes**

148x91x348  
Réf. 310.503.3

**95 F + port**

pour 16 mini  
cassettes



**Media Box Disquettes 5"1/4**

182x178x348 mm  
Réf. 310.501.1

**180 F + port**

pour 70 disquettes, livré  
avec séparations et index



**Media Box  
Compact Disc**

148x135x348 mm  
Réf. 310.502.6

**125 F + port**

pour 13 compacts discs simples

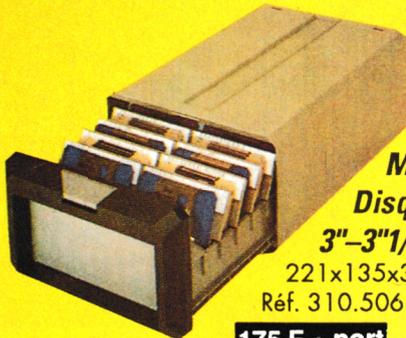


**Media Box  
Disquettes  
3"-3"1/2**

221x135x348 mm  
Réf. 310.506.4

**175 F + port**

pour 150 disquettes



**Media-Box  
Data Cartridge**

222x135x348 mm  
Réf. 310.518.7

**160 F + port**

pour 11 Data Cartridges  
type 3M



**Media Box  
Cassettes Vidéo  
VHS-C**

148x91x348 mm  
Réf. 310.505.7

**102 F + port**

pour 8 cassettes



**Media Box  
Cassettes  
Vidéo 8**

148x91x348 mm  
Réf. 310.531.6

**102 F + port**

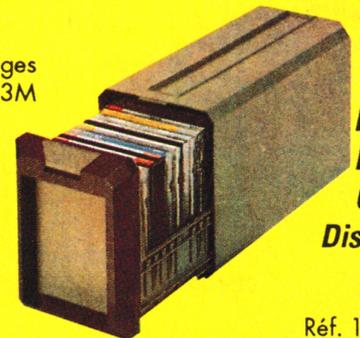
pour 12 cassettes



**Media  
Box  
Compact  
Disc Multi**

148x177  
x348 mm  
Réf. 100.525.0

pour 23 CD simples  
ou 11 CD doubles



**Media Box  
Photo**

222x135  
x348 mm  
Réf. 310.501.9

**155 F + port**

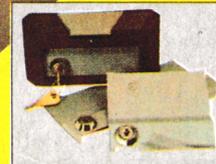
pour 140 tirages  
format maximum 13x18 plus



**Module serrure**

**60 F + port**

(bien indiquer la boîte)



**Nouveau ! les mêmes en noir**



Mini cassettes  
Réf: 310 . 620 . 7  
95 F + PORT

Vidéo VHS (L)  
Réf: 310 . 623 . 8  
160 F + PORT

Compact Disc  
réf: 310 . 621 . 4  
125 F + PORT

C D Multi  
Réf: 310 . 622 . 1  
190 F + PORT

Port + 25 F par article  
port + 40 F par article pour l'étranger

Voir bon de commande page 24