

Micro
Informatique
Standards AMSTRAD

CPC

infos



SQUARES :
stratégie en
couleur

MARK ERROR
triez vos disquettes

M 2604 - 31 - 22,00 F



Mensuel n° 31
AVRIL 1991

SOMMAIRE

N°31

3

FANZ' INFOS

6

PSG

9

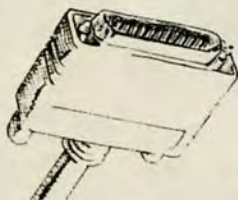
SQUARES

18

MARK ERROR

24

CHRONIQUE
DU TELECHARGEMENT



26

DRIVER LA SOURIS

30

DATALOAD

35

PRATIQUE DE
L'ASSEMBLEUR

37

ANTI-ERREUR

40

TRUCS EN VRAC

44

AXYS

CPC infos

La Hale de Pan - 35170 BRUZ
Tél : 99.52.98.11 - Télécopie : 99.52.78.57
Serveurs : 3615 ARCADES et 3615 MHZ
Terminal NMPP E83
Gérant, directeur de publication
Sylvio FAUREZ

REDACTION

Rédacteur spécialisé
Olivier SAOLETTI
Secrétaire
Catherine FAUREZ

Ont collaboré à ce numéro :

M. BENARD, Cyril CAUCHOIS, Fabien FESSARD, Emmanuel GUILLARD, Jean-Pierre MARQUET, Hervé MONCHATRE, Stéphane SAINT-MARTIN

FABRICATION

Directeur de fabrication
Edmond COUDERT
Maquettiste
Jean-Luc AULNETTE

ABONNEMENTS

Nathalie FAUREZ - Tél. : 99.52.98.11

PUBLICITE

IZARD Création
15, rue St-Melaine
35000 RENNES - Tél. 99.38.95.33

GESTION RESEAU : NMPP

SORACOM E. COUDERT
Fax. 99.52.78.57 - Terminal E83

SOCIETE MAYENNAISE D'IMPRESSION 53100 MAYENNE

Les articles et programmes que nous publions dans ce numéro bénéficient, pour une grande part, du droit d'auteur. De ce fait, ils ne peuvent être imités, contrefaits, copiés par quelque procédé que ce soit, même partiellement sans l'autorisation écrite de la Société SORACOM et de l'auteur concerné. Les opinions exprimées n'engagent que la responsabilité de leurs auteurs. Les différents montages présentés ne peuvent être réalisés que dans un but privé ou scientifique mais non commercial. Ces réserves s'appliquent également aux logiciels publiés dans la revue.

CPC infos est édité par les Editions SORACOM, editrice de PCCompatibles Magazine et MEGAHERTZ Magazine. RCS Rennes B 319 816 302.

AMSTRAD est une marque déposée.
CPC infos est une revue mensuel totalement indépendante d'AMSTRAD GB et d'AMSTRAD FRANCE.

ATTENTION

Durant les mois d'avril et de mai, les permanences techniques du mercredi après-midi sont supprimées. Rendez-vous donc **mercredi 5 juin** pour vos questions.



FANZ'Z'

infos

N°2



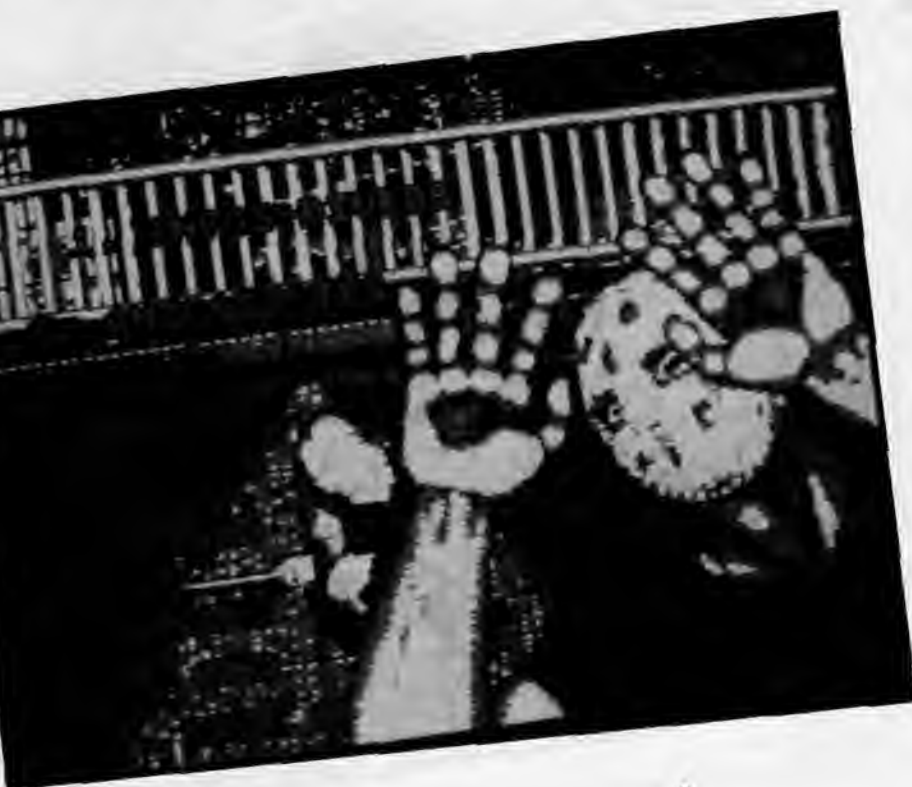
Le
tive-
assez,

Je rappelle
bliés dans

qu'il n'est pas question de les distribuer sur disquette (et même sur papier d'ailleurs) sans obtenir l'autorisation des Editions Soracom et celle de l'auteur. Je pense en particulier à The Fanzine qui a mis dans son numéro 1 le programme Motyf sans en parler à personne. J'espère que ce programme sera retiré du fanzine et surtout que Megasound ne sera pas dans le numéro 2, OK ?

mois dernier je lançais un appel aux fanzines et j'ai effectivement reçu des nouveaux venus. Mais ce n'est pas encore il m'en faut toujours plus et tout de suite car je dois préparer deux numéros à l'avance.

à ceux qui l'oublie parfois que les programmes pu-
CPC Infos ne sont pas dans le domaine public et donc



AMSTUS

Un fanzine sur disque qui comprend de nombreuses images digitalisées sur Amiga. Enfin, je veux dire que les images Amiga passent à travers un digitaliseur CPC. Le résultat n'est pas toujours terrible (je le sais par expérience) mais c'est original. Sinon, il y a beaucoup de textes avec de la musique par dessus. Justement les textes, il faudrait peut-être les réduire car les fautes seraient moins visibles, ou alors il s'agit d'un concours et là je suis sûr que vous avez gagné.

Mais ce n'est pas une raison pour ne pas leur écrire donc :

Frédéric Bazin
5, av des Pyramides
77420 Champs

MICROMAG

Dans le n° 1, j'insiste beaucoup pour que les lecteurs participent : ce sera sans doute très important pour que le fanz vive. Il faut que les lecteurs écrivent, envoient leurs dessins, leurs blagues, leurs tops... Bref, tout ce qu'ils veulent ! Les quatre pages du premier Micro Mag ont été écrites en quelques heures mais les prochains numéros seront beaucoup plus gros : 8, 12 voire 16 pages et le travail à fournir sera plus important et plus long. Micro Mag est un fanzine mensuel qui sortira vraisemblablement au milieu du mois. Pour recevoir Micro Mag il suffit d'envoyer une enveloppe timbrée à 3,80 francs et portant l'adresse du destinataire (ne pas oublier III).

MICRO MAG
Olivier MAILLARD
123, avenue des Saules
59910 Bondues

- Micro Mag -

OLIVIER MAILLARD 123 AVENUE DES SAULES 59910 BONDUES POSTE : 3,80 FRANCS
MENSUEL - NUMERO 2 - MARS 1991 - ENVOI POSTAL 3,80 FRANCS - MENSUEL - NUMERO 2

SOMMAIRE EN PAGE 8
DES BIDOUILLES
DES TESTS DE JEUX
DES JEUX
DES MENS
DES PETITES ANNONCES
DES TRUCS

TESTS EN PAGE 6 !

EN MACHIN I EN PAGE 2 IL Y A LES FANZINES ET RUNSTRAD !

Ouais ! Je sais !



EDITORIAL

Eh oui ! Déjà le numéro 2 de Micro Mag ! Un magazine qui a doublé en nombre de pages : on passe de 4 à 8 pages avec, plus tard sûrement 12 voire 16 pages. Micro Mag numéro 2 est bien mieux, à mon avis que le premier : il est plus complet, mieux mis en page et vous propose aussi un abonnement ! N'oubliez pas de continuer à m'envoyer vos dessins, test, bref le courrier. Tout sera publié ! J'espère que vous passerez un bon moment en lisant ce Micro Mag. On se retrouve tout à l'heure au Mot de la Fin ! Atchao !

LE CHEF : OLIVIER

PAGE 1 - MICRO MAG

- PAGE 1 - MICRO MAG - N°2 - PAGE 1 - MICRO MAG - N°2

SOMMAIRE EN PAGE 8 - SOMMAIRE EN PAGE 8 - SOMMAIRE EN PAGE 8

FREWARE

FREWARE EST REALISE PAR : JEROME ROSSI, 7 GRAND RUE, 54150 MANCE... TEL 06.46.14.79... PRIX : 0 FRS... RESPONSABLE N°1

EDITORIAL

Eh oui, encore un nouveau fanzine ou direz-vous seulement FREWARE à quelques chose de special. Il ne parle que des démos et des logiciels. Mais ce n'est pas tout car toutes les démos et tous les logiciels testés dans FREWARE sont disponibles gratuitement. Pour les recevoir il suffit d'envoyer une disquette avec un timbre à 3,80 frs, génial non ? Comme je vous le disais précédemment, contrairement aux autres fanzines, nous testerons rarement des jeux, à moins qu'ils n'en valent vraiment la peine.

POURQUOI CE FANZINE ?

J'ai créé ce fanzine pour deux raisons, la première étant de faire partager ma passion pour les logiciels et les démos. La seconde est de vous aider à choisir votre fanz, car tous se disent être les meilleurs, ce qui n'est pas toujours le cas... Cependant rappelez-vous que si vous ne connaissez pas un logiciel, c'est bien possible que vous puissiez l'avoir. Vous pouvez constater que je donne pour les logiciels toujours l'adresse de son auteur, au cas où vous voudriez leur acheter ou en personne (ce qui n'est pas toujours le cas). FREWARE n'est absolument pas conçu pour critiquer les autres fanzines, mais pour dire comment ils sont réalisés techniquement, ce que vous y trouverez.

REVOLODS... REVOLODS... REVOLODS...

Voilà une demo qui n'est pas très nouvelle et qui n'assure pas des kilos non plus. Elle a été faite par Longshot des LOGON SYSTEM. Quand je dis qu'elle est sûre pas, ça ne veut pas dire non plus qu'elle est sûre mais un scrolling horizontal avec quelques sprites et un autre. Cependant toujours la qualité de ce qui est faible, nous partagerons largement la qualité de ce super jeu que les LOGON SYSTEM sont ceux qui nous offrent les plus belles démos. THE DEMO a été écrite par Longshot des LOGON SYSTEM. Elle n'est pas écrite pour le jeu, mais pour le plaisir de l'écrire. Elle n'est pas écrite pour le jeu, mais pour le plaisir de l'écrire. Elle n'est pas écrite pour le jeu, mais pour le plaisir de l'écrire.



Je viens de recevoir NOIX DE CROCO ? Il est genial... Il se compose d'une disquette et d'un fanzine sur papier (à lui deux pages). Nous allons tout de suite passer en détail le fanzine disquette, vous pouvez le consulter au numéro 1 de Micro Mag qui s'explique facilement par le supplément papier qui lui est bourré de texte.

Bonne dans cette disquette plein de routines au montage, des files un peu partout, et de toutes les couleurs imaginables qui s'écrit un peu partout. Bref c'est du tout bon. Cependant, si vous voulez voir ce que vous avez à la main, descendez au premier scrolling quand vous verrez à la fin de la disquette ce qui sera votre surprise quand vous aurez à l'ouvrir. Cependant que ce sera votre surprise quand vous aurez à l'ouvrir. Cependant que ce sera votre surprise quand vous aurez à l'ouvrir.

Je viens de vous offrir la face B, sur la face B on y retrouve un scrolling à longueur de texte. Cependant que ce sera votre surprise quand vous aurez à l'ouvrir. Cependant que ce sera votre surprise quand vous aurez à l'ouvrir. Cependant que ce sera votre surprise quand vous aurez à l'ouvrir.

Le numéro 2 s'est fait attendre six mois mais je pense que ça valait la peine vu le résultat.

NOIX DE CROCO : MOIX DE CROCO, 2 RUE SEBASTIEN, 59000 DULLENS

SOMMAIRE	
Page 1	Les deux jeux
Page 2	Les tests de jeux
Page 3	Les tests de jeux
Page 4	Les tests de jeux
MENSAGES	
7 RUE SEBASTIEN, 59000 DULLENS	

FREWARE LE FANZINE QUI FAIT DU 300 Km/h



REMARQUE IL Y A MOIN BIEN, MAIS C'EST PAYANT... PHOTOCOPIEZ-NOUS POUR NE PAS DEVENIR VIEUX... FREWARE IL Y A MOIN BIEN MAIS C'EST MEILLEUR.

FREWARE

En fait, c'est l'ex-Microcool qui se cache derrière ce nouveau titre. Le fanz est principalement axé sur les démos, les autres fanzines, les tests de jeux et les fautes d'orthographe (c'est vrai qu'il n'est pas tout seul). Il y a aussi pas mal de découpages pour illustrations en provenance de revues je pense, ou alors des jeux originaux mais j'ai un petit doute quand même.

Jerome Rossi
7, Grand rue
54150 Mance



CPC
utilitaire

PSG



Mesdames et messieurs, merci d'être venu si nombreux écouter cette conférence, au cours de laquelle je compte vous entretenir du PSG. Mais j'aperçois soudain un certain nombre de personnes qui se lèvent et gagnent en hâte la sortie. Des non-sportifs, sans doute. Je vous en prie, chers amis, un instant de patience, je crois que nous nous sommes mal compris.

Par "PSG", mesdames et messieurs, je n'entendais pas le "Paris-Saint-Germain", mais plutôt le Générateur de Sons programmable, qu'Outre-Manche on nomme, me semble-t-il, "Programmable Sound Generator", ou quelque chose d'approchant, mais ces Britanniques ne font rien comme tout le monde. Oui, il s'agit bien de ce circuit, répondant par ailleurs au sobriquet si poétique de AY-3-8912, qui équipe votre Amstrad, et qui se charge de produire des sons.

Peut-être avez-vous déjà lu de la documentation sur cette puce, comme disent les marchands de machines à laver ; ne serait-ce que les pages qui lui sont consacrées dans "la Bible du 6128" de Micro-Application. Mais ce distingué bouquin, œuvre sans doute de techniciens s'adressant à des techniciens, est plus exhaustif que pédagogique, et il est possible que vous ayez baissé les bras, légèrement découragé. Je vous propose un autre examen de la question. Et, à propos, je prends dès maintenant l'engagement d'être clair, et, en premier lieu d'éviter le jargon : je ne vous parlerai donc ni de "port I/O", ni de "chip", ni de "low byte", ni de "pin",

ni même de "flip-flop". Restons français. Thank you. En revanche, et comme rien ne vaut un bon exemple, je vous offrirai en prime un petit RSX de sons préprogrammés, qui rappelleront peut-être quelque chose aux anciens possesseurs d'un ORIC-A ou d'un ATMOS, et qui vous permettront de concurrencer agréablement le BIP de l'Amstrad, le fameux PRINT CHR\$(7), dont la sécheresse et la pauvreté ne sont plus à vanter.

Le PSG est donc capable de produire des sons, et de travailler indépendamment du microprocesseur ; pour cela, il ne réclame qu'une chose : qu'on lui fournisse les données nécessaires, il se débrouille ensuite tout seul, laissant le Z-80 s'occuper de sa tâche de son côté. Il possède seize registres, dont seulement quatorze concernent la production des sons et des bruits. Ces registres seront donc numérotés de R0 à R13. Votre travail va consister à envoyer les renseignements idoines à chacun de ces quatorze registres, en principe ; mais, dès maintenant, une remarque s'impose : une fois placée dans le registre voulu, une donnée ne s'efface pas,

elle y reste jusqu'à ce qu'une autre donnée vienne l'y remplacer ; il n'est donc pas indispensable de renseigner TOUS les registres chaque fois qu'on veut produire un son.

Comment envoyer une donnée à un registre ? Hélas, uniquement en assembleur ! Et puis, le Basic de l'Amstrad est très efficace, mais, malgré sa rapidité, on peut avoir besoin de travailler encore plus rapidement, et je vous le prouverai tout à l'heure à propos du RSX dont je parlais ; donc, l'assembleur s'impose doublement. Mais, rassurez-vous, le mode d'emploi n'a rien de compliqué : il suffit de placer dans l'accumulateur (le registre A du Z-80) le numéro du registre visé, et dans C la donnée à envoyer. Ensuite, on exécute avec un CALL &BD34.

Tout de suite, un exemple. Vous voulez envoyer la valeur &3E au registre R7 du PSG ? Faites ceci :

```
LD A,&07
LD C,&3E
CALL &BD34
```

Le registre R7 contient à présent la valeur &3E, et elle y demeure jusqu'à ce que vous la remplaciez par



une autre. Compris ? Bien. Quel est donc le rôle des quatorze registres du PSG ?

Voici, dans l'ordre :

- R0 et R1 sont associés pour déterminer la HAUTEUR DU SON qui sortira sur le canal A de l'Amstrad ; R0 contient le poids faible (sur 8 bits), R1 le poids fort (sur seulement 4 bits). Vous pouvez donc choisir une valeur qui occupera au maximum douze bits, par conséquent dans la fourchette 000000000000 à 111111111111 ; en décimal, de 0 à 4095 ; en hexa, de &0000 à &0FFF. Je vous expliquerai plus loin ce que désignent en réalité ces valeurs, qui ne sont ni la fréquence, ni la période du son, du moins telles que les entendent les musiciens.

- R2 et R3 font exactement la même chose pour le canal B.

- R4 et R5 font exactement la même chose pour le canal C.

Arrêtons un instant la revue des registres du PSG, et tenons nos promesses : que désigne la valeur que vous envoyez dans ces trois couples de registres ? Il nous faut dire un mot d'acoustique.

Un son est produit par un objet qui vibre : corde de piano ou de violon, branche de diapason, membrane de timbale, etc. Plus l'objet vibre rapidement, plus le son est aigu. La note LA du diapason, dite "LA International", est produite par 440 vibrations par seconde (on dit 440 Hertz, en abrégé 440 Hz) : ce nombre s'appelle la fréquence. La durée d'une seule vibration est appelée la période, et c'est toujours un nombre très petit : pour le même LA, une seconde divisée par 440, soit 0.002272727 seconde environ.

Lorsqu'on monte d'une octave (par exemple, quand on saute de ce LA-là au LA suivant dans l'aigu), la fréquence double, elle passe donc à 880 Hz, et la période est évidemment deux fois plus courte. Si on monte "trop" haut, au-delà de 10000 à 12000 Hz, le son est de moins en moins audible, il devient bientôt un "ultra-son" ; si on descend "trop" bas, le son ressemble de plus en plus à un simple bruit, il devient de plus en plus sourd, c'est bientôt un "in-fra-son".

Comment obtenir le LA International par l'intermédiaire du PSG ? En envoyant aux registres intéressés la valeur 440 désignant la fréquence ? Ou la valeur 0.0022727 désignant la période ? Ni l'une ni l'autre ! (Es-

sayez, d'ailleurs, de "poker" quelque part le nombre 0.0022727...). Les concepteurs du PSG ont adopté la convention suivante, dictée par des raisons électroniques :

Fréquence x Période = 62500

(et non plus 1, comme avec la période au sens musical du mot)

Soit dit en passant, le "Manuel de l'Utilisateur" de l'Amstrad prétend que c'est 125000, et la faute n'a jamais été corrigée depuis des années, malgré de multiples rééditions, pas plus que les noms des notes n'ont été francisés. Ce devait être trop difficile...

Par conséquent, sachant que le LA International a pour fréquence 440 Hz, la "période", ou du moins ce qu'on appelle ainsi chez monsieur PSG, est : 62500 / 440 = 142.04545, qu'on arrondit évidemment à la valeur entière la plus proche, 142.

Convertissons en hexa : 142 = &008E

Nous enverrons donc &8E, poids faible, au registre R0, et &00, poids fort, au registre R1. Voilà définie la hauteur du son.

(Remarquons que les valeurs extrêmes, proches de 0 ou de 4095, ne sont valables que théoriquement ; au-dessous de 5, plus rien n'est audible, sauf peut-être pour votre chien, et c'est heureux, car la justesse des notes devient très approximative avec des valeurs qu'on ne peut plus affiner, tant elles sont proches ; et au-dessus de 3000, qui donne un son de 21 Hz, on n'a guère plus que du bruit. Vérifiez-le avec du Basic : SOUND 1,3000 et SOUND 1,4000 n'offrent guère de différence...).

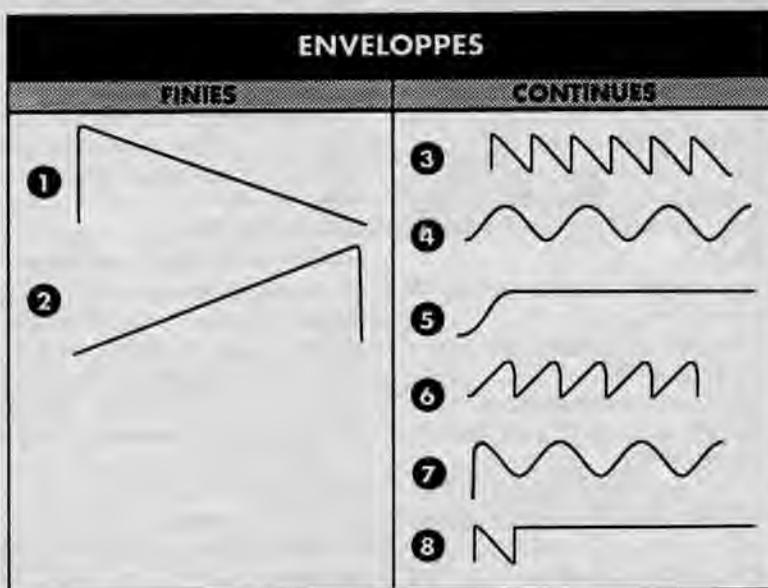
Fin du cours sur la hauteur des sons ; reprenons la revue des registres du PSG. Il en reste huit.

- R6 définit la HAUTEUR DU BRUIT (et non plus du son), car même les bruits ont une hauteur : comme seuls cinq

```

10 dep=&BE80:lig=100
30 s=lig:FOR i=0 TO 15:READ a$:POKE dep+i,VAL("&"a$):s=s+PEEK(dep+i):NEXT
40 READ b$:IF VAL("&"b$)<>s THEN PRINT"Erreur en ligne"lig:END
50 dep=dep+16:lig=lig+10
80 IF lig<240 THEN 30
70 SAVE"onomatop",b,&BE80,211
100 DATA 21,B5,BE,01,89,BE,C3,D1,BC,8D,BE,C3,B9,BE,C3,BF,AA7
110 DATA BE,C3,C4,BE,C3,C9,BE,C3,CE,BE,C3,D3,BE,50,49,4E,B45
120 DATA C7,50,41,CE,42,4F,55,CD,43,4C,49,C3,43,4C,41,C3,77F
130 DATA 50,49,4F,D5,00,FC,A6,89,BE,21,FF,BE,C3,EE,BE,21,986
140 DATA 0D,BF,18,F8,21,1B,BF,18,F3,21,29,BF,18,EE,21,37,8D5
150 DATA BF,18,B9,21,45,BF,CD,EE,BE,0E,00,AF,C5,CD,34,BD,834
160 DATA C1,06,00,10,FE,0C,78,FE,70,20,F0,C3,A7,BC,06,0E,7B2
170 DATA 3E,FF,3C,4E,F5,C5,CD,34,BD,C1,F1,23,10,F4,C8,18,9A3
180 DATA 00,00,00,00,00,00,3E,10,00,00,00,0F,00,00,00,111
190 DATA 00,00,00,0F,07,10,10,10,00,08,00,00,00,00,00,10C
200 DATA 00,1F,07,10,10,10,00,18,00,1F,00,1F,00,1F,00,193
210 DATA 3E,10,10,10,AF,00,00,2F,00,00,00,00,00,00,3E,10,28C
220 DATA 00,00,AF,00,00,00,00,00,00,00,00,00,00,3E,0F,00,00,1D8
230 DATA 00,00,00,00,FF,FF,FF,FF,00,00,00,00,00,FF,FF,FF,6DE
240 "LOAD"onomatop":CALL &BE80
250 "PING,{PAN},{BOUM},{CLIC},{CLAC},{PIOU}

```





bits fonctionnent sur ce registre, vous ne pouvez lui envoyer que les valeurs décimales 0 à 31. Toute autre valeur est "décapitée" selon le procédé modulo (32 donne 0, 33 donne 1, etc).

- R7 est une sorte de poste de commandement, qui va décider de ce qui sortira ou ne sortira pas sur le haut-parleur de votre ordinateur. Ici, il faut détailler chaque bit, ou plutôt les six premiers (ne touchez pas aux bits 6 et 7, laissez-les à zéro) :

- les bits 0, 1 et 2 déterminent si un SON sortira sur les canaux, respectivement, A, B et C : s'ils contiennent 0, le son est produit ; s'ils contiennent 1, silence, quand bien même vous auriez envoyé des données aux registres R0 à R5. C'est R7 qui commande !

- les bits 3, 4 et 5 déterminent si un BRUIT sortira sur les mêmes canaux, avec la même convention : 0 = oui, 1 = non.

Vite, un exemple : en envoyant à R7 la valeur &3E, qui s'écrit en binaire et sur six bits 111110, vous autorisez la sortie d'un son sur le canal A, puisque le Bit 0 contient 0, interdisez toute sortie de son sur B et C, puisque les bits 1 et 2 contiennent 1, et toute sortie de bruit sur les trois canaux, puisque les bits 3, 4 et 5 contiennent 1. Donc, un son sans bruit sera produit sur le canal A.

Vous êtes toujours là ? Je continue.

- R8 définit, sur les bits 0 à 3, le VOLUME (l'intensité sonore) de ce qui sortira sur le canal A. Comme en Basic, vous avez ainsi droit aux valeurs décimales 0 à 15. Mais attention ! Le bit 4 peut aussi être employé : s'il contient 0, le volume sera ce que vous avez choisi, de 0 à 15, et surtout il sera constant. Mais, si vous le mettez à 1, le PSG ne tiendra pas compte du volume défini par les bits 0 à 3, il en produira un autre ! Lequel ? Celui, modulé (c'est-à-dire d'intensité variable, et non plus constante), déterminé par l'ENVELOPPE, dont je vous entretiens plus loin, à propos des registres 11, 12 et 13. Patience.

- R9 fait pour le canal B le même travail que R8.

- R10 fait pour le canal C le même travail que R8 et R9.

A présent, parlons enveloppe. Vous le voyez, je ne vous ai pas fait attendre longtemps.

- R11 et R12 sont employés conjointement, et ils déterminent la durée totale du son ou du bruit, valable pour UNE période d'enveloppe

(certaines enveloppes n'ont qu'une période ; d'autres se répètent indéfiniment, comme nous allons le voir) ; comme d'habitude, poids faible dans R11, poids fort dans R12. Sur seize bits, vous pouvez choisir une valeur de 0 à 65535.

- R13 contrôle l'ENVELOPPE, c'est-à-dire que, par cet intermédiaire, et en choisissant un nombre décimal de 0 à 15, vous sélectionnez l'une des huit enveloppes préprogrammées que connaît le PSG. Je vois d'ici des mouvements divers dans la foule : vous ignorez qu'il existait des enveloppes préprogrammées, toutes faites, dans votre Amstrad ? Le Manuel de l'utilisateur qu'on vous a vendu avec votre ordinateur ne vous en a rien dit ? Eh bien non, il a fait l'impasse sur la question (et sur quelques autres, du reste...).

Sachez donc que le PSG est capable de produire lui-même, sans que vous utilisiez l'instruction Basic ENV, deux enveloppes finies, limitées dans le temps (c'est-à-dire ne s'exécutant qu'une fois, avec une seule période : bref, s'arrêtant toutes seules), et six enveloppes continues, de celles qu'on est obligé d'arrêter si on ne veut pas devenir à la longue légèrement nerveux, mais la chose est aisée. La figure 1 les schématise approximativement, mais un petit commentaire n'est pas inutile.

Prenons par exemple l'enveloppe 1 : elle est FINIE, c'est-à-dire qu'elle n'est exécutée qu'une fois ; elle se compose d'une attaque courte (le son monte d'un seul coup à son maximum), puis l'intensité sonore diminue progressivement jusqu'à tomber à zéro. Le tout est produit une seule fois : une seule période, dont la durée est fixée par R11 et R12, comme expliqué plus haut. L'enveloppe 2 est également finie, mais ici, le son monte progressivement, puis retombe brusquement à zéro.

L'enveloppe 5, elle, est CONTINUE : le son monte progressivement, puis, arrivé à son maximum, reste stable, et se prolonge jusqu'à perpétuité si vous ne vous faites rien pour le stopper. L'enveloppe 3 ressemble à l'enveloppe 1, mais, dès qu'une période est achevée, une autre recommence, et là aussi jusqu'à ce que vous interveniez.

Comment obtenir ces différentes enveloppes ? En utilisant le mini-tableau suivant :

ENVELOPPE VOULUE	VALEURS DECIMALES A ENVOYER A R13
1	0 - 1 - 2 - 3 - 9 (au choix)
2	4 - 5 - 6 - 7 - 15 (au choix)
3	8
4	14
5	13
6	12
7	10
8	11

Voilà donc passés en revue les quatorze registres utiles du PSG.

A présent, le cadeau Bonux.

Comme je l'ai fait observer perfidement, le PRINT CHR\$(7) de l'Amstrad est d'une indigence rare, alors qu'il était si facile de produire des sons plus harmonieux ou plus sophistiqués. Je vous ai donc concocté, à titre d'illustration, un petit RSX qui vous apportera six sons supplémentaires, désignés par des onomatopées suffisamment parlantes : PING, CLIC, CLAC, PAN, BOUM et PIOU. Le chargeur Basic présent quelque part dans ces pages vous l'installera. Vous pouvez vous amuser à le désassembler, il est très simple à comprendre. Les données à fournir au PSG se trouvent en fin de programme, dans un tableau de six fois quatorze octets, et vous pouvez aussi tenter de modifier les sons en remplaçant par d'autres valeurs les valeurs non nulles.

Autre chose : placé en &BE80, ce petit programme résiste au RESET, et même, si vous avez un MICRO-SAVE, à l'extinction de l'ordinateur ; il faut seulement le réactiver avec un CALL &BE80.

Deux derniers détails : je vous disais qu'il était facile de mettre fin à un son dont l'enveloppe est continue. Il suffit d'appeler le vecteur &BCA7. Cherchez-le, il s'y trouve. Et puis, pour vous démontrer que le Basic est trop lent, sachez que PIOU est l'équivalent en langage-machine de la ligne Basic suivante :

```
FOR I=0 TO 112:SOUND 1,,1,15:NEXT  
Chronométrez avec la variable TIME : le langage-machine est sept fois plus rapide !
```

Bon courage, et envoyez vos productions à CPC-INFOS, que tout le monde en profite.

Jean-Pierre MARQUET

SQUARES

Cyril CAUCHOIS & Frédéric DUPUIS

Valable pour CPC 664 et 6128

Connaissez-vous le théorème des 4 couleurs ? Non ? Eh bien il s'agit d'un théorème qui dit qu'avec seulement 4 couleurs, on peut remplir n'importe quelle carte divisée en cases sans que 2 couleurs semblables se touchent. Vous ne nous croyez pas ? Eh bien constatez vous-même !!!

Après le chargement, l'utilisateur pourra soit sélectionner le clavier, soit le joystick, soit la redéfinition des touches, soit regarder le tableau des scores ou bien quitter le jeu.

SQUARES est composé de 3 fichiers :

- **SQUARES.BAS** (Programme)
- **SQUARES.BIN** (Routine)
- **SQUARES.REC** (Tableau des scores)

Le fichier SQUADATA.BAS crée le fichier SQUARES.RTN.

Le fichier TABLEAU.BAS crée le fichier SQUARES.REC.

LE JEU

Au début de la partie, après avoir choisi soit le joystick, soit le clavier (touches directionnelles

et COPY par défaut), l'ordinateur vous demande une touche pour la PAUSE et ensuite à quel niveau vous voulez commencer sachant que :

- **niveau 1** : vous avez 60 couleurs de plus que le nombre total de cases
- **niveau 2** : vous avez 40 couleurs de plus que le nombre total de cases
- **niveau 3** : vous avez 20 couleurs de plus que le nombre total de cases
- **niveau 4** : vous avez autant de couleurs que le nombre total de cases

Pour chaque tableau, vous avez 5 minutes pour le finir (pour une plus grande difficulté de jeu, vous ne voyez pas le temps qui s'écoule).

Bonne chance et prévoyez quelques tubes d'aspirine pour le niveau 4 (on vous le promet, tous les tableaux ont plusieurs solutions).

TABLEAU .BAS

```
1# CLS:MODE 1
2# DIM sc(11),nm$(11)
3# sc(1)=5000:nm$(1)="DAVE"
4# sc(2)=10000:nm$(2)="STEEVE"
5# sc(3)=15000:nm$(3)="CHRIS"
6# sc(4)=20000:nm$(4)="MIKE"
7# sc(5)=25000:nm$(5)="IAN"
8# sc(6)=30000:nm$(6)="NICK"
9# sc(7)=35000:nm$(7)="CYRIL"
10# sc(8)=40000:nm$(8)="BOB"
11# sc(9)=45000:nm$(9)="PETER"
12# sc(10)=50000:nm$(10)="FRED"
13# OPENOUT"SQUARES.REC"
14# FOR a=1 TO 10
15# WRITE #9,sc(a),nm$(a)
16# NEXT
17# CLOSEOUT
18# END
```

SQUARES .BAS

```
1# ***** >LA
2# ** >LB
3# ** SQUARES ** >LC
4# ** >LD
5# ** BY ** >LE
6# ** >LF
7# ** FREDERIC DUPUIS ** >LG
8# ** >LH
9# ** AND ** >LJ
10# ** >RB
11# ** CYRIL CAUCHOIS ** >RC
12# ** >RD
13# ***** >RE
14# KEY DEF 66,0,0,0,0:POKE &BDEE,&C9 >CB
15# MEMORY &9857:LOAD"SQUARES.RTN",&9858 >KX
16# CALL &9C4# >KK
17# DIM sc(11),nm$(11) >PJ
18# OPENIN"SQUARES.REC" >UT
19# FOR a=1 TO 10 >BE
20# INPUT #9,sc(a),nm$(a) >TB
21# NEXT >DJ
22# CLOSEIN >EF
23# DEF FN fm(x,y)=&C000+(y-1)*80+(x-1)*2 >GG
24# up#=CHR$(240):down#=CHR$(241):lft#=CHR$(242):rgt#=C >GB
HR$(243):fir#=CHR$(224)
25# CLS:MODE 0:INK 0,0:INK 1,4:INK 4,17:INK 5,26:INK 6, >JF
8:BORDER 0
```



```

260 PLOT 165,370,5:DRAWR 30,0:DRAWR 0,-10:DRAWR -20,0:D >LR
RAWR 0,-10:DRAWR 20,0:DRAWR 0,-30:DRAWR -30,0:DRAWR 0,1
0:DRAWR 20,0:DRAWR 0,10:DRAWR -20,0:DRAWR 0,30:MOVE 170
,360:FILL 5
270 PLOT 205,370,4:DRAWR 30,0:DRAWR 0,-60:DRAWR -10,0:D >NF
RAWR 0,10:DRAWR -20,0:DRAWR 0,50:DRAWR 10,-10:DRAWR 10,
0:DRAWR 0,-30:DRAWR -10,0:DRAWR 0,30:MOVE 210,360:FILL
4
280 PLOT 245,370,6:DRAWR 10,0:DRAWR 0,-40:DRAWR 10,0:DR >LP
AWR 0,40:DRAWR 10,0:DRAWR 0,-50:DRAWR -30,0:DRAWR 0,50:
MOVE 250,360:FILL 6
290 PLOT 285,370,1:DRAWR 30,0:DRAWR 0,-50:DRAWR -10,0:D >WC
RAWR 0,20:DRAWR -10,0:DRAWR 0,-20:DRAWR -10,0:DRAWR 0,5
0:DRAWR 10,-10:DRAWR 10,0:DRAWR 0,-10:DRAWR -10,0:DRAWR
0,10:MOVE 290,360:FILL 1
300 PLOT 325,370,6:DRAWR 30,0:DRAWR 0,-30:DRAWR -10,0:D >AT
RAWR 0,-10:DRAWR 10,0:DRAWR 0,-10:DRAWR -10,0:DRAWR 0,1
0:DRAWR -10,0:DRAWR 0,-10:DRAWR -10,0:DRAWR 0,50:DRAWR
10,-10
310 DRAWR 10,0:DRAWR 0,-10:DRAWR -10,0:DRAWR 0,10:MOVE >TP
330,360:FILL 6:MOVE 350,325:FILL 6
320 PLOT 365,370,4:DRAWR 30,0:DRAWR 0,-10:DRAWR -20,0:D >QB
RAWR 0,-10:DRAWR 10,0:DRAWR 0,-10:DRAWR -10,0:DRAWR 0,-
10:DRAWR 20,0:DRAWR 0,-10:DRAWR -30,0:DRAWR 0,50:MOVE 3
70,360:FILL 4
330 PLOT 405,370,5:DRAWR 30,0:DRAWR 0,-10:DRAWR -20,0:D >LH
RAWR 0,-10:DRAWR 20,0:DRAWR 0,-30:DRAWR -30,0:DRAWR 0,1
0:DRAWR 20,0:DRAWR 0,10:DRAWR -20,0:DRAWR 0,30:MOVE 410
,360:FILL 5
340 RESTORE 3670 >LE
350 RESTORE 3670 >LF
360 FOR b=1 TO 7:GOSUB 3890:NEXT >ZG
370 CLEAR INPUT >LW
380 f$=INKEY$:IF f$="" THEN 380 >XQ
390 IF f$="1" THEN jsk=0:GOTO 1360 >AY
400 IF f$="2" THEN jsk=1:GOTO 1360 >AR
410 IF f$="3" THEN 550 >NY
420 IF f$="4" THEN sc=0:GOTO 910 >YZ
430 IF f$="5" THEN MODE 1:INK 0,1:INK 1,24:PEN 1:BORDER >DX
1:END
440 IF f$=fir$ THEN jsk=0:GOTO 1360 >BB
450 IF INKEY(76)<>-1 THEN jsk=1:GOTO 1360 >HB
460 IF INKEY(64)<>-1 THEN jsk=0:GOTO 1360 >HY
470 IF INKEY(65)<>-1 THEN jsk=1:GOTO 1360 >HB
480 IF INKEY(57)<>-1 THEN 550 >WZ
490 IF INKEY(56)<>-1 THEN sc=0:GOTO 910 >FR
500 IF INKEY(49)<>-1 THEN MODE 1:INK 0,1:INK 1,24:PEN 1 >ND
:BORDER 1:END
510 GOTO 380 >ZD
520 ***** >RH
530 ** REDEFINITION TOUCHES ** >RJ
540 ***** >RK
550 CLS >UE
560 CLEAR INPUT >LI
570 x1=18:y1=1:a$="UP":GOSUB 3900 >AA
580 f$=INKEY$:IF f$="" THEN 580 >XV
590 IF JOY(0)<># THEN 580 >RJ
600 up$=f$ >LK
610 x1=17:y1=25:a$="DOWN":GOSUB 3900 >DX
620 f$=INKEY$:IF f$="" THEN 620 >XJ
630 IF JOY(0)<># THEN 620 >RY
640 IF f$=up$ THEN 620 >PG
650 down$=f$ >FE
660 x1=0:y1=12:a$="LEFT":GOSUB 3900 >CU
670 f$=INKEY$:IF f$="" THEN 670 >XV
680 IF JOY(0)<># THEN 670 >RJ
690 IF f$=up$ THEN 670 >PT
700 IF f$=down$ THEN 670 >RY
710 lft$=f$ >VH
720 x1=35:y1=12:a$="RIGHT":GOSUB 3900 >FW
730 f$=INKEY$:IF f$="" THEN 730 >XN
740 IF JOY(0)<># THEN 730 >RC
750 IF f$=up$ THEN 730 >PL
760 IF f$=down$ THEN 730 >RB
770 IF f$=lft$ THEN 730 >QY
780 rgt$=f$ >XB
790 x1=17:y1=12:a$="FIRE":GOSUB 3900 >DH
800 f$=INKEY$:IF f$="" THEN 800 >XJ
810 IF JOY(0)<># THEN 800 >RY
820 IF f$=up$ THEN 800 >PG
830 IF f$=down$ THEN 800 >RX
840 IF f$=lft$ THEN 800 >QU
850 IF f$=rgt$ THEN 800 >QC
860 fir$=f$ >VJ
870 GOTO 250 >ZJ
880 ***** >TG
890 ** TABLEAU DES SCORES ** >TH
900 ***** >RK
910 CLS:MODE 0:INK 0,0:INK 1,4:INK 4,17:INK 5,26:INK 6, >JJ
8:BORDER 0
920 IF sc<sc(1) THEN 1240 >TG
930 CLS:MODE 0 >UK
940 x1=9:y1=12:a$="ENTER YOUR NAME :":GOSUB 3900 >UB
950 SPEED KEY 10,5 >LF
960 FOR k=1 TO 6:letr$(k)="" :NEXT k >BZ
970 x1=28:FOR n=1 TO 6 >PJ
980 CLEAR INPUT >LD
990 f$=INKEY$:IF f$="" THEN 990 >XF
1000 IF INKEY(79)<>-1 AND n>1 THEN x1=x1-i:n=n-1:letr$( >QX
n)="" :a$="" :GOSUB 3900:GOTO 990
1010 IF INKEY(18)<>-1 THEN 1090 >XT
1020 f$=UPPER$(f$) >MB
1030 IF f$<"0" OR f$>"Z" THEN 990 >XL
1040 IF f$<"A" AND f$>"9" THEN 990 >XH
1050 letr$(n)=f$ >LC
1060 a$=f$:GOSUB 3900 >PZ
1070 x1=x1+1 >BA
1080 NEXT n >VK
1090 nm$=letr$(1)+letr$(2)+letr$(3)+letr$(4)+letr$(5)+l >JC
etr$(6)
1100 IF nm$="" THEN nm$="M<X" >VA
1110 FOR s=10 TO 1 STEP -1 >RG
1120 IF sc<sc(s) THEN 1230 >TU
1130 FOR v=2 TO s >GB
1140 sc(v-1)=sc(v):nm$(v-1)=nm$(v) >CG
1150 NEXT v >WF
1160 sc(s)=sc:nm$(s)=nm$ >TT
1170 OPENDOUT "SQUARES.REC" >WZ

```



```
118# FOR k=1 TO 10 >LG RAWR #,-76
119# WRITE #9,sc(k),nm#(k) >UI 167# PLOT b,325,9:DRAWR 18#,#:DRAWR #,72:DRAWR -18#,#:D >XD
120# NEXT k >VA RAWR #,-72
121# CLOSEOUT >YE 168# NEXT >LG
122# GOTO 124# >LK 169# FOR a=242 TO 398 STEP 4:PLOT a,1#,#:DRAWR #,56:NEX >ZB
123# NEXT s >WB T
124# CLS:MODE # >AF 170# PLOT 23#,#,9:DRAWR 18#,#:DRAWR #,76:DRAWR -18#,#:D >KF
125# x1=12:y1=8:a#="SCORE NAME":GOSUB 39##:y1=9 >UD RAWR #,-76
126# FOR c=1# TO 1 STEP -1 >RV 171# PLOT 23#,2,9:DRAWR 18#,#:DRAWR #,72:DRAWR -18#,#:D >KA
127# x1=18-LEN(STR$(sc(c))):y1=y1+i:a#=MID$(STR$(sc(c)) >WF RAWR #,-72
,2,LEN(STR$(sc(c)))-1):GOSUB 39##
128# x1=19:a#=nm#(c):GOSUB 39## >YH 172# RESTORE 364#:FOR b=1 TO 6:GOSUB 389#:NEXT >PU
129# NEXT >LD 173# RESTORE 179# >LD
130# PLOT 176,3#4,4:DRAW 176,8#:DRAW 416,8#:DRAW 416,3# >PX 174# x=2#8:y=377 >LH
4:DRAW 176,3#4 >LH 175# READ b:PLOT x,y,b >QR
131# CLEAR INPUT >LH 176# READ b:IF b=-1 THEN y=y-4:GOTO 175# ELSE IF b=-2 T >EY
132# IF INKEY#(<)"** THEN 25# ELSE 132# >BU 177# DRAWR 4,#,b:MOVER -4,-2:DRAWR 4,#,b:MOVER #,2 >TG
133# '***** >XK 178# GOTO 176# >NH
134# '** DESSIN DECOR ** >YA 179# DATA 3,3,8,8,8,8,3,3,3,3,8,8,8,3,3,3,3,8,8,3,3,8,8 >ZY
135# '***** >YB ,3,3,3,3,8,8,3,3,8,8,8,8,8,8,3,3,8,8,8,8,8,8,3,3,3,3,
136# CLS:MODE # >AJ 8,8,8,8,3,3,-1
137# x1=16:y1=12:a#="PAUSE":GOSUB 39## >FK 180# DATA 3,8,8,3,3,8,8,3,3,8,8,3,3,8,8,3,3,8,8,3,3,8,8 >ZY
138# CLEAR INPUT >LQ ,3,3,3,8,8,8,8,3,3,3,8,8,3,3,8,8,3,3,8,8,3,3,8,8,3,3,8,8,
139# f#=INKEY#:IF f#="" THEN 139# >ZY 8,3,3,8,8,3,-1
140# IF JOY(#)<# THEN 139# >UP 181# DATA 3,8,8,3,3,3,3,8,8,3,3,3,8,8,3,3,8,8,3,3,8,8,3,3,8,8 >ZC
141# IF f#=up# THEN 139# >QA ,3,3,8,8,3,3,8,8,3,3,8,8,3,3,8,8,3,3,8,8,3,3,8,8,3,3,3,8,
142# IF f#=down# THEN 139# >TV 8,3,3,3,3,3,-1
143# IF f#=lft# THEN 139# >RP 182# DATA 3,3,8,8,8,8,3,3,8,8,3,3,3,8,8,3,3,8,8,3,3,8,8 >ZL
144# IF f#=rgt# THEN 139# >RY ,3,3,8,8,3,3,8,8,3,3,8,8,8,8,8,8,3,3,8,8,8,8,3,3,3,3,3,
145# IF f#=fir# THEN 139# >RL 8,8,8,8,3,3,-1
146# psd#=# >CA 183# DATA 3,3,3,3,3,8,8,3,8,8,3,8,8,3,8,8,3,8,8,3,3,8,8 >ZV
147# CLS:MODE # >BA ,3,3,8,8,8,8,8,8,8,3,3,8,8,3,8,8,3,3,3,8,8,3,8,8,3,3,3,3,3,
148# x1=2:y1=1#:#a#="A QUEL NIVEAU VOULEZ VOUS COMMENCEZ >XV 3,3,3,8,8,3,-1
?":GOSUB 39##
149# x1=12:y1=12:a#="1 2 3 4":GOSUB 39## >NJ 184# DATA 3,8,8,3,3,8,8,3,8,8,3,8,8,3,3,8,8,3,3,8,8,3,3,8,8 >ZC
150# CLEAR INPUT >LJ 8,3,3,8,8,3,-1
151# f#=INKEY#:IF f#="" THEN 151# >ZK 185# DATA 3,3,8,8,8,8,3,3,3,8,8,8,3,8,8,3,3,3,8,8,8,8,8,8,3 >ZG
152# IF f#<"1" OR f#>"4" THEN 151# >XC ,3,3,8,8,3,3,8,8,3,8,8,3,8,8,3,3,8,8,8,8,8,8,8,8,8,3,3,3,
153# niv=VAL(f#) >LY 8,8,8,8,3,3,-2
154# CLS:MODE # >AJ 186# FOR c=# TO 42# STEP 42# >UH
155# RESTORE 156#:FOR a=# TO 15:READ b:INK a,b:NEXT:BOR >EG 187# FOR b=12 TO 116 STEP 1#4 >UW
DER # >WB 188# FOR a=# TO 88 STEP 4 >QE
156# DATA #,5,18,2,14,2#,5,13,26,1,24,6,17,#,#,# >MD 189# PLOT a+b+c,1#,#:DRAWR #,58 >YI
157# IF np=1 THEN !SCREENCOPY,1,4:GOTO 2#5# >MF 190# NEXT a,b >JC
158# FOR a=# TO 18# STEP 4:PLOT a,399,3:DRAWR #,-76:NEX >ZP 191# PLOT c,#,9:DRAWR #,76:DRAWR 218,#:DRAWR #,-76:DRAW >BU
T >WB R -218,#:PLOT c,2,9:DRAWR #,72:DRAWR 218,#:DRAWR #,-72:
159# FOR a=192 TO 448 STEP 4:PLOT a,399,3:DRAWR #,-76:N >BK DRAWR -218,#:PLOT 1#8+c,#,9:DRAWR #,76
EXT >WB 192# DEG >YD
160# FOR a=46# TO 64# STEP 4:PLOT a,399,3:DRAWR #,-76:N >BT 193# FOR b=4# TO 144 STEP 1#4 >UY
EXT >WB 194# ORIGIN b+c,4# >MP
161# FOR a=# TO 218 STEP 4:PLOT a,#,3:DRAWR #,76:NEXT >WH 195# MOVE #,26,8 >BF
162# FOR a=23# TO 41# STEP 4:PLOT a,#,3:DRAWR #,76:NEXT >YA 196# FOR a=# TO 36# STEP 1# >TK
163# FOR a=422 TO 64# STEP 4:PLOT a,#,3:DRAWR #,76:NEXT >YK 197# DRAW 26*SIN(a),26*COS(a),8 >YQ
164# FOR b=# TO 46# STEP 46# >UQ 198# NEXT a,b >KA
165# FOR a=12 TO 168 STEP 4:PLOT a+b,333,#:DRAWR #,56:N >BZ 199# ORIGIN #,#:NEXT c >QI
EXT >WB 200# MOVE 4#,4#:FILL 11 >QU
166# PLOT b,323,9:DRAWR 18#,#:DRAWR #,76:DBAWR -18#,#:D >LA 201# MOVE 144,4#:FILL 2 >QZ
202# MOVE 46#,4#:FILL 12 >RX
203# MOVE 564,4#:FILL 1# >RB
```



```
204# np=1:SCREENCOPY,4,1 >WX 257# ***** >YG
205# xj=38-LEN(STR$(sc(1#))):yj=4:a#MID$(STR$(sc(1#)), >EL 258# AFTER 150##,3 GOSUB 422# >VJ
2,LEN(STR$(sc(1#)))-1) 259# PLOT x,y+4,#:DRAWR 8,#:PLOT x+4,y:DRAWR 8,8 >RY
206# GOSUB 390# >XC 260# IF nrg(niv)=# AND nvr(niv)=# AND nrs(niv)=# AND nj >DX
207# lvl=1:lf=3:sc=# >PV n(niv)=# THEN GOSUB 421#
208# WINDOW #1,2,19,6,2#:CLS #1 >XQ 261# IF aft=1 THEN 208# >QJ
209# RESTORE 213# >LR 262# IF aft=2 THEN 439# >QT
210# FOR a=1 TO 4 >AG 263# IF jsk=1 THEN 272# >QB
211# READ b:PLOT b,12,#:DRAWR 8,56:DRAWR 88,#:DRAWR 8,- >QR 264# f#=INKEY#:IF f#="" THEN 261# >ZT
56:DRAWR -88,# 265# IF f#=psd# THEN GOSUB 377# >XF
212# NEXT >KG 266# IF f#=up# AND y<296 THEN 280# >ZP
213# DATA 12,116,432,536 >QP 267# IF f#=down# AND y>96 THEN 281# >AX
214# x1=16:y1=1#a#="LEVEL"+STR$(lvl):GOSUB 390# >TR 268# IF f#=lft# AND x>76 THEN 282# >ZG
215# x1=16:y1=12#a#="NIVEAU"+STR$(niv):GOSUB 390# >UZ 269# IF f#=rgt# AND x<556 THEN 283# >AX
216# FOR a=1 TO 250#:NEXT >TF 270# IF f#=fir# THEN 287# >RP
217# aft=#:FOR a=32# TO 6# STEP -4:PLOT a,9#,7:DRAWR 8, >XT 271# GOTO 261# >MG
22#:PLOT 64#-a,9#,7:DRAWR 8,22#:NEXT 272# f#=INKEY#:IF f#="" THEN 261# >ZR
218# FOR a=6# TO 58# STEP 4#:PLOT a,9#,8:DRAWR 8,22#:NE >AX 273# IF f#=psd# THEN GOSUB 377# >XE
IT 274# IF INKEY(72)<>-1 AND y<296 THEN 280# >FZ
219# FOR a=9# TO 31# STEP 2#:PLOT 68#,a,#:DRAWR 52#,8:NE >AP 275# IF INKEY(73)<>-1 AND y>96 THEN 281# >FH
XT 276# IF INKEY(74)<>-1 AND x>76 THEN 282# >FH
220# ***** >XG 277# IF INKEY(75)<>-1 AND x<556 THEN 283# >FG
221# ** LEVEL ** >XH 278# IF INKEY(76)<>-1 THEN 287# >XV
222# ***** >XJ 279# GOTO 261# >NE
223# ON lvl GOSUB 225#,226#,227#,228#,229#,230#,231#,23 >ZQ 280# cs=TEST(x,y+2#):IF cs<># THEN w=y+2#:GOTO 284# ELS >EC
28,233#,234#,235#,236# E 261#
224# GOTO 237# >MH 281# cs=TEST(x,y-2#):IF cs<># THEN w=y-2#:GOTO 284# ELS >EH
225# RESTORE 449#:RETURN >UK E 261#
226# RESTORE 453#:RETURN >UF 282# cs=TEST(x-4#,y):IF cs<># THEN v=x-4#:GOTO 284# ELS >FC
227# RESTORE 457#:RETURN >UL E 261#
228# RESTORE 461#:RETURN >UG 283# cs=TEST(x+4#,y):IF cs<># THEN v=x+4#:GOTO 284# ELS >EH
229# RESTORE 465#:RETURN >UM E 261#
230# RESTORE 469#:RETURN >UH 284# cl=TEST(x,y) >MH
231# RESTORE 473#:RETURN >UD 285# PLOT x,y+4,cl:DRAWR 8,#:PLOT x+4,y:DRAWR 8,8 >UH
232# RESTORE 477#:RETURN >UJ 286# x=v:y=w:GOTO 259# >RB
233# RESTORE 481#:RETURN >UE 287# a=12:a1=12:col=11 >QH
234# RESTORE 485#:RETURN >UK 288# PLOT a,12,#:DRAWR 8,56:DRAWR 88,#:DRAWR 8,-56:DRA >GD
235# RESTORE 489#:RETURN >UQ R -88,#
236# RESTORE 493#:RETURN >UL 289# PLOT a1,12,8:DRAWR 8,56:DRAWR 88,#:DRAWR 8,-56:DRA >HX
237# FOR a=64 TO 54# STEP 4# >UU WR -88,#
238# READ b:IF b=1 THEN 240# >UX 290# a=a1 >CD
239# PLOT a,b,7:DRAWR 34,#:GOTO 238# >DK 291# IF aft=1 THEN 208# >QH
240# NEXT >KH 292# IF aft=2 THEN 439# >QU
241# FOR a=92 TO 292 STEP 2# >UM 293# IF jsk=1 THEN 301# >QI
242# READ b:IF b=1 THEN 244# >UW 294# f#=INKEY#:IF f#="" THEN 291# >ZZ
243# PLOT b,a,7:DRAWR 8,16:GOTO 242# >CH 295# IF f#=lft# AND a=12 THEN 294# >ZY
244# NEXT >LB 296# IF f#=rgt# AND a=536 THEN 294# >AY
245# READ a,b >EJ 297# IF f#=lft# THEN 300# >RY
246# IF a=1 THEN 249# >NY 298# IF f#=rgt# THEN 311# >RA
247# MOVE a,b,#:FILL # >QB 299# IF f#=fir# THEN 314# >RB
248# GOTO 245# >NC 300# GOTO 291# >MC
249# FOR a=1 TO 4 >BJ 301# f#=INKEY#:IF f#="" THEN 291# >ZH
250# READ nrg(a),nvr(a),nrs(a),n#n(a) >FQ 302# IF INKEY(74)<>-1 AND a=12 THEN 301# >EK
251# NEXT >KK 303# IF INKEY(75)<>-1 AND a=536 THEN 301# >FK
252# READ nb# >KC 304# IF INKEY(74)<>-1 THEN 300# >XB
253# GOSUB 329# >XG 305# IF INKEY(75)<>-1 THEN 311# >XI
254# x=316:y=196:w=y:v=x >UM 306# IF INKEY(76)<>-1 THEN 314# >XC
255# ***** >YE 307# GOTO 291# >MK
256# ** LE JEU ** >YF 308# IF a=432 THEN a1=116:col=2:GOTO 288# >HF
```



```
309# IF a=536 THEN a1=432:col=12:GOTO 288# >JC 367# DATA 12,1#, "1<KEYBOARD" >WN
310# IF a=116 THEN a1=12:col=11:GOTO 288# >GA 368# DATA 12,12, "2<JOYSTICK" >WB
311# IF a=432 THEN a1=536:col=1#:GOTO 288# >HB 369# DATA 12,14, "3<REDEFINE KEYS" >BJ
312# IF a=116 THEN a1=432:col=12:GOTO 288# >HY 370# DATA 12,16, "4<SEE HIGH SCORES" >DF
313# IF a=12 THEN a1=116:col=2:GOTO 288# >GK 371# DATA 12,18, "5<QUIT" >RX
314# IF col=2 AND nvr(niv)=# THEN 291# >EA 372# DATA #,24, "PROGRAMMING AND GRAPHICS BY FRED AND BO >FE
315# IF col=1# AND njn(niv)=# THEN 291# >EY B"
316# IF col=11 AND nrg(niv)=# THEN 291# >EB 373# DATA 6,25, "MUSIC ADAPTED BY FRED AND BOB" >QH
317# IF col=12 AND nrs(niv)=# THEN 291# >ER 374# ***** >YG
318# cl=TEST(x,y) >HK 375# ** PAUSE ** >YH
319# IF cl=2 THEN nvr(niv)=nvr(niv)+1 >FZ 376# ***** >YJ
320# IF cl=1# THEN njn(niv)=njn(niv)+1 >FW 377# mf=REMAIN(3) >MV
321# IF cl=11 THEN nrg(niv)=nrg(niv)+1 >FA 378# ;SCREENCOPY,2,1 >QX
322# IF cl=12 THEN nrs(niv)=nrs(niv)+1 >GV 379# WINDOW #1,2,19,6,2#:CLS #1 >XA
323# IF col=2 THEN nvr(niv)=nvr(niv)-1:GOTO 328# >TB 380# x1=14:y1=12:a#="GAME PAUSED":GOSUB 39## >NB
324# IF col=1# THEN njn(niv)=njn(niv)-1:GOTO 328# >TU 381# CLEAR INPUT >LQ
325# IF col=11 THEN nrg(niv)=nrg(niv)-1:GOTO 328# >TY 382# f#=(INKEY#:(F f#="" THEN 382# >ZY
326# IF col=12 THEN nrs(niv)=nrs(niv)-1:GOTO 328# >UT 383# ;SCREENCOPY,1,2 >QT
327# GOTO 291# >NB 384# AFTER mf,3 GOSUB 422# >UA
328# PLOT #,2##,#:MOVE x,y:FILL col:GOTO 338# >NX 385# RETURN >FK
329# x1=4:y1=24:a#=(MID$(STR$(nrg(niv)),2,2):GOSUB 39## >ZD 386# ***** >YK
330# IF nrg(niv)=9 THEN x1=5:y1=24:a#=" ":GOSUB 39## >VR 387# ** ROUTINE D'AFFICHAGE ** >ZA
331# x1=1#:y1=24:a#=(MID$(STR$(nvr(niv)),2,2):GOSUB 39## >AL 388# ***** >ZB
332# IF nvr(niv)=9 THEN x1=11:y1=24:a#=" ":GOSUB 39## >WG 389# READ x1,y1,a# >WT
333# x1=3#:y1=24:a#=(MID$(STR$(nrs(niv)),2,2):GOSUB 39## >AM 390# a#=(UPPER$(a#) >MA
334# IF nrs(niv)=9 THEN x1=31:y1=24:a#=" ":GOSUB 39## >WH 391# FOR a=1 TO LEN(a#) >QN
335# x1=37:y1=24:a#=(MID$(STR$(njn(niv)),2,2):GOSUB 39## >AH 392# z=(ASC(MID$(a#,a,1)))-48 >WK
336# IF njn(niv)=9 THEN x1=38:y1=24:a#=" ":GOSUB 39## >WD 393# IF z<# OR z>42 THEN z=43 >VB
337# RETURN >FG 394# CALL &9858, FN fm(x1,y1)+(a#2),&9878+(z#16) >NJ
338# PLOT a,12,#:DRAWR #,56:DRAWR 88,#:DRAWR #,-56:DRAW >GZ 395# NEXT >LJ
R -88, # 396# RETURN >GB
339# GOSUB 329# >YB 397# ***** >ZB
340# ***** >XX 398# ** GAGNE ** >ZC
341# ** VERIFICATION ** >YA 399# ***** >ZD
342# ***** >YB 400# FOR a=# TO 32# STEP 4 >RY
343# mf=REMAIN(3) >MN 401# PLOT a,8#, #:DRAWR #,24#:PLOT 64#-a,8#, #:DRAWR #,24 >ZX
344# FOR bb=76 TO 556 STEP 4# >VQ # >KH
345# FOR aa=96 TO 296 STEP 2# >VQ 402# NEXT >KH
346# g=TEST(bb,aa) >NL 403# |v|=|v|+1:IF |v|=13 THEN niv=niv+1:|v|=1 >QY
347# IF g=7 THEN AFTER mf,3 GOSUB 422#:GOTO 259# >PV 404# sc=sc+CINT(mf/5) >QQ
348# NEXT aa,bb >BH 405# IF sc<1# THEN x1=7:GOTO 41## >ZR
349# nbc=# >MJ 406# IF sc<1## THEN x1=6:GOTO 41## >AU
350# FOR b=96 TO 296 STEP 2# >UY 407# IF sc<1### THEN x1=5:GOTO 41## >BI
351# FOR a=76 TO 516 STEP 4# >UT 408# IF sc<1#### THEN x1=4:GOTO 41## >CB
352# g1=TEST(a,b) >MH 409# x1=3 >DE
353# g2=TEST(a+4#,b) >PK 410# sc#=(MID$(STR$(sc),2,LEN(STR$(sc))-1) >KD
354# IF g1=g2 THEN nbc=nbc+1 >VV 411# y1=4:a#=#:GOSUB 39## >VF
355# NEXT a,b >JF 412# IF niv<>5 THEN 417# >RP
356# FOR b=76 TO 556 STEP 4# >UD 413# x1=13:y1=1#:a#="CONGRATULATION":GOSUB 39## >TK
357# FOR a=96 TO 276 STEP 2# >UC 414# x1=5:y1=12:a#="YOU>VE FINISHED ALL THE SCREEN":GOS >NG
358# g1=TEST(b,a) >MP UB 39##
359# g2=TEST(b,a+2#) >PP 415# FOR a=1 TO 25##:NEXT >TG
360# IF g1=g2 THEN nbc=nbc+1 >VR 416# GOTO 91# >FG
361# NEXT a,b >JC 417# GOTO 288# >MK
362# IF nbc<>nb# THEN AFTER mf,3 GOSUB 422#:GOTO 259# >WJ 418# ***** >YF
363# GOTO 4## >MD 419# ** PERDU ** >YG
364# DATA 3,2, "SCORE",3,4, "####" >ZP 420# ***** >XJ
365# DATA 3#,2, "HIGHSCORE",32,4, "####" >FW 421# aft=3 >NA
366# DATA 17,22, "CREDIT",19,24, "#3" >BQ 422# mf=REMAIN(3) >MK
```



```
423# FOR a=6# TO 32# STEP 4 >TD
424# PLOT a,9#,#;DRAW# #,22#;PLOT 64#-a,9#,#;DRAW# #,22 >ZA
#
425# NEXT >LC
426# IF=IF-1 >DE
427# IF aft=3 THEN 429# >QW
428# x1=16:y1=1#:a#="TIME UP":GOSUB 39##:GOTO 432# >UK
429# x1=#:y1=11:a#="YOU>VE NO COLOR LEFT = YOU LOSE A C >AZ
REDIT":GOSUB 39##
430# FOR a=1 TO 25##:NEXT >TD
431# WINDOW #1,1,2#,6,2#;CLS #1 >XD
432# x1=9:y1=12 >AH
433# IF If>1 THEN a#="YOU HAVE":b#-STR$(If):c#=" CREDIT >GC
S LEFT":a#-a#+b#+c#:GOSUB 39##:GOTO 436#
434# IF If>-1 THEN a#="YOU HAVE":b#-STR$(If):c#=" CREDI >FU
T LEFT":a#-a#+b#+c#:GOSUB 39##:GOTO 436#
435# a#="YOU HAVE NO CREDIT LEFT":GOSUB 39##:GOTO 438# >AI
436# x1=2#:y1=24:a#="MID$(STR$(If),2,LEN(STR$(If))-1):GO >KE
SUB 39##
437# FOR a=1 TO 25##:NEXT:aft=1:RETURN >GF
438# FOR a=1 TO 25##:NEXT:aft=2:RETURN >GH
439# FOR a=32# TO 6# STEP -4 >UE
440# PLOT a,9#,#;DRAW# #,22#;PLOT 64#-a,9#,#;DRAW# #,22 >ZY
#
441# NEXT >LA
442# x1=15:y1=11:a#="GAME OVER":GOSUB 39## >KN
443# FOR a=1 TO 25##:NEXT >TH
444# GOTO 91# >FH
445# '***** >YF
446# '** LES TABLEAU ** >YG
447# '***** >YH
448# '** TABLEAU 1 ** >YJ
449# DATA 25#,23#,17#,11#,1,29#,27#,15#,1,25#,13#,1,29# >AU
,23#,17#,1,27#,25#,15#,13#,1,29#,23#,17#,11#,1,29#,27#,
25#,23#,17#,15#,13#,11#,1,29#,23#,17#,11#,1,27#,25#,15#
,13#,1,29#,23#,17#,1,25#,13#,1,29#,27#,15#,1,25#,23#,17
#,11#,1
450# DATA 22#,26#,38#,42#,1,18#,46#,1,14#,22#,26#,38#,4 >TC
2#,5##,1,1##,54#,1,18#,22#,42#,46#,1,1##,14#,18#,22#,26
#,38#,42#,46#,5##,54#,1,14#,26#,38#,5##,1,18#,46#,1,14#
,22#,42#,5##,1,1##,54#,1,18#,22#,42#,46#,1
451# DATA 1,1,22,32,3#,27,17,27,25,22,12,22,2#,17,7,17, >EQ
15,12,92
452# '** TABLEAU 2 ** >YD
453# DATA 25#,21#,19#,15#,1,19#,17#,1,25#,23#,19#,1,21# >PT
,17#,15#,1,25#,21#,19#,1,23#,17#,1,25#,21#,19#,15#,1,23
#,17#,1,25#,21#,19#,1,21#,17#,15#,1,25#,23#,19#,1,19#,1
7#,1,25#,21#,19#,15#,1
454# DATA 1,14#,26#,38#,5##,1,1##,3##,34#,54#,1,18#,22# >XB
,42#,46#,1,3##,34#,1,18#,26#,38#,46#,1,1##,3##,34#,54#,
1,1,14#,22#,26#,38#,42#,5##,1,14#,26#,38#,5##,1,1
455# DATA 1,1,41,36,31,25,36,31,26,2#,31,26,21,15,26,21 >FB
,16,1#,7#
456# '** TABLEAU 3 ** >YH
457# DATA 27#,23#,21#,19#,17#,13#,1,29#,15#,11#,1,27#,2 >GJ
3#,21#,19#,1,29#,21#,17#,15#,11#,1,25#,19#,13#,1,29#,25
#,23#,17#,15#,11#,1,29#,27#,21#,19#,13#,11#,1,29#,23#,1
7#,11#,1,21#,13#,1,27#,23#,21#,19#,17#,11#,1,19#,13#,1,
25#,17#,1,29#,27#,23#,21#,19#,17#
```



```

388,288,388,248,388,288,388,328,388,368,388,488,388,448
,388,488,388,528,388,268,288,388,288,1,1,25,16,24,21,28
,11,19,16,15,6,14,11,18,1,9,6,149
476# '*** TABLEAU 8 ** >YK
477# DATA 298,278,258,238,218,198,178,158,138,118,1,1,2 >QJ
18,158,138,1,278,238,198,178,1,238,218,198,178,1,1,258
,238,178,158,1,1,238,218,198,178,1,258,218,138,1,278,158
,138,1,1,298,278,258,238,218,198,178,158,138,118,1
478# DATA 1,188,268,388,348,388,468,1,228,428,468,1,268 >WP
,388,1,188,1,268,388,1,188,1,268,388,468,1,228,428,1,18
8,268,388,348,388,468,1,1
479# DATA 288,228,268,188,368,228,368,188,1,1,38,31,35, >BA
25,33,26,38,28,28,21,25,15,23,16,28,18,71
480# '*** TABLEAU 9 ** >YE
481# DATA 298,278,238,218,178,138,118,1,298,258,198,158 >GG
,118,1,278,238,1,298,218,138,1,278,238,198,178,158,118
,1,258,1,278,238,218,178,158,118,1,258,198,1,278,238,218
,178,158,118,1,298,258,198,138,118,1,278,238,218,178,1,
298,258,198,158,118,1,298,278,238
482# DATA 218,138,118,1,188,148,228,388,588,548,1,188,1 >NZ
88,468,548,1,588,1,188,188,428,548,1,228,388,588,1,148
,188,268,388,428,1,148,388,348,468,588,1,188,188,268,388
,428,548,1,148,228,588,1,188,188,348,388,468,548,1,188
,148,228,268,388,348,388,428,588,548
483# DATA 1,128,388,528,388,128,188,528,188,1,1,24,25,2 >ZT
4,19,19,28,19,14,14,15,14,9,9,18,9,4,111
484# '*** TABLEAU 10 ** >YJ
485# DATA 158,118,1,218,198,178,1,198,178,158,138,118,1 >BG
,278,258,218,178,158,138,1,298,218,198,178,158,138,1,27
8,1,178,158,138,1,278,258,118,1,138,118,1,298,258,1,278
,178,1,298,238,198,158,138,1,278,258,178,158,138,118,1
486# DATA 148,228,268,388,348,388,428,468,588,548,1,188 >GN
,268,388,468,588,1,188,148,428,1,588,1,188,348,388,428
,468,548,1,228,588,1,188,228,268,388,348,388,428,468,548
,1,188,148,268,348,388,588,1,188,188,228,388,428,548,1,
268,348,388,588,1,188,148,188,228
487# DATA 428,468,1,1,1,38,22,22,27,25,17,17,22,28,12,1 >TG
2,17,15,7,7,12,184
488# '*** TABLEAU 11 ** >ZC
489# DATA 278,258,238,218,198,178,158,138,1,298,118,1,2 >BP
78,258,238,218,198,178,158,138,1,238,218,198,178,1,218
,1,1,1,1,198,1,238,218,198,178,1,278,258,238,218,198,178
,158,138,1,298,118,1,278,258,238,218,198,178,158,138,1
490# DATA 188,188,228,268,388,348,388,428,468,548,1,148 >GX
,588,1,188,228,268,388,348,388,428,468,1,228,268,388,34
8,388,428,1,268,388,348,388,1,1,268,388,348,388,1,228,2
68,388,348,388,428,1,188,228,268,388,348,388,428,468,1,
148,588,1,188,188,228,268,388,348
491# DATA 388,428,468,548,1,1,1,38,26,26,17,25,21,21,12 >DH
,28,16,16,7,15,11,11,2,186
492# '*** TABLEAU 12 ** >YH
493# DATA 298,278,238,218,198,178,138,118,1,238,218,198 >KH
,178,1,238,178,1,218,198,1,1,238,178,1,258,218,198,158
,1,238,178,1,1,218,198,1,238,178,1,238,218,198,178,1,298
,278,238,218,198,178,138,118,1
494# DATA 228,268,388,348,388,428,1,1,188,148,188,228,2 >GV
68,388,348,388,428,468,588,548,1,188,148,228,268,388,42
8,588,548,1,188,148,228,428,588,548,1,188,188,388,348,4
68,548,1,188,148,228,428,588,548,1,188,148,228,268,388,

```

```

428,588,548,1,188,148,188,228,268
495# DATA 388,348,388,428,468,588,548,1,1,228,268,388,3 >TG
48,388,428,1,128,288,528,288,168,388,478,388,188,188,47
8,188,1,1,15,21,28,35,18,16,23,38,5,11,18,25,8,6,13,28,
118

```

SQUADATA .BAS

```

18 '*****
28 '*** DATAS POUR SQUARES.RTN **
38 '*****
48 MODE 2:adr=39888:nl=138
58 RESTORE 138
68 tot=:FOR a=1 TO 15:READ a#
78 POKE adr,VAL("&"a#):tot=tot+PEEK(adr):adr=adr+1
88 NEXT:READ som#:IF VAL("&"som#)<>tot THEN 128
98 LOCATE 1,1:PRINT "LIGNE";nl;"CORRECTE"
188 nl=nl+18:IF nl=1698 THEN 118 ELSE 68
118 SAVE"SQUARES.RTN",b,8988,898:END
128 LOCATE 1,1:PRINT"LIGNE";nl;"INCORRECTE":END
138 DATA DD,5E,88,DD,58,81,DD,6E,82,DD,66,83,86,88,CS,505
148 DATA ES,86,82,1A,77,23,13,18,FA,E1,CD,26,BC,C1,18,61F
158 DATA EF,C9,58,F8,58,F8,18,18,18,18,14,14,14,14,48,588
168 DATA C8,88,88,88,88,88,58,88,88,28,88,28,88,28,88,28,288
178 DATA 48,C8,88,88,58,F8,58,58,88,18,88,18,88,28,14,38C
188 DATA 88,48,C8,88,88,58,F8,58,58,88,18,88,18,88,3C,38C
198 DATA 88,14,48,C8,88,88,88,58,88,F8,88,38,18,18,14,288
288 DATA 3C,88,14,88,48,88,88,58,F8,58,88,18,88,88,38,268
218 DATA 88,14,14,14,48,C8,88,88,58,F8,58,58,18,88,18,38C
228 DATA 38,14,14,14,14,48,C8,88,88,58,F8,88,58,88,18,328
238 DATA 88,28,88,28,14,88,48,88,88,88,58,F8,58,58,18,28C
248 DATA 18,18,38,14,14,14,14,48,C8,88,88,58,F8,58,58,388
258 DATA 18,18,18,38,88,14,88,14,48,C8,88,88,88,88,88,188
268 DATA 88,88,88,88,88,88,28,88,88,88,88,88,88,88,88,888
278 DATA 88,88,88,88,88,88,88,28,88,88,88,88,48,88,88,888
288 DATA 88,88,88,88,88,88,88,88,88,88,28,88,88,88,88,888
298 DATA 88,88,88,88,88,88,88,88,88,88,28,88,88,48,888
388 DATA 88,88,88,88,88,18,88,88,88,88,88,88,88,88,158
318 DATA 88,88,88,88,58,58,18,18,88,18,88,28,88,88,88,198
328 DATA 88,88,88,88,88,88,88,28,88,28,88,28,88,88,228
338 DATA 88,88,88,88,58,F8,58,F8,18,18,18,18,14,3C,14,3A4
348 DATA 14,48,48,88,88,58,F8,58,F8,18,18,18,18,14,28,388
358 DATA 14,14,48,C8,88,88,58,F8,58,F8,18,88,18,88,14,30C
368 DATA 88,14,88,48,C8,88,88,58,88,58,F8,18,18,18,18,384
378 DATA 14,14,14,14,48,C8,88,88,58,F8,58,F8,18,88,18,3F8
388 DATA 88,14,28,14,88,48,C8,88,88,58,F8,58,F8,18,88,3E8
398 DATA 18,88,14,28,14,88,48,88,88,88,58,F8,58,F8,18,338
488 DATA 18,18,88,14,88,14,14,48,C8,88,88,58,58,58,58,29C
418 DATA 18,18,18,18,14,3C,14,14,48,48,88,88,58,F8,88,278
428 DATA 88,88,28,88,28,88,28,88,28,48,C8,88,88,58,F8,378
438 DATA 88,88,28,88,28,88,28,88,28,48,88,88,88,58,248
448 DATA 58,58,58,18,18,18,28,14,28,14,14,48,48,88,88,224
458 DATA 58,88,58,88,18,88,18,88,14,88,14,88,48,C8,88,1E8
468 DATA 88,58,58,58,F8,18,38,18,18,14,14,14,14,48,48,318
478 DATA 88,88,58,58,58,58,18,38,18,38,14,3C,14,14,48,278
488 DATA 48,88,88,58,F8,58,58,18,18,18,18,14,14,14,14,288
498 DATA 48,C8,88,88,58,F8,58,F8,18,18,18,18,14,3C,14,424
588 DATA 88,48,88,88,88,58,F8,58,F8,18,18,18,18,14,14,328

```



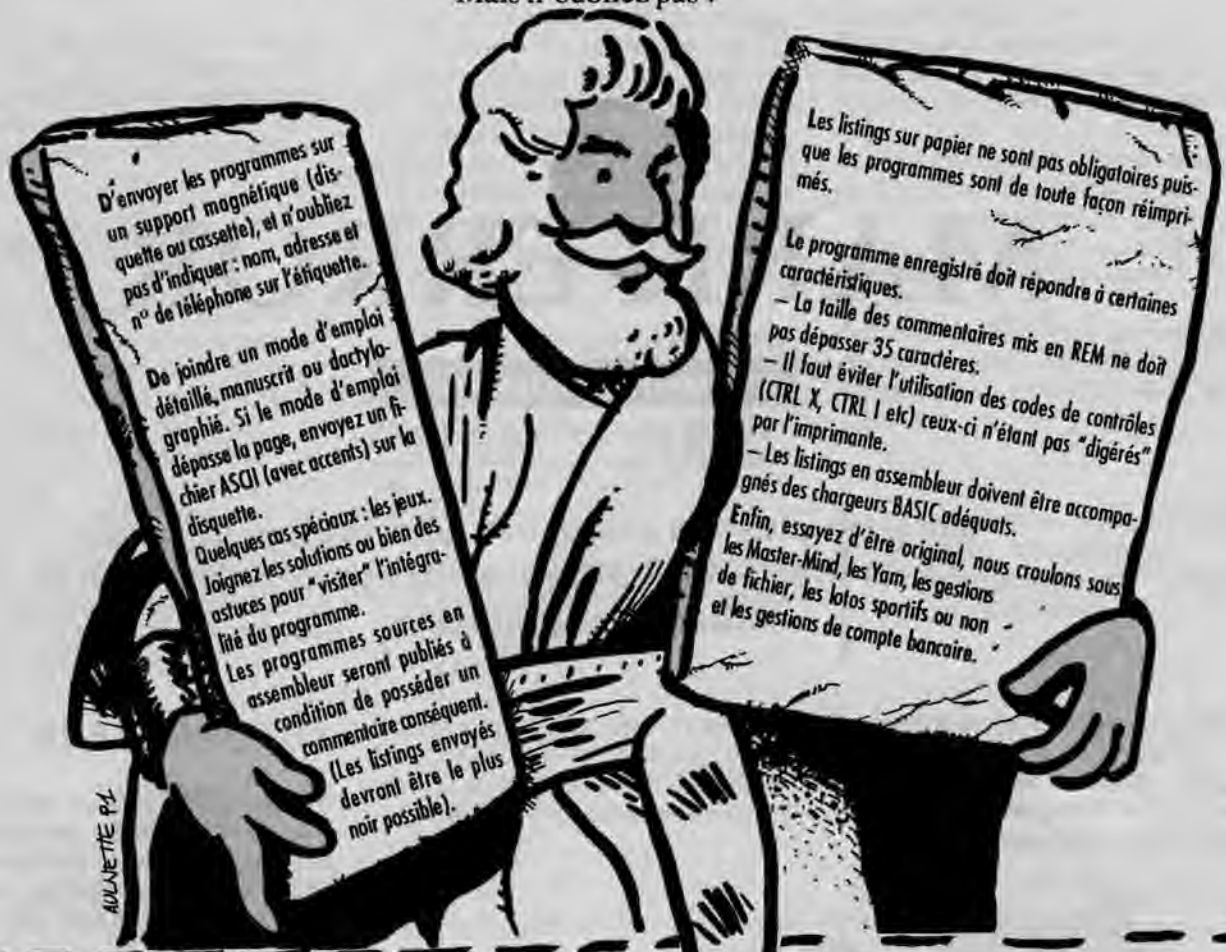
51# DATA 14, 14, 4# ,C# ,## ,## ,5# ,F# ,5# ,F# ,1# ,1# ,1# ,1# ,14, 47C
52# DATA 2# ,14, 14, 4# ,4# ,## ,## ,5# ,F# ,5# ,F# ,1# ,## ,1# ,## ,37#
53# DATA 14, 3C, ## ,14, 4# ,C# ,## ,## ,5# ,F# ,5# ,F# ,## ,2# ,## ,4#
54# DATA 2# ,## ,2# ,## ,2# ,## ,## ,## ,5# ,5# ,5# ,5# ,1# ,1# ,25#
55# DATA 1# ,1# ,14, 14, 14, 14, 4# ,C# ,## ,## ,5# ,5# ,5# ,5# ,1# ,2C#
56# DATA 1# ,1# ,1# ,14, 14, 14, 14, ## ,## ,## ,## ,5# ,5# ,5# ,5# ,24#
57# DATA 1# ,1# ,1# ,1# ,14, 3C, 14, 3C, 4# ,4# ,## ,## ,5# ,5# ,5# ,25#
58# DATA 5# ,1# ,1# ,## ,2# ,## ,2# ,14, 14, 4# ,4# ,## ,## ,5# ,5# ,2##
59# DATA 5# ,5# ,1# ,1# ,## ,2# ,## ,2# ,## ,2# ,## ,## ,## ,5# ,2##
60# DATA F# ,## ,5# ,## ,1# ,## ,2# ,## ,2# ,14, ## ,4# ,C# ,## ,## ,2AC
61# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
62# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
63# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
64# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
65# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
66# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
67# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
68# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
69# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
70# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
71# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
72# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
73# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
74# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
75# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
76# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
77# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
78# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##
79# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,F5, 21, E1, E9, 22, 3#2
80# DATA 3# ,## ,F7, EB, 21, 4E, ## ,19, 4E, 23, 46, 23, 7# ,## ,28, 4C5
81# DATA 14, E5, 6# ,6# ,19, E5, 4E, 23, 46, 6# ,6# ,19, 4A, 4D, E1, 5CB
82# DATA 71, 23, 7# ,E1, 1# ,E4, 3E, C9, 32, F# ,F# ,F1, 87, 2# ,#3, 7DC
83# DATA 32, 6# ,#1, 21, 3A, #1, CD, D4, BC, D# ,21, F# ,FF, 11, ## ,65#
84# DATA 8# ,87, ED, 52, 3# ,#6, 21, FE, ## ,C3, 13, #5, #1, 17, #1, 4BF
85# DATA 21, #3, #1, C3, D1, BC, 24, ## ,2# ,## ,35, ## ,4# ,## ,43, 37C
86# DATA ## ,46, ## ,49, ## ,17, #1, 1A, #1, 1D, #1, 2# ,#1, 23, #1, 125
87# DATA 26, #1, 29, #1, 2C, #1, 2F, #1, 32, #1, 7# ,#1, 8# ,#1, 8# ,266
88# DATA #1, 9A, #1, AC, #1, B4, #1, C2, #1, D3, #1, E# ,#1, E7, #1, 55E
89# DATA ED, #1, F# ,#1, F3, #1, FB, #1, #C, #2, #F, #2, 1# ,#2, 3A, 442
90# DATA #2, 5A, #2, 65, #2, 76, #2, AF, #2, CB, #2, 2E, #3, 3A, #3, 323
91# DATA 4# ,#3, 51, #3, 62, #3, 65, #3, 68, #3, 6B, #3, 6E, #3, 7A, 328
92# DATA #3, 8# ,#3, 96, #3, A8, #3, AB, #3, B1, #3, BA, #3, BD, #3, 4B4
93# DATA C1, #3, CB, #3, F5, #3, FE, #3, 1# ,#4, 6E, #4, 77, #4, 9A, 52E
94# DATA #4, C2, #4, CE, #4, #B, #5, 2E, ## ,86, #1, 95, #1, BA, #1, 382
95# DATA C7, #1, #B, #2, A5, #2, C7, #2, 5F, #3, 76, #3, AE, #3, FB, 4C6
96# DATA #3, 55, #4, 5A, #4, 71, #4, C9, #4, E5, #4, F6, #4, 11, #5, 3F5
97# DATA ## ,## ,54, 6F, 6F, 2# ,4C, 6F, 77, #D, #A, ## ,## ,## ,42, 2DD
98# DATA 61, 64, 2# ,43, 6F, 6D, 6D, 61, 6E, 64, #D, #A, 34, #1, C3, 483
99# DATA #4, #2, C3, ## ,#2, C3, 2B, #3, C3, EE, #3, C3, 67, #4, C3, 561
100# DATA 43, #3, C3, 75, #1, C3, A9, #1, C3, EA, #1, 53, 43, 52, 45, 5C7
101# DATA 45, 4E, 53, 57, 41, D# ,53, 43, 52, 45, 45, 4E, 43, 4F, 5# ,4F#
102# DATA D9, 42, 41, 4E, 4B, 4F, 5# ,45, CE, 42, 41, 4E, 4B, 57, 52, 56C
103# DATA 49, 54, C5, 42, 41, 4E, 4B, 52, 45, 41, C4, 42, 41, 4E, 4B, 536
104# DATA 46, 49, 4E, C4, 56, 49, 45, D7, 56, 44, D5, 56, 44, 55, 8# ,66A
105# DATA ## ,FE, #1, C2, 1# ,#5, DD, 7E, ## ,FE, #2, D2, 1# ,#5, B7, 5CF
106# DATA 2# ,#F, CD, D# ,#1, E5, F5, 2A, FB, ## ,CD, #5, BC, 3E, C# ,76B
107# DATA 1# ,#D, CD, D# ,#1, E5, F5, 2A, F9, ## ,CD, #5, BC, 3E, 4# ,6D4
108# DATA CD, #B, BC, F1, E1, C3, 55, BD, FE, #1, C2, 1# ,#5, DD, 7E, 869
109# DATA ## ,FE, #2, D2, 1# ,#5, B7, 2# ,#D, CD, D# ,#1, 3E, C# ,CD, 644

110# DATA #B, BC, 2A, FB, ## ,1# ,#F, CD, D# ,#1, AF, CD, 5B, BD, 3E, 68B
111# DATA 4# ,CD, #B, BC, 2A, F9, ## ,C3, #5, BC, CD, #B, BC, FE, C# ,7CA
112# DATA 2# ,#4, 22, FB, ## ,C9, FE, 4# ,C# ,22, F9, ## ,C9, FE, #1, 6EB
113# DATA C2, 1# ,#5, CD, 11, #3, D2, 1# ,#5, DD, 7E, ## ,FE, FF, CA, 6C1
114# DATA 1# ,#5, C3, 5B, BD, F5, AF, 1# ,#3, F5, 3E, FF, 32, FD, ## ,71#
115# DATA F1, CD, FC, #2, D2, 1# ,#5, FE, #3, 2# ,6F, FE, #2, C2, 1# ,7#D
116# DATA #5, #1, ## ,4# ,6# ,69, 11, ## ,C# ,3E, FF, DD, BE, ## ,2# ,4E#
117# DATA 1A, DD, BE, #2, 2# ,1# ,#1, ## ,## ,21, ## ,4# ,#9, C5, #1, 32B
118# DATA ## ,#1, CD, A4, #2, C1, #4, CB, 7# ,2# ,EF, C# ,DD, 7E, #2, 6B1
119# DATA EB, 1# ,#3, DD, 7E, ## ,FE, FF, C# ,CD, 5B, BD, F5, 3A, FD, 937
120# DATA ## ,B7, 2# ,#5, F1, ED, ## ,1# ,23, F1, C5, #1, ## ,#1, D5, 632
121# DATA 11, F9, FF, E5, ED, B# ,D1, E1, E5, #1, ## ,#1, ED, B# ,EB, 9AC
122# DATA D1, E5, 21, F9, FF, #1, ## ,#1, ED, B# ,E1, C1, 1# ,DE, C3, 8C1
123# DATA 5B, BD, DD, 7E, #4, E6, 3F, 47, #E, ## ,21, ## ,C# ,#9, EB, 5C6
124# DATA 21, ## ,4# ,#9, #1, ## ,#1, 3E, FF, DD, BE, ## ,2# ,AA, D# ,4ED
125# DATA BE, #2, 2# ,A5, 11, F9, FF, DD, 7E, ## ,CD, 5B, BD, F5, 3A, 8#5
126# DATA FD, ## ,B7, 2# ,12, D5, E5, C5, ED, B# ,C1, D1, E1, DD, 7E, 9D#
127# DATA #2, CD, 5B, BD, ED, B# ,1# ,#7, CD, D1, #2, EB, CD, D1, #2, 7CE
128# DATA F1, C3, 5B, BD, D5, E5, #1, 8# ,## ,ED, B# ,E1, DD, 7E, #2, 8E2
129# DATA CD, 5B, BD, E5, #1, 8# ,## ,ED, B# ,D1, E1, E5, D5, #1, 8# ,8D5
130# DATA ## ,ED, B# ,D1, DD, 7E, ## ,CD, 5B, BD, #1, 8# ,## ,ED, B# ,7CC
131# DATA E1, C9, F5, DD, 7E, #2, B7, 2# ,#A, D6, #2, 3# ,#6, C6, #4, 6C5
132# DATA FE, #B, 3# ,1# ,DD, 77, #2, F1, F5, DD, 7E, ## ,B7, 2# ,#A, 6D1
133# DATA D6, #2, 3# ,#6, C6, #4, FE, #B, 3# ,#6, DD, 77, ## ,F1, 37, 5#B
134# DATA C9, F1, B7, C9, FE, #1, C2, 1# ,#5, DD, 7E, #1, B7, 2# ,F7, 83A
135# DATA DD, 7E, ## ,32, #7, #1, 21, ## ,## ,22, ## ,#1, C9, 21, FF, 3CA
136# DATA FF, FE, #2, 2# ,14, FE, #3, 2# ,1# ,FE, #4, C2, 1# ,#5, DD, 62A
137# DATA 6# ,#1, DD, 6E, ## ,3D, DD, 23, DD, 23, 22, #1, #1, CD, B7, 597
138# DATA #4, D2, 4E, #4, 22, FE, ## ,CD, F5, #4, 32, ## ,#1, EB, 2# ,554
139# DATA #2, ED, B# ,3A, ## ,#1, 4F, 2A, FE, ## ,EB, 7A, 41, E6, C# ,69D
140# DATA #7, #7, C6, #4, 4F, CD, 5B, BD, B7, C2, 41, #4, CB, F2, CB, 752
141# DATA BA, 7# ,B7, 2# ,#D, 21, F9, FF, EB, 1A, B7, 2# ,#3, BE, 2# ,6FC
142# DATA #D, 1# ,37, AF, CD, 5B, BD, ED, 5B, ## ,#1, C3, 5D, #4, 2A, 587
143# DATA FE, ## ,3A, #7, #1, 4F, #6, ## ,#9, 3# ,1A, 22, FE, ## ,2A, 33A
144# DATA #B, #1, ED, 5B, #1, #1, B7, ED, 52, 3# ,## ,19, 23, 22, #B, 3EA
145# DATA #1, AF, CD, 5B, BD, 1# ,A2, AF, CD, 5B, BD, 1# ,7# ,23, 13, 6A1
146# DATA 7D, B7, 2# ,BA, 7C, E6, 3F, 2# ,B5, F6, 4# ,67, 79, 3C, CD, 7A3
147# DATA 5B, BD, 1# ,AB, FE, #2, 2# ,#5, FE, #3, C2, 1# ,#5, CD, B7, 664
148# DATA #4, 3# ,52, E5, CD, F5, #4, EB, 2# ,#4, C5, ED, B# ,C1, D1, 83C
149# DATA 7A, 41, E6, C# ,#7, #7, C# ,#4, 4F, CD, 5B, BD, B7, 2# ,2A, 66E
150# DATA 21, F9, FF, C# ,F2, C# ,BA, 7# ,B7, 2# ,#4, 7E, 12, 1# ,#6, 75C
151# DATA AF, CD, 5B, BD, 1# ,27, 23, 13, 7# ,B7, 2# ,F# ,7A, E6, 3F, 6EA
152# DATA 2# ,EB, F6, 4# ,57, 79, 3C, CD, 5B, BD, 1# ,E1, CD, 5B, BD, 81#
153# DATA 11, FD, FF, 1# ,14, 11, FE, FF, 1# ,#F, 11, FF, FF, 1# ,#A, 69F
154# DATA ED, 5B, ## ,#1, 13, ED, 53, ## ,#1, 1# ,DD, 6E, #2, DD, 66, 55#
155# DATA #3, 73, 23, 72, C9, FE, #2, 2# ,#5, FE, #3, C2, 1# ,#5, CD, 5A6
156# DATA B7, #4, 3# ,D9, E5, CD, F5, #4, D1, 7A, 41, E6, C# ,#7, #7, 7AF
157# DATA C6, #4, 4F, CD, 5B, BD, B7, 2# ,B# ,CB, F2, C# ,BA, 7# ,B7, 8FE
158# DATA 2# ,#4, 1A, 77, 1# ,#D, AF, CD, 5B, BD, CD, F5, #4, 2# ,85, 611
159# DATA ED, B# ,1# ,B1, 23, 13, 7# ,B7, 2# ,E9, 7A, E6, 3F, 2# ,E4, 77A
160# DATA F# ,4# ,57, 79, 3C, CD, 5B, BD, 1# ,DA, FE, #2, 2# ,#D, DD, 72B
161# DATA 6E, ## ,DD, 66, #1, 22, ## ,#1, DD, 23, DD, 23, 3A, #7, #1, 41F
162# DATA 47, ED, 5B, ## ,#1, 21, ## ,## ,CB, 3# ,#3, 19, 3# ,## ,34#
163# DATA 2# ,## ,CB, 23, CB, 12, 3# ,F1, B7, C9, 3A, #7, #1, 5F, AF, 5EC
164# DATA 57, ED, 5A, 2# ,#2, 3# ,F2, B7, ED, 52, 37, C9, 3A, #7, #1, 62A
165# DATA 4F, 4F, #6, ## ,DD, 6E, ## ,DD, 66, #1, 7E, B# ,3# ,#1, 4F, 4EA
166# DATA 23, 5E, 23, 5# ,21, F9, FF, 79, B7, C9, 21, #A, #1, 7E, 23, 5D#
167# DATA CD, 5A, BB, FE, #A, 2# ,F7, ## ,## ,## ,## ,## ,## ,## ,## ,## ,##
168# DATA ## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,## ,##

GUIDE DU PROGRAMMEUR

Votre programme est maintenant fin prêt, il ne vous reste plus qu'à envoyer votre œuvre à CPC Infos.

Mais n'oubliez pas :



L E 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 _____

L E PROGRAMME

Nom : _____

Catégorie Jeu Utilitaire Educatif

Taille : _____

Périphériques utilisées : _____

Support Cassette Disquette

Comptabilité (testée) avec : _____

464 664 6128 PCW 8256

Signature _____

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

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

Participez à CPC Infos

MARK ERROR

Henri MONCHATRE

Valable pour CPC 664 et 6128

**Ne jetez plus vos disquettes défectueuses !!!
Grâce à MARK ERROR, vous allez pouvoir les
utiliser sans problème.**

Ce programme de formatage a en effet un plus par rapport à ses concurrents, il permet de "marquer" les secteurs illisibles d'une disquette.

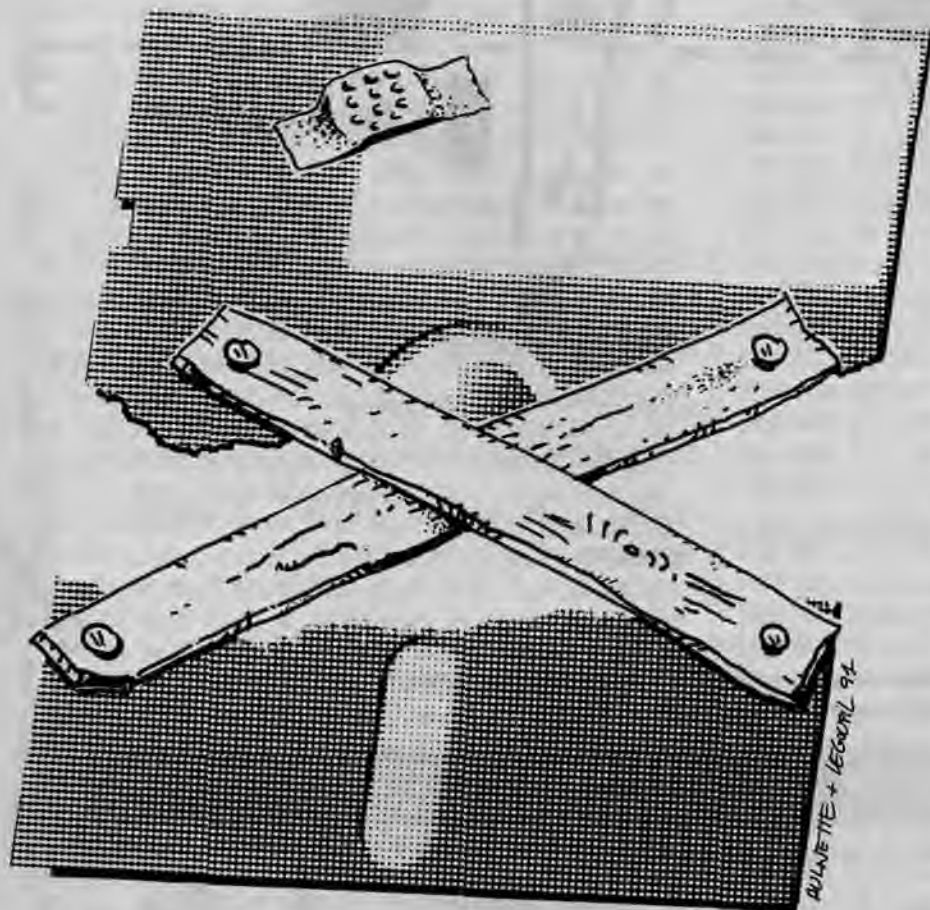
Comme d'habitude, il vous faudra taper les listings MARK.BAS et MARK.DAT, puis exécuter ce dernier qui va créer le fichier MARK.BIN.

LE FORMATEUR

MARK ERROR permet de formater vos disquettes selon les formats les plus employés sur notre bon vieux CPC, les formats DATA (40 pistes, 178 ko) et VENDOR (38 pistes, 169 ko).

Après le formatage proprement dit, la disquette va être testée, chaque secteur va faire l'objet d'une tentative de chargement. Si des secteurs ne peuvent être lus, le programme, après s'être assuré que les secteurs où le catalogue est logé ne sont pas défectueux, va procéder au marquage de la disquette. Il va tout simplement créer un fichier fantôme (qui ne peut être lu ou effacé), regroupant tous les secteurs illisibles. Lorsque l'ordinateur écrira sur la disquette, il croira que ces secteurs sont déjà occupés par un fichier, donc il ne s'en servira pas. Adieu READ FAIL !!!

Cependant, il existe quelques limites à l'emploi de cette technique. Si les secteurs où le catalogue est "écrit" sont défectueux, il ne sera pas possible de procéder au marquage. Essayez alors un autre formatage, les secteurs concernés, suivant que le format soit DATA ou VENDOR ne sont pas les mêmes. Autre chose, pas question de faire des copies physiques sur une dis-





quette "marquée". Il est seulement possible de copier des fichiers, ce qui n'est déjà pas si mal.

Petite chose en passant. Il est possible d'annuler un formatage, en appuyant sur ESC, quand le message vous demandant d'introduire une disquette dans le lecteur est affiché.

LE VERIFICATEUR

Cette option permet tout simplement de tester des disquettes formatées. Si le programme détecte un secteur non conforme, il le signale, et arrête la vérification. Attention !! Ce n'est pas parce que le vérificateur trouve un secteur non conforme, que votre disquette est forcément défectueuse. N'essayez donc pas de vérifier des jeux du commerce qui ont pratiquement tous des formatages non standards, ou des secteurs trafiqués.

LE COPIEUR

Il s'agit d'un copieur rapide uniquement réservé au lecteur A, et qui nécessite 128 ko. Son principal intérêt réside dans le fait qu'il copie toujours une disquette en deux passes, contrairement à DISCOLOGY, par exemple, lorsque la disquette est totalement remplie. Là encore, seul les formats DATA et VENDOR sont reconnus.

Si vous avez une disquette défectueuse avec de précieux fichiers, vous pouvez essayer de la "récupérer" grâce au copieur, en modifiant dans le fichier MARK.BAS la valeur de la variable tentative, comme suit : tentative=0.

Le résultat n'est pas garanti, mais cela peut permettre de les restaurer. Ne vous affolez pas si le lecteur "broute", c'est normal.

LES OPTIONS

Ce sous menu vous permet de sélectionner le lecteur par défaut (les fonctions formatage et vérification tournent en effet sur un second lecteur). La fonction vérification/marquage du formateur peut-être aussi invalidée, ce qui permet de gagner du temps, si vous êtes absolument sûr de l'état de vos disquettes.

Dernière petite chose. Les sélections dans les menus se font grâce aux flèches HAUT et BAS, et à la barre ESPACE.

Ce programme a été écrit sur un CPC 6128, mais, comme d'habitude, il devrait "tourner" avec les CPC 464 et 664.

MARK .BAS

```

1 * MARK ERROR * CHARGEUR BASIC Version 1.1      >FA
2 '                                                 >FB
3 * H.MONCHATRE * NOV.199#                       >FC
4 '                                                 >FD
10 MODE #:OPENOUT "d":MEMORY &1FFF:CLOSEOUT     >PQ
20 MODE #:FOR I=# TO 10:READ a:INK I,a:NEXT      >KR
30 DATA #,26,11,15,6,24,2#,8,10,9,15          >AC
40 BORDER #:WINDOW 1,2#,9,21                   >WL
50 tentative=1                                   >LF
60 FOR I=&BE8# TO &BE88:READ a#:POKE I,VAL("&"#a#):NEXT >YL

70 DATA 3e,#1,df,86,be,c9,#3,c6,#7             >AZ
80 POKE &BE81,tentative:CALL &BE8#              >CD
90 FOR I=&BE8# TO &BE88:READ a#:POKE I,VAL("&"#a#):NEXT >YP

100 DATA 3e,ff,df,86,be,c9,72,ca,#7            >CD
110 CALL &BE8#:LOAD "mark.bin",&2#00:CALL &2#00  >RL
120 FOR I=&BE8# TO &BE89:READ a#:POKE I,VAL("&"#a#):NEX >DV
T:INK I,#:I:B:CALL &BE8#
130 DATA DF,87,BE,32,8#,BE,C9,DD,CD,#7        >EH
140 CLS:INK 1,26:IF PEEK(&BE8#)<>1 THEN POKE &9F81,&A# >PD
POKE &9F82,&94 ' drive B non detecte
150 CALL &809#                                    >KC
160 POKE &99C#,&D3:POKE &9939,&A#:POKE &9A1C,&9#:POKE & >HD
9A1D,&BE
170 FOR I=&BE9# TO &BE9#+7:READ a#:POKE I,VAL("&"#a#):N >BP
EXT
180 DATA 3e,fe,32,fd,97,c3,6c,bb              >ZT
190 CALL &9A#                                       >LA

```



MARK .DAT



```
1 * GENERATEUR DU FICHIER MARK.BIN * H.WONCHATRE © 1990
2 *
3 CALL &BB4E:MODE 2:BORDER @:INK @,@:INK 1,26:PRINT "Patience
...
4 OPENOUT"d":MEMORY &8FFF:CLOSEOUT
5 ligne=1@@:adr=&9@@@
6 FOR l=1 TO 332:b=ligne:FOR j=@ TO 15:READ a$:a=VAL("&"a$):
POKE adr,a:adr=adr+1
7 b=b+a*(j+1):NEXT:READ a$:IF VAL("&"a$)<>b THEN PRINT "erreur
r ligne ";ligne:END
8 ligne=ligne+1@:NEXT
9 PRINT:PRINT "Appuyez sur une touche pour sauver le fichier
MARK.BIN.":CALL &BB18:SAVE "mark.bin",b,&9@@@,&14BF:PRINT:END
1@@ DATA 11,@@,6@,21,2C,2@,7E,23,32,14,2@,7A,FE,A@,CA,2B,36@@
11@@ DATA 2@,7E,47,FE,A@,CA,1F,2@,7B,12,13,23,C3,@B,2@,23,25FE
12@@ DATA 7E,23,46,23,12,13,1@,FC,C3,@B,2@,C9,29,29,@@,FF,32@@E
13@@ DATA 29,@@,ff,29,@@,FF,29,@@,FF,29,@@,FF,29,@@,FF,29,35@@
14@@ DATA @@,FF,29,@@,FF,29,@@,FF,29,@@,FF,29,@@,FF,29,@@,2F9@
15@@ DATA FF,29,@@,FF,29,@@,FF,29,@@,FF,29,@@,FF,29,@@,FF,39C@
16@@ DATA 29,@@,FF,29,@@,FF,29,@@,FF,29,@@,FF,29,@@,FF,29,359@
17@@ DATA @@,FF,29,@@,FF,29,@@,FF,29,@@,FF,29,@@,FF,29,@@,2FB@
18@@ DATA FF,29,@@,FF,29,@@,FF,29,@@,FF,29,@@,FF,29,@@,FF,39E@
19@@ DATA 29,@@,FF,29,@@,FF,29,@@,FF,29,@@,FF,29,@@,FF,29,35B@
2@@@ DATA @@,FF,29,@@,FF,29,@@,FF,29,@@,@@,@@,11,@@,@@,21,BC,338@
21@@ DATA @9,7E,23,32,A4,@9,7A,FE,@@,CA,@B,@9,7E,47,FE,@1,44C@
22@@ DATA CA,AF,@9,7B,12,13,23,C3,@B,@9,23,7E,23,46,23,12,295@
23@@ DATA 13,1@,FC,C3,@B,@9,C9,@1,@1,@@,5@,14,1C,@@,@@,14,1A4@
24@@ DATA 1C,@1,@@,1E,14,@1,@C,@6,1C,@1,@@,1D,@8,@1,@@,@@,@@B@
25@@ DATA 2C,@C,@4,@C,2C,@C,@4,@C,2B,@1,@@,1B,2C,@C,@1,3C,@C8@
26@@ DATA @3,2B,@1,@@,25,2C,@C,@8,@@,@4,@C,@@,@@,@4,@C,2B,@78@
27@@ DATA 14,@C,@@,@@,@4,2B,14,@C,@@,@@,@4,2B,14,@C,@@,@@,@7A@
28@@ DATA 2B,@1,@@,@@,2C,@C,@1,@@,@@,7,2C,@8,@@,@4,1C,@@,14,@7D@
29@@ DATA @C,@@,@@,@C,2B,@@,2C,@8,@@,@@,@4,1C,@@,14,@C,@@,@@,@6A@
3@@@ DATA @C,2B,@1,@@,@@,14,@C,@8,@1,@@,@@,@@,@@,@@,@@,@@,@@B@
31@@ DATA @@,@4,1C,@@,2C,@8,@@,@@,@4,1C,@@,2C,@8,@@,@@,@@,@@,@@9@
32@@ DATA @@,@4,14,@1,@C,@7,2B,14,@C,@@,@@,@4,1C,@@,2C,@8,@@C@
33@@ DATA @@,@@,@C,2B,@@,2C,@1,@C,@3,2B,@@,2C,@8,@@,@@,@@,@@,@@B@
34@@ DATA 2B,@1,@@,@@,3B,4@,@1,@@,19,@1,3@,5@,@1,@@,@@,FF,@1,@@,17D@
35@@ DATA FF,@1,@@,FF,@1,@@,FF,@1,@@,@@,@4,@1,CC,5@,@1,@@,@@,22C@
36@@ DATA 4@,@1,@@,13,@@,@1,@@,27,@@,@1,@@,@@,@@,@@,@1,@@,14,125@
37@@ DATA @C,@4,@@,@@,@@,@@,@@,@@,@4,@@,@@,@@,@@,@@,@@,@@,@@71@
38@@ DATA @@,@C,@4,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@3@
39@@ DATA @@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@D@
4@@@ DATA 14,1C,@@,@@,14,1C,@1,@@,1E,14,@C,@@,@1,@@,2B,2C,@ACE@
41@@ DATA @C,@4,@C,2C,@C,@4,@C,2B,@1,@@,1B,2C,@1,@C,@4,2B,@B1@
42@@ DATA @1,@@,24,14,@C,@8,@1,@@,@@,@@,@@,@@,@@,@4,@C,1C,@@,@@5@
43@@ DATA 2C,@8,@@,@@,@4,1C,@@,2C,@8,@@,@@,@@,2C,@C,1C,@1,@@C@
44@@ DATA @@,@@,14,@C,@8,@1,@@,@@,@@,@@,14,@C,@@,@@,@@,2B,@@A@
45@@ DATA @@,2C,@8,@@,@4,1C,@@,14,@C,@@,@@,@@,2B,@@,2C,@8,@@9@
46@@ DATA @@,@4,1C,@1,@@,@@,14,@C,@8,@1,@@,@@,@@,@@,1C,@@,2C,@@F@
```

```
47@@ DATA @@,@@,@4,1C,@@,2C,@8,@@,@@,@4,1C,@@,2C,@8,@@,@@,@@,2B,@@A@
48@@ DATA @1,@@,@4,14,@1,@C,@7,2B,14,@C,@@,@@,@4,1C,@@,2C,@@A@
49@@ DATA @@,@@,@@,@C,2B,@@,14,@C,@C,1C,@@,@@,2C,@8,@@,@@,@@,@@@
5@@@ DATA @C,2B,@1,@@,17,4@,@1,@@,3D,@1,3@,5@,@1,C@,5@,@1,1B9@
51@@ DATA @@,FF,@1,@@,FF,@1,@@,FF,@1,@@,FF,@1,@@,14,@1,C@,292@
52@@ DATA 5@,@1,@@,FF,@1,@@,31,@@,@1,@@,47,2C,@C,2B,@@,2C,166@
53@@ DATA @C,2B,@1,@@,1D,2C,@C,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,123@
54@@ DATA @@,18,2C,@C,@4,@C,2C,@C,@4,@C,2B,@@,@@,@@,@@,@@,@@,@@@
55@@ DATA @@,@@,@1,3C,@4,2B,@@,3C,2B,@@,14,3C,@1,@@,@@,@@,2C,@@E@
56@@ DATA @1,@C,@4,2B,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@F@
57@@ DATA @3,@1,@@,@@,14,3C,3C,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
58@@ DATA @A,14,@C,@8,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@E@
59@@ DATA @@,@4,1C,@@,2C,@C,@C,1C,@@,@@,@@,2C,@C,@C,2B,@1,@@,@@A@
6@@@ DATA @4,@8,@@,@@,14,@C,@8,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
61@@ DATA @@,2C,@1,@C,@3,2B,@@,14,@C,@@,@@,@@,2B,@@,2C,@1,@@A@
62@@ DATA @C,@3,2B,@1,@@,@@,7D,@@,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
63@@ DATA @1,@@,FF,@1,@@,FF,@1,@@,FF,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@@
64@@ DATA 3@,5@,@1,@@,5F,@@,@1,@@,75,4@,@1,@@,52,2C,@C,2B,18A@
65@@ DATA @@,2C,@C,2B,@1,@@,1D,2C,@C,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@@
66@@ DATA 14,@C,@8,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
67@@ DATA @@,2C,@1,@C,@3,1C,@@,2C,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
68@@ DATA @1,@C,@4,1C,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
69@@ DATA @C,1C,2B,@@,@@,@@,2C,@C,@C,2B,@@,@@,@@,2C,@C,@C,1C,2B,@@@
7@@@ DATA @1,@@,@@,14,@C,@8,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
71@@ DATA 2C,@1,@C,@3,1C,@@,2C,@1,@C,@3,2B,@@,2C,@C,@C,1C,@@C@
72@@ DATA @1,@@,@@,14,@C,@8,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
73@@ DATA 2C,@1,@C,@3,2B,@@,14,@C,@@,@@,@@,2B,@@,2C,@1,@C,@@@
74@@ DATA @3,2B,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
75@@ DATA @@,FF,@1,@@,FF,@1,@@,FF,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@@
76@@ DATA 12,C@,4@,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
77@@ DATA C@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
78@@ DATA @1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
79@@ DATA 2B,@1,@@,@@,2B,@1,@@,1F,@1,3C,@7,2B,@1,@@,@@,@@,@@,@@@
8@@@ DATA @1,@@,1D,2C,@C,2B,@@,2C,@C,2B,@1,@@,@@,1D,2C,@C,@1,@@@
81@@ DATA @@,2B,14,@C,@8,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
82@@ DATA @3,1C,@@,2C,@1,@C,@3,1C,@@,2C,@8,@@,@@,@@,@@,@@,@@,@@@
83@@ DATA @@,14,@C,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
84@@ DATA @C,@3,2B,@@,14,@1,@C,@3,1C,@@,@@,@@,@@,@@,@@,@@,@@,@@@
85@@ DATA @1,@@,@@,2C,@C,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
86@@ DATA 14,@1,@C,@3,1C,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
87@@ DATA @1,@@,@@,2C,@1,@C,@3,2B,@@,14,@1,@C,@3,1C,@@,@@,@@,@@@
88@@ DATA 2C,@@,@@,@@,1C,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
89@@ DATA 5@,@1,@@,FF,@1,@@,FF,@1,@@,FF,@1,@@,@@,@@,@@,@@,@@,@@@
9@@@ DATA @1,F@,5@,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
91@@ DATA @@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
92@@ DATA C@,4@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
93@@ DATA 4@,@1,@@,@@,2B,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
94@@ DATA 2C,@1,@C,@6,2B,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
95@@ DATA 14,@C,@C,1C,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
96@@ DATA 2B,14,@C,@8,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
97@@ DATA 1C,@@,2C,@8,@@,@@,@@,@@,2C,@8,@@,@@,@@,@@,@@,@@,@@,@@@
98@@ DATA 14,@C,@8,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
99@@ DATA @C,2B,@@,14,@C,@8,@C,1C,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
1@@@ DATA @@,@@,2C,@C,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
1@@@ DATA 3C,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
1@@@ DATA 2C,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
1@@@ DATA @1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
1@@@ DATA @1,@@,FF,@1,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@,@@@
```


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	26 F (unité)		Franco Franco	
DISQUETTES CPC Infos - MEGA SOUND - N° 31 (AmStar 48 & CPC Infos 24) - N° 32 (CPC Infos 25 & 26) - N° 33 (CPC Infos 27 & 28) - N° 34 (CPC Infos 29 & 30)	Abonné : 110 F (unité) Non abonné : 140 F (unité)		Franco 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 Franco	
ANCIENS Nos HORS-SERIE CPC N° 4 - 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				
<input type="checkbox"/> 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 31

PAYEZ PAR CARTE BANCAIRE

Date d'expiration _____ Signature _____

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

Date et signature _____ Signature _____

Nom : _____ Prénom : _____

Adresse : _____

Code Postal : _____ Ville : _____

ECRIRE EN MAJUSCULES

Afin de faciliter le traitement des commandes, nous remercions notre aimable clientèle de ne pas agraffer 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 précision 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 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.

chronique du TELECHARGEMENT



Rendez-vous avec de nouveaux logiciels sur le 36-15 ARCADES. Pour ceux qui ne connaissent pas encore ce serveur, il est conseillé de se rendre au numéro précédent, qui explique beaucoup de choses et surtout de lire le mode d'emploi sur le minitel.

Ceci dit, voici la liste des programmes que vous pourrez trouver fin avril sur le serveur :

AZERTY : c'est un tout petit programme qui permet de simuler le clavier AZERTY sur un clavier QWERTY par redéfinition des touches.

FORMLIB : deux sous-programmes permettant la sortie et l'entrée de données.

Pour la sortie des données, il faut mettre un attribut (voir plus loin) dans la variable `attr%`, la chaîne de caractères à imprimer dans `a$` et ensuite faire `GOSUB 10000`.

Les attributs suivants peuvent être utilisés :

- 0 - normal
- 1 - gras
- 2 - souligné
- 4 - inverse
- 8 - à l'intérieur d'une boîte

Par exemple, un texte en gras et souligné, s'obtiendra par `attr%=2+1`.

Pour l'entrée de données, il faut définir les variables suivantes :

`maxlen%` = longueur maximale de l'entrée en caractères.
`type%` = type d'entrée, 0 = chaîne de caractères, 1 = nombre entier, 2 = nombre décimal.

`default$` = valeur par défaut de l'entrée.

`box%` = 1 si l'entrée doit être mise dans une boîte sinon = 0
`help$` = message d'aide qui apparaîtra en ligne 25 si on appuie sur la touche COPY.

Pour lancer la routine, `GOSUB 10200`.

Ces routines peuvent être chargées en utilisant `LOAD «FORMLIB»`, ou bien encore ajoutées à vos programmes par `MERGE «FORMLIB»`. Essayez le fichier `FORMDEM` pour une démonstration.

GRAPHLIB : une bibliothèque graphique produisant des histogrammes ou des «camemberts».

Pour les histogrammes, il faut définir les variables suivantes :
`basex%` = coordonnée gauche de la première barre.

`basey%` = coordonnée du bas de la première barre sur l'écran.

`wid%` = largeur de la barre en pixels.

`units%` = nombre de pixels par unité

`bar%` = nombre de barres.

`height%` = hauteur de la barre en unités.

`shade%` = lignes d'ombrage (0 = pas d'ombrage)

`GOSUB 30000` pour lancer la routine.

Pour dessiner des «camemberts» :

`basex%` et `basey%` = coordonnées du centre.

`radius%` = rayon du cercle.

`sang%` = angle de départ du segment (en degrés).

`fang%` = angle de fin du segment (en degrés).

`GOSUB 30200`.

La méthode d'installation est la même que pour `FORMLIB` et le programme de démonstration se lance par `RUN «GRAPHDEM»`.

HACK : un ensemble de `RSX` qui permet de fouiller dans un programme binaire placé en mémoire.

`ùCHEATFIND, début, longueur`.

Cherche dans un programme (chargé en mémoire) les instructions `INC A`, `DEC A`, `INC (HL)` ou `DEC (HL)` qui sont en général utilisées pour décrémenter les vies ou l'énergie dans un jeu. Il suffit souvent de `POKE` un 0 à l'adresse trouvée pour obtenir des vies infinies.

`ùBREAKP.ON, adresse`.

Place un point d'arrêt à l'adresse donnée. Le programme stoppe à cette adresse et retourne au basic.

`ùBREAKP.OFF, adresse`.

Annule la commande précédente.

`ùDEEK, adresse, variable entière`.

Prend une valeur sur 16 bits (adresse et adresse+1) et la place dans la variable entière.

`ùDOKE, adresse, valeur`.

Poke la valeur dans adresse et adresse+1.

`ùDISCREAD, piste, secteur, variable entière`.

Charge le secteur spécifié et envoie l'adresse du premier octet dans la variable entière.

`ùDISCWRITE, piste, secteur`.

Ecrit le secteur en mémoire sur la piste et sur le secteur donné.

`ùFILEVIEW, chaîne`.

Affiche le contenu ASCII du fichier «chaîne» sur l'écran.

`ùHELP` Affiche les commandes disponibles.

`ùMCOPY, source, destination, longueur`.

Copie une portion de mémoire de taille donnée de l'adresse source vers l'adresse destination.

`ùMDUMP, départ, longueur`.

Affiche le contenu de la mémoire suivant les paramètres donnés.

`ùPRINTER.ON`.

Dirige les sorties vers l'imprimante.

`ùPRINTER.OFF`.

Annule l'instruction précédente.

↳RAMVIEW.

Affiche une représentation graphique de la mémoire.

↳SCREENVIEW,départ,mode.

Affiche un bloc de données comme une page écran, dans le mode précisé.

↳SCRINIT

Initialise les paramètres par défaut de l'écran.

↳TEXTSEARCH,départ,longueur.

Cherche, dans la région spécifiée de la mémoire, les occurrences de 3 (ou plus) caractères alphabétiques. Cela permet de découvrir des messages cachés par exemple.

Le listing source est disponible sous la forme d'un fichier ASCII intitulé : HACKBAS.ASM.

PILOT : un petit interpréteur de langage PILOT (doc en anglais, faire RUN «PILOTINF»).

PONTOON : un jeu de cartes graphique dans lequel il ne faut pas dépasser une certaine valeur en additionnant la valeur de ses cartes.

THINCHAR : une police de caractères très fins.

MENUMAKE : un exemple de ce qu'il est possible de réaliser avec FORMLIB. Il s'agit ici d'un créateur de menus pour vos disquettes.

Maintenant passons aux démos qui ne sont pas nombreuses mais de qualité :

UGLYDEMO : elle devait passer le mois dernier mais pour

cause de problèmes techniques elle est reportée à un peu plus tard.

OMICRON DEMO : première démo d'Omicron, pas mal mais un peu courte peut-être.

AMIGA DEMO 4 : Assez impressionnante avec un gros scrolling et des rasters à l'intérieur.

C'est au tour de la disquette 34 regroupant les numéros 28 et 29 de CPC Infos :

GRAFFITI ET SUPertext : deux utilitaires qui permettent la création de fontes et le scrolling de messages.

JAQUETTE : imprime les catalogues de vos disquettes sur papier. C'est un programme très simple d'emploi mais qui ne fonctionne que sur 664 et 6128.

THE SPEAKER : votre ordinateur parle ! Sans interface mais seulement en anglais, sorry.

POUSNIK : Un jeu qui ressemble à qui-vous-savez et réalisé par qui-vous-savez.

CIRCUS FOLIES : Un autre Le Moullec prenant pour base le jeu Spherical.

Il reste bien sûr Axyx mais ce programme ne sera mis sur le téléchargement qu'après la publication de toutes les parties dans CPC Infos.

A bientôt sur le 36-15 ARCADES !

N'oubliez pas !!!

36 • 15 ARCADES

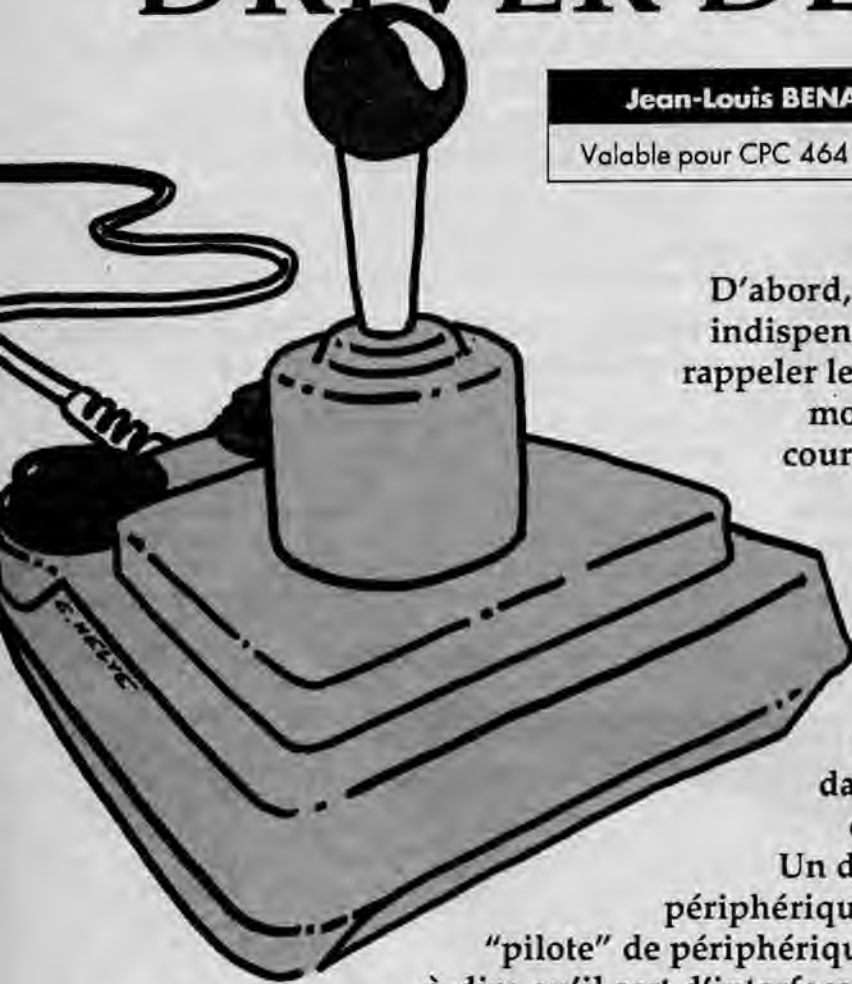
Des centaines de logiciels pour CPC à télécharger avec le kit ARCADES.

Voir bon de commande page 23

DRIVER DE SOURIS

Jean-Louis BENARD

Valable pour CPC 464 - 6128



D'abord, il paraît indispensable de rappeler le sens du mot driver, couramment utilisé dans le monde du PC, et très peu dans celui du CPC.

Un driver de périphérique est un "pilote" de périphérique, c'est-à-dire qu'il sert d'interface entre le hardware que constitue le périphérique et le software : votre programme.

Ici, on supposera que l'on dispose simplement d'un joystick et non d'une souris ; le driver, appelé par le biais d'une interruption assembleur, va scruter le joystick, modifier le pointeur à l'écran en conséquence et renvoyer en mémoire des informations susceptibles d'être utilisées par le programme qui utilise le driver.

L'interruption choisie est celle liée au contrôleur vidéo, à savoir qu'elle est appelée tous les 1/50 de seconde ; on aurait pu utiliser un appel tous les

1/300, mais ce qui a été choisi ici est amplement suffisant.

Ainsi donc que se passe-t-il tous les 1/50 de seconde ?

Première étape : la routine fait appel à `Joy(0)` ; si aucune action n'a été exécutée au niveau du joystick, le driver revient au programme (basic ou assembleur) qui l'utilise.

Dans le cas contraire, le driver modifie la position du pointeur à l'écran, en restituant le caractère masqué par le pointeur et en stockant en mémoire celui sur lequel le pointeur va se placer.

Un tel driver serait inutile s'il ne renvoyait pas des informations au programme qui l'utilise ; voici donc la table d'informations (modifiable par le programme) ; toutes les données liées au driver seront situées à partir de l'adresse 39000 :

- En 39000 est placée la coordonnée Y du pointeur affiché.
- En 39001, la coordonnée X.

- En 39002 est stocké le caractère effacé provisoirement par le pointeur.

- Le contenu de 39003 permet de régler la vitesse du pointeur ; vous constaterez en effet, si vous utilisez le driver dans un programme en assembleur, que le pointeur se déplace trop rapidement ; pour le ralentir, il suffit d'augmenter le contenu de 39003.

- 39004 : réservé au driver.

- 39005 et 39006 contiennent les lignes minimum et maximum entre lesquelles le pointeur de souris peut évoluer ; elles sont bien entendu modifiables par le programme qui utilise le driver.

- 39007 et 39008 contiennent les colonnes minimum et maximum entre lesquelles la souris évolue.

- 39009 : contient initialement 0 ; mais si l'utilisateur appuie sur le bouton fire 1, contient 255 jusqu'à modification par le programme qui utilise le driver.

Concrètement, comment utiliser le driver ? Vous devez d'abord charger le programme `MOUSE.BAS`, puis l'exécuter. Celui-ci va stocker sur disquette un fichier `MOUSE.BIN` ; c'est ce fichier qui va être utilisé dans vos programmes.

Ainsi, lorsque vous voulez utiliser le driver dans votre programme, vous devez inclure les séquences suivantes :

- D'abord mettre un `MEMORY`, par exemple `MEMORY 16383`, puis mettre un `LOAD*MOUSE.BIN` ; le fichier binaire va se charger à partir de 38600.

- A partir de là, deux possibilités : soit le programme lance immédiatement le driver par un call &96c8 (38600) ; dans ce cas, c'est la routine elle-même qui initialise la table de données : 39000 et 39001 contiennent les coordonnées du curseur : le pointeur démarre du curseur ; 39002 contient le caractère qui se trouvait au niveau du curseur ; 39003 contient 2 (vitesse moyenne, il faudra souvent la mettre à 1) ; 39005, 39006, 39007, 39008 contiennent les limites de l'écran, 39009 contient 0 : soit vous faites un call &96f0 APRES AVOIR VOUS-MEME initialisé la table de valeurs ; si vous ne l'avez pas initialisé, c'est le plantage assuré. Ce second call vous permet donc de faire démarrer le pointeur d'où vous voulez, puisque, il faut le rappeler, pointeur et curseur sont complètement indépendants.

- Une fois l'un des deux calls exécuté, le pointeur apparaît à l'écran, peut être manipulé par le joystick sans que le programme qui utilise le driver ait à s'en occuper.

- Pour inhiber le driver, le programme devra faire un call &9817 : le pointeur disparaît, le caractère masqué est restitué.

Quelques mises en garde :

- Le pointeur de souris est un pointeur de souris texte : en conséquence, s'il passe sur du graphique, il ne reconnaîtra pas de caractère, d'où mauvaise restitution.

- Supposons que vous affichez à l'écran une information alors que le driver est actif : l'information pourrait

recouvrir le pointeur, sans que le driver le sache : il y aurait alors une mauvaise restitution de caractère : c'est pourquoi, de même que lorsque vous effacez l'écran, il est conseillé, avant de faire un affichage, d'inhiber le driver par un call &9817, puis, après l'affichage, de refaire un call &96f0.

- Il peut se produire des problèmes de saturation : si vous manipulez le joystick pendant que le programme fait des affichages répétés, l'ordinateur doit, en plus de ces affichages, tous les 1/50 de seconde afficher le pointeur ; d'où un débordement avec des aberrations à l'écran.

Le listing assembleur est assez simple : à partir de 38600, on a d'abord les initialisations de la table (qu'on peut, je vous le rappelle, éviter). Puis le JP CRTIN initialise l'interruption : tous les 1/50 de seconde, on a un appel à MOUSE.

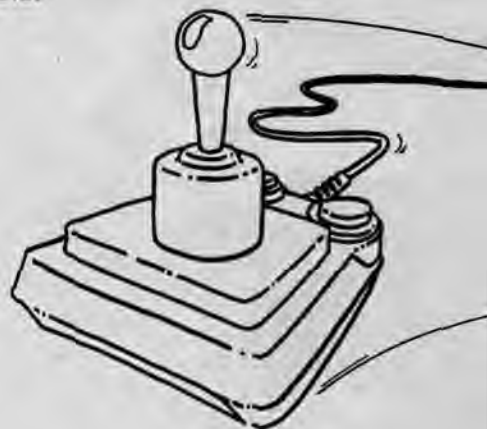
MOUSE, en fonction de la valeur contenue en 39003, et en 39004 (compteur) retourne au basic ou continue la routine assembleur. De là, on a un appel à JOY : le résultat est dans l'accumulateur, que l'on stocke dans sto ; on examine bit après bit la valeur de cet accumulateur.

Si le bit 0 est à 1, on exécute HAUT : stockage provisoire dans la pile des coordonnées du curseur ; on vérifie que le pointeur ne déborde pas de la limite. Puis on restitue le caractère effacé par le pointeur ; on modifie les coordonnées du pointeur, on stocke le caractère sur le point d'être effacé :

on affiche le pointeur à la nouvelle position ; on restitue les coordonnées du curseur. Puis on répète ces opérations avec le bit 1 (BAS) et ainsi de suite.

En &9817 débute la routine d'inhibition : restitution du caractère effacé, puis annulation de l'interruption.

Je vous propose un exemple pour illustrer ceci : il s'agit d'un exemple de fenêtre de saisie : la fenêtre avec les options oui et non est d'abord dessinée ; puis le programme initialise lui-même la table de données de telle sorte que le pointeur ne sorte pas de la fenêtre ; puis appel du driver avec un call &96f0 ; l'utilisateur vient se placer sur l'une des 2 options OUI ou NON avec le joystick, appuie sur fire : 39009 vaut alors 255 : le programme examine les coordonnées du pointeur : s'il se trouve dans l'une des fenêtres OUI ou NON il affiche un message en conséquence, sinon il s'arrête.



EXEMPLE.

B A S

```

1# REM EXEMPLE >CJ
2# MEMORY 16383:LOAD "MOUSE.BIN" >AE
3# MODE 2 >BG
4# LOCATE 19,5:PRINT "*****" >GD
*****
5# FOR i=6 TO 19:LOCATE 19,i >VM
6# PRINT "i" >UP
i"
7# NEXT >YH
8# LOCATE 19,29:PRINT "*****" >HV
*****
9# LOCATE 29,7:PRINT "EX. DE FENETRE DE SAISIE" >TU
10# LOCATE 22,19:PRINT "Choisissez avec le joystick OUI >YQ
ou NON"
11# LOCATE 23,12:PRINT "En vous positionnant puis avec >WJ
FIRE!"
12# LOCATE 28,16:PRINT " * OUI * * NON *" >PR
13# LOCATE 28,14:PRINT "*****" >NN
14# LOCATE 28,18:PRINT "*****" >NU
15# LOCATE 28,15:PRINT " * * * " >LQ
16# LOCATE 28,17:PRINT " * * * " >LU
17# REM UTILISATION MOUSE:ON REDEFINIT LA TABLE >TC
18# POKE 39##13,13:POKE 39##1,4# :POKE 39##2,8:POKE 39##3 >QH
,1:POKE 39##4,8
19# POKE 39##5,5:POKE 39##6,2# :POKE 39##7,19:POKE 39##8 >BA
,61
20# POKE 39##9,8 >AF
21# CALL &96F0 >LA
22# WHILE PEEK(39##9)<>255 >UG
23# WEND >CD
24# L1=PEEK(39##0):C1=PEEK(39##1) >AN
25# IF L1>13 AND L1<19 AND C1>27 AND C1<35 THEN LOCATE >UH
29,22:PRINT "VOUS AVEZ CHOISI OUI"
26# IF L1>13 AND L1<19 AND C1>45 AND C1<53 THEN LOCATE >UG
29,22:PRINT "VOUS AVEZ CHOISI NON"

```

MOUSE.

B A S

```

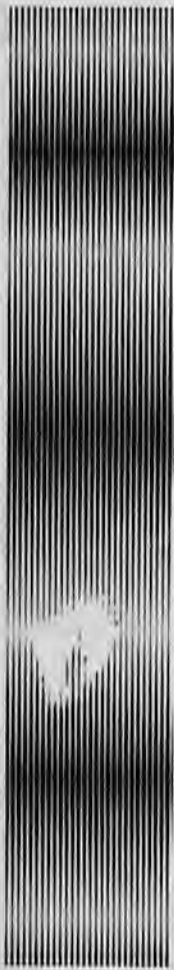
1# REM CHARGEUR MOUSE
2# FOR I=386# TO 389#
3# READ A#:A#="&"+A#:A=VAL(A#)
4# POKE I,A
5# NEXT
6# SAVE "MOUSE.BIN",B,386#,399
1#1# DATA CD,78,BB,22,58,98,3E,2,32,5B,98,3E,0,32,5C,
98,3E,1,32,5D
1#2# DATA 98,32,5F,98,CD,17,BC,78,32,60,98,79,32,5E,9
8,3E,0,32,61,98
1#3# DATA 2A,58,98,CD,75,BB,CD,60,BB,32,5A,98,3E,F0,C
D,5A,BB,21,36,98
1#4# DATA 6,81,E,0,11,E,97,C3,D7,BC,3A,5C,98,3C,32,5C
,98,47,3A,5B
1#5# DATA 9B,88,28,1,C9,3E,0,32,5C,98,CD,24,BB,FE,0,2
,0,1,C9,32,3E
1#6# DATA 98,E6,1,28,31,CD,78,BB,E5,2A,58,98,3A,5D,98
,BD,20,2,E1,C9
1#7# DATA CD,75,BB,3A,5A,98,CD,5A,BB,2A,58,98,2D,22,5
8,98,CD,75,BB,CD

```

```

>RT
>PR
>XB
>VA
>YF
>ZM
>ME
>PH
>TW
>LR
>LJ
>PH
>TA
>RH
>RA
>XA
>PV
1#8# DATA 60,BB,32,5A,98,3E,F0,CD,5A,BB,E1,CD,75,BB,3
A,3E,98,E6,2,28
1#9# DATA 31,CD,78,BB,E5,2A,58,98,3A,5E,98,BD,20,2,E1
,C9,CD,75,BB,3A
1#10# DATA 5A,98,CD,5A,BB,2A,58,98,2C,22,58,98,CD,75,B
B,CD,60,BB,32,5A
1#11# DATA 98,3E,F0,CD,5A,BB,E1,CD,75,BB,3A,3E,98,E6,4
,28,31,CD,78,BB
1#12# DATA E5,2A,58,98,3A,5F,98,BC,20,2,E1,C9,CD,75,BB
,3A,5A,98,CD,5A
1#13# DATA BB,2A,58,98,25,22,58,98,CD,75,BB,CD,60,BB,3
,2,5A,98,3E,F0,CD
1#14# DATA 5A,BB,E1,CD,75,BB,3A,3E,98,E6,8,28,31,CD,78
,BB,E5,2A,58,98
1#15# DATA 3A,60,98,BC,20,2,E1,C9,CD,75,BB,3A,5A,98,CD
,5A,BB,2A,58,98
1#16# DATA 24,22,58,98,CD,75,BB,CD,60,BB,32,5A,98,3E,F
0,CD,5A,BB,E1,CD
1#17# DATA 75,BB,3A,3E,98,E6,10,28,5,3E,FF,32,61,98,C9
,CD,78,BB,E5,2A
1#18# DATA 58,98,CD,75,BB,3A,5A,98,CD,5A,BB,3E,0,32,5A
,98,E1,CD,75,BB
1#19# DATA 21,36,98,C3,DD,BC,0,0,0,0,0,0,0,0,0,0,0,0
,0
1#20# DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

```



```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19 96C8 CD78BB
20 96CB 225898
21 96CE 3E02
22 96D0 325B98
23 96D3 3E00
24 96D5 325C98
25 96D8 3E01
26 96DA 325D98
27 96DD 325F98
28 96E0 CD17BC
29 96E3 78
30 96E4 326098
31 96E7 79
32 96EB 325E98
33 96EB 3E00
34 96ED 326198
35 96F0 2A5898
36 96F3 CD75BB
37 96F6 CD60BB
38 96F9 325A98
39 96FC 3E00

```

```

ORG 38600
LOAD 38600
*****
INTERRUPTIONS...
*****
CRTIN      : E  QU  OBCD7H
CRTOFF     : E  QU  OBCDDH
JOY        : E  QU  OBB24H
LOCATE     : E  QU  OBB75H
LOCATE?    : E  QU  OBB79H
PRINT      : E  QU  OBE5AH
READ       : E  QU  OBB60H
BLIMITS    : E  QU  OBC17H

```

; INITIALISATION

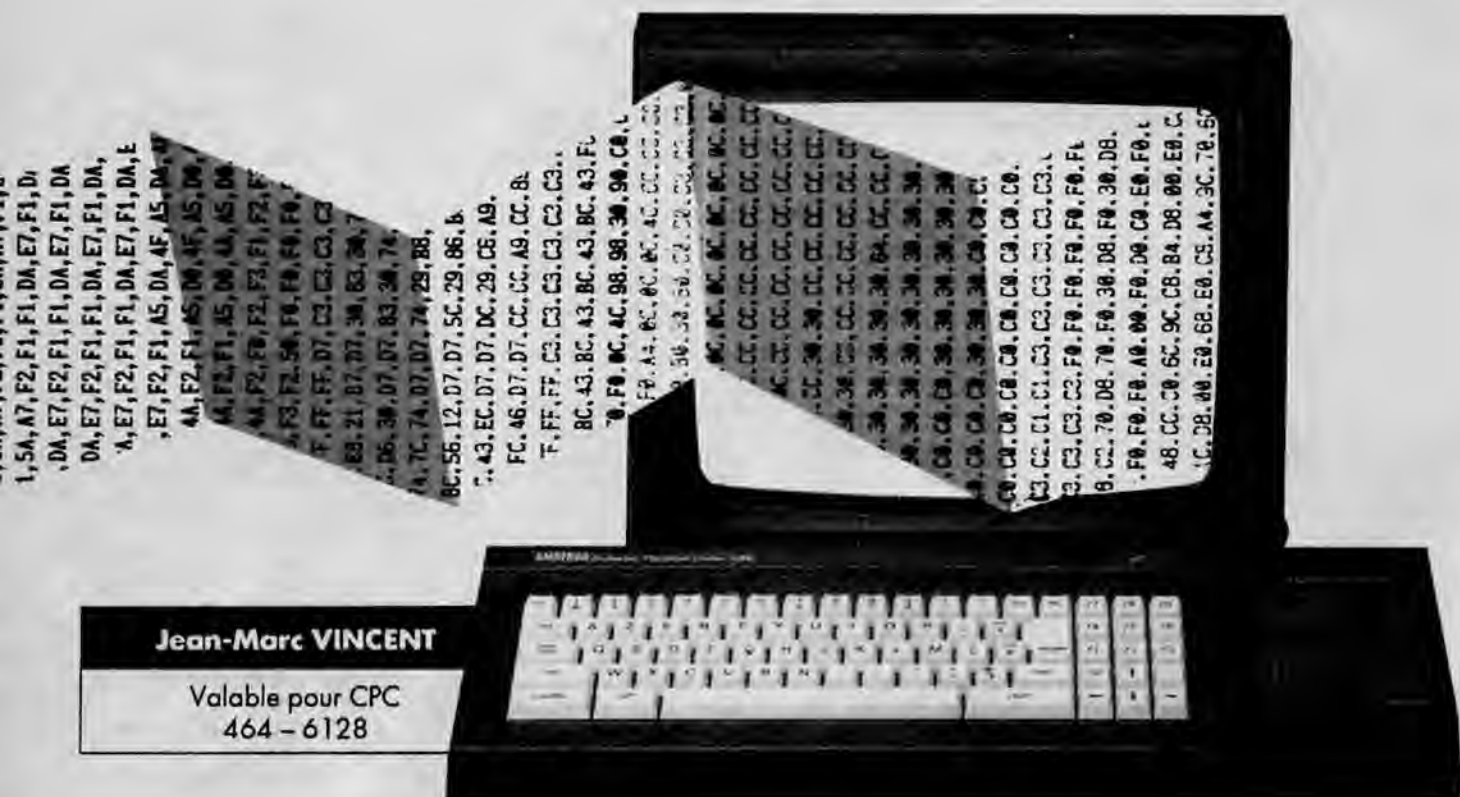
```

CALL LOCATE?
LD (39000),HL
LD A,2
LD (39003),A
LD A,0
LD (39004),A
LD A,1
LD (39005),A
LD (39007),A
CALL BLIMITS
LD A,B
LD (39008),A
LD A,C
LD (39006),A
LD A,0
LD (39009),A
LD HL,(39000)
CALL LOCATE
CALL READ
LD (39002),A
LD A,240

```



DATALOAD



Jean-Marc VINCENT

Valable pour CPC
464 - 6128

Voici un petit programme qui va faciliter la vie de beaucoup de lecteurs.

En effet, nombre de programmes publiés dans nos pages sont très intéressants mais leur utilisation nécessite des heures de travail ininterrompues devant votre CPC pour saisir les dizaines de pages de DATAs.

Ce programme vous permet maintenant de charger les lignes de DATAs dans un fichier ASCII et de s'arrêter à tout moment. Ainsi le fichier de DATAs original pourra être reconstitué en utilisant les commandes LOAD pour le premier fichier, MERGE pour les suivants et enfin SAVE pour le sauvegarder.

FONCTIONNEMENT DU PROGRAMME :

- Le programme demande sur quel support (cassette ou disquette) doit se trouver le fichier.
- S'il s'agit de la cassette, le programme demande s'il est besoin d'initialiser les commandes cassette (TAPE : dans le cas d'un 464 avec DDI-1 ou 6128 avec cassette).
- S'il s'agit de la disquette, le programme affiche la liste de tous les fichiers de type ASCII (*.asc).
- Le programme demande ensuite le nom du fichier en sortie.
- Ensuite on demande le numéro de ligne de départ et l'incrément pour les lignes suivantes.
- Puis le nombre de DATAs par ligne.
- Ensuite si le signe "&" est utilisé ou non.

- Le programme demande alors la longueur de la chaîne somme. Si RETURN pas de somme en fin de ligne sinon le programme propose le choix de la formule de calcul pour cette somme. (Il est d'ailleurs possible d'en ajouter facilement de nouvelles. c.f lignes 2350 à 2470).

- Et si cette somme doit être réinitialisée à chaque ligne.

- On peut alors entrer les DATAs. Il faut taper RETURN après chaque DATA. La touche DEL peut être utilisée en cas d'erreur.

- A la fin de la ligne, si besoin est, on entre la somme de contrôle (le programme l'ayant calculé lui-même tout au long de la ligne). Ici aussi on peut utiliser la touche DEL.

- Si il y a erreur sur la somme le programme propose de recommencer en effaçant la ligne ou de corriger en passant sur chaque DATA ainsi que sur la somme.

- Si la somme est correcte le programme propose de continuer ou de terminer.

```

1000 ' >XD
1010 ' ##### >XE
1020 ' # >XF
1030 ' # Chargement de DATAs # >XG
1040 ' # CPC 464/664/6128 # >XH
1050 ' # ←————→ # >XJ
1060 ' # | Version 1.0.0 | # >XK
1070 ' # ←————→ # >YA
1080 ' # >YB
1090 ' ##### >YC
1100 ' >XE
1110 ' (c) J.-M. VINCENT Fevrier 1998 >XF
1120 ' >XG
1130 ' Toutes les lignes commentees peuvent etre efface >XH
es : tous les points
1140 ' d'entree pour GOTO ou GOSUB ne sont pas des lign >XJ
es commentees.
1150 ' >XK
1160 ' ←————→ >YA
1170 ' | >YB
1180 ' | Liste des variables | >YC
1190 ' | ————— | >YD
1200 ' ←————→ >XF
1210 ' >XG
1220 ' Alphetiques >XH
1230 ' ===== >XJ
1240 ' >XK
1250 ' inval$ : caractere lu par INKEY$ >YA
1260 ' valin$ : chaine de caracteres composee >YB
a partir de inval$
1270 ' support$ : nom du support : cassette ou d >YC
isque
1280 ' file$ : nom du fichier en sortie >YD
1290 ' >YE
1300 ' Numeriques entieres >XG
1310 ' ===== >XH
1320 ' >XJ
1330 ' caZ : fenetre de sortie ecran >XK
1340 ' lenin$ : longueur d'une chaine a lire >YA
1350 ' nL : numero de ligne de depart >YB
1360 ' inc% : increment des numeros de ligne >YC

1370 ' ndati% : nombre de DATAs par ligne >YD
1380 ' lsum% : longueur de la chaine sua a li >YE
re
1390 ' posx% / posy% : position du curseur en debut d >YF
e chaque ligne
1400 ' posxd% / posyd% : position du curseur au debut d >XH
e chaque DATA par ligne
1410 ' iZ : utilise par les boucles FOR >XJ
1420 ' >XK
1430 ' Numeriques reelles >YA
1440 ' ===== >YB
1450 ' >YC
1460 ' valdat() : tableau de DATAs pour une lign >YD
e
1470 ' savsua : sauvegarde de la somme cumulee >YE

```

```

courante si erreur
1480 ' sua : somme calculee (et cumulee) po >YF
ur la ligne courante
1490 ' valsum : somme lue pour la ligne couran >YB
te
1500 ' ERR : numero d'erreur >XJ
1510 ' ERL : ligne en erreur >XK
1520 ' >YA
1530 ' Elags >YB
1540 ' ===== >YC
1550 ' >YD
1560 ' amp% : | si 'a' utilise sinon 0 >YE
1570 ' csua% : | si sua a zero en debut de ch >YF
aque ligne sinon 0
1580 ' correct% : | si correction de la ligne DA >YB
TA courante sinon 0
1590 ' inpsua% : | si entree de la somme de con >YH
trole sinon 0
1600 ' >XK
1610 ' Fonction >YA
1620 ' ===== >YB
1630 ' >YC
1640 ' FN sua : definit la formule de calcul p >YD
OUR SUM
1650 ' >YE
1660 ' Debut du programme >YF
1670 ' ===== >YB
1680 ' >YH

```

N'oubliez pas!!!

36 • 15

ARCADES

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

Voir bon de commande page 23

```

1690 KEY DEF 66,0,0,0,0:POKE &B0EE,&C9 ' Annule BREAK & >DU
CTRL+SHIFT+ESC
1700 MODE 2:ON BREAK GOSUB 3180:ON ERROR GOTO 3230 >TA
1710 ' >YB
1720 ' Initialisation des variables pour le chargement >YC
des DATAs
1730 ' ----- >YD
-----
1740 ' >YE
1750 WINDOW #1,1,80,1,24:CLS #1:WINDOW #2,1,80,25,25:CL >BL
S #2
1760 PRINT #1,CHR$(24);SPACE$(14);"Utilitaire de charge >XT
ment de datas dans un programme.";SPACE$(14);CHR$(24);
1770 PRINT #2,"Cassette ou Disquette (C/D) ? ";CHR$(24) >XF
";CHR$(24);CHR$(8);
1780 inpval$="":WHILE inpval$="":inpval$=UPPER$(INKEY$) >UE
:MEND:CLS #2
1790 IF inpval$="C" THEN support$="cassette":GOTO 1820 >CC
1800 IF inpval$(">"D" THEN PRINT CHR$(7);:GOTO 1770 >TU
1810 support$="disquette" >ZY
1820 CLS #2:PRINT #2,"Appuyer sur n'importe quelle touc >NM
he lorsque vous aurez inserer une ";support$;
1830 inpval$="":WHILE inpval$="":inpval$=UPPER$(INKEY$) >UA
:MEND:CLS #2
1840 IF support$="disquette" THEN !DIR,"*.asc":GOTO 189 >GM
0
1850 CLS #2:PRINT #2,"Avez-vous besoin d'initialiser le >MK
s commandes cassette (ITAPE) (O/N) ? ";:GOSUB 2850
1860 IF inpval$="D" THEN ITAPE >YR
1870 CLS #2:PRINT #2,"Press REC and PLAY then any key: >BX
";CHR$(24);" ";CHR$(24);CHR$(8);
1880 inpval$="":WHILE inpval$="":inpval$=UPPER$(INKEY$) >UF
:MEND:CLS #2
1890 INPUT #2,"Quel fichier voulez-vous creer (le type >XC
'.asc' est implicite) ";file$
1900 IF file$="" OR LEN(file$)>8 THEN PRINT CHR$(7);:CL >PN
S #2:GOTO 1890
1910 CLS #2:PRINT #2,"A quel numero de ligne voulez-vous >ED
s commencer ? ";ca%=2:ilenp%=5:GOSUB 2940
1920 nl%=VAL(valinp$) >QR
1930 CLS #2:PRINT #2,"Quel increment voulez-vous utilis >NB
er ? ";ca%=2:ilenp%=5:GOSUB 2940
1940 inc%=VAL(valinp$) >RD
1950 CLS #2:INPUT #2,"Nombre de datas par ligne ";ndat1 >XF
%
1960 IF ndat1%=0 THEN PRINT CHR$(7);:GOTO 1950 >NM
1970 CLS #2:PRINT #2,"Utilisez-vous le signe '&' pour l >DK
es datas (O/N) ? ";CHR$(24);" ";CHR$(24);CHR$(8);:GOSUB
2850
1980 IF inpval$="O" THEN amp%=1 ELSE amp%=0 >XM
1990 CLS #2:INPUT #2,"Longueur de la chaine 'somme' en >RE
fin de ligne ";lsum%
2000 IF lsum%(>8 THEN GOSUB 2390:CLS #2:PRINT #2,"Doit- >BM
on remettre a zero la chaine 'somme' au debut de chaque
ligne (O/N) ? ";CHR$(24);" ";CHR$(24);CHR$(8);:GOSUB 2
850:IF inpval$="O" THEN csum%=1 ELSE csum%=0
2010 ' >XF
2020 ' Ouverture du fichier choisi precedemment >XG
2030 ' ----- >XH
2040 ' >XJ
2050 CLS #1:CLS #2 >XW
2060 DIM valdat(ndat1%):sum=0 >YR
2070 OPENOUT "!"+"file$+".asc" >YJ
2080 ' >YC
2090 ' Entree des DATAs, de la somme et verification >YD
2100 ' ----- >XF
2110 ' >XG
2120 correct%=0:GOSUB 2520 >XU
2130 ' >XJ
2140 ' Ecriture dans le fichier choisi precedemment >XK
2150 ' ----- >YA
2160 ' >YB
2170 PRINT #9,RIGHT$(STR$(nl%),LEN(STR$(nl%))-1);" DATA >EU
";
2180 FOR i%=0 TO ndat1%-1 >RG
2190 IF amp%=1 THEN PRINT #9,"&"; >ZE
2200 PRINT #9,MID$("00",1,2-LEN(HEX$(valdat(i%)))):HEX$ >RF
(valdat(i%));
2210 IF i%(>ndat1%-1 THEN PRINT #9,""; ELSE IF lsum%(> >NZ
0 THEN PRINT #9,""; ELSE PRINT #9
2220 NEXT i% >AA
2230 IF lsum%=0 THEN GOTO 2300 >XM
2240 IF amp%=1 THEN PRINT #9,"&"; >ZA
2250 PRINT #9,MID$(STRING$(lsum%,"0"),1,LEN(STRING$(lsum >ML
%,"0"))-LEN(HEX$(valsum)));HEX$(valsum)
2260 ' >YC
2270 ' Test pour continuer ou sortir >YD
2280 ' ----- >YE
2290 ' >YF
2300 PRINT #2,CHR$(24);" Appuyer sur <F> pour sortir ou >GH
une autre touche pour continuer. ";CHR$(24);
2310 inpval$="":WHILE inpval$="":inpval$=UPPER$(INKEY$) >KM
:MEND
2320 CLS #2:IF inpval$="F" THEN CLOSEOUT:WINDOW 1,80,1, >MM
25:CLS:KEY DEF 66,0,&FC,&FC:POKE &B0EE,&C3:LOCATE 1
,1:PRINT "Vous pouvez charger votre fichier en tapant :
":PRINT:PRINT "LOAD ";CHR$(34);UPPER$(file$);".ASC";CHR
$(34):PRINT:END
2330 nl%=nl%+inc% >LX
2340 GOTO 2120 >XB
2350 ' >YC
2360 ' Sous-programme du choix de la formule de calcul >YD
de la somme
2370 ' ----- >YE
-----
2380 ' >YF
2390 CLS #2:PRINT #2,"Utilisez-vous cette formule de ca >JE
lcul (O/N) : TOT=TOT+PEEK(AD) ? ";:GOSUB 2850:IF inpval
$="O" THEN DEF FN sum=sum+valdat(i%):RETURN
2400 CLS #2:PRINT #2,"Utilisez-vous cette formule de ca >MJ
lcul (O/N) : S=S+K+65536*(S+K>32767) ? ";:GOSUB 2850:IF
inpval$="O" THEN DEF FN sum=sum+valdat(i%)+65536*(sum+
valdat(i%)>32767):RETURN
2410 CLS #2:PRINT #2,"Utilisez-vous cette formule de ca >QY
lcul (O/N) : soe=som+a*j ? ";:GOSUB 2850:IF inpval$="O"
THEN DEF FN sum=sum+valdat(i%)*(i%+1):RETURN
2420 ' >YA
2430 ' Formule de calcul suivante >YB

```



```

2440 ' ..... >YC
2450 ' Derniere formule de calcul >YD
2460 ' >YE
2470 PRINT CHR$(7);GOTO 2390 >XC
2480 ' >YG
2490 ' Sous-programme d'entree et de verification des D >YH
ATAAs
2500 ' ----- >XK
-----
2510 ' >YA
2520 inpsum%=0;IF correct%=0 THEN savsum=sum ELSE sum=s >XM
avsum
2530 posxl%=POS(#1);posyl%=VPOS(#1);IF csum%=1 THEN sum >DE
=0
2540 PRINT #1,RIGHT$(STR$(n1%),LEN(STR$(n1%))-1);" DATA >EL
";
2550 FOR i%=0 TO ndatl%-1 >RH
2560 IF amp%=1 THEN PRINT #1,"&"; >YF
2570 posxd%=POS(#1);posyd%=VPOS(#1);ca%=1;leninp%=2;IF >BK
correct%=0 THEN GOSUB 2940 ELSE valinp%=HEX$(valdat(i%
),leninp%);PRINT #ca%,valinp%;GOSUB 2950
2580 valdat(i%)=VAL("&"+valinp%) >AJ
2590 IF LEN(valinp%)<=2 THEN LOCATE #1,posxd%,posyd%;PR >UP
INT #1,HEX$(valdat(i%),2); ELSE PRINT CHR$(7);LOCATE #
1,posxd%,posyd%;PRINT #1,SPACE$(LEN(HEX$(valdat(i%),2)
));LOCATE #1,posxd%,posyd%;GOTO 2570
2600 IF i%<ndatl%-1 THEN PRINT #1," "; ELSE IF lsum%<> >NB
0 THEN PRINT #1," "; ELSE PRINT #1
2610 IF lsum%<>0 THEN sum=FN sum >AY
2620 NEXT i% >AE
2630 ' >YD
2640 ' Entree de la somme (si besoin) >YE
2650 ' ----- >YF
2660 ' >YG
2670 inpsum%=1;IF lsum%=0 THEN RETURN >GE
2680 IF amp%=1 THEN PRINT #1,"&"; >ZA
2690 posxd%=POS(#1);posyd%=VPOS(#1);ca%=1;leninp%=lsum% >RL
;IF correct%=0 THEN GOSUB 2940 ELSE valinp%=HEX$(valsum
,leninp%);PRINT #ca%,valinp%;GOSUB 2950
2700 valsum=VAL("&"+valinp%) >XU
2710 IF LEN(valinp%)<=lsum% THEN LOCATE #1,posxd%,posyd >DD
%;PRINT #1,HEX$(valsum,lsum%) ELSE PRINT CHR$(7);LOCAT
E #1,posxd%,posyd%;PRINT #1,SPACE$(LEN(HEX$(valsum,lsum
%)));LOCATE #1,posxd%,posyd%;GOTO 2690
2720 ' >YD
2730 ' Verification de la somme (si besoin) >YE
2740 ' ----- >YF
2750 ' >YG
2760 IF valsum=sum THEN RETURN >AX
2770 PRINT #2,CHR$(24);" Erreur dans les datas. Appuyer >QR
sur <R> pour recommencer ou <C> pour corriger. ";CHR$(
24);correct%=0
2780 inpval$="";WHILE inpval$="";inpval%=UPPER$(INKEY$) >XH
;WEND
2790 IF inpval$="R" THEN CLS #2;LOCATE #1,posxl%,posyl% >6X
;PRINT #1,SPACE$(80);LOCATE #1,posxl%,posyl%;sum=savsu
m;GOTO 2520
2800 IF inpval$="C" THEN correct%=1;LOCATE #1,posxl%,po >MK
syl%;GOTO 2520 ELSE PRINT CHR$(7);GOTO 2770
2810 ' >YD
2820 ' Sous-programme d'entree O (oui) / N (non) >YE
2830 ' ----- >YF
2840 ' >YG
2850 PRINT #2,CHR$(24);" ";CHR$(24);CHR$(8); >LU
2860 inpval%=UPPER$(INKEY$);IF inpval$="" THEN 2860 >MK
2870 IF inpval$<>"O" AND inpval$<>"N" THEN PRINT CHR$(7 >NZ
);GOTO 2860
2880 PRINT #2,inpval%;CHR$(24);" ";CHR$(24);CHR$(8); >MR
2890 RETURN >GC
2900 ' >YD
2910 ' Sous-programme d'entree de caracteres hexadecima >YE
ux et de controle
2920 ' ----- >YF
-----
2930 ' >YG
2940 valinp$="" >CF
2950 inpval$="" >CG
2960 WHILE inpval$<>CHR$(13) >XH
2970 IF correct%=0 THEN PRINT #ca%,CHR$(24);" ";CHR$(24 >BB
);
2980 IF correct%=1 AND inpsum%=0 THEN PRINT #ca%,CHR$(2 >UY
4);" ";CHR$(24);
2990 IF correct%=1 AND inpsum%=1 THEN PRINT #ca%,CHR$(2 >UM
4);" ";CHR$(24);
3000 inpval%=UPPER$(INKEY$);IF inpval$="" THEN 3000 >MG
3010 IF ASC(inpval%)=0 THEN 3000 ' Touche BREAK >ZA
3020 IF inpval%=CHR$(13) THEN 3100 ' Touche RETURN >AP
3030 IF inpval$<>CHR$(127) THEN 3070 ' Touche DEL >CC
3040 IF LEN(valinp%)=0 THEN PRINT CHR$(7);inpval$="";G >XH
OTO 3000
3050 valinp%=MID$(valinp$,1,LEN(valinp%)-1) >NJ
3060 PRINT #ca%,CHR$(8);CHR$(8);" ";CHR$(8);CHR$(8);i >ZA
npval$="";GOTO 2970
3070 IF LEN(valinp%)=leninp% THEN PRINT CHR$(7);inpval >XF
$="";GOTO 3000
3080 IF (inpval$>"0" AND inpval$<="9") OR (inpval$>"A >AE
" AND inpval$<="F") THEN PRINT #ca%,CHR$(8);" ";CHR$(8)
;inpval%; ELSE PRINT CHR$(7);GOTO 3000
3090 valinp%=valinp$+inpval$ >YZ
3100 IF LEN(valinp%)=0 THEN PRINT CHR$(7);inpval$="";G >XE
OTO 3000
3110 WEND >MK
3120 PRINT #ca%,CHR$(8);" ";CHR$(8); >DE
3130 RETURN >FA
3140 ' >YA
3150 ' Sous-programme de traitement du BREAK >YB
3160 ' ----- >YC
3170 ' >YD
3180 CLOSEOUT;WINDOW 1,80,1,25;LOCATE 1,25;PRINT SPACE$ >LF
(80);LOCATE 1,24;GOTO 3260
3190 ' >YF
3200 ' Sous-programme de traitement des erreurs >XH
3210 ' ----- >XJ
3220 ' >XK
3230 CLOSEOUT;WINDOW 1,80,1,25;CLS >DM
3240 PRINT "Erreur #";ERR;"a la ligne #";ERL >JB
3250 RESUME 3260 >6K
3260 END >YK

```


PRATIQUE DE L'ASSEMBLEUR

LE HARDWARE DU CPC

Emmanuel GUILLARD

Hé, non ! Ce n'est pas le listing de l'éditeur ligne que vous aurez ce mois ci, il a subi quelques problèmes et dégâts qui ont engendrés son retard. Ce mois-ci donc, on va commencer à s'attarder sur le matériel, que dis-je, le «hardware» du CPC. Amstrad à cœur ouvert !

Avez-vous essayé un jour de démonter votre CPC ? On découvre une carte électronique parsemée de plusieurs circuits Intégrés (ces petites boîtes noires entourées d'énormément de pattes). Les électroniciens avertis auront bien sûr reconnu le micro-processeur Z-80, la RAM, la ROM et quelques circuits spéciaux supplémentaires.

Le schéma suivant vous montre les différents circuits et la façon dont ils sont liés sur le CPC 464, les 664 et les 6128 étant assez similaires.

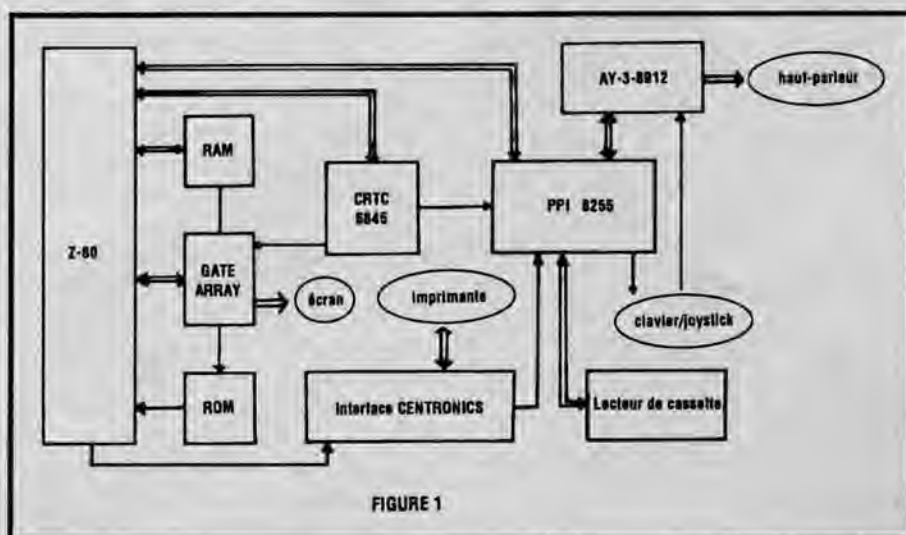


FIGURE 1

Description des différents éléments.

Le Z-80

C'était juste histoire d'en parler ! Notre cher micro-processeur 8 bits possède quand même 8 lignes d'adresse, ce qui lui permet (chaque ligne pouvant prendre la valeur «1» (+5 V) ou «0» (0 V)) d'adresser 2^8 soit 65536 positions mémoire. Ces positions mémoire étant des octets, le Z-80 peut donc adresser 65536 octets soit 64 Koctets ($65536/1024 = 64$). Ça vous la coupe, hein !

Mémoire ROM/RAM

Vous le savez maintenant, la ROM est une mémoire qu'on ne peut que lire et la RAM est une mémoire où l'on peut aussi bien lire qu'écrire.

Suivant les modèles de CPC, on a effectivement :

- * 64 Ko de RAM et 32 Ko de ROM pour le 464
- * 64 Ko de RAM et 48 Ko de ROM pour le 664
- * 128 Ko de RAM et 48 Ko de ROM pour le 6128

Ces capacités dépassent alors les 64 Ko que peut adresser notre Z-80. Il est cependant possible de «tricher» et d'utiliser plus de mémoire que prévu en utilisant le procédé de commutation de banques.

Sur Amstrad, la mémoire est paginée, c'est à dire divisée en plusieurs pages ou banques. Chaque banque représente 16 Ko. Prenons l'exemple du CPC 464 :

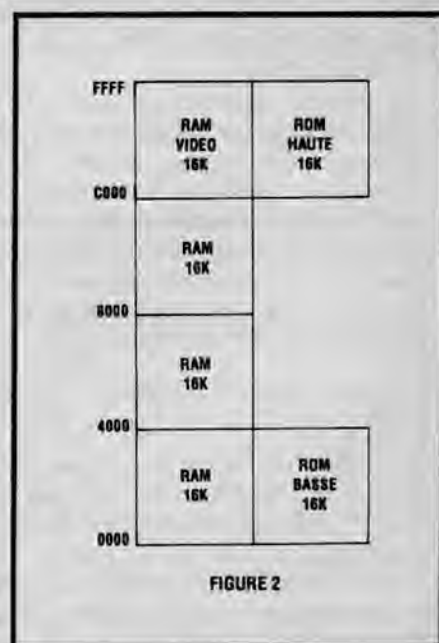


FIGURE 2

On voit ici que sur les zones 0000-3FFF et C000-FFFF se superposent respectivement RAM et ROM basse, et RAM vidéo et ROM haute. Pour accéder à l'une des deux ROM, le logiciel commute les banques nécessaires.

A noter que les instructions d'écriture, dans la plupart des cas, sélectionneront obligatoirement les banques RAM.

La ROM basse contient le système d'exploitation et les routines arithmétiques.

La ROM supérieure ou ROM haute contient l'interpréteur BASIC et peut elle-même être commutée avec 252 ROM supplémentaires !

Gate array.

Ce circuit est un peu la grosse bête du système des CPC. Ne cherchez pas dans les magasins, ce circuit a été spécialement développé pour le CPC et ses concepteurs n'en ont dévoté que très peu de choses.

Voici ses fonctions principales :

- * Générations des différentes fréquences d'horloge pour les autres circuits.

- * Commande les accès à la RAM, et génère les signaux servant à l'exploitation de la RAM dynamique..

- * Gestion de la connexion et la déconnexion de la ROM.

- * Génération des signaux vidéos, RVB pour le moniteur, du mode écran, stockage des couleurs de chaque encre.

- * Génération de l'impulsion d'inter-rup-tion.

Ce circuit reste assez mystérieux mais possède un intérêt énorme pour la gestion de tout le reste du système. C'est un peu le responsable des fantastiques possibilités du CPC.

CRTC 6845

Le CRTC (Cathode Ray Tube Controller, ou pour les français purs et durs, contrôleur vidéo) est en fait destiné à la «gestion» de l'écran. Le Gate array lui envoie un signal d'horloge de 1 MHz, et le CRTC l'utilise pour générer les signaux de syn-

chronisation avec le moniteur.

Ce circuit est programmable par l'utilisateur, à l'aide de 19 registres :

AR : registre d'adresse. Il sert à sélectionner les 18 autres registres (5 bits).

R0 : registre 8 bits dans lequel on envoie le nombre de caractères par ligne.

R1 : registre 8 bits comprenant le nombre de caractères réellement affichables à l'écran. Il doit évidemment être inférieur à R0.

R2 : registre 8 bits servant à la synchronisation horizontale.

R3 : registre 4 bits servant aussi à la synchronisation horizontale.

R4 : registre 7 bits contenant le nombre de caractères affichables verticalement.

R5 : registre 5 bits servant à l'ajustement vertical.

R6 : registre 7 bits contenant le nombre de lignes réellement affichées à l'écran.

R7 : registre 7 bits servant à la synchronisation verticale.

R8 : registre 2 bits servant à donner le mode de fonctionnement du CRTC.

R9 : registre 5 bits contenant le nombre de lignes par caractères.

R10 : registre 7 bits servant à programmer l'affichage du curseur.

R11 : registre 5 bits contenant la ligne où finit le curseur.

R12 : registre 6 bits contenant le poids fort de l'adresse de départ de la mémoire écran

R13 : registre 8 bits contenant le poids faible de l'adresse de départ de la mémoire écran (adresse basse).

R14 : registre 6 bits contenant l'adresse haute de la position du curseur.

R15 : registre 8 bits contenant l'adresse basse de la position du curseur.

R16 : registre 6 bits fournissant l'adresse haute fournie par le crayon optique lorsqu'il est activé.

R17 : registre 8 bits fournissant l'adresse basse fournie par le crayon optique.

Vous voyez donc que ce registre possède de nombreuses fonctions, même si certaines vous semblent floues. Rassurez-vous, tout ceci sera expliqué ultérieurement avec beaucoup plus de détails.

PPI 8255.

Ou Programmable Parallel Interface, soit Interface Parallèle programmable.

Ce circuit dispose de 24 canaux, accessibles en entrée ou en sortie, répartis en 3 ports 8 bits, dont un est programmable.

Il s'occupe du clavier, du lecteur de cassette (commande du moteur, écriture et lecture des données), du chip sonore, du CRTC 6845, du test de l'imprimante, du connecteur d'extension et possède quelques autres fonctions.

Chip sonore AY-3-8912.

Je suppose que vous connaissez assez bien les possibilités sonores du CPC :

- * 3 canaux sonores indépendants A, B et C.
- * 1 générateur de bruit.
- * Enveloppes (ton et volume) programmables.

Eh bien, vous l'aurez deviné, le chip sonore s'occupe de toutes ces fonctions à l'aide de 16 registres dont 15 sont utilisables :

Registres 0 et 1 :	fréquence du son sur canal A
Registres 2 et 3 :	idem canal B.
Registres 4 et 5 :	idem canal C.
Registre 6 :	registre servant au générateur de bruit.
Registre 7 :	registre multi-fonctions servant principalement à couper ou mettre le bruit et/ou le son indépendamment sur les canaux A, B et C.
Registre 8 :	Volume sur le canal A.
Registre 9 :	idem R8 pour le canal B.
Registre 10 :	idem R8 pour le canal C.
Registres 11 et 12 :	Période de la courbe d'enveloppe.
Registre 13 :	forme de la courbe du générateur de courbe d'enveloppe.

Voilà donc une présentation globale des circuits principaux du CPC. Certes cette présentation est rapide mais il vaut mieux, d'après moi, avoir une vue générale de tout le système avant d'attaquer chaque circuit de façon plus détaillée.

On se retrouve le mois prochain, avec le fameux listing de l'éditeur ligne. **A bientôt !**

ANTI ERREURS

Grégory NOE

Valable pour CPC 464,
CPC 664, CPC 6128

Il est difficile de taper un programme d'une revue sans commettre d'erreurs. Quand les erreurs sont signalées (Syntax error In..., Lines does not exist In...), la correction est facile puisque la ligne est indiquée, mais lorsqu'au "point de vue" du CPC, le programme est correct, il peut en être tout autrement pour le lecteur.

Si le fonctionnement du programme ne correspond pas vraiment au programme original, alors il est impossible de retrouver l'erreur commise, à moins de comparer les lignes de votre programme avec celles du journal, mais c'est très long et on risque de passer sur une erreur sans la voir.

C'est pourquoi ce programme a été créé : il est à utiliser si vous ne trouvez pas vos erreurs.

La frappe d'un programme de votre journal favori (CPC) terminée, vous faites "RUN" et là, vous corrigez toutes les erreurs signalées par le CPC en vous référant au journal pour comparer. Si malgré cela, il ne tourne toujours pas, suivez cette procédure :

- sauvez votre programme en ASCII grâce à l'instruction : Save "nom prog", A.

Puis, si vous êtes sur K7, rembobinez la bande jusqu'au début du fichier ;
- chargez Anti-Erreurs ;



- entrez le nom de votre programme et insérez le support où il se trouve (K7 ou disquette).

Après quelques secondes, les numéros de ligne vont défilier. Vous remarquerez qu'à chaque ligne correspond un code de 2 lettres sous cette forme : >XX ; c'est le code qui va vous permettre de savoir d'où proviennent les erreurs. Pour cela, il faut que vous compariez les codes du journal à ceux de l'écran. Si un code diffère, notez sur papier le numéro de ligne qui correspond à ce code. Continuez ainsi jusqu'à la fin du listage. Le défilement terminé, recherchez votre programme et corrigez les lignes que vous avez notées en comparant avec le journal.

Faites "RUN" et, à miracle !, ça marche. Attention, lors de la frappe du programme à traiter, n'omettez aucun caractère car Anti-Erreurs les prend en compte et bien sûr, ne tapez pas les codes de contrôle.

PETITES annonces

SUR MINITEL

36 15

Code **MHZ**

```

10 ' >LA
20 ' >LB
30 ' ANTI - ERREURS >LC
40 ' >LD
50 ' VERSION 2.0 >LE
60 ' >LF
70 ' >LG
80 ' >LH
90 ' (c) CPC & G. NOE 1987 >LI
100 ' >RB
110 ' >RC
    
```


ABONNEZ-VOUS !

11 numéros
210 F

32 F
D'ECONOMIE



5% de remise
sur les produits
du catalogue
SORACOM



OUI, je désire m'abonner à **CPC Infos** pour
un an au prix de 210 F (11 numéros)

Nom : _____ Prénom : _____

Adresse : _____

Code postal : _____ Ville : _____

Date : _____ Signature obligatoire

ATTENTION !
SEUL CE BULLETIN
D'ABONNEMENT
EST VALABLE

NOUVEAU !
PAYEZ AVEC
VOTRE
CARTE
BANCAIRE

- Les abonnements ne sont pas rétroactifs
- Envoi par avion + 120 F

Je désire payer avec une carte bancaire

Date d'expiration

Envoyez votre bulletin accompagné du règlement à : Editions SORACOM - Service abonnement - BP 88 - 35170 BRUZ

TRUCS EN VRAC

Encore une fourmée de lettres
made in Jean-Pierre MARQUET. Et
puisque je suis un peu maso dans
l'âme, on commence par des
reproches !

● Merci à Guy DUBUS, qui fournit,
et pour la troisième fois dans le
même journal (lequel ? NDLR) !...
un truc complètement faux, celui
permettant d'écrire sur une dis-
quette formatée en 42 pistes (page
13, en haut de la colonne de gau-
che). Ce "truc", publié jadis par
CPC n°23, en Juin 87, consiste, entre
autres, à ajouter 13 au contenu de
l'octet &A895, qui contient en
temps ordinaire le nombre de blocs
+1 que peut exploiter l'AMSDOS.
Comme si formater DEUX pistes sup-
plémentaires pouvait faire gagner
TREIZE Ko !

Autre bourde : si par inadvertance
vous lancez plusieurs fois le pro-
gramme qu'il donne sous le nom
d'INIT, ces 13 K seront ajoutés cha-
que fois, et le CAT d'une disquette
vierge vous donnera successive-
ment : 191 K, 204 K, 217 K libres, etc.
Or, faites l'essai de sauver 11 pa-
ges-écran : 11 x 17 187 ; les 42 pistes
de la disquette sont alors pleines, et
vous n'y sauvez même plus un
simple "10 REM". Vous n'obtiendrez
qu'un WRITE FAIL...

Le seul et vrai truc consiste à utiliser
le format Data (pourquoi en choisir
un autre, à moins d'être maso ?), et
à taper une fois pour toutes POKE
&A8A8,255 : POKE &A895,188. Avec
le format System, si vraiment vous y
tenez, POKE &A8A8,255 : POKE
&a895,179. Et, bien sûr, pas d'OPE-
NOUT ni de CLOSEOUT. Je ne vous
demande pas de me croire sur pa-
role. Essayez...



Ne passez
plus de temps à
taper les listings
et évitez d'user

vos petits doigts manucurés avec les disquettes
CPC Infos.

Dans la disquette 34 vous trouverez les pro-
grammes de CPC Infos n° 29 et n° 30 :

• GRAFFITI & SUPertext • CIRCUS FOLIES
• JAQUETTE • Pousnik • SPEAKER • VA-
RIABLES LOCALES • OPERATEURS LOGI-
QUES • LES PREMIERS LISTINGS D'AXYS.

CPC INFOS N° 34

DISQUE

TRUCS EN VRAC



On continue avec quelques méthodes pour éviter les fatigues excessives lors des frappes prolongées.

● Bravo pour le programme MODEX du numéro 26 ! Enfin, on va pouvoir utiliser des images en mode 0 avec OXFORD PAO ! Merci à Stéphane et Michael. Mais, une fois de plus, CPC Infos a oublié de donner le chargeur Basic permettant de mettre en œuvre ce logiciel en langage machine ; et, comme cette charmante habitude est presque systématique (allons bon, NDLR), je propose, plutôt que de se lamenter, d'en faire l'objet d'un concours à l'intention de vos lecteurs perspicaces, dans l'esprit de feu le "Listing truqué" qui passait naguère : le concours consisterait naturellement à trouver le lanceur Basic manquant. Voici ma réponse à ce concours hypothétique :

Je suppose que Stéphane et Michael, les auteurs de MODEX, ont fourni un lanceur du genre de :

```
10 MEMORY &1FFF
20 LOAD"modex1.bin",&2000
30 LOAD"modex2.bin",&2000+&1C00
40 CALL &598B
```

Fort bien ; mais cela fait trois fichiers (le lanceur Basic et les deux fichiers binaires). Temps de mise en œuvre : 10 secondes ; occupation sur la disquette : 17 K. Je propose plutôt ceci ; taper successivement en mode manuel :

```
- MEMORY &1FFF
- LOAD"modex1.bin",&2000
- LOAD"modex2.bin",&3C00
- POKE &599C,&C9
- CALL &598B
- SAVE"modex",B,&2000,&1C00*2,&5938
```

Vous pouvez alors effacer MODEX1.BIN et MODEX2.BIN. Le MODEX.BIN qui restera sur la disquette n'occupe plus que 15 K, se lance en 6 secondes par un simple RUN*MODEX, et le MEMORY n'est plus nécessaire.

Voici, en prime, comment je tente de me faciliter la vie lorsque j'ai à taper les chargeurs Basic, des excellents mais gigantesques (quelques 90 K de Basic pour le VUMETRE du numéro 25 !) fichiers binaires de CPC Infos : comme ils commencent généralement par quatre lignes de REMs, je remplace ces quatre lignes par les suivantes :

```
10 z$="lanceur":y$=""_bak":KEY 150,"save z$:
üera,y$:cat"+CHRS(13):KEY DEF 68,0,150:
SPEED KEY 10,2
20 BORDER 13:INK 0,13:INK 1,0
30 KEY 10,"":KEY 11,CHRS(13)+data"
40 KEY DEF 26,0,65:KEY DEF 17,0,66:
KEY DEF 28,0,67:KEY DEF 19,0,68:
KEY DEF 30,0,69:KEY DEF 22,0,70
```

Explications : la ligne 10 vous permettra de sauvegarder périodiquement votre travail par un simple appui sur la touche TAB, et accélèrera la vitesse du curseur pour faciliter les corrections. La ligne 20 établit les couleurs d'encre les moins fatigantes pour vos pauvres yeux surmenés. La ligne 30 transforme en virgule le point du pavé numérique, et redéfinit la touche ENTER pour vous éviter d'avoir à taper DATA cinq cent fois. Et la ligne 40 oblige les touches les plus proches du pavé numérique à écrire les lettres A-B-C-D-E-F, nécessaires pour l'hexadécimal. Vous pourrez ainsi tout taper d'une main, la droite hélas ! mais ce monde n'est pas fait pour les gauchers... Naturellement, votre programme Basic s'appellera LANCEUR. Lorsqu'il aura fait son office et aura installé en mémoire le fichier binaire, ne l'effacez pas : supprimez simplement par DELETE les seules lignes de DATAs, et sauvegardez-le, il resservira pour la prochaine fois. (On peut aussi utiliser le programme DataLoad NDLR).

Maintenant, pourquoi ne pas modifier CATEDIT et rectifier Pratique de l'assembleur ?

● Je propose aux utilisateurs de l'excellent CATEDIT, publié dans le numéro 46, deux petites bidouilles. La première concerne l'option "Formatage". En effet, ce logiciel, qui doit dater un peu, formate les disquettes en 40 pistes, alors que tout le monde, aujourd'hui (du moins je l'espère), utilise le format en 42 pistes. Mettez-vous au goût du jour en chargeant CATEDIT à son adresse habituelle, sans faire RUN, puis tapez en mode direct : PRINT PEEK (&93D2). Si vous obtenez 39, faites un POKE &93D2,41. Essayez en formatant une disquette vierge : si cela marche, faites un RESET, recommencez, et sauvez avec la modification, par un :
SAVE "catedit",B,&7000,&24AD,&7000

(Vous avez remarqué que j'ai supprimé la page de présentation, réduisant le programme à 10 K, comme le conseille Serge BREUZIN dans le numéro 27 de décembre 1990. Temps de mise en œuvre : 5 secondes).

Si vous préférez modifier CATEDIT avec DISCOLOGY, recherchez dans le dernier bloc la chaîne hexa 3A-2E-8F-FE-27-28-12-3C-32-2E-8F (c'était la ligne 3470 du chargeur Basic CAT3), et remplacez le 27 par un 29.

Par ailleurs, pourquoi ne pas se servir d'une bidouille ancienne, parue dans le numéro 29 d'AMSTAR & CPC, page 109, et qui permettait de doubler la vitesse d'affichage du texte en mode 2 ? Vous trouverez ci-joint le programme Basic qui ajoute à CATEDIT les 32 octets nécessaires, juste avant le programme principal (cette bidouille, telle quelle, est donc incompatible avec celle de Serge BREUZIN cité plus haut ; mais vous pouvez la placer ailleurs). A présent, permettez-moi de manier le bâton : dans le numéro 27 de décembre, Emmanuel GUILLARD présente un multi-tableau regroupant les instructions du Z-80, mais il commet l'erreur, en ce qui concerne le calcul du temps d'exécution de chaque instruction, de confondre le nombre de "cycles mémoire" avec ce que les livres spécialisés appellent le "nombre d'états", seul à prendre en compte pour le calcul, puisqu'il représente le nombre

de cycles d'horloge nécessaires pour décoder l'instruction, puis pour l'exécuter. Bref, aucune instruction ne s'exécute en UN cycle d'horloge, comme il le laisse supposer pour LD r1,R2 ou pour DAA. Encore plus fort, LDIR ne fait pas son travail en 5 cycles d'horloge, car ici, le temps dépend du nombre d'octets à transférer ! En réalité, LDIR transfère un octet en 21 cycles d'horloge tant que BC est différent de zéro, puis en 16 cycles le dernier octet, quand BC est nul. Ce qui n'est pas du tout la même chose. Par ailleurs, l'horloge du CPC tourne bien à 4 MHz, mais, pour des raisons intrinsèques, la vitesse effective est réduite à 3,3 MHz, détail peu connu, quoique mentionné dans le manuel technique de la bécaune. Enfin, il faut tenir compte des interruptions, qui prennent

9 % du temps d'exécution, donc multiplier la durée globale par 1,09.

Pour vous convaincre de tout cela, faites l'expérience de bricoler le CLS "maison" suivant :

```
LD HL,&C000
LD (HL),0
LD DE,&C001
LD BC,&3FFF
LDIR
RET
```

Ici, LDIR est exécuté 16383 fois, à raison de 21 cycles d'horloges à 3,3 MHz : le calcul théorique donne : $21 * 16383 * 1,09 / 3\ 300\ 000 = 0,1136$ secondes

Chronométrez en vous servant de la variable TIME : vous obtenez bien 0,11 secondes. CQFD.

```
10 MEMORY &6FDF:LOAD"catedit
20 DATA e5,cd
30 DATA d4 : REM : pour 664 : d0 ; pour 464 : d3
40 DATA 12,eb,e1,cd
50 DATA 6a : REM : pour 664 : 66 ; pour 464 : 64
60 DATA 0b,3a
70 DATA 30,b7 : REM : pour 664 : 30,b7 ; pour 464 : 90,b2
80 DATA 4f,06,08,1a,a9,13,77,7c,ce,08,67,10,f6,c9
90 DATA 21,e0,6f,22,d4,bd
100 FOR i=&6FEO TO &6FFF:READ a$:POKE i,VAL("&"+a$):NEXT
110 CALL 47896 : REM : attente si on veut changer de disquette
120 SAVE"essai",b,&6FEO,&24C0,&6FFA
130 REM : Essayez ESSAI avant de le renommer CATEDIT !●
```

**Enfin, un truc qui n'a jamais été
publié ailleurs et révélera la
personnalité profonde de votre
CPC.**

● Cher CPC Infos,

J'ai récemment entrepris de mettre au point une petite bidouille de derrière les fagots, que tu recevras peut-être un de ces jours, quand j'y aurais mis la touche finale. Cette bidouille, lorsque tu la réceptionneras, je l'aurai sauvée sous le nom d'ALEXIS.

Pourquoi ALEXIS, me diras-tu ? C'est une excellente question, et je te remercie de me l'avoir posée. Voici donc l'explication. Accroche-toi.

Figure-toi qu'il y a longtemps que je rêve, lorsque je fais un CAT, de voir s'étaler sous mes yeux éblouis la liste (quasi-interminable, à condition toutefois d'être en mode 2) de mes conquêtes féminines. J'avais donc résolu, désormais, de ne plus sauver mes programmes que sous des noms féminins qui m'étaient chers (ô combien ! Mon grand-père lui-même, et son père avant lui, se ruinaient pour les dames). Lorsque j'ai voulu sauver, par conséquent la bidouille dont à propos

de laquelle que je t'entretenais quelques lignes plus haut, j'ai fait, étant en Basic, un SAVE"PRISCILLA", du nom de ma petite amie du moment.

Quelle ne fut pas ma contrariété d'obtenir un vexant "Bad command" ! Têtu, j'essayai alors, successivement, JACQUELINE, CHRISTELLE, GENEVIEVE, ALEXANDRA, FLORENTINE, CONCEPCION (j'ai passé mes dernières vacances en Espagne), CONTRACEPCION, VERONIQUE, BERNADETTE (elle est très chouette !), CLOTHILDE, HENRIETTE, PHILAMINTE (merci Mollère), et même, puisque je vis en Afrique FETE NATIONALE : en vain le bide !

C'est d'autant plus contrariant que la sauvegarde marche très bien avec PIERRE, MAURICE, FREDERIC, EMILE, JULIEN, LUCIEN, ERIC, THOMAS, GILLES, ALBERT, FABRICE, DENIS, FRANCK, RENAUD, avec les noms des quatre évangélistes, LUC, MATHIEU, MARC et JEAN, ou encore les noms des quatre frères de Jésus, JOSEPH, JACQUES, JUDE et SIMON.

J'ai même tenté le coup avec les prénoms de la famille Giscard (je suis resté fidèle à Sa Mémoire) : l'ordinateur accepte VALERY, LOUIS ou RITON, mais refuse ANNE-AYMONE et VALERIE-ANNE !

C'est ainsi que j'ai découvert que mon Amstrad était sexiste.

De guerre lasse, je m'apprêtais, ô toi la providence des désespérés, à t'écrire pour te demander conseil, lorsque j'eus l'idée géniale de sauver sous le nom d'"ALEXIS" : en effet, mon 6128, qui ne regarde pas Dynastie à la télévision, ignore que, depuis l'épisode 15 de cette fabuleuse sagaga, c'est aussi un prénom féminin. Et ça a marché, cette anecdote n'y a vu que du feu !

Je suis très fier de ma ruse, et je ne doute pas qu'elle rendra service à tes lecteurs. Afin d'augmenter encore la portée, crois-tu que je devrais l'envoyer à MICRO-MAG ?



TRUCS EN VRAC

Un lecteur d'EPROM 8 Ko (2764) et 16 Ko (27128) tout fait pour Amstrad CPC 6128, 664 et 464.

- Fournitures de base :
- Interface LE HACKER de DUCHET COMPUTERS.
 - Assembleur-Désassembleur ZEN.

- Possibilités :
- Lire le contenu d'une EPROM 2764 (8 Ko) ou 27128 (16 Ko).
 - Sauver le contenu sur disc ou K7.
 - Tester la virginité de l'EPROM.

- MANIPULATION :
- Pour cela il faut :
- Repérer l'encoche sur l'EPROM à lire.
 - Faire de même avec celle du HACKER puis la retirer de son support en prenant soin de ne pas abîmer les broches.
 - Placer l'EPROM à lire sur le support d'EPROM du HACKER en respectant la position (encoche de l'EPROM vers le bouton poussoir du HACKER).
 - Installer le HACKER (non validé) sur le port d'extension du CPC.
 - Mettre en route l'ordinateur.
 - Lancer ZEN par MEMORY 16383:LOAD*ZEN*:CALL 16384.
 - Au curseur taper le programme suivant :

```

Pour une EPROM 16 Ko
K
E
1 ORG 1000H
2 LOAD 1000H
3 LD C,00H
4 CALL 0B90FH
5 LD HL,0C000H
6 LD DE,06700H
7 LD BC,04000H
8 LDIR
9 JP 4000H
10 END
11 .
    
```

```

Pour une EPROM 8 Ko
K
E
1 ORG 1000H
2 LOAD 1000H
3 LD C,00H
4 CALL 0B90FH
5 LD HL,0C000H
6 LD DE,06700H
7 LD BC,02000H
8 LDIR
9 JP 4000H
10 END
11 .
    
```

- Assembler ce programme par les commandes A puis V pour obtenir l'assemblage à l'écran.
- Valider LE HACKER.
- Lancer ce programme par la commande G1000H.

Le programme en 1000H valide la ROM 00 qui est la position occupée par le HACKER quand il est validé et déplace au choix 4000H ou 2000H octets de C000H (adresse de départ du HACKER) en 6700H.

Quand ZEN redonne la main on peut :

- Lire le contenu de l'EPROM par la commande Q6700H.
- Sauvegarder le contenu par la commande WB avec les paramètres
16 Ko DEBUT>6700H FIN>A6FFH
START>6700H NAME>EPROM16.BIN
8 Ko DEBUT>6700H FIN>86FFH
START>6700H NAME>EPROM8.BIN
- Tester la virginité de l'EPROM en vérifiant si toutes les positions sont à FF en plaçant à l'aide de la commande M2000H les octets

Pour une 16 Ko, voir listing 1.

Pour une 8 Ko, voir listing 2.

- On lance le programme de test par G2000H
- Si toutes les positions mémoires de l'EPROM testée :
- Sont à FF on a le message TEST VIRGINITE OK
 - Ne sont pas à FF on a le message EPROM PAS VIERGE.

On peut maintenant installer une autre EPROM.

- ATTENTION !
MONTAGE ET DEMONTAGE DES EPROM SE FONT :
- HORS TENSION
 - EN RESPECTANT LE SENS DE MONTAGE DANS LE HACKER
 - EN PRENANT SOIN DE NE PAS ABÎMER LES BROCHES DES EPROMS.

Jean-Michel NIETO

LISTING 1

Pour une 16 Ko

```

2000 21 00 67 01 FF 3F 7E FE FF 20 0F 23 0B 78 B1 20  !.q..?`.. .#.x.
2010 F5 21 3D 20 06 15 CD 23 20 C9 21 2B 20 06 14 CD  .!= ...# .!+ ...
2020 23 20 C9 7E CD 5A 8B 23 10 F9 C9 0D 0A 45 50 52  # .".Z.#.....EPR
2030 4F 4D 20 50 41 53 20 56 49 45 52 47 45 0D 0A 54  OM PAS VIERGE..T
2040 45 53 54 20 56 49 52 47 49 4E 49 54 45 20 4F 4B  EST VIRGINITE OK
2050 0D 0A 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
```

LISTING 2

Pour une 8 Ko

```

2000 21 00 67 01 FF 1F 7E FE FF 20 0F 23 0B 78 B1 20  !.q... .. .#.x.
2010 F5 21 3D 20 06 15 CD 23 20 C9 21 2B 20 06 14 CD  .!= ...# .!+ ...
2020 23 20 C9 7E CD 5A 8B 23 10 F9 C9 0D 0A 45 50 52  # .".Z.#.....EPR
2030 4F 4D 20 50 41 53 20 56 49 45 52 47 45 0D 0A 54  OM PAS VIERGE..T
2040 45 53 54 20 56 49 52 47 49 4E 49 54 45 20 4F 4B  EST VIRGINITE OK
2050 0D 0A 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
```



AXYS

T H E • L A S T • B A T T L E

3ème partie

Fabien FESSARD & Stéphane ST-MARTIN

Valable pour CPC 6128, 464 + ext. 64 ko

Il y a encore du chemin pour arriver à AXYS. Pour ceux qui seraient découragés par la masse de Datas à taper, il y aura une disquette contenant AXYS en entier.

LOADER



```

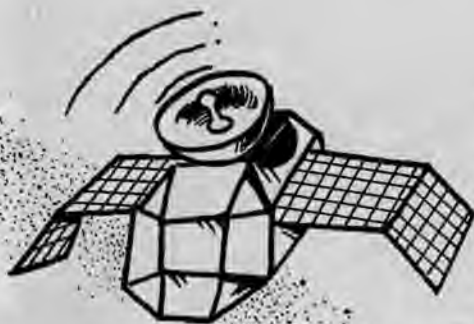
10 ' LOADER AXYS
20 ' (C) 1990 CPC INFOS
30 ' S.ST-MARTIN & F.FESSARD
40 '
50 MODE 2:AD=&A000:NL=110
60 FOR A=1 TO 20:B=0:FOR C=1 TO 15:READ A#
70 POKE AD,VAL("&"+A#):B=B+PEEK(AD):AD=AD+1
80 NEXT:READ E#:IF VAL("&"+E#)<>B THEN 100
90 NL=NL+10:NEXT:SAVE"AXYS.BIN",B,&A000,&120,&A000:CLS:END
100 CLS:PRINT"LIGNE"NL;" INCORRECTE":END
110 DATA 01,C0,7F,ED,49,21,FF,B0,11,40,00,0E,07,CD,CE,647
120 DATA BC,3E,FF,CD,6B,BC,06,09,21,C7,A0,11,F4,6C,CD,7C2

```

```

130 DATA BB,A0,CD,F4,6C,06,09,21,D0,A0,11,40,01,CD,BB,702
140 DATA A0,CD,F5,75,06,0B,21,0C,A1,11,40,00,CD,BB,A0,62F
150 DATA 06,08,21,04,A1,11,40,1D,CD,BB,A0,06,0A,21,D9,474
160 DATA A0,11,3F,21,CD,BB,A0,06,08,21,E3,A0,11,00,3E,53A
170 DATA CD,BB,A0,06,0B,21,F4,A0,11,00,40,CD,BB,A0,21,600
180 DATA 00,40,AF,77,01,C4,7F,ED,49,2F,77,01,C0,7F,ED,683
190 DATA 49,BE,28,37,01,C4,7F,ED,49,06,09,21,EB,A0,11,5AC
200 DATA 00,40,CD,BB,A0,3E,30,32,FF,A0,11,00,48,21,FF,620
210 DATA A0,34,3E,37,BE,28,0E,06,05,D5,CD,BB,A0,E1,11,637
220 DATA 00,00,19,EB,18,E9,01,C0,7F,ED,49,AF,18,02,3E,58A
230 DATA 01,32,3B,00,C3,40,00,D5,CD,77,BC,E1,CD,83,BC,733
240 DATA CD,7A,BC,C9,53,54,41,52,54,2E,41,58,53,49,4E,60B
250 DATA 54,52,4F,2E,41,58,53,53,50,52,49,54,45,2E,41,455
260 DATA 58,53,4D,41,53,48,2E,41,58,53,53,4F,55,4E,44,47A
270 DATA 2E,41,58,53,47,52,41,50,48,45,53,2E,41,58,53,43E
280 DATA 30,2E,41,58,53,57,41,56,45,2E,41,58,53,50,52,439
290 DATA 4F,47,52,41,4D,2E,41,58,53,00,00,00,00,00,290
300 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0

```



DATAS3 .BAS

```

1# DATAS3 AXYS
2#
3# MODE 2:AD=&C000:NL=98
4# FOR A=1 TO 103:B=0:FOR C=1 TO 15:READ A#
5# POKE AD,VAL("&*+A#"):B=B+PEEK(AD):AD=AD+1
6# NEXT:READ E#:IF VAL("&*+E#")<B THEN 8#
7# NL=NL+1#:NEXT:SAVE"SPRITE.AXS",B,&C000,&601:CLS:END
8# CLS:PRINT"LIGNE"NL;" INCORRECTE":END
9# DATA ED,73,86,21,ED,7B,3B,00,08,26,3F,0A,6F,1A,FD,5A7
10# DATA 6F,A6,B5,12,03,1C,0A,6F,1A,FD,67,FD,E5,A6,B5,72F
11# DATA 12,03,D5,1D,3E,08,02,57,E6,38,20,14,7A,D6,40,508
12# DATA 57,7B,C6,40,5F,30,0A,14,7A,E6,07,20,04,7A,D6,560
13# DATA 08,57,08,3D,20,C6,ED,73,3B,00,31,00,00,C9,ED,50C
14# DATA 73,E7,21,ED,7B,3B,00,08,26,3F,0A,6F,1A,FD,6F,58A
15# DATA A6,B5,12,03,1C,0A,6F,1A,FD,67,FD,E5,A6,B5,12,6D2
16# DATA 03,1C,0A,6F,1A,FD,6F,A6,B5,12,03,1C,0A,6F,1A,43D
17# DATA FD,67,FD,E5,A6,B5,12,03,D5,21,FD,07,19,7C,E6,02B
18# DATA 38,20,14,7C,D6,40,67,7D,C6,40,6F,30,0A,24,7C,531
19# DATA E6,07,20,04,7C,D6,08,67,EB,08,3D,20,AF,ED,73,631
20# DATA 3B,00,31,00,00,C9,ED,73,54,22,ED,7B,3B,00,08,486
21# DATA 26,3F,0A,6F,25,6E,24,1A,FD,6F,A6,B5,12,03,1C,4A7
22# DATA 0A,6F,25,6E,24,1A,FD,67,FD,E5,A6,B5,12,03,1C,61C
23# DATA 0A,6F,25,6E,24,1A,FD,6F,A6,B5,12,03,1C,0A,6F,4BB
24# DATA 25,6E,24,1A,FD,67,FD,E5,A6,B5,12,03,D5,21,FD,77A
25# DATA 07,19,7C,E6,38,20,14,7C,D6,40,67,7D,C6,40,6F,5D9

```

```

26# DATA 30,0A,24,7C,E6,07,20,04,7C,D6,08,67,EB,08,3D,4DC
27# DATA 20,A3,ED,73,3B,00,31,00,00,C9,ED,73,CB,22,ED,692
28# DATA 7B,3B,00,08,26,3F,0A,6F,1A,FD,6F,A6,B5,12,03,492
29# DATA 1C,0A,6F,1A,FD,67,FD,E5,A6,B5,12,03,1C,0A,6F,5FA
30# DATA 1A,FD,6F,A6,B5,12,03,1C,0A,6F,1A,FD,67,FD,E5,6EB
31# DATA A6,B5,12,03,1C,0A,6F,1A,FD,6F,A6,B5,12,03,1C,517
32# DATA 0A,6F,1A,FD,67,FD,E5,A6,B5,12,03,D5,21,FB,07,741
33# DATA 19,7C,E6,38,20,14,7C,D6,40,67,7D,C6,40,6F,30,602
34# DATA 0A,24,7C,E6,07,20,04,7C,D6,08,67,EB,08,3D,20,4CC
35# DATA 99,ED,73,3B,00,31,00,00,C9,ED,73,58,23,ED,7B,671
36# DATA 3B,00,08,26,3F,0A,6F,1A,FD,6F,A6,B5,12,03,1C,433
37# DATA 0A,6F,1A,FD,67,FD,E5,A6,B5,12,03,1C,0A,6F,1A,5F8
38# DATA FD,6F,A6,B5,12,03,1C,0A,6F,1A,FD,67,FD,E5,A6,777
39# DATA B5,12,03,1C,0A,6F,1A,FD,6F,A6,B5,12,03,1C,0A,47B
40# DATA 6F,1A,FD,67,FD,E5,A6,B5,12,03,1C,0A,6F,1A,FD,6EB
41# DATA 6F,A6,B5,12,03,1C,0A,6F,1A,FD,67,FD,E5,A6,B5,72F
42# DATA 12,03,D5,21,F9,07,19,7C,E6,38,20,14,7C,D6,40,584
43# DATA 67,7D,C6,40,6F,30,0A,24,7C,E6,07,20,04,7C,D6,596
44# DATA 08,67,EB,08,3D,20,03,ED,73,3B,00,31,00,00,C9,4D7
45# DATA ED,73,FE,23,ED,7B,3B,00,08,26,3F,0A,6F,25,6E,59D
46# DATA 24,1A,FD,6F,A6,B5,12,03,1C,0A,6F,25,6E,24,1A,480
47# DATA FD,67,FD,E5,A6,B5,12,03,1C,0A,6F,25,6E,24,1A,61C
48# DATA FD,6F,A6,B5,12,03,1C,0A,6F,25,6E,24,1A,FD,67,5A6
49# DATA FD,E5,A6,B5,12,03,1C,0A,6F,25,6E,24,1A,FD,6F,624
50# DATA A6,B5,12,03,1C,0A,6F,25,6E,24,1A,FD,67,FD,E5,61C
51# DATA A6,B5,12,03,1C,0A,6F,25,6E,24,1A,FD,6F,A6,B5,59D
52# DATA 12,03,1C,0A,6F,25,6E,24,1A,FD,67,FD,E5,A6,B5,61C
53# DATA 12,03,D5,21,F9,07,19,7C,E6,38,20,14,7C,D6,40,584
54# DATA 67,7D,C6,40,6F,30,0A,24,7C,E6,07,20,04,7C,D6,596
55# DATA 08,67,EB,08,3D,C2,03,23,ED,73,3B,00,31,00,00,483

```





560 DATA C9, ED, 73, A2, 24, ED, 7B, 3B, 00, 00, 26, 3F, 0A, 6F, 1A, 592
 570 DATA FD, 6F, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 67, FD, E5, A6, 777
 580 DATA B5, 12, 03, 1C, 0A, 6F, 1A, FD, 6F, A6, B5, 12, 03, 1C, 0A, 47B
 590 DATA 6F, 1A, FD, 67, FD, E5, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 6EB
 600 DATA 6F, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 67, FD, E5, A6, B5, 72F
 610 DATA 12, 03, 1C, 0A, 6F, 1A, FD, 6F, A6, B5, 12, 03, 1C, 0A, 6F, 435
 620 DATA 1A, FD, 67, FD, E5, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 6F, 6EB
 630 DATA A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 67, FD, E5, A6, B5, 12, 6D2
 640 DATA 03, D5, 21, F7, 07, 19, 7C, E6, 38, 20, 14, 7C, D6, 40, 67, 5D7
 650 DATA 7D, C6, 40, 6F, 30, 0A, 24, 7C, E6, 07, 20, 0A, 7C, D6, 00, 537
 660 DATA 67, EB, 00, 3D, C2, 09, 24, ED, 73, 3B, 00, 31, 00, 00, C9, 51B
 670 DATA ED, 73, 5C, 25, ED, 7B, 3B, 00, 08, 26, 3F, 0A, 6F, 1A, FD, 581
 680 DATA 6F, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 67, FD, E5, A6, B5, 72F
 690 DATA 12, 03, 1C, 0A, 6F, 1A, FD, 6F, A6, B5, 12, 03, 1C, 0A, 6F, 435
 700 DATA 1A, FD, 67, FD, E5, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 6F, 6EB
 710 DATA A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 67, FD, E5, A6, B5, 12, 6D2
 720 DATA 03, 1C, 0A, 6F, 1A, FD, 6F, A6, B5, 12, 03, 1C, 0A, 6F, 1A, 43D
 730 DATA FD, 67, FD, E5, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 6F, A6, 777
 740 DATA B5, 12, 03, 1C, 0A, 6F, 1A, FD, 67, FD, E5, A6, B5, 12, 03, 62F
 750 DATA 1C, 0A, 6F, 1A, FD, 6F, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 537
 760 DATA 67, FD, E5, A6, B5, 12, 03, D5, 21, F5, 07, 19, 7C, E6, 38, 75E
 770 DATA 20, 14, 7C, D6, 40, 67, 7D, C6, 40, 6F, 30, 0A, 24, 7C, E6, 5D7
 780 DATA 07, 20, 0A, 7C, D6, 00, 67, EB, 00, 3D, C2, AD, 24, ED, 73, 60F
 790 DATA 3B, 00, 31, 00, 00, C9, ED, 73, 2C, 26, ED, 7B, 3B, 00, 00, 492
 800 DATA 26, 3F, 0A, 6F, 1A, FD, 6F, A6, B5, 12, 03, 1C, 0A, 6F, 1A, 483
 810 DATA FD, 67, FD, E5, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 6F, A6, 777
 820 DATA B5, 12, 03, 1C, 0A, 6F, 1A, FD, 67, FD, E5, A6, B5, 12, 03, 62F
 830 DATA 1C, 0A, 6F, 1A, FD, 6F, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 537

840 DATA 67, FD, E5, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 6F, A6, B5, 72F
 850 DATA 12, 03, 1C, 0A, 6F, 1A, FD, 67, FD, E5, A6, B5, 12, 03, 1C, 596
 860 DATA 0A, 6F, 1A, FD, 6F, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 67, 582
 870 DATA FD, E5, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 6F, A6, B5, 12, 6DA
 880 DATA 03, 1C, 0A, 6F, 1A, FD, 67, FD, E5, A6, B5, 12, 03, 1C, 0A, 58E
 890 DATA 6F, 1A, FD, 6F, A6, B5, 12, 03, 1C, 0A, 6F, 1A, FD, 67, FD, 675
 900 DATA E5, A6, B5, 12, 03, D5, 21, F3, 07, 19, 7C, E6, 38, 20, 14, 62C
 910 DATA 7C, D6, 40, 67, 7D, C6, 40, 6F, 30, 0A, 24, 7C, E6, 07, 20, 5D2
 920 DATA 0A, 7C, D6, 00, 67, EB, 00, 3D, C2, 67, 25, ED, 73, 3B, 00, 5DE
 930 DATA 31, 00, 00, C9, ED, 73, 44, 26, ED, 7B, 3B, 00, E1, C1, 70, 679
 940 DATA 2D, 71, 3D, 20, F8, ED, 73, 3B, 00, 31, 00, 00, C9, ED, 73, 5E8
 950 DATA 61, 26, ED, 7B, 3B, 00, E1, C1, 70, 2D, 71, 2D, C1, 70, 2D, 665
 960 DATA 71, 3D, 20, F3, ED, 73, 3B, 00, 31, 00, 00, C9, ED, 73, 83, 639
 970 DATA 26, ED, 7B, 3B, 00, E1, C1, 70, 2D, 71, 2D, C1, 70, 2D, 71, 675
 980 DATA 2D, C1, 70, 2D, 71, 3D, 20, EE, ED, 73, 3B, 00, 31, 00, 00, 513
 990 DATA C9, ED, 73, AA, 26, ED, 7B, 3B, 00, E1, C1, 70, 2D, 71, 2D, 779
 1000 DATA C1, 70, 2D, 71, 2D, C1, 70, 2D, 71, 2D, C1, 70, 2D, 71, 3D, 604
 1010 DATA 20, E9, ED, 73, 3B, 00, 31, 00, 00, C9, ED, 73, D6, 26, ED, 6E7
 1020 DATA 7B, 3B, 00, E1, C1, 70, 2D, 71, 2D, C1, 70, 2D, 71, 2D, C1, 650
 1030 DATA 70, 2D, 71, 2D, C1, 70, 2D, 71, 2D, C1, 70, 2D, 71, 3D, 20, 563
 1040 DATA E4, ED, 73, 3B, 00, 31, 00, 00, C9, ED, 73, 07, 27, ED, 7B, 66F
 1050 DATA 3B, 00, E1, C1, 70, 2D, 71, 2D, C1, 70, 2D, 71, 2D, C1, 70, 645
 1060 DATA 2D, 71, 2D, C1, 70, 2D, 71, 2D, C1, 70, 2D, 71, 2D, C1, 70, 5F4
 1070 DATA 2D, 71, 3D, 20, DF, ED, 73, 3B, 00, 31, 00, 00, C9, ED, 73, 5CF
 1080 DATA 3D, 27, ED, 7B, 3B, 00, E1, C1, 70, 2D, 71, 2D, C1, 70, 2D, 642
 1090 DATA 71, 2D, C1, 70, 2D, 71, 2D, C1, 70, 2D, 71, 2D, C1, 70, 2D, 5F4
 1100 DATA 71, 2D, C1, 70, 2D, 71, 2D, C1, 70, 2D, 71, 3D, 20, DA, ED, 68D
 1110 DATA 73, 3B, 00, 31, 00, 00, C9, 00, 00, 00, 00, 00, 00, 00, 1A0

DATAS 4 .BAS

```
10 ' DATAS4 AXYS
20 '
30 MODE 2:AD=&C000:NL=90
40 FOR A=1 TO 35:B=0:FOR C=1 TO 15:READ A*
50 POKE AD,VAL("&"+A*):B=B+PEEK(AD):AD=AD+1
60 NEXT:READ E$:IF VAL("&"+E$)<>B THEN 80
70 NL=NL+10:NEXT:SAVE"MASK.AXS",B,&C000,&200:CLS:END
80 CLS:PRINT"LIGNE"NL;" INCORRECTE":END
90 DATA 00,02,04,06,08,0A,0C,0E,10,12,14,16,18,1A,1C,D2
100 DATA 1E,20,22,24,26,28,2A,2C,2E,30,32,34,36,38,3A,29A
110 DATA 3C,3E,40,42,44,46,48,4A,4C,4E,50,52,54,56,58,45E
120 DATA 5A,5C,5E,60,62,64,66,68,6A,6C,6E,70,72,74,76,61B
130 DATA 7B,7A,7C,7E,80,82,84,86,88,8A,8C,8E,90,92,94,7DA
140 DATA 96,98,9A,9C,9E,AA,A2,A4,A6,AB,AA,AC,AE,B0,B2,99C
150 DATA BA,B6,B8,BA,BC,BE,C0,C2,C4,C6,C8,CA,CC,CE,D0,B5E
160 DATA D2,D4,D6,D8,DA,DC,DE,E0,E2,E4,E6,EB,EA,EC,EE,D20
170 DATA F0,F2,F4,F6,F8,FA,FC,FE,01,03,05,07,09,0B,0D,7E9
180 DATA 0F,11,13,15,17,19,1B,1D,1F,21,23,25,27,29,2B,1B3
190 DATA 2D,2F,31,33,35,37,39,3B,3D,3F,41,43,45,47,49,375
200 DATA 4B,4D,4F,51,53,55,57,59,5B,5D,5F,61,63,65,67,537
210 DATA 69,6B,6D,6F,71,73,75,77,79,7B,7D,7F,81,83,85,6F9
220 DATA 87,89,8B,8D,8F,91,93,95,97,99,9B,9D,9F,A1,A3,8BB
230 DATA A5,A7,A9,AB,AD,AF,B1,B3,B5,B7,B9,BB,BD,BF,C1,A7D
240 DATA C3,C5,C7,C9,CB,CD,CF,D1,D3,D5,D7,D9,DB,DD,DF,C3F
250 DATA E1,E3,E5,E7,E9,EB,ED,EF,F1,F3,F5,F7,F9,FB,FD,E01
260 DATA FF,FF,AA,55,00,AA,AA,00,00,55,00,55,00,00,4FB
270 DATA 00,00,AA,AA,00,00,AA,AA,00,00,00,00,00,00,2AB
280 DATA 00,00,00,55,00,55,00,00,00,00,55,00,55,00,15A
290 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
300 DATA 00,00,00,00,00,AA,AA,00,00,AA,AA,00,00,00,2AB
310 DATA 00,00,00,00,00,00,AA,AA,00,00,AA,AA,00,00,00,2AB
320 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
330 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
340 DATA 00,00,00,00,00,00,00,00,00,55,00,55,00,00,AA
350 DATA 00,00,55,00,55,00,00,00,00,00,00,00,00,00,AA
360 DATA 00,00,00,00,00,00,00,00,00,00,55,00,55,00,AA
370 DATA 00,00,00,00,55,00,55,00,00,00,00,00,00,00,AA
380 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
390 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
400 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
410 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
420 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
430 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
```

DATAS 5.BAS

```
10 ' DATAS5 AXYS
20 '
30 MODE 2:AD=&C000:NL=90
40 FOR A=1 TO 129:B=0:FOR C=1 TO 15:READ A*
50 POKE AD,VAL("&"+A*):B=B+PEEK(AD):AD=AD+1
60 NEXT:READ E$:IF VAL("&"+E$)<>B THEN 80
70 NL=NL+10:NEXT:SAVE"SOUND.AXS",B,&C000,&783:CLS:END
80 CLS:PRINT"LIGNE"NL;" INCORRECTE":END
90 DATA C3,06,40,C3,2A,40,CD,1A,40,21,E3,44,87,16,00,542
100 DATA 5F,19,7E,32,07,44,23,4E,C3,1A,43,F5,3A,1F,44,496
110 DATA B7,20,07,32,0B,44,3C,32,1F,44,F1,C9,3E,0A,21,450
120 DATA B9,43,32,95,40,22,92,40,3A,06,44,B7,28,48,F5,597
130 DATA DD,21,C6,43,CD,84,40,22,73,40,32,81,40,DD,6E,60B
140 DATA 0A,DD,66,0B,DD,5E,0E,16,00,19,7E,FE,E1,20,04,551
150 DATA AF,32,06,44,F1,87,4F,06,00,21,BD,43,09,44,4D,4B3
160 DATA 0A,03,16,00,5F,21,AF,43,19,11,00,00,73,23,72,2C7
170 DATA 0A,21,AF,43,16,00,5F,19,3E,00,77,21,04,44,7E,347
180 DATA 32,B6,43,23,7E,CB,3F,32,B5,43,21,B9,43,16,0A,53D
190 DATA AF,5E,06,F4,ED,51,01,C0,F6,ED,49,ED,79,06,F4,892
200 DATA ED,59,01,80,F6,ED,49,ED,79,2B,15,F2,97,40,C9,82B
210 DATA AF,DD,77,20,DD,7E,1A,B7,28,07,3D,DD,77,1A,C3,6EC
220 DATA 0E,41,DD,6E,12,DD,66,13,DD,5E,30,16,00,19,7E,59A
230 DATA FE,E1,CA,8B,41,FE,E0,20,0F,23,7E,E6,3F,16,00,75E
240 DATA 5F,DD,6E,12,DD,66,13,19,7E,FE,E2,20,0C,AF,DD,741
250 DATA 77,0E,3C,DD,77,17,23,1C,18,D7,FE,E9,20,18,23,59C
260 DATA 7E,CB,3F,32,BA,43,3E,0D,32,95,40,01,BC,43,ED,5F6
270 DATA 43,92,40,23,1C,1C,18,8B,FE,E7,20,1C,23,7E,26,52B
280 DATA 00,54,5D,6F,01,03,45,29,09,5E,23,56,EB,AF,DD,4E9
290 DATA 77,30,DD,75,12,DD,74,13,18,9B,FE,E8,20,0D,23,658
300 DATA 7E,DD,77,1A,1C,1C,DD,73,30,C3,B8,40,FE,E4,20,781
310 DATA 0E,AF,DD,77,2A,23,7E,DD,77,20,23,1C,1C,18,2D,4F0
320 DATA FE,E5,20,0B,DD,36,2A,01,1C,DD,73,30,C3,B8,40,6A3
330 DATA FE,E6,20,0B,DD,36,2A,02,23,1C,18,12,FE,E3,20,5B5
340 DATA 0E,1C,1C,1C,23,7E,DD,77,1B,23,7E,DD,77,1C,23,4A6
350 DATA 7E,DD,77,2B,1C,DD,73,30,DD,7E,19,B7,28,06,3D,62F
360 DATA DD,77,19,18,43,DD,35,17,20,3E,DD,7E,18,DD,77,616
370 DATA 17,DD,6E,0A,DD,66,0B,DD,5E,0E,16,00,19,7E,FE,5AE
380 DATA E8,20,0D,DD,34,0E,DD,34,0E,23,7E,DD,77,19,18,579
390 DATA CB,FE,E1,28,16,FE,E0,20,0B,23,7E,E6,3F,D6,05,792
400 DATA DD,77,0E,18,CF,7E,DD,77,2D,DD,34,0E,DD,7E,2B,6ED
410 DATA B7,FA,EA,41,DD,86,08,DD,86,2C,E6,7F,87,4F,16,827
420 DATA 00,5F,21,23,44,19,5E,23,58,79,32,AA,42,ED,53,4AE
430 DATA B5,42,26,00,DD,6E,21,01,9B,43,09,4E,23,46,23,44B
440 DATA 5E,23,56,DD,7E,2A,B7,20,0A,3A,04,44,AA,A2,32,533
450 DATA 04,44,18,30,FE,01,20,24,3A,04,44,B1,A2,32,04,30E
460 DATA 44,DD,7E,08,DD,77,20,DD,7E,2B,B7,F2,40,42,E6,7B2
470 DATA 7F,DD,77,20,18,10,DD,86,20,DD,77,20,18,08,3A,56C
480 DATA 04,44,AA,B3,32,04,44,DD,7E,20,B7,28,07,66,1F,57B
490 DATA EE,1F,32,05,44,DD,7E,1E,B7,28,06,3D,DD,77,1E,595
500 DATA 10,51,DD,7E,1C,07,47,DD,4E,1D,DD,7E,2E,B7,F2,72B
510 DATA 7C,42,CB,47,20,23,CB,6F,20,0F,79,DD,96,1B,4F,5D2
520 DATA 30,15,DD,CB,2E,EE,0E,00,18,0D,79,DD,86,1B,4F,502
530 DATA 8B,3B,05,DD,CB,2E,AE,48,DD,71,1D,CB,38,79,90,730
```



```

540 DATA 6F,26,00,30,01,25,3E,00,C6,A0,38,05,29,C6,18,3D3
550 DATA 30,FB,11,00,00,19,18,03,2A,85,42,DD,7E,2E,EE,508
560 DATA 01,DD,77,2E,DD,CB,09,6E,28,40,E5,DD,7E,1F,5F,6C8
570 DATA 0E,00,87,F2,D7,42,0D,CB,23,CB,11,CB,10,CB,23,670
580 DATA CB,11,CB,10,CB,23,CB,11,CB,10,CB,23,CB,11,CB,6F1
590 DATA 10,DD,7E,39,83,DD,77,39,30,01,03,DD,6E,3A,DD,64A
600 DATA 66,3B,09,DD,75,3A,DD,74,3B,EB,E1,B7,ED,52,DD,861
610 DATA 7E,2D,DD,96,3C,F2,15,43,AF,CB,3C,CB,1D,C9,87,7C2
620 DATA 28,7D,DD,21,C6,43,DD,77,08,3A,08,44,5F,87,87,5FB
630 DATA DD,77,21,7B,3C,32,06,44,AF,DD,77,16,DD,77,19,62E
640 DATA DD,77,1A,DD,77,1F,DD,77,39,DD,77,3A,DD,77,3B,78B
650 DATA DD,77,0E,DD,77,0F,DD,77,30,DD,77,31,DD,36,2E,70F
660 DATA 40,79,E6,1F,26,00,6F,11,F1,45,29,19,5E,23,56,4B3
670 DATA EB,7E,DD,77,17,DD,77,18,23,4E,23,7E,DD,77,1B,6C1
680 DATA 23,7E,DD,77,1C,DD,77,1D,23,7E,DD,77,1E,23,DD,695
690 DATA 75,0A,DD,74,0B,26,00,69,11,03,45,29,19,5E,23,386
700 DATA 56,DD,73,12,DD,72,13,C9,01,FE,08,F7,02,FD,10,6F0
710 DATA EF,04,FB,20,DF,01,00,00,00,00,00,E1,00,00,00,3CF
720 DATA 00,00,00,00,00,3F,00,00,00,00,00,0A,00,00,00,49
730 DATA 00,02,09,04,0A,00,00,00,00,00,00,00,00,00,21
740 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
750 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
760 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
770 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,01,1
780 DATA 02,03,04,05,06,07,08,09,0A,0B,0F,0F,0F,0F,0F,BC
790 DATA 0F,00,00,00,00,00,00,00,00,00,EE,0E,17,0E,4D,0D,18A
800 DATA 8E,0C,D9,0B,2F,0B,8E,0A,F7,09,67,09,E0,08,61,509
810 DATA 00,08,07,77,07,0B,07,A6,06,47,06,EC,05,97,05,40D
820 DATA 47,05,FB,04,B3,04,70,04,30,04,F4,03,8B,03,85,4E4
830 DATA 03,53,03,23,03,F6,02,CB,02,A3,02,7D,02,59,02,3C3
840 DATA 38,02,18,02,FA,01,DD,01,C2,01,A9,01,91,01,7B,4A7
850 DATA 01,65,01,51,01,3E,01,2C,01,1C,01,0C,01,FD,00,24C
860 DATA EE,00,E1,00,D4,00,C8,00,BD,00,B2,00,AB,00,9F,621
870 DATA 00,96,00,8E,00,86,00,7E,00,77,00,70,00,6A,00,379
880 DATA 64,00,5E,00,59,00,54,00,4F,00,4B,00,47,00,43,293
890 DATA 00,3F,00,3B,00,38,00,35,00,32,00,2F,00,2C,00,174
900 DATA 2A,00,27,00,25,00,23,00,21,00,1F,00,1D,00,1C,112
910 DATA 00,1A,00,19,00,17,00,16,00,15,00,13,00,12,00,9A
920 DATA 11,00,10,00,0F,00,06,00,10,01,1B,02,04,03,15,80
930 DATA 04,20,05,11,06,1A,07,06,08,09,09,30,0A,14,05,D4
940 DATA 0E,0B,01,0C,3B,0D,30,0E,21,45,37,45,42,45,4C,261
950 DATA 45,58,45,6C,45,79,45,87,45,8F,45,A0,45,A9,45,5C4
960 DATA 89,45,CA,45,D8,45,E9,45,E2,E5,97,E6,20,1E,1C,7F6
970 DATA 1A,18,16,14,12,10,0E,0C,0A,08,06,04,02,00,E1,197
980 DATA E2,E5,97,E6,20,1F,1E,1D,1C,E0,05,E2,E5,97,E6,803
990 DATA 10,0E,0C,0A,08,04,E2,E5,9F,E6,22,20,1E,E6,1A,5C4
1000 DATA 18,E0,05,E2,E6,20,1F,1E,1D,1C,1B,1A,19,18,17,3D8
1010 DATA 16,15,14,13,12,11,E0,02,E2,E5,93,E6,00,02,E5,57E
1020 DATA 95,91,8D,93,81,E1,E2,E5,97,E6,20,1C,18,14,10,764
1030 DATA 0C,08,04,00,E1,E2,E4,93,03,02,01,00,E1,E2,E6,601
1040 DATA 00,01,02,03,04,05,06,07,08,09,0A,0B,0C,E0,02,130
1050 DATA E2,E4,93,00,02,04,02,E0,03,E2,E6,00,00,00,00,50C
1060 DATA 04,04,04,04,07,07,07,07,E0,02,E2,E5,05,E6,12,3D2
1070 DATA 10,E4,03,0E,0C,E5,08,06,04,02,00,E1,E2,E5,83,535
1080 DATA E4,84,20,1F,1E,1D,E5,88,53,E0,02,E2,E6,0C,0C,664
1090 DATA 0B,0B,0B,0A,0A,0A,09,08,07,06,04,E0,02,E2,E9,30E
1100 DATA 0F,E4,93,00,00,E1,0F,46,21,46,39,46,51,46,68,4A1
1110 DATA 46,7E,46,B2,46,C3,46,CC,46,E2,46,F8,46,0E,47,6D8
1120 DATA 3D,47,53,47,75,47,01,00,00,00,00,0D,0F,0E,0D,212

```

```

1130 DATA 0C,0B,0A,08,06,04,02,00,E1,01,01,00,00,00,0D,125
1140 DATA 0F,0F,0E,0D,0C,0B,0A,09,08,07,06,05,04,03,02,86
1150 DATA 01,00,E1,01,02,00,00,00,0D,0F,0F,0E,0D,0C,0B,142
1160 DATA 0A,09,08,07,06,05,04,03,02,01,00,E1,01,03,00,11C
1170 DATA 00,00,0D,0F,0E,0D,0C,0B,0A,09,08,07,06,05,04,7F
1180 DATA 03,02,01,00,E1,03,04,00,00,00,0F,0E,0D,0C,0B,12F
1190 DATA 0A,09,08,07,06,05,04,03,02,01,00,E1,01,05,00,11E
1200 DATA 00,00,0D,0F,0F,0D,0D,0D,0D,0C,0C,0C,0B,0B,0B,AA
1210 DATA 0A,0A,0A,09,09,09,08,08,08,08,07,07,07,06,06,7A
1220 DATA 06,05,05,05,04,04,04,03,03,03,03,02,02,02,01,3A
1230 DATA 01,01,00,E1,01,06,00,00,00,0D,0F,0E,0D,0C,0A,137
1240 DATA 06,04,02,01,00,E1,01,07,00,00,00,0F,0D,00,E1,1F3
1250 DATA 04,00,00,00,00,0F,0E,0D,0C,0B,0A,09,08,07,06,75
1260 DATA 05,04,03,02,01,00,E1,01,09,00,00,00,0F,0E,0D,124
1270 DATA 0C,0B,0A,09,08,07,06,05,04,03,02,01,00,E1,04,133
1280 DATA 0A,00,00,00,0F,0E,0D,0C,0B,0A,09,08,07,06,05,78
1290 DATA 04,03,02,01,00,E1,01,0B,00,00,00,0D,0F,0F,0F,131
1300 DATA 0E,0D,0C,0B,0B,0B,0A,0A,0A,09,09,09,08,08,08,99
1310 DATA 07,07,07,06,06,06,05,05,05,04,04,04,03,03,03,4B
1320 DATA 02,02,02,01,01,01,00,E1,07,0C,00,00,00,0F,0E,11A
1330 DATA 0D,0C,0B,0A,09,08,07,06,05,04,03,02,01,00,E1,13C
1340 DATA 04,0D,10,20,00,0D,0E,0D,0E,0D,0E,0D,0E,0D,0D,C7
1350 DATA 0C,0C,0B,0B,0A,0A,09,09,08,08,07,06,05,04,03,7D
1360 DATA 02,01,00,E1,08,0E,02,05,00,10,10,10,10,10,161
1370 DATA 10,00,E1,00,00,00,00,00,00,00,00,00,00,00,00,F1

```

DATAS8 .BAS

```

10 ' DATAS8 AXYS
20 '
30 MODE 2:AD=&C000:NL=90
40 FOR A=1 TO 68:B=0:FOR C=1 TO 15:READ A#
50 POKE AD,VAL("&"+A#):B=B+PEEK(AD):AD=AD+1
60 NEXT:READ E#:IF VAL("&"+E#)<>B THEN 00
70 NL=NL+10:NEXT:SAVE"WAVE.AXS",B,&C000,&3FB:CLS:END
80 CLS:PRINT"LIGNE"NL;" INCORRECTE":END
90 DATA 64,1D,6E,1D,7C,1D,8A,1D,98,1D,A4,1D,B2,1D,C0,551
100 DATA 1D,CE,1D,D6,1D,DA,1D,DE,1D,E2,1D,F2,1D,02,1E,61B
110 DATA 12,1E,20,1E,2A,1E,19,04,2E,1E,2E,1E,2E,1E,2E,1E5
120 DATA 1E,0A,06,75,1E,8A,1E,75,1E,8A,1E,75,1E,8A,1E,3DF
130 DATA 19,06,F3,1E,F3,1E,F3,1E,F3,1E,F3,1E,F3,1E,19,69E
140 DATA 06,DB,1E,DB,1E,DB,1E,DB,1E,DB,1E,DB,1E,01,05,5E2
150 DATA 9F,1E,AB,1E,B7,1E,C3,1E,CF,1E,19,06,05,1F,26,492
160 DATA 1F,05,1F,26,1F,05,1F,26,1F,19,06,47,1F,47,1F,1DC
170 DATA 47,1F,47,1F,47,1F,47,1F,19,06,62,1F,62,1F,62,31B
180 DATA 1F,62,1F,62,1F,62,1F,19,03,7D,1F,7D,1F,7D,1F,392
190 DATA 19,01,9B,1F,01,01,BC,1F,01,01,CE,1F,19,07,E0,3A0
200 DATA 1F,E0,1F,E0,1F,E0,1F,E0,1F,E0,1F,E0,1F,1E,07,63E
210 DATA 19,20,19,20,19,20,19,20,19,20,19,20,19,20,1E,1AD
220 DATA 07,37,20,37,20,37,20,37,20,37,20,37,20,37,20,268
230 DATA 1E,06,D0,20,D0,20,D0,20,D0,20,D0,20,D0,20,01,5C5

```


DATAS 6A.BAS

```
240 DATA 04,55,20,70,20,94,20,C4,20,19,01,5B,1E,12,00,346
250 DATA 04,0F,09,21,1E,F0,0A,01,02,0A,FF,02,0A,01,02,270
260 DATA 0A,FF,02,0A,01,02,0A,FF,02,0A,01,02,0A,FF,02,33B
270 DATA 0A,01,02,0A,FF,02,0A,01,02,0A,FF,02,FF,2E,14,371
280 DATA 0E,41,13,21,19,E0,50,00,02,19,FF,00,19,01,FE,3FE
290 DATA 19,01,02,19,FF,00,D0,00,FE,10,00,04,10,15,21,35C
300 DATA 0E,EF,20,01,02,20,FF,02,20,01,02,20,FF,02,FF,484
310 DATA 10,00,04,10,15,21,2E,EF,20,FF,02,20,01,02,20,20B
320 DATA FF,02,20,01,02,FF,20,00,04,0F,10,21,04,F0,7F,403
330 DATA 00,02,FF,20,00,04,0F,19,21,11,F0,55,00,03,FF,3C6
340 DATA 20,00,04,0F,19,21,1E,F0,40,00,04,FF,20,00,04,2E2
350 DATA 0F,19,21,20,F0,55,00,03,FF,20,00,04,0F,19,21,328
360 DATA 30,F0,7F,00,02,FF,21,00,04,0E,10,21,1E,EF,20,44C
370 DATA 00,02,15,00,FD,1E,FF,02,3C,01,02,14,00,02,FF,307
380 DATA 20,00,04,0F,19,21,1E,F0,1C,FF,02,3A,01,02,22,2F7
390 DATA 00,02,FF,22,00,04,0F,1D,21,2E,F0,0A,00,02,0A,2A8
400 DATA FE,02,0A,FF,02,0A,00,02,0A,02,02,0A,01,02,0A,23C
410 DATA 00,02,19,FF,04,FF,22,00,04,0F,1D,21,0E,F0,0A,39B
420 DATA 00,02,0A,02,02,0A,01,02,0A,00,02,0A,FE,02,0A,13D
430 DATA FF,02,0A,00,02,19,01,04,FF,23,00,04,0F,1F,21,2A0
440 DATA 0E,E4,14,00,FC,14,01,FF,14,01,FE,14,00,FE,14,54F
450 DATA FE,FE,0A,00,FF,FF,24,00,04,0F,21,21,2E,E4,14,5A3
460 DATA 00,FC,14,FF,FF,14,FF,FE,14,00,FE,14,02,FE,0A,74F
470 DATA 00,FF,FF,2C,00,0C,0F,23,21,0A,E0,14,00,02,14,3A5
480 DATA 02,02,14,FE,02,14,02,02,14,FE,02,14,02,02,10,26C
490 DATA 00,02,FF,20,04,00,1F,25,21,37,E0,1A,FE,FF,1A,4E3
500 DATA 02,FF,1A,FE,FF,1A,02,FF,1A,FE,FF,1A,02,FF,1A,77F
510 DATA FE,FF,4A,00,FF,FF,20,06,00,1F,27,21,0E,E0,20,5F0
520 DATA 01,02,20,FF,02,3F,00,02,FF,20,06,00,1F,27,21,301
530 DATA 2E,E0,20,FF,FE,20,01,FE,3F,00,FE,FF,10,00,04,6A5
540 DATA 0F,29,21,1E,F0,0C,00,02,0C,FF,02,0C,01,02,0C,29D
550 DATA 01,FE,0C,FF,FE,0A,01,02,0A,FF,02,0A,01,02,0A,437
560 DATA FF,02,0A,01,02,0A,FF,02,0A,01,02,0A,FF,02,0A,33B
570 DATA 01,02,0A,FF,02,0A,01,02,FF,17,01,04,0D,2D,21,291
580 DATA 32,F0,19,00,03,19,00,FE,19,00,02,19,00,FF,19,3A1
590 DATA FF,02,19,01,02,34,00,02,FF,19,01,04,10,20,21,2CC
600 DATA 0C,EF,19,00,03,19,00,FE,19,00,02,19,00,FF,19,37A
610 DATA 01,02,19,FF,02,34,00,02,FF,1A,00,04,10,31,21,2D2
620 DATA 0C,EB,18,00,FE,06,01,FF,0C,01,FF,06,01,02,0C,434
630 DATA FF,02,0C,01,02,FF,1A,00,04,10,31,21,0C,E9,3C,3C0
640 DATA 00,FE,06,01,FF,0C,01,FF,06,01,02,0C,FF,02,0C,432
650 DATA 01,02,0C,FF,02,0C,01,02,0C,FF,02,FF,1A,00,04,349
660 DATA 10,31,21,0C,E7,00,00,FE,06,01,FF,0C,01,FF,06,4CB
670 DATA 01,02,0C,FF,02,0C,01,02,0C,FF,02,0C,01,02,0C,247
680 DATA FF,02,0C,01,02,0C,FF,02,0C,01,02,06,FF,02,FF,432
690 DATA 25,0A,00,1F,33,21,0C,E0,7F,00,FE,FF,1A,00,04,430
700 DATA 10,31,21,2E,0F,0F,00,FE,0F,FE,00,0F,00,FE,0F,4A5
710 DATA 02,00,0F,00,FE,0F,FE,00,0F,00,FE,0F,02,00,0F,349
720 DATA 00,FE,0F,FE,00,0F,00,FE,0F,02,00,0F,00,FE,0F,445
730 DATA FE,00,0F,00,FE,0F,02,00,FF,12,04,13,04,14,04,360
740 DATA 15,04,96,04,AE,FF,10,06,91,06,AE,FF,A1,FF,A2,6EE
750 DATA FF,A3,FF,A4,FF,AC,FF,A9,FF,A8,FF,9B,FF,99,FF,C70
760 DATA 17,04,98,04,9A,FF,25,0C,26,06,27,0C,A6,06,00,38C
770 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
```

```
10 ' DATAS6A AXYS
20 '
30 MODE 2:AD=&C000:NL=90
40 FOR A=1 TO 547:B=0:FOR C=1 TO 15:READ A#
50 POKE AD,VAL("&*+A#"):B=B+PEEK(AD):AD=AD+1
60 NEXT:READ E#:IF VAL("&*+E#")<>B THEN 00
70 NL=NL+1#:NEXT:SAVE"GRAPHESA",B,&C000,&2000:CLS:END
80 CLS:PRINT"LIGNE"NL;" INCORRECTE":END
90 DATA 3C,20,F0,0C,C0,00,50,F0,F0,E4,00,3C,3C,3C,20,710
100 DATA F0,A0,14,20,50,F0,E4,00,00,3C,3C,3C,20,F0,F0,6AC
110 DATA F0,00,14,3C,3C,3C,00,50,F0,F0,E4,14,3C,3C,3C,594
120 DATA 3C,3C,3C,00,3C,3C,3C,3C,00,F0,F0,00,14,3C,3C,410
130 DATA 20,00,3C,3C,0C,00,44,10,0C,04,04,04,00,4C,1EB
140 DATA 4C,18,40,0C,30,24,64,64,90,00,00,00,00,E4,00,440
150 DATA C0,C0,00,0C,04,04,00,64,10,04,10,90,00,00,00,3DC
160 DATA 30,40,54,FC,FC,10,E4,E4,90,64,CC,10,10,0C,04,69C
170 DATA 00,00,30,E4,90,64,E4,00,0C,0C,CC,0C,0C,4C,0C,65B
180 DATA 0C,0C,4C,CC,0C,40,40,C4,BE,FF,5D,7D,00,00,00,5FF
190 DATA 00,00,C0,40,4C,90,0C,CC,CC,64,CC,CC,CC,CC,0C,700
200 DATA 90,0C,4C,0C,0C,CC,C0,C0,04,04,4C,0C,0C,0C,40,594
210 DATA 04,40,C0,C0,04,0C,0C,0C,10,64,0C,00,CC,00,00,540
220 DATA 00,00,00,00,00,00,00,00,40,0C,40,FA,AE,E0,58,3EC
230 DATA 5D,F5,A0,00,00,00,C0,A0,CC,0C,00,00,00,00,00,3D2
240 DATA 00,00,00,00,00,00,00,00,04,C0,00,0F,0F,0F,0F,18C
250 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,CC,0C,15B
260 DATA 0F,0C,00,00,00,00,00,00,00,00,00,00,04,10,40,F7
270 DATA 0C,0C,0C,0C,0C,0C,00,00,00,00,E4,40,40,C0,00,2FC
280 DATA 00,00,00,00,00,00,00,00,00,00,00,00,44,A4,40,41,171
290 DATA C3,C3,10,4C,00,00,00,00,00,00,00,00,00,00,1EA
300 DATA 00,00,90,0C,00,C0,CC,F0,F0,F0,F0,CC,F0,F0,30,94C
310 DATA F0,0C,C0,40,4C,30,64,F0,00,F0,00,00,00,40,68C
320 DATA 01,40,40,00,00,00,00,00,00,00,00,00,00,00,00,00,81
330 DATA 04,0C,C0,C0,C0,04,04,0C,00,00,00,00,00,00,00,36C
340 DATA 00,00,00,00,00,00,00,A4,C0,00,00,4D,20,00,00,AE,3A7
350 DATA 20,00,00,51,20,00,00,52,20,00,00,53,20,00,00,396
360 DATA 54,20,00,00,55,20,00,00,56,20,00,00,59,20,00,00,378
370 DATA 00,5A,20,00,00,5B,20,00,00,5C,20,00,00,5D,10,3F6
380 DATA 04,C0,5D,10,04,00,5E,10,04,40,5E,10,04,00,5E,337
390 DATA 10,04,C0,5E,10,04,00,5F,10,04,40,5F,10,04,00,2EC
400 DATA 5F,10,04,C0,5F,10,04,00,61,10,04,C0,61,10,04,3D0
410 DATA 00,62,10,04,40,62,10,04,00,62,10,04,C0,62,10,354
420 DATA 04,00,63,10,04,40,63,10,04,00,63,10,04,C0,63,34C
430 DATA 10,04,00,64,10,04,40,64,10,04,00,64,10,04,C0,2FC
440 DATA 64,10,04,00,65,10,04,40,65,10,04,00,65,10,04,2A3
450 DATA C0,65,10,04,00,66,10,04,40,66,10,04,00,66,10,363
460 DATA 04,C0,66,10,04,00,67,10,04,40,67,10,04,00,67,35B
470 DATA 10,04,C0,67,10,04,00,6F,10,04,C0,6F,10,04,00,3D5
480 DATA 57,10,04,00,69,20,00,00,6A,20,00,00,6B,20,00,00,341
490 DATA 00,4C,20,00,00,69,04,02,00,69,12,02,04,69,0E,413
500 DATA 02,D0,69,0D,02,EA,69,07,02,F0,69,09,02,0A,6A,486
510 DATA 13,04,56,6A,06,04,6E,6A,07,04,8A,6A,09,04,AE,373
520 DATA 6A,09,04,D2,6A,09,04,F6,6A,00,04,16,6B,0A,04,3BB
530 DATA 3E,6B,0E,04,76,6B,09,04,9A,6B,10,04,DA,6B,0F,416
```

540 DATA 04, 16, 6C, 0F, 04, 52, 6C, 0F, 04, 8E, 6C, 0F, 04, CA, 6C, 3AD
550 DATA 0F, 04, 06, 6D, 0F, 04, 42, 6D, 10, 04, 82, 6D, 10, 04, C2, 321
560 DATA 6D, 0D, 04, F6, 6D, 10, 04, 36, 6E, 0F, 04, 72, 6E, 10, 04, 3A0
570 DATA B2, 6E, 10, 04, F2, 6E, 10, 04, 32, 6F, 10, 04, 72, 6F, 0F, 44D
580 DATA 04, AE, 6F, 0E, 04, 80, 71, 0F, 04, BC, 71, 0F, 04, F8, 71, 4E0
590 DATA 0F, 04, 34, 72, 1F, 08, 2C, 73, 1F, 08, 24, 74, 1F, 08, 1C, 281
600 DATA 75, 1F, 08, 14, 76, 1F, 08, 0C, 77, 17, 08, 80, 79, 0F, 06, 2FD
610 DATA DA, 79, 0F, 0C, 8E, 7A, 15, 0A, 60, 7B, 41, 0E, 00, 00, 00, 3BF
620 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 0
630 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 0
640 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 0
650 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 0
660 DATA 00, 00, 00, 00, 00, C3, 82, 02, 02, 02, A8, A8, 28, 28, 02, 02, 2ED
670 DATA C3, 82, 00, 00, 41, 00, 03, 00, 54, 00, 14, 00, 01, 00, C3, 2B5
680 DATA 82, 00, 00, C3, 82, 02, 02, 00, A8, 3C, 28, 02, 00, C3, 82, 41E
690 DATA 00, 00, C3, 82, 02, 02, 54, A8, 00, 28, 02, 02, C3, 82, 00, 3B6
700 DATA 00, 82, 00, 02, 02, A8, A8, 3C, 28, 00, 02, 00, 82, 00, 00, 2BE
710 DATA C3, 82, 02, 00, FC, A8, 00, 28, 02, 02, C3, 82, 00, 00, C3, 51F
720 DATA 82, 02, 00, FC, A8, 28, 28, 02, 02, C3, 82, 00, 00, C3, 82, 506
730 DATA 00, 02, 00, A8, 14, 00, 01, 00, 41, 00, 00, 00, C3, 82, 02, 247
740 DATA 02, FC, A8, 28, 28, 02, 02, C3, 82, 00, 00, C3, 82, 02, 02, 408
750 DATA A8, A8, 3C, 28, 00, 02, C3, 82, 00, 00, 41, 00, 02, 02, A8, 3E8
760 DATA A8, 3C, 28, 02, 02, 82, 82, 00, 00, C3, 00, 02, 02, FC, 00, 3D7
770 DATA 28, 28, 02, 02, C3, 00, 00, 00, C3, 82, 02, 02, A8, 00, 28, 330
780 DATA 00, 02, 02, C3, 82, 00, 00, C3, 00, 02, 02, A8, A8, 28, 28, 3B0
790 DATA 02, 02, C3, 00, 00, 00, C3, 82, 02, 00, FC, 00, 28, 00, 02, 334
800 DATA 00, C3, 82, 00, 00, C3, 82, 02, 00, FC, 00, 28, 00, 02, 00, 382
810 DATA 82, 00, 00, 00, C3, 82, 02, 00, A8, 00, 28, 28, 02, 02, C3, 388
820 DATA 82, 00, 00, 82, 82, 02, 02, FC, A8, 28, 28, 02, 02, 82, 82, 406
830 DATA 00, 00, C3, 82, 01, 00, 54, 00, 14, 00, 01, 00, C3, 82, 00, 2F4
840 DATA 00, 00, 82, 00, 02, 00, A8, 28, 28, 02, 02, C3, 82, 00, 00, 2C5
850 DATA 82, 82, 02, 02, FC, 00, 28, 28, 02, 02, 82, 82, 00, 00, 82, 3DE
860 DATA 00, 02, 00, A8, 00, 28, 00, 02, 00, C3, 82, 00, 00, 82, 82, 31D
870 DATA 03, 02, A8, A8, 28, 28, 02, 02, 82, 82, 00, 00, C3, 82, 02, 3F4
880 DATA 02, A8, A8, 28, 28, 02, 02, 82, 82, 00, 00, 41, 00, 02, 02, 2EF
890 DATA A8, A8, 28, 28, 02, 02, 41, 00, 00, 00, C3, 82, 02, 02, A8, 306
900 DATA A8, 3C, 28, 02, 00, 82, 00, 00, 00, C3, 82, 02, 02, A8, A8, 429
910 DATA 28, 28, 03, 02, C3, 82, 00, 82, C3, 82, 02, 02, A8, A8, 3C, 4F1
920 DATA 00, 02, 02, 82, 82, 00, 00, C3, 82, 02, 00, FC, A8, 00, 28, 41B
930 DATA 00, 02, C3, 82, 00, 00, C3, 82, 01, 00, 54, 00, 14, 00, 01, 2F6
940 DATA 00, 41, 00, 00, 00, 82, 82, 02, 02, A8, A8, 28, 28, 02, 02, 2ED
950 DATA C3, 82, 00, 00, 82, 82, 02, 02, A8, A8, 28, 28, 01, 00, 41, 42F
960 DATA 00, 00, 00, 82, 82, 02, 02, A8, A8, 28, 28, 03, 02, 82, 82, 3B1
970 DATA 00, 00, 82, 82, 02, 02, 54, 00, 14, 00, 02, 02, 82, 82, 27B
980 DATA 00, 82, 82, 02, 02, A8, A8, 14, 00, 01, 00, 41, 00, 00, 00, 2AE
990 DATA C3, 82, 00, 02, 54, 00, 14, 00, 02, 00, C3, 82, 00, 00, 00, 2F6
1000 DATA 00, 00, 00, 00, 00, 00, 00, 01, 00, 41, 00, 00, 00, 00, 00, 42
1010 DATA 00, 00, 00, 00, 00, 00, 00, 02, 00, 82, 41, 00, 41, 00, 01, 107
1020 DATA 00, 54, 00, 14, 00, 00, 00, 41, 00, 00, 00, 00, 82, 01, 00, 12C
1030 DATA 54, 00, 14, 00, 01, 00, 00, 82, 00, 00, 41, 00, 00, 02, 00, 12E
1040 DATA A8, 00, 28, 00, 02, 41, 00, 00, 00, 00, 00, 01, 00, 00, 00, 114
1050 DATA 00, 00, 01, 00, 00, 00, 00, 00, 41, 00, 01, 00, A8, 00, 00, EB
1060 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, C0, 80, C0, 80, C0, 80, 3C0
1070 DATA 80, 80, 08, 08, 08, 08, 20, 20, 20, 20, A0, A0, A0, A0, 28, 448
1080 DATA 28, 28, 28, A8, A8, A8, A8, 02, 02, 03, 02, C3, 82, C3, 82, 5AB
1090 DATA 00, 00, 40, 00, 40, 00, 40, 00, C0, 80, 0C, 00, 0C, 00, 10, 1A8
1100 DATA 00, 10, 00, 50, 00, 50, 00, 14, 00, 14, 00, 54, 00, 54, 00, 100
1110 DATA 01, 00, 03, 02, C3, 82, C3, 82, 00, 00, C0, 80, C0, 80, C0, 5D0
1120 DATA 80, 80, 80, 08, 08, 08, 08, 20, 00, 20, 00, A0, F0, A0, 410

1130 DATA 3C, 28, 3C, 28, A8, 00, A8, 00, 02, 00, 03, 02, C3, 82, C3, 427
1140 DATA 82, 00, 00, C0, 80, C0, 80, C0, 80, 80, 08, 08, 08, 08, 562
1150 DATA 10, 20, 10, 20, 50, A0, 00, A0, 00, 28, 00, 28, A8, A8, A8, 438
1160 DATA A8, 02, 02, 03, 02, C3, 82, C3, 82, 00, 00, 80, 00, 80, 00, 43B
1170 DATA 80, 00, 80, 80, 00, 08, 08, 08, 20, 20, 20, 20, A0, A0, F0, 450
1180 DATA A0, 3C, 28, 3C, 28, 00, A8, 00, A8, 00, 02, 00, 02, 00, 82, 33E
1190 DATA 00, 82, 00, 00, C0, 80, C0, 80, C0, 80, 80, 00, 00, 00, 00, 4D2
1200 DATA 00, 30, 20, 30, 20, F0, A0, 00, A0, 00, 28, 00, 28, A8, A8, 470
1210 DATA A8, A8, 02, 02, 03, 02, C3, 82, C3, 82, 00, 00, C0, 80, C0, 5E3
1220 DATA 80, C0, 80, 80, 00, 08, 00, 08, 00, 30, 20, 30, 20, F0, A0, 480
1230 DATA A0, A0, 28, 28, 28, 28, A8, A8, A8, A8, 02, 02, 03, 02, C3, 54C
1240 DATA 82, C3, 82, 00, 00, C0, 80, C0, 80, C0, 80, 00, 00, 00, 08, 60F
1250 DATA 00, 08, 00, 20, 00, 20, 00, A0, 50, 00, 14, 00, 14, 00, 54, 1B4
1260 DATA 00, 54, 00, 01, 00, 01, 00, 41, 00, 41, 00, 00, 00, C0, 80, 218
1270 DATA C0, 80, C0, 80, 80, 80, 08, 08, 08, 08, 30, 20, 30, 20, F0, 530
1280 DATA A0, A0, A0, 28, 28, 28, 28, A8, A8, A8, A8, 02, 02, 03, 02, 529
1290 DATA C3, 82, C3, 82, 00, 00, C0, 80, C0, 80, C0, 80, 80, 80, 80, 752
1300 DATA 08, 08, 08, 20, 20, 20, 20, A0, A0, F0, A0, 3C, 28, 3C, 28, 430
1310 DATA 00, A8, 00, A8, 00, 02, 03, 02, C3, 82, C3, 82, 05, 0F, 05, 3FA
1320 DATA 0F, 05, 0F, 05, 0F, 05, 0F, 05, 0F, 05, 0F, 05, 0F, 05, 0F, 9B
1330 DATA 05, 0F, 05, 0F, 05, 0F, 05, 0F, 05, 0F, 05, 0F, 05, 0F, 05, 91
1340 DATA 0F, 05, 0F, 05, 0F, 05, 0F, 41, C3, 41, C3, 41, C3, 41, C3, 45B
1350 DATA 41, C3, 41, C3, 41, C3, 41, C3, 41, C3, 41, C3, 41, C3, 41, 75D
1360 DATA C3, 41, C3, 41, C3, 41, C3, 41, C3, 41, C3, 41, C3, 41, C3, 7DF
1370 DATA 41, C3, 01, 03, 01, 03, 01, 03, 01, 03, 01, 03, 01, 03, 01, 11D
1380 DATA 03, 01, 03, 01, 03, 01, 03, 01, 03, 01, 03, 01, 03, 01, 03, 1F
1390 DATA 01, 03, 01, 03, 01, 03, 01, 03, 01, 03, 01, 03, 01, 03, 54, FC, 18C
1400 DATA FC, 54, FC, 54, FC, 54, FC, 54, FC, 54, FC, 54, FC, 54, FC, A2C
1410 DATA 54, FC, 54, FC, 54, FC, 54, FC, 54, FC, 54, FC, 54, FC, 54, 984
1420 DATA FC, 54, FC, 54, FC, 54, FC, 14, 3C, 14, 3C, 14, 3C, 14, 3C, 62C
1430 DATA 14, 3C, 14, 3C, 14, 3C, 14, 3C, 14, 3C, 14, 3C, 14, 3C, 14, 244
1440 DATA 3C, 14, 3C, 14, 3C, 14, 3C, 14, 3C, 14, 3C, 14, 3C, 14, 3C, 26C
1450 DATA 14, 3C, 00, 00, 00, 00, 00, 00, 3C, 50, E4, 8C, 8C, C0, D0, 468
1460 DATA 64, 98, 30, 60, 14, 3C, 3C, 50, F0, F0, A4, 8C, F0, D8, CC, 80C
1470 DATA 8C, 0C, 14, 3C, 3C, 50, F0, D8, D8, F0, A0, 14, 3C, 00, F0, 6EA
1480 DATA F0, D8, F0, CC, 80, 3C, 3C, 3C, 3C, 3C, 28, F0, 14, 3C, 3C, 6D4
1490 DATA 28, F0, F0, 80, F0, 20, 14, 28, 50, F0, 14, 3C, 8C, 00, 44, 664
1500 DATA 10, 24, 04, 04, 04, 10, 00, 0C, CC, 18, 80, 84, 30, 24, 4C, 2EA
1510 DATA 64, 18, 80, 00, 00, 00, F0, 08, C0, C0, 00, 0C, 04, 04, 00, 390
1520 DATA 18, 04, 04, 04, CC, 20, 30, 00, 98, 48, 54, FC, A8, B0, F0, 588
1530 DATA 24, 98, 64, 98, 30, 04, 48, 04, 08, 08, 30, 98, 98, CC, A4, 518
1540 DATA 80, 8C, 48, 84, CC, 0C, C8, C0, 48, 0C, 4C, C0, 84, 0C, 04, 82C
1550 DATA C4, FA, AE, FF, F5, CC, 08, 00, 00, 00, C8, 40, 4C, CC, C0, 814
1560 DATA 84, CC, 8C, 48, 8C, 48, C0, C0, CC, 4C, C0, C0, 4C, C8, 80, 8A4
1570 DATA 00, C0, C0, 0C, 48, C0, 84, C0, C0, C0, C0, C0, 48, C0, 8A0
1580 DATA C4, 8C, 64, 8C, 80, CC, 00, 00, 00, 00, 00, 00, 00, 00, 38C
1590 DATA 00, 04, 48, 40, BE, AE, C8, 4C, 5D, 7D, 8B, 00, 00, 00, C8, 536
1600 DATA 40, 98, 0C, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, E4
1610 DATA 04, 8C, 80, 0F, 0F, 0F, 0A, 0C, 80, 00, 00, 00, 00, 00, 00, 1D3
1620 DATA 00, 00, 00, 00, 00, 00, CC, 0C, 0F, 98, 00, 00, 00, 00, 00, 17F
1630 DATA 00, 00, 00, 00, 00, 04, 18, 80, 84, 4C, 8C, 0C, 0C, 0C, 80, 29C
1640 DATA 00, 00, 00, F0, 08, C0, C8, 00, 00, 00, 00, 00, 00, 00, 280
1650 DATA 00, 00, 00, 00, 50, C8, 48, 41, C3, 82, B0, 24, 88, 00, 00, 3C2
1660 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 98, 0C, 80, C0, 0C, 1F0
1670 DATA 4C, F0, E4, 0C, C4, D8, 0C, C0, 84, 48, 80, 84, 4C, 64, 4C, 760
1680 DATA F0, 70, F0, 88, 00, 00, 00, 48, 41, 40, 40, 88, 00, 00, 00, 469
1690 DATA 00, 00, 00, 00, 00, 00, 00, 00, 8C, 48, 80, 00, 00, C0, C0, 2D4
1700 DATA 48, 58, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 50, A4, 194
1710 DATA 48, 80, 30, 30, 30, 30, 30, 30, 20, 71, 3F, F3, F3, F3, 5C1

1720 DATA 3F, F3, A2, 35, 71, F3, F3, B7, 7B, B2, A2, 35, FB, F3, 3F, 94B
 1730 DATA 3F, F3, F7, 2A, 35, 7B, B7, 7B, B7, 7B, B7, 2A, 71, 7B, 3F, 773
 1740 DATA FF, FB, 7B, 3F, A2, 71, 3F, 7F, AF, FF, 3F, 7B, A2, 35, 7B, 83F
 1750 DATA AF, AF, 5F, FB, F3, A2, 35, F3, AF, 5B, 5F, FB, F3, A2, 35, 9A3
 1760 DATA 7F, 5B, 1F, 5F, FF, F3, A2, 71, 7F, 5B, 1A, 0F, FF, F3, 2A, 77C
 1770 DATA 71, F7, B2, 1A, 7B, 5F, 7B, 2A, 71, FF, AF, 30, 5F, FF, BF, 81F
 1780 DATA A2, 71, AF, 3A, 34, 7B, 5F, BF, A2, 71, AF, 3F, 3C, 7B, 5F, 6E0
 1790 DATA FB, A2, 71, AF, 7B, 3F, F3, 5F, FB, A2, 35, AF, A7, 1F, 2F, 83F
 1800 DATA 5F, FB, A2, 35, AF, AF, 5B, 7B, 5F, AF, 2A, 35, FF, FF, 5B, 82B
 1810 DATA 5B, 5F, AF, 2A, 71, 7F, FF, F3, 5B, FF, 0F, A2, 71, 7F, FF, 86F
 1820 DATA F7, 0F, FF, 0F, A2, 35, 7F, AF, F3, 0F, FF, 0F, A2, 35, F3, 7F3
 1830 DATA AF, 0F, F7, AF, 0F, A2, 35, 3F, FF, 5F, 5F, AF, 0F, A2, 71, 717
 1840 DATA 3F, F7, FF, FF, 0F, 0F, 2A, 71, F3, 2F, FF, AF, 0F, 5B, 2A, 751
 1850 DATA 71, F3, 3F, 0F, 0F, 0F, 1F, A2, 71, F3, B7, 0F, 0F, 0F, 1F, 4F8
 1860 DATA A2, 71, 35, 7B, A7, 0F, 0F, 3A, A2, 71, BF, F3, F3, 0F, 0F, 69B
 1870 DATA F7, A2, 35, 7B, F3, F3, B7, 7B, F3, A2, 00, 00, 00, 00, 00, 6F6
 1880 DATA 00, 00, 00, 30, 30, 30, 30, 30, 30, 30, 20, 71, 3F, F3, F3, 406
 1890 DATA F3, 3F, F3, A2, 35, 71, F3, F3, B7, 7B, B2, A2, 35, FB, F3, 9FC
 1900 DATA 3F, 3F, F3, F7, 2A, 35, 7B, B7, 7B, B7, 7B, B7, 2A, 71, 7B, 773
 1910 DATA 3F, FF, FB, 7B, 3F, A2, 71, 3F, 7F, AF, FF, 3F, 7B, A2, 35, 803
 1920 DATA 7B, AF, AF, 5F, FB, F3, A2, 35, F3, AF, 5B, 5F, FB, F3, A2, 9E9
 1930 DATA 35, 7F, 5B, 1F, 5F, FF, F3, A2, 71, 7F, 5B, 1A, 0F, FF, F3, 787
 1940 DATA 2A, 71, F7, B2, 1A, 7B, 5F, 7B, 2A, 71, FF, AF, 30, 5F, FF, 78A
 1950 DATA BF, A2, 71, AF, 3A, 34, 7B, 5F, BF, A2, 71, AF, 3F, 3C, 7B, 740
 1960 DATA 5F, FB, A2, 71, AF, 7B, 3F, F3, 5F, FB, A2, 35, AF, A7, 1F, 86F
 1970 DATA 0F, 5F, FB, A2, 35, AF, B7, 5B, 5F, 5F, AF, 2A, 35, FF, 5B, 727
 1980 DATA 5B, FF, FF, AF, 2A, 71, 7F, 5B, F3, FF, FF, 0F, A2, 71, 7F, 90F
 1990 DATA 0F, A7, FF, FF, 0F, A2, 35, 7F, AF, 0F, F7, FF, 0F, A2, 35, 783
 2000 DATA F3, AF, 0F, 0F, AF, 0F, A2, 35, 3F, FF, 5F, 5F, AF, 0F, A2, 6B1
 2010 DATA 71, 3F, F7, FF, FF, 0F, 0F, 2A, 71, F3, 2F, FF, AF, 0F, 5B, 798
 2020 DATA 2A, 71, F3, 3F, 0F, 0F, 0F, 1F, A2, 71, F3, B7, 0F, 0F, 0F, 503
 2030 DATA 1F, A2, 71, 35, 7B, A7, 0F, 0F, 3A, A2, 71, BF, F3, F3, 0F, 6A8
 2040 DATA 0F, F7, A2, 35, 7B, F3, F3, B7, 7B, F3, A2, 00, 00, 00, 00, 705
 2050 DATA 00, 00, 00, 00, 30, 30, 30, 30, 30, 30, 30, 20, 71, 3F, F3, 313
 2060 DATA F3, F3, 3F, F3, A2, 35, 71, F3, F3, B7, 7B, B2, A2, 35, FB, 9FC
 2070 DATA F3, 3F, 3F, F3, F7, 2A, 35, 7B, B7, 7B, B7, 7B, B7, 2A, 71, 7EB
 2080 DATA 7B, 3F, FF, FB, 7B, 3F, A2, 71, 3F, 7F, 5F, 5F, 3F, 7B, A2, 759
 2090 DATA 35, 7B, FF, 5B, 5F, FB, F3, A2, 35, F3, FF, F3, 0F, FB, F3, A10
 2100 DATA A2, 35, 7F, FF, FB, A7, FF, F3, A2, 71, 7F, FF, AF, A7, FF, ACF
 2110 DATA F3, 2A, 71, F7, 5F, BA, 7B, 5F, 7B, 2A, 71, FF, 5F, 30, 5F, 77B
 2120 DATA FF, BF, A2, 71, AF, 3A, 34, 7B, 5F, BF, A2, 71, AF, 3F, 3C, 7C4
 2130 DATA 7B, 5F, FB, A2, 71, AF, 7B, 3F, F3, 5F, FB, A2, 35, AF, A7, 8CB
 2140 DATA 1F, 2F, 5F, FB, A2, 35, AF, B7, 5B, 7B, 5F, AF, 2A, 35, FF, 727
 2150 DATA 5B, 5B, 5B, 5F, AF, 2A, 71, 7F, 5B, 5B, 5B, FF, 0F, A2, 71, 66B
 2160 DATA 7F, 0F, 5B, 0F, FF, 0F, A2, 35, 7F, AF, 0F, 0F, FF, 0F, A2, 5D9
 2170 DATA 35, F3, AF, 0F, 5F, AF, 0F, A2, 35, 3F, FF, AF, FF, AF, 0F, 784
 2180 DATA A2, 71, 3F, F7, FF, FF, 0F, 0F, 2A, 71, F3, 2F, FF, AF, 0F, 7DF
 2190 DATA 5B, 2A, 71, F3, 3F, 0F, 0F, 0F, 1F, A2, 71, F3, B7, 0F, 0F, 54F
 2200 DATA 0F, 1F, A2, 71, 35, 7B, A7, 0F, 0F, 3A, A2, 71, BF, F3, F3, 6A8
 2210 DATA 0F, 0F, F7, A2, 35, 7B, F3, F3, B7, 7B, F3, A2, 00, 00, 00, 714
 2220 DATA 00, 00, 00, 00, 00, 30, 30, 30, 30, 30, 30, 20, 71, 3F, 220
 2230 DATA F3, F3, F3, 3F, F3, A2, 35, 71, F3, F3, B7, 7B, B2, A2, 35, 9F4
 2240 DATA FB, F3, 3F, 3F, F3, F7, 2A, 35, 7B, B7, 7B, B7, 7B, B7, 2A, 875
 2250 DATA 71, 7B, 3F, FF, FB, 7B, 3F, A2, 71, 3F, 7F, 5F, 5F, 3F, 7B, 728
 2260 DATA A2, 35, 7B, AF, 5B, 5F, FB, F3, A2, 35, F3, AF, F3, 5F, FB, 96F
 2270 DATA F3, A2, 35, 7F, 5B, F3, FF, FF, F3, A2, 71, 7F, 5B, A7, FF, A1B
 2280 DATA FF, F3, 2A, 71, F7, B2, 1A, FF, 5F, 7B, 2A, 71, FF, AF, 30, 8A2
 2290 DATA 5F, 5F, BF, A2, 71, AF, 3A, 34, 7B, 5F, BF, A2, 71, AF, 3F, 747
 2300 DATA 3C, 7B, 5F, FB, A2, 71, AF, 7B, 3F, F3, 5F, FB, A2, 35, AF, 860

2310 DATA A7, 1F, 2F, 5F, FB, A2, 35, AF, B7, 5B, 7B, 5F, AF, 2A, 35, 6CF
 2320 DATA FF, 5B, 5B, 5B, 5F, AF, 2A, 71, 7F, 5B, 5B, 5B, FF, 0F, A2, 6F9
 2330 DATA 71, 7F, 0F, 5B, 0F, FF, 0F, A2, 35, 7F, AF, 0F, 0F, FF, 0F, 5A8
 2340 DATA A2, 35, F3, AF, 0F, 5F, AF, 0F, A2, 35, 3F, FF, AF, FF, AF, 817
 2350 DATA 0F, A2, 71, 3F, F7, FF, FF, 0F, 0F, 2A, 71, F3, 2F, FF, AF, 7DF
 2360 DATA 0F, 5B, 2A, 71, F3, 3F, 0F, 0F, 0F, 1F, A2, 71, F3, B7, 0F, 54F
 2370 DATA 0F, 0F, 1F, A2, 71, 35, 7B, A7, 0F, 0F, 3A, A2, 71, BF, F3, 5C4
 2380 DATA F3, 0F, 0F, F7, A2, 35, 7B, F3, F3, B7, 7B, F3, A2, 00, 00, 807
 2390 DATA 00, 00, 00, 00, 00, 00, 30, 30, 30, 30, 30, 30, 75, 71, 236
 2400 DATA 3F, F3, F3, F3, 3F, F3, F7, 35, 71, F3, F3, B7, 7B, B2, F7, AAB
 2410 DATA 35, FB, F3, 3F, 3F, F3, F7, 7F, 35, 7B, B7, 7B, B7, 7B, 8D5
 2420 DATA 7F, 71, 7B, 3F, FF, FB, 7B, 3F, F7, 71, 3F, 7F, 5F, FF, 3F, 821
 2430 DATA 7B, F7, 35, 7B, AF, 5F, 5F, FB, F3, F7, 35, F3, AF, F7, 0F, 951
 2440 DATA FB, F3, F7, 35, 7F, 5B, 7F, 0F, FF, F3, F7, 71, 7F, 5B, 7F, 935
 2450 DATA A7, FF, F3, 7F, 71, F7, B7, 75, A7, 5F, 7B, 7F, 71, AF, B7, 983
 2460 DATA 75, 2F, 5F, BF, F7, 71, AF, B2, 7D, 7B, 5F, BF, F7, 71, AF, 888
 2470 DATA B2, 7D, 7B, 5F, FB, F7, 71, FF, FF, FF, FF, FF, FB, F7, 35, 88E
 2480 DATA AF, B7, 75, 7B, 5F, FB, F7, 35, AF, B7, 7F, 7B, 5F, AF, 7F, 8C9
 2490 DATA 35, FF, 5B, 7F, A7, 5F, AF, 7F, 71, 7F, 5B, F7, A7, FF, 0F, 839
 2500 DATA F7, 71, 7F, 0F, F7, 0F, FF, 0F, F7, 35, 7F, AF, 5F, 0F, FF, 7D1
 2510 DATA 0F, F7, 35, F3, AF, 5F, 5F, AF, 0F, F7, 35, 3F, FF, 5F, 5F, 7B1
 2520 DATA AF, 0F, F7, 71, 3F, F7, FF, FF, 0F, 0F, 7F, 71, F3, 2F, FF, 889
 2530 DATA AF, 0F, 5B, 7F, 71, F3, 3F, 0F, 0F, 0F, 1F, F7, 71, F3, B7, 699
 2540 DATA 0F, 0F, 0F, 1F, F7, 71, 35, 7B, A7, 0F, 0F, 3A, F7, 71, BF, 58A
 2550 DATA F3, F3, 0F, 0F, F7, F7, 35, 7B, F3, F3, B7, 7B, F3, F7, FF, AA3
 2560 DATA FF, FF, FF, FF, FF, FF, FF, 30, 30, 30, 30, 30, 30, 75, 88E
 2570 DATA 71, 3F, F3, F3, F3, 3F, F3, F7, 35, 71, F3, F3, B7, 7B, B2, A22
 2580 DATA F7, 35, FB, F3, 3F, 3F, F3, F7, 7F, 35, 7B, B7, 7B, B7, 7B, 915
 2590 DATA B7, 7F, 71, 7B, 3F, FF, FB, 7B, 3F, F7, 71, 3F, 7F, FF, FF, 939
 2600 DATA 3F, 7B, F7, 35, 7B, AF, FF, 5F, FB, F3, F7, 35, F3, AF, FF, A29
 2610 DATA 0F, FB, F3, F7, 35, 7F, 5B, FF, 0F, FF, F3, F7, 71, 7F, 1F, 909
 2620 DATA FF, A7, FF, F3, 7F, 71, F7, B7, FF, A7, 5F, 7B, 7F, 71, AF, A55
 2630 DATA 3A, FF, 2F, 5F, BF, F7, 71, AF, 3A, FF, 7B, 5F, BF, F7, 71, 8D7
 2640 DATA FF, FF, FF, FF, FF, FF, FB, F7, 71, FF, FF, FF, FF, FF, FB, F7, E4B
 2650 DATA 35, FF, FF, FF, FF, FF, FF, FB, F7, 35, AF, B7, FF, 7B, 5F, AF, 845
 2660 DATA 7F, 35, FF, 5B, FF, A7, 5F, AF, 7F, 71, 7F, 5B, FF, A7, FF, 931
 2670 DATA 0F, F7, 71, 7F, 0F, FF, 0F, FF, 0F, F7, 35, 7F, AF, FF, 0F, 789
 2680 DATA FF, 0F, F7, 35, F3, AF, FF, 5F, AF, 0F, F7, 35, 3F, FF, FF, 961
 2690 DATA 5F, AF, 0F, F7, 71, 3F, F7, FF, FF, 0F, 0F, 7F, 71, F3, 2F, 7E9
 2700 DATA FF, AF, 0F, 5B, 7F, 71, F3, 3F, 0F, 0F, 0F, 1F, F7, 71, F3, 6E1
 2710 DATA B7, 0F, 0F, 0F, 1F, F7, 71, 35, 7B, A7, 0F, 0F, 3A, F7, 71, 582
 2720 DATA BF, F3, F3, F3, 3F, F7, F7, 35, 7B, F3, F3, B7, 7B, F3, F7, B77
 2730 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, 5F, 5F, 5F, 5F, 5F, 5F, A91
 2740 DATA 5F, F7, F7, F7, F7, F7, F7, F7, F7, 7F, 7F, 7F, 7F, 7F, 7F, B11
 2750 DATA 7F, 7F, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 7F, 6F9
 2760 DATA 7F, 7F, 7F, 7F, 7F, 7F, 7F, 7F, 7F, 7F, 7F, 7F, 7F, 7F, B31
 2770 DATA 5F, 5F, 5F, 5F, 5F, 5F, 5F, 5F, 5F, 5F, 5F, 5F, 5F, 5B, 3A, 7B, 804
 2780 DATA 5F, 5B, 3A, 7B, 5F, 5B, 3E, 7B, 5F, 5B, 3E, 7B, 5F, 5B, 3E, 54D
 2790 DATA 7B, 5F, 5B, 3E, 7B, 5F, 5B, 3A, 7B, 5F, 5B, 3A, 7B, 5F, 5B, 586
 2800 DATA 3A, 7B, 5F, 5B, 3A, 7B, 5F, 5B, 3E, 7B, 5F, 5B, 3A, 7B, 5F, 565
 2810 DATA 5B, 3A, 7B, 5F, 5B, 3A, 7B, 5F, 5B, 3A, 7B, 5F, FF, FF, FF, 74A
 2820 DATA FF, 28, CC, 0C, 4C, 8C, 4C, C0, 0C, CC, 0C, 0C, 00, 3C, 3C, 5CB
 2830 DATA 50, F0, CC, F0, F0, E4, 8C, C0, 0C, 0C, 00, 3C, 3C, 50, 98, 79C
 2840 DATA CC, CC, D0, F0, F0, F0, E4, CC, F0, E4, F0, 8C, 3C, 3C, BA8
 2850 DATA 3C, 3C, 3C, 50, E4, A0, 14, 28, 50, F0, CC, CC, CC, D0, A0, 7E0
 2860 DATA 50, F0, F0, A0, 3C, 24, 0C, 44, 04, 30, 04, 04, 00, 18, 04, 3D8
 2870 DATA 0C, CC, B4, B0, B4, 98, 8C, 4C, 30, 0C, 80, 00, 00, 00, E4, 570
 2880 DATA 08, C0, A8, 04, 0C, 04, 04, 04, 18, 00, 0C, 04, 44, 20, 4C, 204
 2890 DATA 04, CC, C0, 01, 03, 50, 30, D8, CC, CC, 30, 98, 30, 04, 48, 5C8

2900 DATA 04, 00, 04, 30, 24, 18, CC, C8, C0, 0C, 00, 00, 00, 00, 3D4
 2910 DATA 00, 00, 00, 00, 00, C0, 18, 04, CC, BA, EE, FF, 75, 4C, 00, 518
 2920 DATA 00, 00, 00, 00, 40, 4C, 60, 80, 00, 00, 00, 00, 00, 1F4
 2930 DATA 00, 00, 00, 00, C0, CC, 48, 41, C3, 00, 40, 0C, 4B, 00, 00, 36C
 2940 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 44, 64, 8C, 41, 4C, 00, 1C1
 2950 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 04, 48, 40, BE, 2C, E0, 256
 2960 DATA 58, 1C, 7D, 00, 00, 00, 00, 8C, 40, 18, 0C, 00, 00, 00, 269
 2970 DATA 00, 00, 00, 00, 00, 00, 00, 00, 04, 8C, 00, 0F, 0F, 0F, 0A, 147
 2980 DATA 0C, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, CC, 158
 2990 DATA 8C, 0F, 30, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 04, 84, 153
 3000 DATA 00, 04, 4C, CC, 8C, 48, 48, 80, 00, 00, 00, E4, 00, CA, 48, 580
 3010 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 50, 0C, C0, 11C
 3020 DATA 01, 03, 50, 30, 8C, 00, 00, 00, 00, 00, 00, 00, 00, 00, 118
 3030 DATA 00, 00, 00, 98, 0C, C0, 40, 84, 8C, E4, 40, 8C, 0C, 4C, E0, 524
 3040 DATA 00, C0, C0, 00, 84, CC, 64, D8, CC, F0, E4, 88, 00, 00, 7B4
 3050 DATA 48, 41, 82, 40, D8, D0, F0, E4, CC, F0, F0, F0, F0, D8, 823
 3060 DATA CC, 8C, 48, 41, C3, C3, 00, 40, 84, 4C, F0, F0, 80, 70, F0, 867
 3070 DATA F0, F0, 80, F0, 70, F0, F0, C8, 48, 41, 30, 30, 30, 30, 811
 3080 DATA 30, 30, 75, 71, 3F, F3, F3, F3, 3F, F3, F7, 35, 71, F3, F3, 913
 3090 DATA B7, 7B, B2, F7, 35, FB, F3, 3F, 3F, F3, F7, 7F, 35, 7B, B7, 94C
 3100 DATA 7B, B7, 7B, B7, 7F, 71, 7B, 3F, FF, FB, 7B, 3F, F7, 71, 3F, 869
 3110 DATA 7F, FF, FF, 3F, 7B, F7, 35, 7B, AF, FF, FF, FB, F3, F7, 35, AA5
 3120 DATA F3, AF, FF, AF, FB, F3, F7, 35, 7F, 5B, FF, AF, FF, F3, F7, BDB
 3130 DATA 71, 7F, 1F, FF, AF, FF, F3, 7F, 71, F7, B2, FF, FB, 5F, 7B, A1C
 3140 DATA 7F, 71, AF, 3A, FF, FB, 5F, BF, F7, 71, FF, FF, FF, FF, B54
 3150 DATA BF, F7, 71, FF, FF, FF, FF, FF, FB, F7, 71, FF, FF, FF, D81
 3160 DATA FF, FB, F7, 35, FF, FF, FF, FF, FF, FB, F7, 35, FF, FF, D45
 3170 DATA FF, FF, AF, 7F, 35, FF, B7, FF, FB, 5F, AF, 7F, 71, 7F, 5B, 9E9
 3180 DATA FF, AF, FF, 0F, F7, 71, 7F, 5B, FF, AF, FF, 0F, F7, 35, 7F, 965
 3190 DATA AF, FF, AF, FF, 0F, F7, 35, F3, AF, FF, FF, AF, 5B, F7, 35, A6D

3200 DATA 3F, FF, FF, FF, AF, 5B, F7, 71, 3F, F7, FF, FF, 0F, 5B, 7F, 9CB
 3210 DATA 71, F3, 2F, FF, AF, 0F, F3, 7F, 71, F3, 3F, 0F, 0F, 0F, B7, 749
 3220 DATA F7, 71, F3, B7, A7, 0F, 1F, B7, F7, 71, 35, 7B, F3, F3, 3F, 8DB
 3230 DATA 3A, F7, 71, BF, F3, F3, F3, 3F, F7, F7, 35, 7B, F3, F3, B7, AB4
 3240 DATA 7B, F3, F7, FF, FF, FF, FF, FF, FF, FF, 30, 30, 30, 30, B1D
 3250 DATA 30, 30, 30, 75, 71, 3F, F3, F3, F3, 3F, F3, F7, 35, 71, F3, 850
 3260 DATA F3, B7, 7B, B2, F7, 35, FB, F3, 3F, 3F, F3, F7, 7F, 35, 7B, 988
 3270 DATA B7, 7B, B7, 7B, B7, 7F, 71, 7B, 3F, FF, FB, 7B, 3F, F7, 71, 8E1
 3280 DATA 3F, 7F, FF, FF, 3F, 7B, F7, 35, 7B, AF, FF, FF, FB, F3, F7, AAF
 3290 DATA 35, F3, AF, FF, AF, FB, F3, F7, 35, 7F, 5B, FF, AF, FF, F3, B19
 3300 DATA F7, 71, 7F, 1F, FF, AF, FF, F3, 7F, 71, F7, FF, FF, FF, FF, B89
 3310 DATA 7B, 7F, 71, FF, FF, FF, FF, FF, BF, F7, 71, FF, FF, FF, FF, C89
 3320 DATA FF, BF, F7, 71, FF, FF, FF, FF, FF, FB, F7, 71, FF, FF, FF, D81
 3330 DATA FF, FF, FB, F7, 35, FF, FF, FF, FF, FF, FB, F7, 35, FF, FF, D45
 3340 DATA FF, FF, FF, FB, 7F, 35, FF, FF, FF, FF, FF, FB, 7F, 71, 7F, C11
 3350 DATA FF, FF, FF, FF, 5B, F7, 71, 7F, 5B, FF, AF, FF, 5B, F7, 35, ACD
 3360 DATA 7F, AF, FF, AF, FF, 5B, F7, 35, F3, AF, FF, FF, AF, 5B, F7, B03
 3370 DATA 35, 3F, FF, FF, FF, AF, F3, F7, 71, 3F, F7, FF, FF, 0F, F3, AB1
 3380 DATA 7F, 71, F3, 2F, FF, AF, 5B, F3, 7F, 71, F3, 3F, 0F, 0F, F3, 841
 3390 DATA B7, F7, 71, F3, B7, F3, 3F, F3, B7, F7, 71, 35, 7B, F3, B7, A67
 3400 DATA 3F, 3A, F7, 71, BF, F3, F3, F3, 3F, F7, F7, 35, 7B, F3, F3, A3C
 3410 DATA B7, 7B, F3, F7, FF, FF, FF, FF, FF, FF, FF, FF, 30, 30, 30, BA4
 3420 DATA 30, 30, 30, 30, 75, 71, 3F, F3, F3, F3, 3F, F3, F7, 35, 71, 78D
 3430 DATA F3, F3, B7, 7B, B2, F7, 35, FB, F3, 3F, 3F, F3, F7, 7F, 35, A00
 3440 DATA 7B, B7, 7B, B7, 7B, B7, 7F, 71, 7B, 3F, FF, FB, 7B, 3F, F7, 8EB
 3450 DATA 71, 3F, 7F, FF, FF, 3F, 7B, F7, 35, 7B, FF, FF, FF, FB, F3, A79
 3460 DATA F7, 35, F3, FF, FF, FF, FB, F3, F7, 35, 7F, FF, FF, FF, FF, CB1
 3470 DATA F3, F7, 71, 7F, FF, FF, FF, FF, F3, 7F, 71, F7, FF, FF, FF, CAD
 3480 DATA FF, 7B, 7F, 71, FF, FF, FF, FF, FF, BF, F7, 71, FF, FF, FF, C89
 3490 DATA FF, FF, BF, F7, 71, FF, FF, FF, FF, FF, FB, F7, 71, FF, FF, D81
 3500 DATA FF, FF, FF, FB, F7, 35, FF, FF, FF, FF, FF, FB, F7, 35, FF, D45
 3510 DATA FF, FF, FF, FF, FB, 7F, 35, FF, FF, FF, FF, FF, FB, 7F, 71, C91
 3520 DATA 7F, FF, FF, FF, FF, 5B, F7, 71, 7F, FF, FF, FF, FF, 5B, F7, C0B
 3530 DATA 35, 7F, FF, FF, FF, FF, F3, F7, 35, F3, FF, FF, FF, AF, F3, CB1
 3540 DATA F7, 35, 3F, FF, FF, FF, FB, F3, F7, 71, 3F, F7, FF, FF, 5B, B4D
 3550 DATA F3, 7F, 71, F3, 2F, FF, AF, F3, F3, 7F, 71, F3, 3F, 0F, 5B, 925
 3560 DATA F3, B7, F7, 71, F3, B7, F3, 3F, F3, B7, F7, 71, 35, 7B, F3, AA3
 3570 DATA B7, 3F, 3A, F7, 71, BF, F3, F3, F3, 3F, F7, F7, 35, 7B, F3, A00
 3580 DATA F3, B7, 7B, F3, F7, FF, FF, FF, FF, FF, FF, FF, 30, 30, C67
 3590 DATA 30, 30, 30, 30, 30, 75, 71, 3F, F3, F3, F3, 3F, F3, F7, 35, 74C
 3600 DATA 71, F3, F3, B7, 7B, B2, F7, 35, FB, F3, 3F, 3F, F3, F7, 7F, A3C
 3610 DATA 35, 7B, B7, 7B, B7, 7B, B7, 7F, 71, 7B, 3F, FF, FB, 7B, 3F, 829
 3620 DATA F7, 71, 3F, 7F, AF, FF, 3F, 7B, F7, 35, 7B, AF, AF, 5F, FB, 8ED
 3630 DATA F3, F7, 35, F3, AF, 5B, 5F, FB, F3, F7, 35, 7F, 5B, 1F, 5F, 8ED
 3640 DATA FF, F3, F7, 71, 7F, 5B, 1A, 0F, FF, F3, 7F, 71, F7, B2, 1A, 902
 3650 DATA 7B, 5F, 7B, 7F, 71, FF, AF, 30, 5F, FF, BF, F7, 71, AF, 3A, 891
 3660 DATA 34, 7B, 5F, BF, F7, 71, AF, 3F, 3C, 7B, 5F, FB, F7, 71, AF, 84B
 3670 DATA 7B, 3F, F3, 5F, FB, F7, 35, AF, A7, 1F, 2F, 5F, FB, F7, 35, 85D
 3680 DATA AF, B7, 5B, 7B, 5F, AF, 7F, 35, FF, 5B, 5B, 5B, 5F, AF, 7F, 79B
 3690 DATA 71, 7F, 5B, 5B, 5B, FF, 0F, F7, 71, 7F, 0F, FF, AF, FF, 0F, 7C1
 3700 DATA F7, 35, 7F, AF, FF, AF, FF, 0F, F7, 35, F3, AF, FF, FF, AF, A91
 3710 DATA 0F, F7, 35, 3F, FF, FF, FF, AF, 0F, F7, 71, 3F, F7, FF, FF, 9D1
 3720 DATA 0F, 0F, 7F, 71, F3, 2F, FF, AF, 0F, 5B, 7F, 71, F3, 3F, 0F, 679
 3730 DATA 0F, 0F, 1F, F7, 71, F3, B7, 0F, 0F, 0F, 1F, F7, 71, 35, 7B, 583
 3740 DATA A7, 0F, 0F, 3A, F7, 71, BF, F3, F3, 0F, 0F, F7, F7, 35, 7B, 7C8
 3750 DATA F3, F3, B7, 7B, F3, F7, FF, FF, FF, FF, FF, FF, FF, 30, D2A
 3760 DATA 30, 30, 30, 30, 30, 30, 75, 71, 3F, F3, F3, F3, 3F, F3, F7, 747
 3770 DATA 35, 71, F3, F3, B7, 7B, B2, F7, 35, FB, F3, 3F, 3F, F3, F7, 9F2
 3780 DATA 7F, 35, 7B, B7, 7B, B7, 7B, B7, 7F, 71, 7B, 3F, FF, FB, 7B, 869

N'oubliez pas !!!

36 • 15

ARCADES

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

Voir bon de commande page



3790 DATA 3F, F7, 71, 3F, 7F, AF, FF, 3F, 7B, F7, 35, 7B, AF, AF, 5F, 831
 3800 DATA FB, F3, F7, 35, F3, AF, 5B, 5F, FB, F3, F7, 35, 7F, 5B, 1F, 989
 3810 DATA 5F, FF, F3, F7, 71, 7F, 5B, 1A, 0F, FF, F3, 7F, 71, F7, B2, 947
 3820 DATA 1A, 7B, 5F, 7B, 7F, 71, FF, AF, 30, 5F, FF, BF, F7, 71, FF, 8C1
 3830 DATA FF, 34, 7B, 5F, BF, F7, 71, FF, FF, 3C, 7B, 5F, FB, F7, 71, 9AB
 3840 DATA FF, FF, 3F, F3, 5F, FB, F7, 35, FF, FF, 1F, 2F, 5F, FB, F7, A53
 3850 DATA 35, FF, FF, 5B, 7B, 5F, AF, 7F, 35, FF, 5B, 5B, 5B, 5F, AF, 7E9
 3860 DATA 7F, 71, 7F, 5B, 5B, 5B, FF, 0F, F7, 71, 7F, 0F, 5B, 0F, FF, 6ED
 3870 DATA 0F, F7, 35, 7F, AF, 0F, 0F, FF, 0F, F7, 35, F3, AF, 0F, 5F, 6D1
 3880 DATA AF, 0F, F7, 35, 3F, FF, AF, FF, AF, 0F, F7, 71, 3F, F7, FF, 931
 3890 DATA FF, 0F, 0F, 7F, 71, F3, 2F, FF, AF, 0F, 5B, 7F, 71, F3, 3F, 769
 3900 DATA 0F, 0F, 0F, 1F, F7, 71, F3, B7, 0F, 0F, 0F, 1F, F7, 71, 35, 547
 3910 DATA 7B, A7, 0F, 0F, 3A, F7, 71, BF, F3, F3, 0F, 0F, F7, F7, 35, 7C8
 3920 DATA 7B, F3, F3, B7, 7B, F3, F7, FF, FF, FF, FF, FF, FF, FF, D75
 3930 DATA 30, 30, 30, 30, 30, 30, 30, 75, 71, 3F, F3, F3, F3, 3F, F3, 680
 3940 DATA F7, 35, 71, F3, F3, B7, 7B, B2, F7, 35, FB, F3, 3F, 3F, F3, 9F2
 3950 DATA F7, 7F, 35, 7B, B7, 7B, B7, 7B, B7, 7F, 71, 7B, 3F, FF, FB, 8E5
 3960 DATA 7B, 3F, F7, 71, 3F, 7F, FF, FF, 3F, 7B, F7, 35, 7B, AF, FF, 8ED
 3970 DATA FF, FB, F3, F7, 35, F3, AF, FF, FF, FB, F3, F7, 35, 7F, 5B, BAD
 3980 DATA FF, FF, FF, F3, F7, 71, 7F, 5B, FF, AF, FF, F3, 7F, 71, F7, BB9
 3990 DATA B2, 1A, 7B, 5F, 7B, 7F, 71, FF, AF, 30, 5F, FF, BF, F7, 71, 874
 4000 DATA AF, 3A, 34, 7B, 5F, BF, F7, 71, AF, 3F, 3C, 7B, 5F, FB, F7, 814
 4010 DATA 71, AF, 7B, 3F, F3, 5F, FB, F7, 35, AF, A7, 1F, 2F, 5F, FB, 851
 4020 DATA F7, 35, AF, B7, 5B, 7B, 5F, AF, 7F, 35, FF, 5B, 5B, 5B, 5F, 799
 4030 DATA AF, 7F, 71, 7F, 5B, 5B, 5B, FF, 0F, F7, 71, 7F, 0F, 5B, 0F, 69D
 4040 DATA FF, 0F, F7, 35, 7F, AF, 0F, 0F, FF, 0F, F7, 35, F3, AF, 0F, 771
 4050 DATA 5F, AF, 0F, F7, 35, 3F, FF, AF, FF, AF, 0F, F7, 71, 3F, F7, 891
 4060 DATA FF, FF, 0F, 0F, 7F, 71, F3, 2F, FF, AF, 0F, 5B, 7F, 71, F3, 829
 4070 DATA 3F, 0F, 0F, 0F, 1F, F7, 71, F3, B7, 0F, 0F, 0F, 1F, F7, 71, 551

4080 DATA 35, 7B, A7, 0F, 0F, 3A, F7, 71, BF, F3, F3, 0F, 0F, F7, F7, 7C8
 4090 DATA 35, 7B, F3, F3, B7, 7B, F3, F7, FF, FF, FF, FF, FF, FF, CAB
 4100 DATA FF, 0F, 0F, 0F, 5F, 5B, B7, 7B, 5F, 5B, 3E, 7B, 5F, 5B, 3E, 503
 4110 DATA 7B, 5F, 5B, 3E, 7B, 5F, 5B, 3A, 7B, 5F, 5B, 3A, 7B, 5F, 5B, 586
 4120 DATA 3A, 7B, 5F, 5B, 3A, 7B, 5F, 5B, 3A, 7B, 5F, 5B, 3E, 7B, 5F, 565
 4130 DATA 5B, 3E, 7B, 5F, 5B, 3A, 7B, 5F, 1F, 7B, F3, 5F, 0F, 0F, 4FB
 4140 DATA 5F, FF, FF, FF, FF, 00, 00, 00, 00, 00, 00, 00, 00, 00, 45B
 4150 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 0
 4160 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 0
 4170 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 0
 4180 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 50, 8C, 0C, 0C, 0C, 24C
 4190 DATA 8C, 4C, 8C, 0C, 0C, 48, 14, 3C, 50, 58, D8, F0, F0, E4, 4B, 6A0
 4200 DATA 00, 00, 04, 00, 14, 28, D0, 98, 30, CC, E4, CC, D8, CC, CC, 6CC
 4210 DATA CC, CC, D8, E4, D8, CC, 14, 3C, 3C, 3C, 28, F0, F0, F0, A0, 958
 4220 DATA 50, F0, CC, CC, CC, CC, CC, F0, F0, E4, F0, F0, 14, 64, 0C, A64
 4230 DATA 04, 04, 18, 44, 04, 00, 4C, 04, 0C, CC, C8, 80, 84, 8C, 0C, 3F4
 4240 DATA 8C, 98, 0C, 80, 00, 00, 00, E4, 80, 84, 0C, 10, 0C, 04, 04, 3C8
 4250 DATA 04, 0C, 00, 0C, 04, 10, 20, CC, 44, 8C, C0, 88, FC, 50, 30, 4B0
 4260 DATA CC, B0, 30, 30, 30, 18, 40, 48, 40, 80, 04, 18, 24, 30, CC, 4A8
 4270 DATA 8C, C0, 0C, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 230
 4280 DATA 04, 84, BE, 34, 8C, 7D, 4C, 00, 00, 00, 00, 88, 84, 4C, 60, 48F
 4290 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 154
 4300 DATA 05, 0F, 0F, 0A, 0C, 80, 00, 00, 00, 00, 00, 00, 00, 00, 00, B9
 4310 DATA 00, 00, 00, 30, 8C, 05, 4C, 00, 00, 00, 00, 00, 00, 00, 00, 10D
 4320 DATA 00, 00, 40, 20, C0, BE, E0, C8, 4C, D0, 7D, 00, 00, 00, 00, 527
 4330 DATA CC, 40, 18, C0, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 1E4
 4340 DATA 00, 40, 0C, 05, 0F, 0F, 0F, 0A, 0C, 80, 00, 00, 00, 00, 00, 114
 4350 DATA 00, 00, 00, 00, 00, 00, 00, CC, 8C, 05, 24, 00, 00, 00, 00, 181
 4360 DATA 00, 00, 00, 00, 00, 00, 04, 84, 80, 84, C0, 18, 0C, C8, 48, 380
 4370 DATA 80, 00, 00, 00, E4, 80, C4, 48, 00, 00, 00, 00, 00, 00, 00, 2F0

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.

LEAD		SELECTION	MODIFIER	RELEVES	BILAN		
LEAD		SUPPORT/SORTIE		CITERE			
SUPPORT / SORTIE							
C	DATE	TYPE	NO	ORDRE/OBJET	DEBIT	CREDIT	SOLDE
M	02/05/87	RELEVÉ		VERSSEMENT INITIAL		10000.00	10000.00
M	02/05/87	RELEVÉ		SALAIRE MOIS DE MAI		2500.00	12500.00
M	10/05/87	CHÉQUE	10000	MISTRAD CPC 6128	2000.00		10500.00
M	17/05/87	CHÉQUE	10001	CONSULTATION MEDECIN	110.00		10390.00
M	24/05/87	PATR. CAR		PHARMACIE	200.00		10190.00
M	31/05/87	VIREMENT		REBOURSEMENT S.S.		187.00	10377.00
M	11/05/87	PRELEVÉ		TELEPHONE	854.70		9522.30
M	11/05/87	RET. COIC		ELECTRICITE	562.47		8959.83
M	01/06/87	PRELEVÉ		RETRAIT EPARGNE		500.00	8459.83
M	02/06/87	RELEVÉ		RELEVÉ			8459.83
M	03/06/87	CHÉQUE	10002	SUPERMARCHÉ	562.47		7897.36
M	03/06/87	RET. CARTE		POINT ARGENT	300.00		7597.36
M	04/06/87	CHÉQUE	10003	ASSURANCE AUTO	350.00		7247.36
M	05/06/87	PATR. CAR		ESSENCE	250.00		7000.00
20/1	15 N 00	ANNUX TEST		No 10245678901	Occup. 1 X	Solde:	11027.94

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 votre autres produits pour CPC.

Je choisis de régler par :

CHEQUE C.B.

No Expire FIN

Signature

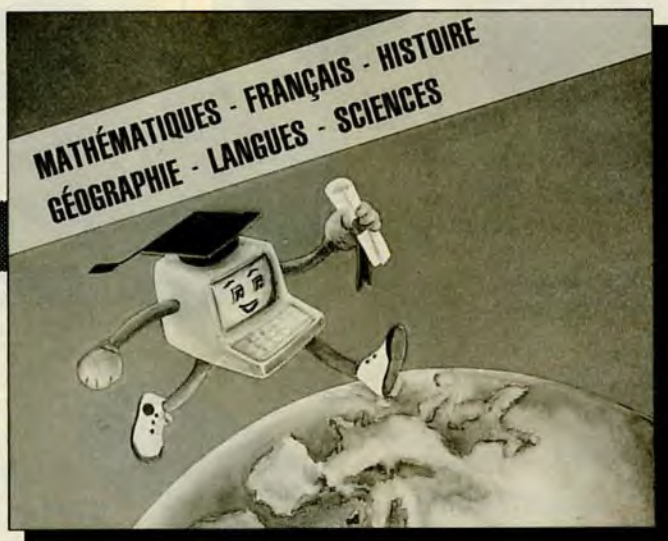
MANDAT CONTRE RBT (+35 Frs)

4380 DATA 00,00,00,00,00,50,24,C0,68,C3,50,30,C6,0B,00,3CF
 4390 DATA 00,00,00,00,00,00,00,00,00,00,00,E4,0C,C0,40,1F0
 4400 DATA C0,0C,0C,0C,C0,C0,0C,80,40,40,00,84,4B,DB,5A0
 4410 DATA E4,F0,F0,F0,DB,00,00,10,80,03,02,40,4C,F0,8C,729
 4420 DATA CC,48,84,DB,E4,E4,CC,4C,48,84,C0,01,03,03,03,6E6
 4430 DATA 02,84,0C,CC,E4,DB,CC,98,F0,E4,CC,CC,70,70,E4,9AE
 4440 DATA C8,C0,01,30,30,30,30,30,30,75,71,3F,F3,F3,5E4
 4450 DATA F3,3F,F3,F7,35,71,F3,F3,B7,7B,B2,F7,35,FB,F3,AA6
 4460 DATA 3F,3F,F3,F7,7F,35,7B,B7,7B,B7,7F,71,7B,81D
 4470 DATA 3F,FF,FB,7B,3F,F7,71,3F,7F,AF,FF,3F,7B,F7,35,8AD
 4480 DATA 7B,AF,AF,5F,FB,F3,F7,35,F3,AF,5B,5F,FB,F3,F7,A93
 4490 DATA 35,7F,5B,1F,5F,FF,F3,F7,71,7F,5B,1A,0F,FF,F3,7DC
 4500 DATA 7F,71,F7,B2,1A,7B,5F,7B,7F,71,FF,AF,30,5F,FF,834
 4510 DATA BF,F7,71,AF,3A,34,7F,FF,BF,F7,71,AF,3F,3C,7F,892
 4520 DATA FF,FB,F7,71,AF,7B,3F,F7,FF,FB,F7,35,AF,A7,1F,A5D
 4530 DATA 7F,FF,FB,F7,35,AF,B7,5B,7F,FF,AF,7F,35,FF,5B,9A1
 4540 DATA 5B,5B,5F,AF,7F,71,7F,5B,5B,5B,FF,0F,F7,71,7F,739
 4550 DATA 0F,5B,0F,FF,0F,F7,35,7F,AF,0F,0F,FF,0F,F7,35,639
 4560 DATA F3,AF,0F,5F,AF,0F,F7,35,3F,FF,AF,FF,AF,0F,F7,89B
 4570 DATA 71,3F,F7,FF,FF,0F,0F,7F,71,F3,2F,FF,AF,0F,5B,7ED
 4580 DATA 7F,71,F3,3F,0F,0F,0F,1F,F7,71,F3,B7,0F,0F,0F,5AD
 4590 DATA 1F,F7,71,35,7B,A7,0F,0F,3A,F7,71,BF,F3,F3,0F,752
 4600 DATA 0F,F7,F7,35,7B,F3,F3,B7,7B,F3,F7,FF,FF,FF,FF,BAB
 4610 DATA FF,FF,FF,FF,30,30,30,30,30,30,75,71,3F,F3,764
 4620 DATA F3,F3,3F,F3,F7,35,71,F3,F3,B7,7B,B2,F7,35,FB,AA6
 4630 DATA F3,3F,3F,F3,F7,7F,35,7B,B7,7B,B7,7F,71,895
 4640 DATA 7B,3F,FF,FB,7B,3F,F7,71,3F,7F,5F,FF,3F,7B,F7,8A3
 4650 DATA 35,7B,AF,5F,5F,FB,F3,F7,35,F3,AF,F7,0F,FB,F3,9CD
 4660 DATA F7,35,7F,FB,7F,F7,FF,F3,F7,71,7F,FF,75,FF,FF,867
 4670 DATA F3,7F,71,F7,F7,FF,FF,5F,7B,7F,71,AF,F7,FF,FB,839
 4680 DATA 5F,BF,F7,71,AF,7F,FF,FF,5F,BF,F7,71,AF,7F,FF,A65
 4690 DATA FF,5F,FB,F7,71,FF,FF,FF,FF,FF,FB,F7,35,AF,F7,C09
 4700 DATA FF,FF,5F,FB,F7,35,AF,F7,FF,FF,5F,AF,7F,35,FF,AE9
 4710 DATA 5F,FF,FF,5F,AF,7F,71,7F,5F,FF,FF,FF,0F,F7,71,9AD
 4720 DATA 7F,FB,7F,AF,FF,0F,F7,35,7F,AF,F7,A7,FF,0F,F7,9B3
 4730 DATA 35,F3,AF,5F,5F,AF,0F,F7,35,3F,FF,5F,5F,AF,0F,739
 4740 DATA F7,71,3F,F7,FF,FF,0F,A7,7F,71,F3,2F,FF,AF,0F,921
 4750 DATA 7B,7F,71,F3,3F,0F,0F,0F,3F,F7,71,F3,B7,0F,0F,639
 4760 DATA 1F,B7,F7,71,35,7B,A7,0F,5B,3A,F7,71,BF,F3,F3,046
 4770 DATA 0F,5B,F7,F7,35,7B,F3,F3,B7,7B,F3,F7,FF,FF,FF,807
 4780 DATA FF,FF,FF,FF,FF,30,30,30,30,30,75,71,F3,824
 4790 DATA F3,B7,7B,F3,F3,F7,35,7B,F3,F3,3F,F3,B7,7F,35,A35
 4800 DATA F3,F3,F3,F3,F3,3F,F7,71,0F,0F,0F,0F,0F,0F,F7,7B7
 4810 DATA 71,AF,0F,0F,0F,0F,5F,F7,71,FB,F3,F3,F3,F3,F7,8E1
 4820 DATA F7,71,5F,0F,0F,0F,0F,AF,F7,71,5F,0F,0F,0F,0F,4B5
 4830 DATA AF,7F,71,0F,FF,FF,FF,FF,0F,7F,35,0F,FB,F3,F3,95D
 4840 DATA F7,0F,F7,71,0F,AF,0F,0F,5F,0F,F7,71,0F,AF,0F,5ED
 4850 DATA 0F,5F,0F,F7,71,0F,AF,FF,FF,5F,0F,F7,71,0F,AF,735
 4860 DATA FF,FF,5F,0F,F7,71,0F,AF,FF,FF,5F,0F,F7,71,0F,875
 4870 DATA AF,FB,F7,5F,0F,F7,71,0F,AF,FB,F7,5F,0F,F7,35,8C1
 4880 DATA 0F,AF,F3,F3,5F,0F,7F,35,0F,AF,30,30,5F,0F,F7,649
 4890 DATA 71,0F,BF,3F,3F,7F,0F,F7,71,0F,BF,3F,3F,7F,0F,58D
 4900 DATA F7,71,0F,F3,F3,F3,F3,0F,F7,71,0F,30,30,30,30,789
 4910 DATA 0F,F7,71,1F,3F,3F,3F,3F,2F,7F,71,1F,3F,3F,3F,48D
 4920 DATA 3F,2F,7F,35,3F,3F,3F,3F,3F,F7,71,3F,3F,3F,4C1
 4930 DATA 3F,3F,3F,F7,71,B7,3F,F3,F3,B7,7B,F7,71,F3,B7,945
 4940 DATA 7B,F3,F3,7B,F7,71,F3,3F,F3,F3,B7,F7,FF,FF,FBF
 4950 DATA FF,FF,FF,FF,FF,FF,30,30,30,30,30,30,75,30,7EF
 4960 DATA 30,30,30,30,30,75,3A,30,30,30,30,30,30,5F,34E

4970 DATA 3A,30,30,30,30,30,30,5F,3F,30,30,30,30,25,30D
 4980 DATA 5F,3F,30,30,30,30,25,5F,3F,3A,30,30,30,30,34B
 4990 DATA 0F,5F,3F,3A,30,30,30,0F,5F,3F,3F,30,30,30,323
 5000 DATA 25,0F,5F,3F,3F,30,30,30,25,0F,5F,3F,3F,30,30,312
 5010 DATA 30,25,0F,5F,3F,3F,3A,30,30,0F,0F,5F,3F,3F,3A,310
 5020 DATA 30,30,0F,0F,5F,3F,3F,3F,30,25,0F,0F,5F,3F,3F,2EA
 5030 DATA 3F,30,25,0F,0F,5F,3F,3F,3F,3A,0F,0F,0F,5F,3F,2D3
 5040 DATA 3F,3F,7B,0F,0F,0F,5F,3F,3F,3F,F3,A7,0F,0F,5F,459
 5050 DATA 3F,3F,3F,F3,A7,0F,0F,5F,3F,3F,7B,F3,F3,0F,0F,5D1
 5060 DATA 5F,3F,3F,7B,F3,F3,0F,0F,5F,3F,3F,F3,F3,F3,A7,7B9
 5070 DATA 0F,5F,3F,3F,F3,F3,F3,A7,0F,5F,3F,3F,F3,F3,F3,831
 5080 DATA A7,0F,5F,3F,7B,F3,F3,F3,F3,0F,5F,3F,7B,F3,F3,8A9
 5090 DATA F3,F3,0F,5F,3F,F3,F3,F3,F3,A7,5F,3F,F3,F3,A7D
 5100 DATA F3,F3,F3,A7,5F,7B,F3,F3,F3,F3,F3,5F,7B,F3,BD9
 5110 DATA F3,F3,F3,F3,F3,5F,F3,F3,F3,F3,F3,F3,F7,FF,DB9
 5120 DATA FF,FF,FF,FF,FF,FF,FF,FF,CC,CC,EE,DD,99,9B,EE,77,CF5
 5130 DATA CD,9B,EE,DF,CD,33,EE,DF,CD,33,EE,DF,CD,33,EE,ABD
 5140 DATA DF,99,9B,EE,77,99,9B,EE,77,99,67,EE,77,99,67,976
 5150 DATA EE,77,99,67,EE,77,99,CF,EE,77,99,CF,EE,77,CD,A31
 5160 DATA 33,EE,DF,CD,33,EE,DF,FF,FF,FF,FF,CC,EE,CC,DD,C2C
 5170 DATA CD,EE,9B,DF,99,EE,9B,77,99,EE,33,77,99,EE,33,989
 5180 DATA 77,99,EE,33,77,99,EE,9B,77,99,EE,9B,77,99,EE,961
 5190 DATA 67,77,99,EE,67,77,CD,EE,67,DF,CD,EE,CF,DF,CD,A7A
 5200 DATA EE,CF,DF,CD,EE,33,DF,99,EE,33,77,FF,FF,FF,FF,B96
 5210 DATA CC,CC,CC,DD,CD,67,33,DF,CD,33,67,DF,FF,FF,FF,ACA
 5220 DATA FF,CC,CC,CC,DD,99,33,67,77,CD,33,33,DF,CD,33,8FC
 5230 DATA 33,DF,99,9B,33,77,99,9B,33,77,99,CF,33,77,CD,7AD
 5240 DATA CF,33,DF,CD,CF,33,DF,99,67,67,77,99,33,CF,77,87F
 5250 DATA FF,FF,FF,FF,CC,CC,CC,DD,99,67,67,77,99,CF,33,AB6
 5260 DATA 77,CD,9B,33,DF,99,9B,33,77,99,9B,33,77,99,67,7AD
 5270 DATA 33,77,99,67,33,77,99,33,9B,77,99,33,9B,77,99,6A9
 5280 DATA 33,CF,77,99,67,CF,77,FF,FF,FF,FF,99,9B,67,77,9CD
 5290 DATA CD,9B,33,DF,FF,FF,FF,FF,30,30,35,7F,71,B7,CC,97E
 5300 DATA DD,35,6E,9B,77,64,99,FF,FF,64,77,CC,DD,99,EE,998
 5310 DATA 9B,77,99,EE,67,77,CD,EE,67,77,CD,EE,33,DF,CD,9AA
 5320 DATA EE,33,DF,99,EE,33,DF,99,EE,67,DF,99,EE,67,DF,A33
 5330 DATA CD,EE,9B,77,99,EE,9B,77,FF,FF,FF,FF,3F,30,30,A01
 5340 DATA 75,CC,9D,F3,F7,33,66,3F,7F,FF,8B,9D,F7,CC,DD,A16
 5350 DATA 66,F7,CD,33,8B,DD,99,9B,EF,DD,99,9B,EF,DD,99,A8E
 5360 DATA 67,8B,DD,CD,67,8B,DD,CD,67,8B,DD,CD,CF,8B,DD,ACB
 5370 DATA 99,CF,8B,DD,CD,33,8B,DD,CD,33,EF,DD,FF,FF,FF,861
 5380 DATA FF,CC,EE,CC,DD,99,EE,9B,DF,CD,EE,9B,77,CD,EE,BEB
 5390 DATA 33,77,99,EE,33,DF,99,EE,67,DF,99,EE,CF,DF,99,9DE
 5400 DATA EE,CF,DF,CD,EE,67,77,99,EE,67,77,64,77,CC,DD,A1E
 5410 DATA 71,99,FF,FF,71,99,67,77,71,E6,CC,DD,71,F3,B7,A0B
 5420 DATA DD,FF,FF,FF,FF,CC,CC,EE,DD,CD,67,8B,DD,CD,33,C00
 5430 DATA 8B,DD,99,33,EF,DD,99,33,EF,DD,99,33,EF,DD,CD,A2D
 5440 DATA 33,8B,DD,CD,33,8B,DD,99,9B,8B,DD,99,9B,8B,DD,9FB
 5450 DATA 99,DF,66,F7,FF,8B,D9,F7,CD,9B,D9,F7,CC,CC,F3,C22
 5460 DATA F7,9D,F3,B7,F7,FF,FF,FF,FF,3F,3F,3F,7F,3F,3F,9EB
 5470 DATA 3F,7F,3F,3F,3F,7F,3F,3F,3F,7F,3F,3F,3F,7F,3F,481
 5480 DATA 3F,3F,7F,3F,3F,3F,7F,3F,3F,3F,7F,3F,3F,3F,7F,4B1
 5490 DATA 3F,3F,3F,7F,3F,3F,3F,7F,3F,3F,3F,7F,3F,3F,3F,471
 5500 DATA 7F,3F,3F,3F,7F,3F,3F,3F,7F,FF,FF,FF,FF,F3,F3,8D9
 5510 DATA F3,F7,F3,F3,F3,F7,F3,F3,F3,F7,F3,F3,F3,F7,F3,E4D
 5520 DATA F3,F3,F7,F3,F3,F3,F7,F3,F3,F3,F7,F3,F3,F3,F7,E4D
 5530 DATA F3,F3,F3,F7,F3,F3,F3,F7,F3,F3,F3,F7,F3,F3,F3,E49
 5540 DATA F7,F3,F3,F3,F7,F3,F3,F3,F7,F3,F3,F3,F7,FF,FF,E65
 5550 DATA FF,FF,FB,00,00,00,00,00,00,00,00,00,00,00,2EE

LE SPECIALISTE DE L'EDUCATIF

notre CATALOGUE des éducatifs



Les détails des logiciels les tarifs

catalogue gratuit

Je 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
editions

Media Box Disquettes 5 1/4

182x178x348 mm

Réf. 310.501.1

230 F + port

pour 70 disquettes, livré avec séparations et index



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 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

155 F + port

pour 11 Data Cartridges type 3M



Media Box Compact Disc Multi

148x177

x348 mm

Réf. 100.525.0

pour 23 CD simples ou 11 CD doubles



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



190 F + port

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)



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

Nouveau ! les mêmes en noir

Mini cassettes Réf: 310 . 620 . 7 95 F + PORT	Vidéo VHS (L) Réf: 310 . 623 . 8 155 F + PORT	Compact Disc réf: 310 . 621 . 4 125 F + PORT	C D Multi Réf: 310 . 622 . 1 190 F + PORT

Voir bon de commande page 23