

Micro  
Informatique  
Standards AMSTRAD

# CPC

infos

**TELECHARGEMENT :**  
**QUOI DE NEUF ?**

**PLUMPY :**  
**LA SUITE !**

M2604 - 30 - 22,00 F



Mensuel  
N° 30 - Mars 1991

La Haie 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 :

Guy DUBUS, Fabien FESSARD,  
Yannick GOUR, Emmanuel GUILLARD,  
Claude LE MOULLEC,  
Stéphane SAINT-MARTIN

## FABRICATION

Directeur de fabrication  
Edmond COUDERT  
Maquettiste  
Jean-Luc AULNETTE  
SOCIETE MAYENNAISE D'IMPRESSION 53100 MAYENNE

## 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  
Tél. 99.52.78.57 - Terminal E83

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.



# SOMMAIRE

## N°30

3

FANZ' INFOS

8

CHRONIQUE  
DU TELECHARGEMENT

11

PLUMPY (suite)

15

CIRCUS FOLIES

22

PRATIQUE  
DE L'ASSEMBLEUR

26

OPERATEURS LOGIQUES

36

VARIABLES LOCALES  
ET RECURSIVITE

41

ABONNEMENT

42

AXYS

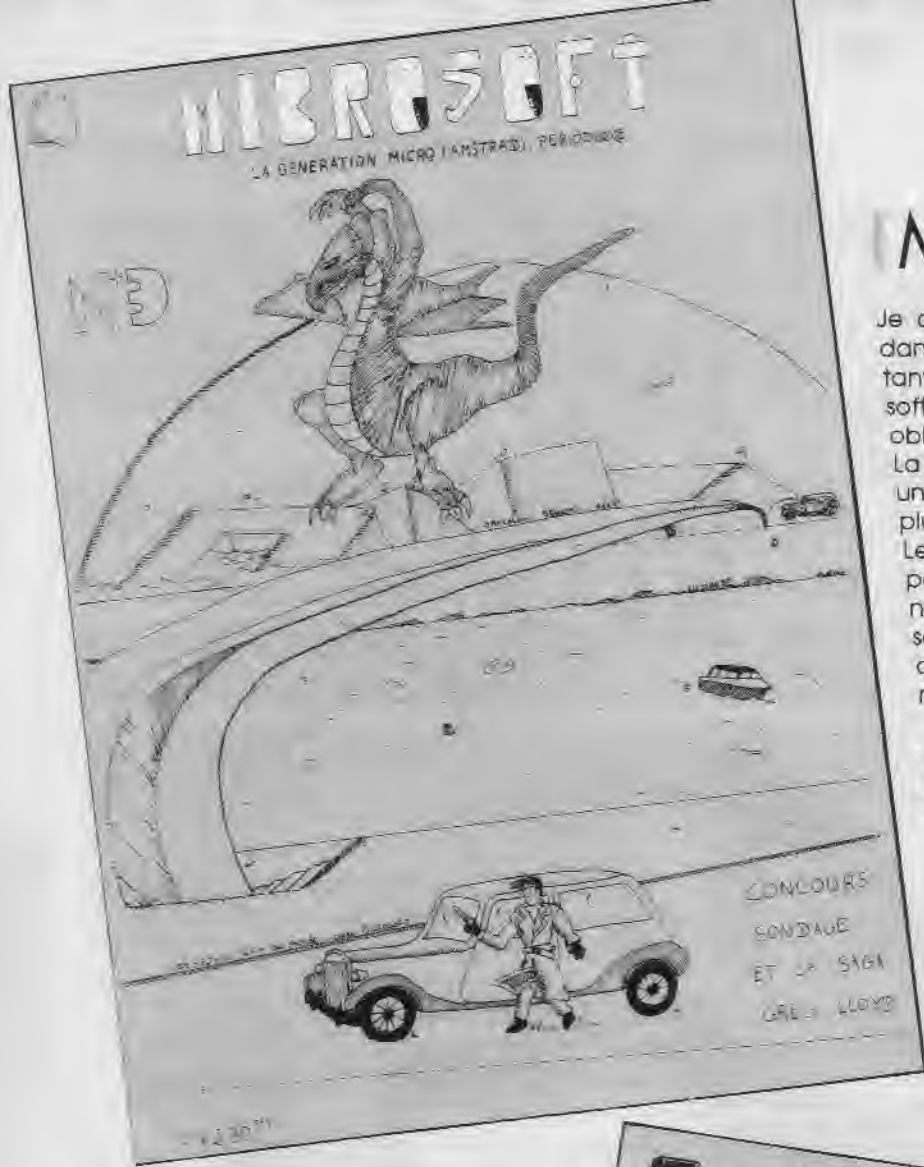


# FANZ'

## infos

Alors quoi, les fanzineux, on s'endort sur ses lauriers ? On ne veut pas sortir de chez soi ? On reste près de la télé à regarder le Club Dorothée et à se gaver d'Ultraman ? Où est-il l'heureux temps qui voyait chaque jour ou presque une avalanche de fanzines déferler sur mon bureau qui n'en demandait pas temps. Bref, il faut un peu se remuer pour m'envoyer les nouveaux numéros aussi bien sur disque que sur papier.





# MICROSOFT

Je crois déjà vous avoir causé de ce fanzine dans un numéro antérieur mais le doute subsistant. Je préfère ne pas m'abstenir. Donc Microsoft prend le parti de la non-PAO (peut-être par obligation d'ailleurs).

La fabrication utilise les services d'un PCW 9512, un Amstrad quand même, et se compose de plusieurs pages recto-verso au nombre de 15. Les tests de jeux sont assez imposants, par rapport aux concurrents bien sûr et le reste de ce numéro est consacré à un dossier sur Luc Beson, à un sondage et à diverses autres choses que vous pourrez bien découvrir par vous-même.

C. Duguay  
Bel-égo  
44850 St Mars du Desert



## THE AMAZING FANZINE

Encore un fanzine que vous devez déjà connaître mais comme les nouveautés ne se bousculent pas, il ne faut pas faire la fine bouche. On remarque tout de suite la compacité des textes et le délire ambiant qui ne facilitent pas toujours la lecture. Mais heureusement, chacun y trouve son bonheur : des tests de jeux, de démos (c'est à la mode), des bidouilles hard, des cours d'Assembleur, de Basic et des listings.

Thierry Prouteau  
71 bis, rue Parmentier  
49000 Angers  
OU  
Emmanuel Brillot  
12, square des Calledes  
49000 Angers.



# SOS FANZINES

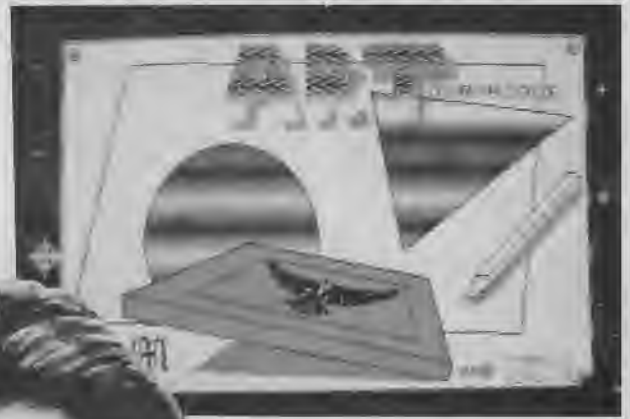
Attention, fanzine sur disque ! Et à mon avis, il est plutôt bon. La page de présentation est ornée d'une digi'te sobre et propre.

Chaque rubrique donne accès à un (trop souvent d'ailleurs) ou plusieurs écrans. Il est vrai que les pages présentées sont parfois superbes avec des écrans en Overscan et des rasters qui défilent.

Le seul regret donc concerne le manque de pages (je sais, c'est long à fabriquer mais on en veut toujours plus et il y a deux faces sur une disquette).

Sinon tentez l'expérience, ça vaut le coup d'oeil.

**SOS fanzines**  
2, Allée de Verdun  
93420 Villepinte



## LE PETIT ELECTRO-JACK ILLUSTRE

A titre spécial, contenu spécial. Bien que vous puissiez trouver dans ces pages quelques trucs pour des jeux, le reste du canard est consacré au Basic. Mais attention des programmes homéopathiques : c'est-à-dire qui ne dépassent pas 10 lignes. Certes l'intention est louable mais malgré tout on ne peut pas tout faire avec le Basic et l'Assembleur est irremplaçable pour certaines applications. Quand à s'abonner aux fanzines à la place des revues «du commerce» (voir édito du no 2), c'est sans doute une bonne idée mais quelle est la durée de vie moyenne d'un fanzine ?

Dernière petite chose, le programme 255 PAPER différents ne permet pas d'obtenir d'autres couleurs que les 16 normalement présentes (à comparer avec les 27 réellement affichables simultanément par un programme en Assembleur)

**Jacques de Lamar**  
3, avenue des Cosmonautes  
45400 Fleury les Aubrais

# GENERATION CPC



Encore un fanzine sur disque avec quantité de pages en Overscan et quelques effets graphiques. En revanche les textes n'occupent pas une place énorme et on peut faire le même reproche que pour SOS, les rubriques d'une seule page sont beaucoup trop courtes. Enfin c'est plutôt intéressant à regarder et à écouter car la musique est omni-présentée (comme pour SOS d'ailleurs). Pendant que j'y suis, j'en profite pour remercier H. Monchâtre qui m'a envoyé ce fanzine.

**Génération CPC**  
17, rue d'Arras  
62120 Aire sur la lys

# THE TWILIGHT STRAD

On termine avec un nouveau, qui ne l'est pas totalement d'ailleurs, car les auteurs ne sont autre que ceux qui avaient déjà commis AM-SING-MAG. Ce numéro 1 semble mieux «fini» que le fanzine précédent. Mais ce n'est pas encore tout à fait ça. Le choix des couleurs en mode 1 n'est pas toujours réussi. Le prochain numéro devrait être



mieux, je pense. D'autant que les digts sont réussies et que la présence des démos est un plus (j'adore les démos).

**Alexis Henaux**  
15, rue des Narcisses  
21300 Chenove





# chronique du TELECHARGEMENT



**Bonjour à tous : les débutants comme les initiés, vont pouvoir télécharger sur le 36-15 Arcades. Pour cela, il suffit de posséder un câble Arcades et le logiciel correspondant.**

**Vous pouvez commander câble et/ou logiciel grâce au bon qui se trouve quelque part dans les pages de CPC Infos.**



**A**ttention ce câble et ce logiciel sont les seuls capables d'utiliser le protocole du serveur. Pas la peine d'essayer le câble X «qui-marche-pourtant-très-bien-avec-le-serveur-Ycharge» ou le logiciel Truc qui télécharge sur SM-Chose.

Lorsque vous êtes en possession de ce petit matériel, il ne vous reste plus qu'à tester votre matériel. En effet, il arrive qu'avec certains Minitel, une fois le câble connecté, on ne puisse plus taper le code ARCADES. L'écran se remplissant de codes bizarres. La seule solution alors est de débrancher le câble et de ne le raccorder qu'au dernier moment. Sinon, vous procédez comme suit : vous allumez l'ordinateur et vous faites RUN «PRG1» avec la disquette de téléchargement dans le lecteur (si vous devez faire RUN «CPC c'est que vous possédez l'ancienne version). Vous allez voir apparaître un écran avec deux choix : Téléchar-

gement ou Mode d'emploi, choisissez Téléchargement et vous aurez ensuite la possibilité de définir le lecteur destination (1 ou 2). Puis la grille de téléchargement s'ouvrira à vos yeux éblouis. Bon, maintenant vous passez sur le minitel : 36-15 ARCADES. Ensuite sélection des options (choix de l'ordinateur et type d'opération à effectuer). Indiquez le numéro ou le nom du programme à télécharger. En effet, depuis peu, il est possible de télécharger un fichier par son nom, ce qui permet une certaine souplesse comme nous le verrons plus loin. Le minitel affiche ensuite le nombre total de Ko et le temps approximatif de téléchargement (environ 2 Ko par minute).

Le message «Lancez Arcades sur le micro» doit devenir pour vous, le signal pour le départ du téléchargement. Vous n'avez alors qu'à appuyer sur la barre d'espace (ou brancher le cordon puis appuyer sur



espace) pour commencer le téléchargement. Normalement, une barre de caractères doit s'afficher en haut de l'écran du Minitel et les numéros de blocs, noms de programmes et autres indications de téléchargement doivent s'afficher sur le CPC.

Pour le reste, lisez le mode d'emploi sur le 36-15 vous devriez y trouver ce que vous cherchez.

## LE FICHER CATA

Ce fichier est un outil indispensable pour le téléchargement. Il est composé en fait de deux fichiers : CATA, et CATALOG.DOC. Le premier contient les explications pour imprimer le second. Le tout se déroule sous CP/M (eh oui, c'est comme ça).

Ce catalogue est indispensable car il contient tous les fichiers présents sur le 36-15 avec indication de la taille et du numéro de la revue correspondante (si nécessaire) car n'oubliez pas que la plupart des programmes d'Arcades ont été publiés dans nos revues et que les modes d'emploi s'y trouvent éventuellement.

Le fait de pouvoir télécharger les fichiers par leur nom permet d'envisager le cas des coupures de liaison. Si votre téléchargement est coupé, il n'est plus nécessaire de reprendre à zéro ; vous regardez dans le catalogue le nom du fichier qui précédait la coupure et vous reprenez le téléchargement à ce niveau-là.

Il faut tout de même savoir que le chaînage entre les fichiers ne se fait plus alors automatiquement : il faudra entrer les noms de tous les fichiers suivants. De même, il n'est pas possible de télécharger seul le premier fichier appartenant à un groupe de fichier car le chaînage se fera automatiquement.

Avec le catalogue, vous pourrez ainsi prévoir le temps de téléchargement et la place à réserver sur votre disquette de destination.

Ce catalogue sera remis à jour à chaque nouvel arrivage sur le serveur. Vous serez tenu au courant par la revue et par la date indiquée près du fichier CATA.

En ce qui concerne les fichiers .MDM regardez le mode d'emploi sur le 36-15 Arcades.

Il est certain que ne paraîtront pas tous les mois les paragraphes précédents qui sont destinés à «mettre en condition» les usagers d'Arcades.

Arcades contient plus de 450 fichiers et il est certainement le serveur, non-dédié au jeu, le plus important de France. Au fil des mois vous verrez de nouveaux logiciels. Tous les 2 mois, il y aura les programmes parus dans CPC (ce rythme correspondant à la parution des disquettes CPC), vous trouverez aussi des programmes du Domaine Public. La plupart de ces programmes fonctionnant sous CP/M, il faudra vous munir de votre disquette système (celle livrée avec votre CPC). Il y a un autre problème : ces programmes sont tous en Anglais, il faudra donc faire un effort de traduction pour les utiliser. A moins qu'un lecteur compréhensif ne possède le temps nécessaire pour effectuer ces traductions, vous ne pourrez compter que sur la présentation succincte du programme dans le cadre de cette rubrique.

Voici donc les programmes que vous pourrez trouver en fin de mois sur le 36-15 ARCADES :

## LES PROGRAMMES DE CPC INFOS

On commence avec la disquette 33 comprenant les numéros 27 et 28 de CPC Infos. La prochaine disquette sera (exceptionnellement) décrite dans la rubrique du mois suivant.

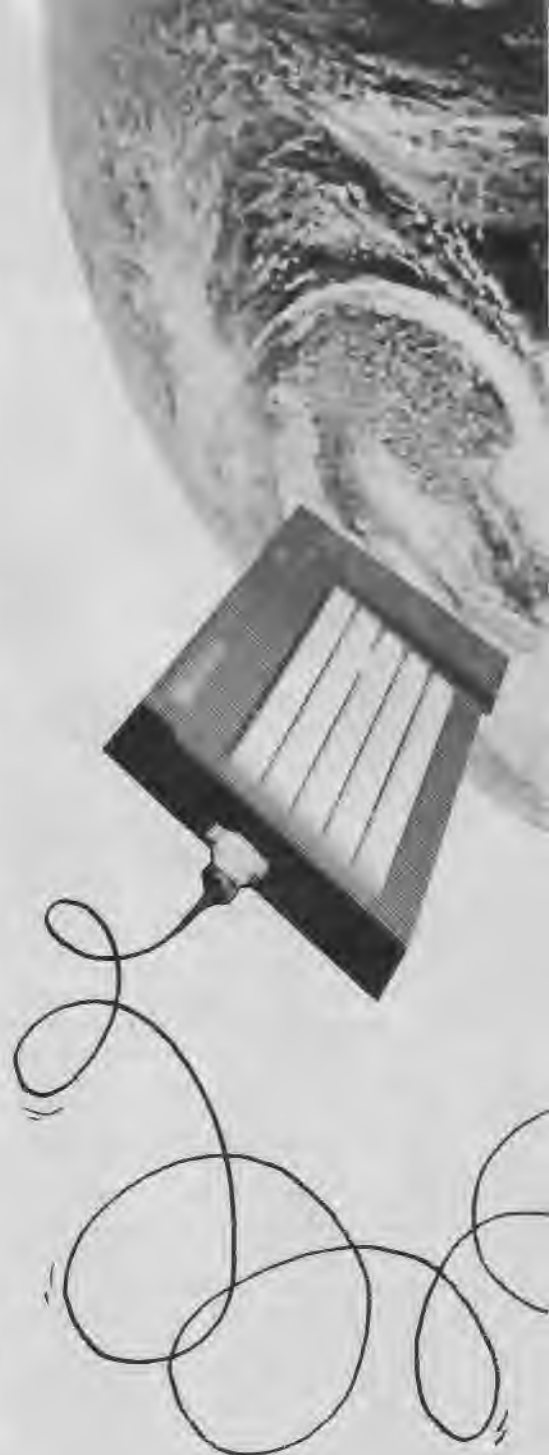
- **PLUMPY**, le célèbre jeu qui a posé tant de problèmes (ceux-ci sont résolus dans ce numéro) aux lecteurs.

- **VIRUS**, un autre jeu de Claude Le Moullec. Simple, en Basic et bien réalisé.

- **VARIABLES LOCALES**, un ensemble de RSX permettant d'obtenir la récursivité sur votre CPC et en Basic.

- **ROUTINES DU CCC**, 4 routines en Assembleur pour différents usages.

Malheureusement, pas tout à fait récentes mais quand on aime on ne compte pas. Donc remerciements à tous ceux qui m'ont envoyé des démos et particulièrement à Gozeur.



- **GHOST DEMO**, il faut être très patient pour lire le scroll et ne pas avoir de problèmes de vue.

- **MUDEBI** demo.

- **UGLYDEMO**, l'alliance de Digit (pour les programmes) et Ugly (pour les graphismes) donne une démo sympathique, sans grandes innovations.

- **ATOMIC DEMO 2**, en trois parties, le tout sans débauches graphiques ou techniques.

- **BAD DEMO**, rien à voir avec Michel.

- **KREATOR DEMO**, un seul écran mais bien réalisé.

• **BRAIN WAVE MEGA DEMO**, en deux parties par le célèbre (si,si) Gozeur.

• **RASTERS DEMO**, 4 parties + 1. Les 4 premières sont de Gozeur et contiennent pas mal d'effets avec les rasters. Il y a aussi la première démo de Gladateur, encourageante pour la suite.

Maintenant les programmes du domaine public sur CPC. Je rappelle que ces programmes sont en Anglais et que certains nécessitent CP/M ou CP/M+.

• **EASYDOS-DESKTOP**, est un utilitaire simulant un environnement graphique tel celui existant sur ST.

Vous trouverez donc des icônes pour les disquettes, pour la corbeille et pour les fichiers exécutables ou non. Il est possible d'utiliser une souris ou les touches du clavier pour déplacer le pointeur. Pour cela, il faut lancer le fichier CONFIG et sélectionner le clavier par exemple. Les menus déroulants se trouvent en haut de l'écran et permettent diverses opérations tels l'affichage des fichiers textes ou l'obtention d'informations sur les fichiers binaires.

Easydos simule également les sous-répertoires ou dossiers pour regrouper vos programmes par catégorie (il utilise en fait les numéros de USER). Pour en savoir plus, il faut lire le fichier DESKINFO (Anglais).

En supplément, vous trouverez deux programmes utilisables sous cet environnement : un formateur (FORMAT) et un «dé-effaceur» de fichiers : OOPS. Ces deux fichiers devront être installés en USER 1. Les noms des dossiers sont placés dans USERS.INF

• **KEYWORD**, permet d'avoir à sa disposition des instructions Basic accessibles par CTRL+ une touche.

Ainsi vous pourrez taper des listings normalement en abrégéant la frappe.

E - Edit, I - Input, S - Save », K - Load », W - Window, O - If, D - Cat, L - List, R - Run, P - Print, F - For, C - Cls, T - Then, A - Auto, G - Goto, M - Mode. Exemple : CTRL+P donnera Print à l'écran.

• **EDITOR**, est un éditeur de secteurs tout à fait classique.

• **BASIC +**, ajoute plusieurs commandes RSX à votre Basic. C'est surtout intéressant pour le 464, car les fonctions supplémentaires du 6128 sont implantées. Voici la liste de ces instructions que vous pouvez obtenir à l'écran avec ÛHELP :

ÛBOX,X,Y,L,H. Permet de tracer une trace une boîte de coordonnées X,Y (point en haut, à gauche), de largeur L et de hauteur H.

ÛCAPSOFF, Contraire de l'Instruction suivante.

ÛCAPSON, Positionne le clavier en mode majuscules

ÛCHARS, Affiche tout le jeu de caractères du CPC.

ÛCLEAR,INPUT, Vide le buffer clavier (comme sur 6128)

ÛCOLOUR,Border,paper,ink, Modifie la couleur du Border, du fond et de l'encre.

ÛCOLUNM,col, Positionne le curseur à la colonne col.

ÛCOPYCHAR,à chaîne

ÛCOPYCHAR,X,Y,à chaîne

ÛCURSOR.

ÛDEEK,adresse,ànn. Récupère le contenu de adresse et de adresse+1 et le place dans nn

ÛDISCREAD,plste,secteur,àa\$. Lit le contenu d'un secteur et place le contenu dans a\$. (Le secteur doit être donné sous forme hexa ; &C1 par exemple pour le format data).

ÛDISCWRITE,plste,secteur

ÛDOKE,adresse,nn. Poke une valeur 16 bits dans adresse et adresse+1

ÛDISARM,BRK, Empêche le Break d'un programme.

ÛFILLBOX,ink,gauche,droite,haut,bas. Trace un rectangle coloré avec ink et défini par gauche, droite, haut et bas.

ÛFINDMODE,àvariable entière

ÛFLUSH

ÛFRAME. Même fonction que pour le 6128 et CALL &BD19.

ÛGET,àA\$. Arrête le programme et attend l'appui sur une touche. La touche enfoncée est placée dans A\$

ÛGMODE,type. Sélectionne le mode graphique (0=normal, 1=XOR, 2=AND, 3=OR)

ÛGPAPER,encr. Détermine la couleur du fond en mode graphique.

ÛGPEN,encr. Détermine la couleur d'écriture en mode graphique.

ÛHELP. Affiche la liste des commandes de Basic +.

ÛINVERSE. Echange la couleur de fond et la couleur d'écriture.

ÛKEYRES

ÛLARGE,àA\$. Affiche le texte contenu dans A\$ en double hauteur.

ÛMCPY,source,destination,longueur. Recopie le contenu de «source» vers la «destination» sur une certaine longueur. Le tout se déroule en mémoire.

ÛMDUMP,départ,longueur. Affiche le contenu de la mémoire à partir de l'adresse «départ» et sur toute la «longueur».

ÛOVER,0 ou 1

ÛPCHECK,àA

ÛPRINTL,àA\$. Affiche le contenu de variable A\$ verticalement de haut en bas

ÛPRINTR,àA\$. Même effet que l'Instruction précédente mais de bas en haut.

ÛSCRINIT. Remet l'écran dans ses couleurs initiales.

ÛSCROLLODOWN. Déplace l'écran d'une ligne texte vers le bas.

ÛSCROLLUP. Déplace l'écran d'une ligne vers le haut.

ÛSHIFTOFF. Action contraire de ÛSHIFTON.

ÛSHIFTON. Simule une mise en place permanente de la touche SHIFT.

ÛSOFTROLL,0-1,encr,gauche,droite,haut,bas

ÛVDU,a,b,c,... etc

ÛWAIT. Stoppe un programme et attend l'appui d'une touche (CALL &BB06)

ÛWAIT,num. Stoppe un programme et attend l'appui sur la touche dont le numéro est dans num.

Vous avez remarqué que certaines instructions ne possèdent pas de commentaires : Je n'ai tout simplement pas eu le temps de rechercher leur fonction. Et puis, il faut bien que vous aussi vous travailliez.

• **MANDEL**. Création de figures fractales. Très long et peu être très joli si les trois chiffres choisis au départ sont corrects.

• **SIREN**. Imité le cri d'une ambulance.

• **MGP**. Traceur de courbes mathématiques.

Voilà, c'est tout pour ce mois-ci. En cas de problèmes téléphonez au 99-52-98-11 le Mercredi après-midi de 14 h à 17 h.

Olivier SAOLETTI



CPC  
jeu

# Plumpy

(suite)

M. FAUQUET - S. ST-MARTIN - C. VEYRE

Volable pour CPC 464 - 664 - 6128

Eh oui, quand y'en a plus, y'en a encore. Le listing DATA7 de Plumpy avait été laissé dans un coin sombre et le voici tout de même à la lumière des projecteurs. L'absence de ce listing explique les erreurs lors de l'accès au jeu proprement dit (voir CPC Infos 27 & 28). Attention, vous devez lancer ce listing uniquement en fin puisqu'il utilise PLUMPY22.BIN et PLUMPY23.BIN. Bon courage !

## DATAS7 .BAS

```

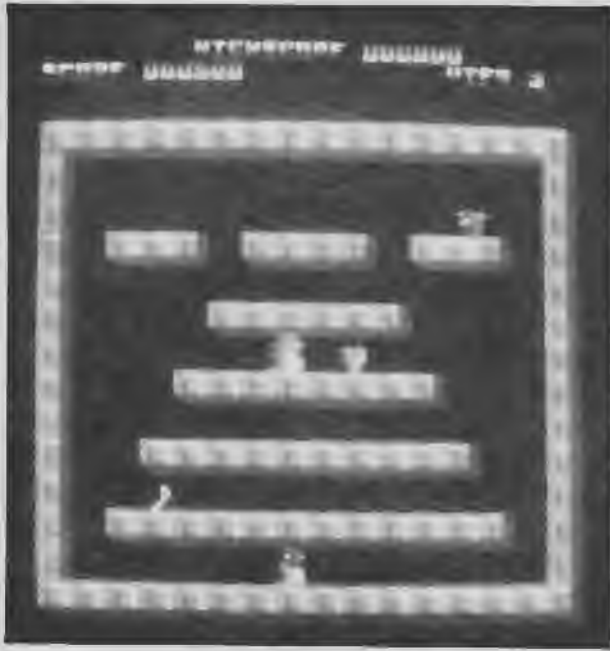
10 ' DATAS 7
20 '
30 MODE 2:AD=&C000:NL=100
40 FOR A=1 TO 365:B=0:FOR C=1 TO 16:READ A#
50 POKE AD,VAL("&"*A#):B=B+PEEK(AD):AD=AD+1
60 NEXT:READ E#:IF VAL("&"*E#1<>B THEN 90
70 NL=NL+10:NEXT:SAVE"PLUMPY23.BIN",B,&C000,&1560
80 MODE 2:LOAD"PLUMPY22.BIN",&C000:LOAD"PLUMPY23.BIN",&E000:ERA,
"PLUMPY22.BIN":!ERA,"PLUMPY23.BIN":SAVE"PLUMPY22.BIN",B,&C000,&35
60:CLS:END
90 CLS:PRINT"LIGNE"NL;" INCORRECTE":END
100 DATA 00,00,00,DB,D1,8A,00,00,45,00,00,45,00,00,45,305
110 DATA 00,00,45,00,00,00,DB,D1,8A,00,00,00,00,00,00,2FB
120 DATA 20,00,00,00,00,00,00,00,8A,00,00,F0,00,45,8A,319
130 DATA 3A,40,F0,00,45,02,30,10,D2,22,45,12,30,00,C3,617
140 DATA 22,45,9A,00,10,C1,02,00,9A,3A,10,C1,00,45,02,620
150 DATA 20,30,00,00,00,12,30,30,00,00,00,45,20,9D,30,304
160 DATA 22,00,CF,21,99,30,22,00,51,DB,CE,32,20,05,14,462
170 DATA 00,10,00,00,00,00,00,00,00,00,40,00,00,00,45,95
180 DATA E7,00,00,00,00,8A,00,8A,00,00,00,8A,00,8A,00,30F
190 DATA 00,00,45,E7,00,00,00,51,00,00,45,00,00,00,00,102
200 DATA 00,00,00,00,8A,00,00,F0,00,45,8A,3A,40,F0,00,4EB
210 DATA 45,02,30,10,D2,22,45,12,30,00,C3,22,45,9A,00,506
220 DATA 10,C1,02,00,9A,3A,10,C1,00,45,02,20,30,00,00,5A7
230 DATA 00,12,30,30,00,00,00,45,20,9D,30,22,00,CF,21,39E
240 DATA 99,30,22,00,51,DB,CE,32,20,03,0D,00,40,C3,00,44A
250 DATA 00,40,C1,00,00,D0,E0,02,00,F0,E0,02,40,F0,C1,776
260 DATA C1,10,F0,C2,D0,10,C0,C0,C3,10,C0,C0,C1,10,00,867
270 DATA C0,C0,00,30,C0,00,00,60,C0,00,00,10,C0,00,00,560
280 DATA 10,30,00,03,0D,00,40,D2,00,00,40,D0,00,00,C0,332
290 DATA CB,A0,00,C5,CF,A0,40,CF,CF,DA,40,4F,0F,DA,10,85F

```

```

300 DATA 07,00,40,10,ES,DA,CB,10,0F,9F,4A,00,65,45,00,609
310 DATA 00,65,45,00,00,10,CA,00,00,10,60,00,03,0D,00,204
320 DATA 40,D2,00,00,40,D0,00,00,CA,63,A0,00,91,33,A0,549
330 DATA 40,33,33,72,40,10,27,72,10,07,23,40,10,B1,72,304
340 DATA 63,10,30,37,62,00,31,11,00,00,31,11,00,00,10,2DB
350 DATA 62,00,00,10,60,00,03,0D,00,40,D2,00,00,40,D0,304
360 DATA 00,00,C0,40,A0,00,05,0F,A0,40,0F,0F,5A,40,25,3FC
370 DATA 1A,5A,10,90,4A,61,10,A5,5A,40,10,2F,1F,4A,00,3C1
380 DATA 25,05,00,00,25,05,00,00,10,4A,00,00,10,60,00,21E
390 DATA 03,0C,00,A2,44,00,45,DB,33,00,45,F3,66,00,45,53B
400 DATA DB,33,00,01,0E,00,A0,01,06,5C,20,01,00,0C,20,44F
410 DATA 00,06,FC,00,00,06,1C,00,00,03,BC,00,00,01,20,20C
420 DATA 00,00,01,20,00,03,00,11,01,44,00,51,F3,DB,A2,34B
430 DATA DB,DB,DB,DB,ED,ED,ED,ED,F2,79,7B,70,58,70,70,AC3
440 DATA 2C,58,70,70,2C,0C,0C,0C,0C,04,06,10,A6,A6,A6,3DC
450 DATA B2,10,53,53,53,12,40,71,F3,F3,60,00,25,53,07,543
460 DATA 20,00,90,0F,1A,00,00,40,30,60,00,02,0C,00,05,23C
470 DATA 0A,00,54,00,00,14,00,00,14,A0,00,00,A0,00,14,1F2
480 DATA A0,00,14,00,00,54,00,00,AC,00,00,20,00,54,00,240
490 DATA 00,04,00,00,03,00,00,99,11,00,44,22,99,66,00,2A9
500 DATA 01,26,44,00,16,06,19,00,7C,03,00,00,A0,03,00,1DB
510 DATA 00,00,03,00,00,0F,03,00,00,00,00,00,0F,03,60
520 DATA 00,00,05,00,00,02,0C,05,00,00,05,06,00,00,03,49
530 DATA 00,05,AA,AB,1B,00,FE,1B,22,FE,1B,00,AE,1B,44,5E3
540 DATA 00,1B,11,23,0A,33,26,0F,33,03,05,00,02,01,0E,1A1
550 DATA 01,5C,01,5C,00,02,40,E1,40,E1,40,E1,40,E1,40,600
560 DATA E1,10,F0,40,E1,40,E1,10,E1,10,C1,00,20,01,00,611
570 DATA 0A,0A,47,20,9A,30,9A,35,A2,30,CE,20,90,20,10,59F
580 DATA C1,10,C3,30,D2,31,93,04,0A,00,31,36,00,00,00,3CF
590 DATA 31,36,00,00,00,31,36,00,00,E1,C0,36,00,00,00,2A5
600 DATA 31,33,6C,20,41,C0,33,99,9C,00,31,33,33,66,00,45E
610 DATA C2,33,33,66,00,0C,AC,FC,FC,00,0F,07,03,03,90,4EA
620 DATA 90,90,30,60,60,30,90,90,60,60,30,90,30,C0,90,600
630 DATA 30,30,F0,3A,3A,6A,6A,3A,3A,3A,3A,3A,3F,60,30,409

```



640 DATA 30,90,30,C3,60,60,30,30,90,90,60,30,30,3F,90,582  
 650 DATA 90,60,30,60,E0,92,92,92,92,00,92,00,E0,C2,3A,876  
 660 DATA 30,60,90,90,90,90,30,60,60,30,90,90,60,60,30,600  
 670 DATA 90,30,C0,90,30,30,F0,3A,3A,6A,6A,3A,3A,3A,3A,590  
 680 DATA 3A,3F,60,30,30,95,35,97,35,35,35,95,95,35,4CD  
 690 DATA 35,35,3F,95,95,35,3F,3F,3F,3F,3F,3F,3F,3F,43F  
 700 DATA 3F,3F,3F,3F,3F,3F,90,90,90,30,60,60,30,90,519  
 710 DATA 90,60,60,3F,3F,3F,3F,3F,30,30,F0,3A,3A,6A,6A,523  
 720 DATA 3A,3A,3A,3A,3F,3F,3F,3F,95,35,97,35,35,35,423  
 730 DATA 35,95,95,35,35,3F,3F,3F,3F,3F,3F,3F,3F,43F  
 740 DATA 3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,90,90,90,30,495  
 750 DATA 60,60,30,90,90,60,60,3F,3F,3F,3F,3F,30,30,F0,55B  
 760 DATA 3A,3A,6A,6A,3A,3A,3A,3A,3F,3F,3F,3F,3F,90,30,42B  
 770 DATA C3,60,60,30,30,90,90,60,30,3F,3F,3F,3F,3F,30,4FE  
 780 DATA 60,E0,92,92,92,92,00,92,00,E0,3F,3F,3F,3F,3F,795  
 790 DATA 90,90,90,30,60,60,30,90,90,60,60,3F,3F,15,15,55B  
 800 DATA 00,30,30,F0,3A,3A,6A,6A,3A,3A,3A,3A,3F,3F,3F,43D  
 810 DATA 3F,3F,90,30,C3,60,60,30,30,90,90,60,30,3F,3F,54F  
 820 DATA 3F,3F,3F,30,60,E0,92,92,92,92,00,92,00,E0,3F,786  
 830 DATA 3F,3F,3F,3F,90,90,90,30,60,60,30,90,90,60,30,57C  
 840 DATA 3F,3F,15,15,00,30,30,F0,3A,3A,6A,6A,3A,3A,3A,3EE  
 850 DATA 3A,3F,3F,3F,3F,3F,95,35,97,35,35,35,95,95,4D4  
 860 DATA 35,35,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,39D  
 870 DATA 3F,3F,3F,3F,3F,3F,3F,3F,90,90,90,30,60,60,30,4CB  
 880 DATA 90,90,60,60,30,90,30,C0,90,30,30,F0,3A,3A,6A,64E  
 890 DATA 6A,3A,3A,3A,3A,3A,3F,60,30,30,95,35,97,35,35,456  
 900 DATA 35,35,95,95,35,35,35,3F,95,95,35,00,00,2A,2A,425  
 910 DATA 3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,90,90,90,4AA  
 920 DATA 30,60,60,30,90,90,60,60,30,90,30,C0,90,30,30,5A0  
 930 DATA F0,3A,3A,6A,6A,3A,3A,3A,3A,3A,3F,60,30,30,90,4E9  
 940 DATA 30,C3,60,60,30,30,90,90,60,30,35,3F,95,95,35,596  
 950 DATA 30,60,E0,92,92,92,92,00,92,00,92,3F,3F,3F,3F,738  
 960 DATA 3F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,0C,06,03,68E  
 970 DATA 0B,0F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,0C,06,666  
 980 DATA 03,0B,0F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,0C,663  
 990 DATA 06,03,0B,0F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,65D  
 1000 DATA 0C,06,03,0B,0F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,56D

1010 DATA FC,0C,06,03,0B,0F,3F,3A,30,60,C0,C3,D2,F0,B4,62D  
 1020 DATA 3C,FC,0C,06,03,0B,0F,3F,3F,35,35,95,97,97,85,4C7  
 1030 DATA B5,3D,0D,1D,17,17,1F,1F,3F,3F,3F,3F,3F,3F,3F1  
 1040 DATA 3F,3F,3F,3F,3F,3F,3F,3F,3F,3A,30,60,C0,C3,4C3  
 1050 DATA D2,F0,B4,3C,FC,3F,3F,3F,3F,3F,3F,3A,30,60,C0,682  
 1060 DATA C3,D2,F0,B4,3C,FC,3F,3F,3F,3F,3F,3F,3F,35,35,694  
 1070 DATA 95,97,97,B5,B5,3D,0D,3F,3F,3F,3F,3F,3F,3F,61F  
 1080 DATA 3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3A,3AC  
 1090 DATA 30,60,C0,C3,D2,F0,B4,3C,FC,3F,3F,3F,3F,3F,3F,730  
 1100 DATA 3A,30,60,C0,C3,D2,F0,B4,3C,FC,3F,3F,3F,3F,3F,736  
 1110 DATA 3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,3F,3F,3F,3F,736  
 1120 DATA 3F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,3F,3F,3F,736  
 1130 DATA 3F,3F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,3F,3F,736  
 1140 DATA 15,15,00,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,3F,6A3  
 1150 DATA 3F,3F,3F,3F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,736  
 1160 DATA 3F,3F,3F,3F,3F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,679  
 1170 DATA FC,3F,3F,3F,3F,3F,3F,3A,30,60,C0,C3,D2,F0,B4,739  
 1180 DATA 3C,FC,3F,3F,15,15,00,3F,3A,30,60,C0,C3,D2,F0,62E  
 1190 DATA B4,3C,FC,3F,3F,3F,3F,3F,3F,35,35,95,97,97,5D2  
 1200 DATA B5,B5,3D,0D,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,519  
 1210 DATA 3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3A,30,60,C0,43F  
 1220 DATA C3,D2,F0,B4,3C,FC,0C,06,03,0B,0F,3F,3A,30,60,5A9  
 1230 DATA C0,C3,D2,F0,B4,3C,FC,0C,06,03,0B,0F,3F,3F,35,613  
 1240 DATA 35,95,97,97,B5,B5,3D,0D,1D,17,17,1F,1F,00,00,4E5  
 1250 DATA 2A,2A,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,387  
 1260 DATA 3A,30,60,C0,C3,D2,F0,B4,3C,FC,0C,06,03,0B,0F,62A  
 1270 DATA 3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,0C,06,03,0B,65A  
 1280 DATA 0F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,1D,17,694  
 1290 DATA 1F,1F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,3F,3F,6F6  
 1300 DATA 3F,3F,3F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,FF,7F6  
 1310 DATA DD,CC,66,33,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,87C  
 1320 DATA FF,DD,CC,66,33,3F,3A,30,60,C0,C3,D2,F0,B4,3C,87F  
 1330 DATA FC,FF,DD,CC,66,33,3F,3A,30,60,C0,C3,D2,F0,B4,93F  
 1340 DATA 3C,FC,FF,DD,CC,66,33,3F,3A,30,60,C0,C3,D2,F0,8C7  
 1350 DATA B4,3C,FC,FF,DD,CC,66,33,3F,3A,30,60,C0,C3,D2,88B  
 1360 DATA F0,B4,3C,FC,FF,DD,CC,66,33,3F,3F,35,35,95,97,831  
 1370 DATA 97,B5,B5,3D,0D,BF,9D,9D,37,37,3F,3F,3F,3F,69D  
 1380 DATA 3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3A,30,60,38E  
 1390 DATA C0,C3,D2,F0,B4,3C,FC,3F,3F,3F,3F,3F,3F,3A,30,715  
 1400 DATA 60,C0,C3,D2,F0,B4,3C,FC,3F,3F,3F,3F,3F,3F,3F,74A  
 1410 DATA 35,35,95,97,97,B5,B5,3D,0D,3F,3F,3F,3F,3F,3F,600  
 1420 DATA 3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3B1  
 1430 DATA 3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,3F,3F,3F,3F,736  
 1440 DATA 3F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,3F,3F,3F,736  
 1450 DATA 3F,3F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,3F,3F,736  
 1460 DATA 3F,3F,3F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,3F,736  
 1470 DATA 3F,3F,3F,3F,3F,3A,30,60,C0,C3,D2,F0,B4,3C,FC,736  
 1480 DATA 3F,3F,15,15,00,3F,3A,30,60,C0,C3,D2,F0,B4,3C,5E6  
 1490 DATA FC,3F,3F,3F,3F,3F,3F,3A,30,60,C0,C3,D2,F0,B4,739  
 1500 DATA 3C,FC,3F,3F,3F,3F,3F,3F,3A,30,60,C0,C3,D2,F0,6C1  
 1510 DATA B4,3C,FC,3F,3F,3F,3F,3F,3A,30,60,C0,C3,D2,685  
 1520 DATA F0,B4,3C,FC,3F,3F,15,15,00,3F,3A,30,60,C0,C3,610  
 1530 DATA D2,F0,B4,3C,FC,3F,3F,3F,3F,3F,3F,3F,35,35,95,666  
 1540 DATA 97,97,B5,B5,3D,0D,3F,3F,3F,3F,3F,3F,3F,3F,5C9  
 1550 DATA 3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3A,30,39D  
 1560 DATA 60,C0,C3,D2,F0,B4,3C,FC,FF,DD,CC,66,33,3F,3A,94B  
 1570 DATA 30,60,C0,C3,D2,F0,B4,3C,FC,FF,DD,CC,66,33,3F,941  
 1580 DATA 3F,35,35,95,97,97,B5,B5,3D,0D,BF,9D,9D,37,37,737  
 1590 DATA 00,00,2A,2A,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,309



1600 DATA 3F, 3F, 3A, 30, 60, C0, C3, D2, F0, B4, 3C, FC, FF, DD, CC, 921  
 1610 DATA 66, 33, 3F, 3A, 30, 60, C0, C3, D2, F0, B4, 3C, FC, FF, DD, 8AF  
 1620 DATA CC, 66, 33, 3F, 3A, 30, 60, C0, C3, D2, F0, B4, 3C, FC, BF, 85E  
 1630 DATA 9D, 9D, 37, 37, 3F, 3A, 30, 60, C0, C3, D2, F0, B4, 3C, FC, 7E2  
 1640 DATA 3F, 3F, 3F, 3F, 3F, C3, F0, 30, 30, 3F, 3F, 30, 30, 30, 30, 48C  
 1650 DATA 30, 30, 30, 30, 30, 30, C3, F0, 70, 70, 7A, 6B, 61, 61, 70, 5CA  
 1660 DATA 70, 70, 70, 61, 61, 61, 70, C3, F0, 35, 35, 3F, 3F, 35, 35, 5E8  
 1670 DATA 35, 35, 35, 35, 35, 35, 35, 35, C3, F0, 30, 30, 3F, 3F, 30, 469  
 1680 DATA 30, 30, 30, 30, 30, 30, 30, 30, C3, F0, 30, 30, 3F, 3F, 441  
 1690 DATA 30, 30, 30, 30, 30, 30, 30, 30, 30, C3, F0, 70, 70, 7A, 4ED  
 1700 DATA 6B, 61, 61, 70, 70, 70, 70, 61, 61, 61, 70, 97, 85, 35, 35, 636  
 1710 DATA 3F, 3F, 35, 35, 35, 35, 35, 35, 35, 35, 35, 3F, 3F, 3F, 34D  
 1720 DATA 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, C3, F0, 4E6  
 1730 DATA 30, 30, 3F, 3F, 30, 30, 30, 30, 30, 3F, 3F, 3F, 3F, C3, 3CC  
 1740 DATA F0, 70, 70, 7A, 6B, 61, 61, 70, 70, 70, 3F, 3F, 3F, 3F, 602  
 1750 DATA 97, 85, 35, 35, 3F, 3F, 35, 35, 35, 35, 35, 3F, 3F, 3F, 3F, 439  
 1760 DATA 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3B1  
 1770 DATA 3F, 3F, C3, F0, 30, 30, 3F, 3F, 30, 30, 30, 30, 30, 3F, 3F, 47D  
 1780 DATA 3F, 3F, 3F, C3, F0, 70, 70, 7A, 6B, 61, 61, 70, 70, 70, 3F, 686  
 1790 DATA 3F, 3F, 3F, 3F, C3, F0, 35, 35, 3F, 3F, 35, 35, 35, 35, 35, 4A0  
 1800 DATA 3F, 3F, 3F, 3F, 3F, C3, F0, 30, 30, 3F, 3F, 30, 30, 30, 30, 48C  
 1810 DATA 30, 3F, 3F, 3F, 3F, 3F, C3, F0, 30, 30, 3F, 3F, 30, 30, 30, 48C  
 1820 DATA 30, 30, 3F, 3F, 15, 15, 00, C3, F0, 70, 70, 7A, 6B, 61, 61, 542  
 1830 DATA 70, 70, 70, 3F, 3F, 3F, 3F, 3F, C3, F0, 35, 35, 3F, 3F, 35, 55B  
 1840 DATA 35, 35, 35, 35, 3F, 3F, 3F, 3F, 3F, C3, F0, 30, 30, 3F, 3F, 4A0  
 1850 DATA 30, 30, 30, 30, 30, 3F, 3F, 3F, 3F, 3F, C3, F0, 30, 30, 3F, 47D  
 1860 DATA 3F, 30, 30, 30, 30, 30, 3F, 3F, 15, 15, 00, C3, F0, 70, 70, 46A  
 1870 DATA 7A, 6B, 61, 61, 70, 70, 70, 3F, 3F, 3F, 3F, 3F, 97, 85, 35, 5B3  
 1880 DATA 35, 3F, 3F, 35, 35, 35, 35, 35, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 375  
 1890 DATA 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, C3, 435  
 1900 DATA F0, 30, 30, 3F, 3F, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 3AE  
 1910 DATA C3, F0, 70, 70, 7A, 6B, 61, 61, 70, 70, 70, 70, 61, 61, 61, 71D  
 1920 DATA 70, 97, 85, 35, 35, 3F, 3F, 35, 35, 35, 35, 35, 35, 35, 35, 44C  
 1930 DATA 35, 35, 00, 00, 2A, 2A, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 3F, 2F5  
 1940 DATA 3F, 3F, 3F, C3, F0, 30, 30, 3F, 3F, 30, 30, 30, 30, 30, 30, 46E  
 1950 DATA 30, 30, 30, 30, C3, F0, 70, 70, 7A, 6B, 61, 61, 70, 70, 70, 64A  
 1960 DATA 70, 61, 61, 61, 70, C3, F0, 35, 35, 3F, 3F, 35, 35, 35, 35, 572  
 1970 DATA 35, 35, 35, 35, 35, 35, C3, F0, 30, 30, 3F, 3F, 30, 30, 30, 45F  
 1980 DATA 30, 30, 3F, 3F, 3F, 3F, 3F, 28, 00, 5C, 00, 00, 09, 5C, 20, 2B4  
 1990 DATA 54, 09, 5C, 78, 54, 09, 5C, 28, 00, 00, 00, 00, 54, 09, 5C, 2CB  
 2000 DATA 28, 54, 09, 5C, 78, 00, 01, 5C, 00, 54, 09, 5C, 28, 54, 00, 2F3  
 2010 DATA 00, 78, 00, 00, 5C, 06, 00, 14, AC, 06, BC, 14, AC, 06, BC, 3DE  
 2020 DATA 14, AC, 06, AB, 00, 00, 00, 00, 14, AC, 06, BC, 14, AC, 06, 3B6  
 2030 DATA BC, 00, 04, 06, 00, 14, AC, 06, BC, 14, AB, 00, BC, 00, 00, 3C0  
 2040 DATA 06, 09, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, F  
 2050 DATA 00, 00, 00, 00, 00, 00, 00, 5C, 00, 00, 00, 00, 00, 7C, 09, E1  
 2060 DATA 00, 00, 00, 00, 5C, 50, 7C, 09, 5C, 00, 00, 09, AC, 00, 14, 25E  
 2070 DATA A0, 00, 00, 14, B4, AC, 00, 14, A0, 00, 00, 00, 00, 00, 00, 2C8  
 2080 DATA 14, B4, AC, 06, 14, B4, AC, 00, 00, B4, 04, 02, 14, A0, 00, 45C  
 2090 DATA 06, 00, B4, AC, 02, 00, 00, AC, 00, 00, 54, 28, 54, 09, 54, 341  
 2100 DATA 78, 7C, 00, 54, 28, 00, 00, 00, 00, 00, 00, 54, 78, 7C, 00, 2C0  
 2110 DATA 54, 78, 7C, 00, 54, 78, 54, 09, 54, 28, 00, 09, 00, 00, 00, 2F6  
 2120 DATA 00, 00, 00, 00, 00, 00, 04, AB, 00, AC, 04, AB, 00, 00, 04, 208  
 2130 DATA AB, 00, 00, 00, 00, 00, 00, 00, 14, B4, 00, 04, AB, 00, 00, 21C  
 2140 DATA 04, AB, 00, AC, 04, AB, 00, AC, 00, 14, B4, 00, 00, 00, 00, 378  
 2150 DATA 78, 00, 01, 5C, 78, 7C, 01, 5C, 78, 7C, 01, 5C, 78, 28, 00, 417  
 2160 DATA 00, 00, 00, 00, 00, 78, 28, 01, 5C, 78, 7C, 01, 5C, 28, 7C, 2F2  
 2170 DATA 01, 5C, 78, 7C, 00, 54, 78, 00, 00, 00, 78, BC, 00, 00, 06, 357  
 2180 DATA BC, A0, 04, 06, BC, B4, 00, 06, BC, A0, 00, 00, 00, 00, 00, 43B



2190 DATA 00, 14, B4, 04, 06, BC, B4, 04, 06, AB, B4, 04, 06, BC, A0, 50E  
 2200 DATA 00, 04, BC, 00, 00, 00, BC, 01, FF, C0, 01, FF, C0, 00, FF, 5FB  
 2210 DATA E0, 00, FF, E0, 00, FF, F0, 00, 7F, F0, 00, 7F, F0, 00, 3F, 7CB  
 2220 DATA F0, 00, 3F, F0, 00, 1F, FC, 00, 1F, FC, 00, 0F, FC, 00, 0F, 57F  
 2230 DATA FE, 00, 0F, FE, 00, 07, FF, 00, 07, FF, 00, 07, FF, 00, 03, 52B  
 2240 DATA FF, 00, 03, FF, 00, 03, FF, C0, 01, FF, C0, 01, FF, C0, 00, 843  
 2250 DATA FF, E0, 00, FF, F0, 00, 7F, F0, 00, 7F, F0, 00, 3F, F0, 00, 7E3  
 2260 DATA 1F, FC, 00, 0F, FE, 00, 07, FE, 00, 07, FF, 00, 03, FF, 00, 5B5  
 2270 DATA 01, FF, 00, 01, FF, C0, 00, FF, E0, 00, 7F, E0, 00, 3F, F0, 7AD  
 2280 DATA 00, 1F, F0, 00, 1F, FC, 00, 0F, FC, 00, 07, FE, 00, 03, FF, 544  
 2290 DATA 00, 01, FF, 00, 00, FF, C0, 00, FF, E0, 00, 3F, F0, 00, 1F, 66C  
 2300 DATA F0, 00, 0F, FE, 00, 07, FE, 00, 03, FF, 00, 01, FF, 00, 00, 58C  
 2310 DATA FF, C0, 00, 7F, E0, 00, 3F, F0, 00, 1F, F0, 00, 0F, FC, 00, 66F  
 2320 DATA 03, FF, 00, 03, FF, 00, 00, FF, C0, 00, 7F, F0, 00, 3F, F0, 6E9  
 2330 DATA 00, 0F, FE, 00, 07, FF, 00, 03, FF, 00, 01, FF, E0, 00, 7F, 5F4  
 2340 DATA F0, 00, 3F, F0, 00, 1F, FC, 00, 07, FF, 00, 03, FF, 00, 01, 5CB  
 2350 DATA FF, C0, 00, FF, E0, 00, 7F, F0, 00, 1F, F0, 00, 0F, FC, 00, 72F  
 2360 DATA 07, FF, 00, 03, FF, 00, 01, FF, 00, 00, FF, E0, 00, 7F, F0, 756  
 2370 DATA 00, 3F, F0, 00, 1F, FC, 00, 07, FE, 00, 03, FF, 00, 01, FF, 559  
 2380 DATA C0, 00, FF, E0, 00, 7F, F0, 00, 1F, FC, 00, 0F, FE, 00, 07, 63D  
 2390 DATA FF, 00, 03, FF, 00, 01, FF, C0, 00, FF, E0, 00, 7F, F0, 00, 78F  
 2400 DATA 1F, FC, 00, 0F, FE, 00, 07, FF, 00, 03, FF, 00, 00, FF, C0, 6EF  
 2410 DATA 00, 7F, F0, 00, 3F, FC, 00, 0F, FC, 00, 07, FF, 00, 03, FF, 5BD  
 2420 DATA C0, 01, FF, C0, 00, FF, E0, 00, 7F, F0, 00, 3F, F0, 00, 0F, 71C  
 2430 DATA FC, 00, 07, FE, 00, 03, FF, C0, 03, FF, E0, 00, FF, F0, 00, 794  
 2440 DATA 7F, F0, 00, 3F, FC, 00, 1F, FE, 00, 0F, FE, 00, 07, FF, 00, 5DA  
 2450 DATA 01, FF, 00, 00, FF, E0, 00, 7F, F0, 00, 1F, FC, 00, 0F, FE, 6F6  
 2460 DATA 00, 03, FF, 00, 00, FF, C0, 00, 7F, E0, 00, 3F, F0, 00, 0F, 566  
 2470 DATA FC, 00, 07, FE, 00, 03, FF, 00, 00, FF, C0, 00, 7F, E0, 00, 5A1  
 2480 DATA 7F, F0, 00, 1F, FC, 00, 0F, FE, 00, 03, FF, 00, 00, FF, 00, 618  
 2490 DATA 00, 3F, C0, 00, 7F, E0, 00, 3F, F0, 00, 1F, FC, 00, 07, FE, 5B5  
 2500 DATA 00, 03, FF, 00, 01, FF, 00, 00, FF, C0, 00, 7F, E0, 00, 3F, 5DF  
 2510 DATA F0, 00, 1F, FC, 00, 07, FF, 00, 03, FF, 00, 01, FF, C0, 00, 5DB  
 2520 DATA 7F, E0, 00, 3F, F0, 00, 0F, FE, 00, 07, FF, 00, 03, FF, 00, 623  
 2530 DATA 01, FF, C0, 00, 7F, F0, 00, 3F, F0, 00, 1F, FC, 00, 07, FE, 686  
 2540 DATA 00, 03, FF, 00, 01, FF, C0, 00, FF, E0, 00, 3F, F0, 00, 1F, 5EF  
 2550 DATA F0, 00, 0F, FC, 00, 07, FE, 00, 03, FF, 00, 00, FF, C0, 18, 661  
 2560 DATA 7F, F0, 00, 3F, F0, 00, 0F, FE, 00, 07, FF, 00, 01, FF, 00, 639



2570 DATA 00, 7F, E0, 00, 3F, F8, 00, 0F, FF, 00, 03, FF, 00, 01, FF, 5A6  
2580 DATA F0, 30, 7F, FC, 00, 3F, FC, 02, 0F, FF, 00, 83, FF, E0, 20, 768  
2590 DATA FF, F8, 18, 3F, FE, 06, 0F, FF, 01, 87, FF, E0, 60, FF, F0, 916  
2600 DATA 18, 7F, FE, 06, 1F, FF, 01, 83, FF, E0, 81, FF, F0, 38, 7F, 823  
2610 DATA FC, 06, 1F, FF, 01, 87, FF, C0, E1, FF, F0, 18, 7F, FE, 06, 8D2  
2620 DATA 1F, FF, 01, 07, FF, C0, C1, FF, F0, 30, 7F, F8, 1C, 3F, FE, 895  
2630 DATA 03, 0F, FF, 81, 83, FF, C0, 60, FF, F8, 38, 7F, FC, 06, 1F, 803  
2640 DATA FF, 03, 87, FF, C0, E1, FF, F0, 18, 3F, FC, 06, 0F, FF, 01, 800  
2650 DATA 83, FF, E0, 60, FF, F8, 0C, 3F, FE, 03, 07, FF, 81, C3, FF, 94E  
2660 DATA E0, 70, FF, F8, 0C, 1F, FF, 03, 07, FF, C0, C1, FF, E0, 30, 90A  
2670 DATA 7F, F8, 0C, 1F, FF, 03, 07, FF, 80, C1, FF, F0, 38, 7F, FC, 88D  
2680 DATA 0E, 1F, FF, 01, 07, FF, C0, E0, FF, F0, 18, 7F, FC, 06, 0F, 76A  
2690 DATA FF, 01, 83, FF, E0, 70, 7F, F8, 1C, 1F, FE, 03, 07, FF, C0, 8CB  
2700 DATA E0, FF, F0, 18, 3F, FE, 07, 07, FF, 81, C1, FF, E0, 38, 3F, 8C9  
2710 DATA FC, 0E, 0F, FF, 81, C1, FF, E0, 38, 7F, FC, 0F, 0F, FF, 81, 88A  
2720 DATA C1, FF, F0, 38, 3F, FE, 07, 07, FF, C0, F0, 7F, F8, 1E, 1F, 896  
2730 DATA FF, 03, 83, FF, C0, 70, 7F, F8, 0E, 0F, FF, 01, C1, FF, E0, 8EB  
2740 DATA 78, 7F, FC, 0F, 0F, FF, 81, C1, FF, E0, 38, 3F, FE, 07, 03, 780  
2750 DATA FF, C0, F0, 7F, F8, 1E, 0F, FF, 03, C1, FF, E0, 78, 3F, FC, 9A8  
2760 DATA 0F, 07, FF, 81, E0, FF, F8, 1E, 0F, FF, 03, C1, FF, F0, 3C, 888  
2770 DATA 1F, FF, 03, 81, FF, E0, 78, 1F, FE, 07, 83, FF, E0, 78, 3F, 836  
2780 DATA FE, 07, 83, FF, E0, 78, 1F, FE, 0F, 83, FF, E0, 38, 1F, FE, 8C2  
2790 DATA 07, 81, FF, E0, 78, 1F, FE, 03, C0, FF, F0, 3C, 0F, FF, 03, 7FB  
2800 DATA C0, FF, F0, 3E, 07, FF, 81, F0, 3F, FC, 1F, 83, FF, E0, 78, 998  
2810 DATA 0F, FF, 07, C0, 7F, F8, 7F, 03, FF, C1, F0, 1F, FF, 0F, E0, 88B  
2820 DATA 7F, FC, 7F, 03, FE, E0, FC, 0F, FF, 87, F0, 3F, FF, 80, E0, 9FB  
2830 DATA FF, FC, 0F, 01, FF, F0, 00, 07, FF, C1, F0, 0F, FF, 87, C0, 906  
2840 DATA 0F, FF, 07, 80, 1F, FF, 07, E0, 1F, FF, 07, 80, 1F, FE, 07, 663  
2850 DATA 80, 1F, FF, 03, 80, 1F, FF, 07, 80, 0F, FF, 03, 80, 07, FF, 5DD  
2860 DATA 83, C0, 03, FF, C1, E0, 01, FF, E0, F0, 00, FF, F0, 38, 00, 8DD  
2870 DATA 3F, FC, 06, 00, 0F, FF, 00, 00, 03, FF, C0, F0, 60, 7F, F8, 6DB  
2880 DATA 0E, 00, 07, FF, 81, C1, E0, 7F, F8, 0C, 1E, FF, FF, FF, FF, 8D3  
2890 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
2900 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
2910 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
2920 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
2930 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
2940 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
2950 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
2960 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
2970 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
2980 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
2990 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3000 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3010 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3020 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3030 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3040 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3050 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3060 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3070 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3080 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3090 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3100 DATA E0, 01, BF, F0, E0, 00, 7C, FF, FC, 00, E0, F0, F8, F0, 00, 99F  
3110 DATA FF, FC, 00, 70, 78, F8, F0, 60, FF, F0, 00, E0, F0, F0, ADA  
3120 DATA E1, E3, F1, C1, C0, 78, F8, E0, C1, E7, F1, C1, C0, 78, F8, C10  
3130 DATA E0, 01, FF, F8, C1, C0, 78, F8, E0, 01, FF, F0, 81, C0, F1, ACB  
3140 DATA F1, C0, 01, F3, F0, 03, C1, C3, E3, C0, 83, E7, E3, 07, 83, 996  
3150 DATA 87, C7, B1, 07, CF, C6, 07, 07, 0F, CF, 02, 0F, 9F, 9C, 0E, 581

3160 DATA 0E, 3F, 1E, 0C, 1F, FF, FC, 00, 1C, 7E, 38, 18, 3E, FF, F0, 5A8  
3170 DATA 00, 61, F9, F0, 60, 79, FF, E0, 01, C7, FF, C0, 01, 8F, FF, 918  
3180 DATA 80, 00, 7F, FE, 00, 0C, 7F, FC, 00, 3F, E0, 03, 1F, C0, 1F, 5A4  
3190 DATA F0, 3F, F8, 00, FF, E0, 3F, F0, 03, FF, 00, 1F, 80, 0F, F8, 7DD  
3200 DATA 00, FF, E0, 07, F0, 03, FE, 00, 3F, F8, 01, FC, 00, FF, 82, 78C  
3210 DATA 0F, FE, 00, 7E, 00, 7F, C1, 87, EF, 02, 1F, 80, 1F, F0, 61, 652  
3220 DATA FB, C1, 87, C0, 0F, F8, 38, 7C, F0, 63, F0, 03, FA, 0E, 3E, 84A  
3230 DATA 3C, 18, FC, 20, FF, 83, 87, 8F, 06, 3F, 08, 3F, F0, E1, E3, 748  
3240 DATA C1, 87, C0, 0F, FC, 38, 78, F0, 61, F0, 83, FF, 0E, 1E, 3C, 7EE  
3250 DATA 18, FC, 20, FF, C3, 87, 9F, 06, 3F, 18, 3F, F0, C1, C7, C1, 7F1  
3260 DATA 0F, 86, 0F, FC, 70, F1, E0, C3, E3, 03, FF, 1C, 3C, F8, 21, 7FA  
3270 DATA F0, C0, FF, 86, 0E, 7C, 18, 7C, E0, 77, E3, 06, 3E, 00, 3C, 78D  
3280 DATA F0, 83, F1, 07, 3F, 00, 3E, F8, 01, FF, 07, 3F, 0C, 3C, F8, 5E8  
3290 DATA 01, FF, 0E, 3E, 0C, 38, F8, 61, FF, 0E, 7E, 18, 78, F8, 61, 65D  
3300 DATA FF, 0C, 7E, 18, 78, F8, 61, FE, 1C, 7E, 18, 79, F8, E1, F2, 866  
3310 DATA 1C, 7E, 18, 38, F8, 60, FB, 0E, 3F, 0C, 1C, 7C, 70, 7C, 07, 521  
3320 DATA 3F, 00, 0C, 3E, 78, 3C, 07, 1F, 82, 06, 3F, FE, 00, 03, 8F, 43A  
3330 DATA C1, 83, 07, FF, 80, C1, C3, F8, F0, 00, FB, C0, 00, 00, 3E, 82F  
3340 DATA F8, 00, 39, C1, 80, 18, 1E, 3C, 00, 1C, E1, 86, 0C, 07, 1E, 498  
3350 DATA 30, 06, 70, C3, 06, 07, CF, 38, 01, F8, C3, 87, 0F, F3, FE, 8C0  
3360 DATA 20, C3, 01, E1, 87, 7C, FF, 80, 63, 83, F8, 00, FF, 7F, E0, 883  
3370 DATA 1C, C3, FE, 00, 3F, FF, F8, C7, 81, FF, 80, 0F, F9, FF, E0, 9C1  
3380 DATA C3, CF, FC, 00, FE, 7C, F0, 00, E7, FF, 02, 3F, FF, FE, 01, 91D  
3390 DATA 8F, FF, C0, 03, FE, FF, E0, 31, FF, FC, 00, 3F, FF, FC, 00, 99A  
3400 DATA 1F, FF, F8, 01, FF, FF, E0, 39, FF, FF, C0, 1F, FF, FC, 01, A07  
3410 DATA DF, FF, FC, 01, FF, FE, FF, CC, 7F, FF, F8, FF, FF, FF, FF, D15  
3420 DATA BD, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EAF  
3430 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3440 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3450 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3460 DATA FF, FF, FF, FF, FE, FF, FF, FE, FF, FF, FF, FF, FF, FF, EF1  
3470 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3480 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3490 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3500 DATA FF, FF, FF, FF, FE, FF, FE, FF, FE, FF, FE, FF, FE, FF, EF1  
3510 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3520 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3530 DATA FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3540 DATA FF, FF, FF, FF, FE, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3550 DATA FF, FE, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, EF1  
3560 DATA FF, FF, FF, FF, FF, FF, FF, FF, CF, FD, EF, FE, C0, 1F, FF, D90  
3570 DATA 0F, F8, 9C, 63, 9C, 77, FF, C4, 73, EE, 31, C7, FF, C0, 88, 97C  
3580 DATA F1, 0E, FE, 33, 8E, FC, C3, 18, E7, FF, E6, 33, 3F, FD, CF, 99F  
3590 DATA 00, 46, FF, FF, 32, 2F, F9, B2, 01, 9F, BF, FC, 60, 4E, F8, 85A  
3600 DATA 26, 53, 07, FF, FE, 64, 9E, E7, CC, E2, 4C, FF, FE, 66, 4F, 912  
3610 DATA 77, F6, 61, 26, F3, FC, 33, 47, 33, 02, 21, 86, 78, FE, 19, 798  
3620 DATA 07, 89, CB, 30, 83, 7D, BE, 4C, 93, D8, E5, 98, C9, 3C, DF, 892  
3630 DATA 26, 41, EE, F2, 4C, 24, DF, EF, 91, 32, 78, 2D, 32, 19, 27, 662  
3640 DATA F3, 6C, CC, 9C, C9, 48, 84, 99, ED, B2, 33, 27, 36, C2, 60, 846  
3650 DATA 26, DF, 6C, 86, 49, EC, 92, C8, 25, 9A, C8, 30, C8, 34, 9A, 7D6  
3660 DATA 49, 04, 13, 4D, E2, 04, C9, 4C, A4, 98, 43, 30, DF, 61, 26, 58D  
3670 DATA 49, 64, 24, C2, 49, A4, D3, 08, 89, 31, 92, 13, 08, 22, 1A, 4FE  
3680 DATA 4F, 21, 92, 61, 34, B2, 08, 46, 33, C8, 21, 99, 21, 94, D3, 5D4  
3690 DATA 08, 26, 18, EC, 22, 22, 4C, 64, 04, C0, 09, 9E, F3, 08, 06, 48D  
3700 DATA 64, 86, 40, 4C, 00, 99, E4, 90, 40, 88, 93, 19, 00, 30, 00, 4A7  
3710 DATA 67, 98, 40, 00, 86, 98, 39, 80, 10, 00, 33, 88, 60, 00, C1, 507  
3720 DATA 11, 67, 30, 02, 00, 06, 71, 00, 00, 18, C4, 4D, 9C, C0, 08, 3AE  
3730 DATA 00, 19, 00, 00, 00, 00, 00, 44, D8, CC, 00, 00, 01, 9E, 40, 2E3  
3740 DATA 00, 98, 44, 45, 2C, 48, 4C, 0D, 80, 41, 2C, 28, 00, 00, 00, 306

# CPC

jeu

# CIRCUS

## folies



**Claude LE MOULLEC**

Valable pour CPC 464  
664 - 6128

Ce jeu reprend le principe de Sphérique mais cette fois, il ne s'agit pas de faire passer une porte par une boule mais de faire rentrer un ours dans une cage. Les règles et déplacements sont inclus dans le jeu. Circus Folies ne se joue qu'au joystick et se compose de 2 fichiers : CIRCUS.BAS, le programme principal et DATACIR, chargeur BASIC qui sauvegardera le programme CIRBIN.

## CIRCUS.BAS

```

10 REM .....:
20 REM :
30 REM : CIRCUS FOLIE :
40 REM :
50 REM : CLAUDE LE MOULLEC :
60 REM : 83 RUE J CURIE :
70 REM : 22420 PLOUARET :
80 REM : TEL 96 38 94 24 :
90 REM :
100 REM .....:
110 MEMORY &9700
120 LOAD "cirbin",&9720
130 REM .....:
140 REM :
150 REM : VARIABLES DE BASE :

```



```

160 REM : : >LB
170 REM .....: >AL
180 GOSUB 3400:REM si vous (tes trop fainfant pour tape >KR
n la presentation ne tapez pas cette ligne
190 DEFINT A-Z:MODE 0:BORDER 0 >YC
200 RESTORE 210:FOR H=0 TO 15:READ A:INK H,A:NEXT >RV
210 DATA 0,4,2,11,9,24,3,6,1,16,15,13,26,20,19,18 >NM
220 DEF FN PO(X,Y)=&C050+(Y-1)*160+(X-1)*4 >HP
230 DEF FN LP(X,Y)=&C000+(Y-1)*80+(X-1)*2 >GR
240 DIM SP(30):FOR H=0 TO 23:SP(H+1)=&9720+(H*64):NEXT >XA
250 DIM JE(20,12):DIM a1*(20):a=1 >AU
260 WINDOW #1,2,19,4,23:VIE=4 >XN
270 ENV 1,100,3,1:ENT 1,100,2,2:ENV 14,1,5,1,10,1,4,5,- >ZD
3,1
280 ENT 2,10,-2,2:ENV 2,10,-1,2 >WT
290 ENV 15,1,5,1,10,1,4,5,-3,1:ENT 14,40,-10,1:ENT 15,5 >MM
,-20,1,60,10,1
300 REM .....: >AF

```



```

310 REM : >KJ
320 REM : DECOR DE BASE : >RF
330 REM : : >LA
340 REM : : : >AK
350 CALL &A00D, FN PO(1,1), SP(1): CALL &A00D, FN PO(20,1), >DA
SP(2)
360 CALL &A00D, FN PO(20,12), SP(3): CALL &A00D, FN PO(1,12 >FB
), SP(4)
370 FOR H=2 TO 19: CALL &A00D, FN PO(H,1), SP(5): CALL &A00 >ER
D, FN PO(H,12), SP(7): NEXT
380 FOR H=2 TO 11: CALL &A00D, FN PO(1,H), SP(8): CALL &A00 >EK
D, FN PO(20,H), SP(8): NEXT
390 a$="SCORE:00000 00 00000:HIGHT": ZL=1: >YT
YL=1: CER=2: GOSUB 420
400 a$=" 20 ": ZL=19: YL=25: CER=2: GOSUB 420 >JT
410 GOTO 500 >YG
420 A$=UPPER$(A$): FOR T=1 TO LEN(A$): PS=(ASC(MID$(A$, T, >HV
1)))-48
430 IF PS<0 THEN PS=43 >QJ
440 CALL &9D20, FN LP(ZL, YL)+(T*2)-CER, &9D40+(PS*16): NEX >MD
T: RETURN
450 REM : : >AM
460 REM : : >LE
470 REM : BRANCHEMENT TABLEAU : >ZN
480 REM : : >LG
490 REM : : >AR
500 ON ta GOTO 510, 520, 530, 540, 550, 560, 570, 580, 590, 600, >JU
610, 620, 630
510 RESTORE 2800: GOSUB 690: GOTO 980 >DA
520 RESTORE 2920: GOSUB 690: GOTO 980 >CE
530 RESTORE 2960: GOSUB 690: GOTO 980 >DB
540 RESTORE 3000: GOSUB 690: GOTO 980 >CV
550 RESTORE 3040: GOSUB 690: GOTO 980 >CB
560 RESTORE 3080: GOSUB 690: GOTO 980 >CG
570 RESTORE 3120: GOSUB 690: GOTO 980 >CC
580 RESTORE 3160: GOSUB 690: GOTO 980 >CH
590 RESTORE 3200: GOSUB 690: GOTO 980 >CD
600 RESTORE 3240: GOSUB 690: GOTO 980 >CZ
610 RESTORE 3280: GOSUB 690: GOTO 980 >CE
620 RESTORE 3320: GOSUB 690: GOTO 980 >CA
630 CLS #1: a$="BRAVO @ @ @": ZL=15: YL=12: GOSUB 420: GOTO >ER
2770
640 REM : : >AN
650 REM : : >LF
660 REM : DESSIN TABLEAU : >UF
670 REM : : >LH
680 REM : : >AT
690 ERASE je: DIM je(20,12): FOR h=1 TO 20: je(h,1)=1: je(h >MQ
,12)=1: NEXT
700 FOR h=2 TO 11: je(1,h)=1: je(20,h)=1: NEXT >KE
710 CLS #1: A$="LIFE ": STR$(VIE): ZL=17: YL=14: CER=2: GOSU >GL
B 420
720 FOR T=1 TO 2000: NEXT: CLS #1: a$=" 20 ": ZL=19: YL=25: C >RZ
ER=2: GOSUB 420
730 a1=9: GOSUB 740: a1=10: GOSUB 740: GOTO 860 >LD
740 FOR h=2 TO 11: READ a1: a1$(h)=BIN$(a1,9): NEXT >QC
750 FOR h=2 TO 11: IF a1$(h)="000000000" THEN 790 >LY
760 FOR g=1 TO 9: b1$=MID$(a1$(h), g, 1): b1=VAL(b1$) >QR
770 IF b1=1 THEN CALL &A00D, FN po(g+1, h), sp(a1): je(g+1, >GY

```

```

h)=a1-8
780 NEXT g >PB
790 SOUND 1, INT(RND*600)+60, 5, 15: NEXT h >GB
800 FOR h=2 TO 11: READ a1: a1$(h)=BIN$(a1,9): NEXT >QZ
810 FOR h=2 TO 11: IF a1$(h)="000000000" THEN 850 >LR
820 FOR g=1 TO 9: b1$=MID$(a1$(h), g, 1): b1=VAL(b1$) >QN
830 IF b1=1 THEN CALL &A00D, FN po(g+10, h), sp(a1): je(g+1 >JV
0, h)=a1-8
840 NEXT g >NJ
850 SOUND 1, INT(RND*600)+60, 5, 15: NEXT h: RETURN >QQ
860 READ mx, my, c, d >MG
870 CALL &A00D, FN po(mx, my), sp(12): je(mx, my)=5 >QC
880 CALL &A00D, FN po(c, d), sp(11): je(c, d)=4 >JJ
890 READ n: IF n=1 THEN READ nx, ny: CALL &A00D, FN po(nx, n >DC
y), sp(16): je(nx, ny)=6
900 READ x, y: CALL &A00D, FN po(x, y), sp(21) >JL
910 ac=21: fin=0: tp=20: m=0: sm=1: sn=1: tps=0: gan=0: ca=0 >ZR
920 GOSUB 2390: RETURN >RA
930 REM : : >AQ
940 REM : : >LH
950 REM : ROUTINE PRINCIPALE : >ZF
960 REM : : >LK
970 REM : : >AV
980 EVERY 50 GOSUB 2360: EVERY 15, 1 GOSUB 1790 >LZ
990 IF fin=1 THEN 2590 >PN
1000 IF n=1 THEN GOSUB 2080 >UD
1010 IF JOY(0)=8 THEN 1100 >TY
1020 IF JOY(0)=4 THEN 1240 >TA
1030 IF JOY(0)=1 THEN 1550 >TC
1040 IF JOY(0)=2 THEN 1620 >TC
1050 IF JOY(0)=20 OR JOY(0)=24 THEN 1690 >EK
1060 IF JOY(0)=16 THEN 1490 >UL
1070 IF INKEY$=CHR$(13) THEN FIN=1 >BC
1080 GOTO 990 >GC
1090 REM :: A DROITE :: >RP
1100 D1: IF ac=21 THEN CALL &A00D, FN po(x, y), sp(ac): CALL >DV
&A00D, FN po(x, y), sp(18): ac=18: GOTO 1220
1110 a=je(x+1, y): IF a=0 THEN 1130 >ZT
1120 IF a<5 THEN 1190 ELSE fin=1: E1: GOTO 990 >KW
1130 CALL &A00D, FN po(x, y), sp(ac): SOUND 1, 300, 5, 1, 1, 1, 1 >ZL
5
1140 FOR h=1 TO 3: CALL &A00D, FN po(x, y)+h, sp(17+h) >RU
1150 FOR t=1 TO 50: NEXT: CALL &A00D, FN po(x, y)+h, sp(17+h) >AN
)
1160 NEXT: x=x+1: CALL &A00D, FN po(x, y), sp(18): ac=18 >VP
1170 IF je(x, y+1)=0 OR je(x, y+1)>4 THEN 1370 ELSE E1: GO >CV
TO 990
1180 REM :: MONTE SUR UNE CAISSE :: >DQ
1190 IF je(x, y-1)=0 AND je(x+1, y-1)=0 THEN 1200 ELSE E1 >FD
: GOTO 990
1200 CALL &A00D, FN po(x, y), sp(ac): SOUND 1, 300, 5, 1, 1, 1, 1 >ZJ
5
1210 y=y-1: GOTO 1140 >NV
1220 FOR t=1 TO 100: NEXT: WHILE INKEY$<>"": WEND: E1: GOTO >EE
990
1230 REM :: A GAUCHE :: >RG
1240 D1: IF ac=18 THEN CALL &A00D, FN po(x, y), sp(ac): CALL >DU
&A00D, FN po(x, y), sp(21): ac=21: GOTO 1220
1250 a=je(x-1, y): IF a=0 THEN 1270 >ZF

```





```

1260 IF a<5 THEN 1330 ELSE fin=1:EI:GOTO 990 >KX
1270 CALL &A00D, FN po(x,y), sp(ac):SOUND 1,300,5,1,1,1 >ZR
5
1280 FOR h=1 TO 3:CALL &A00D, FN po(x,y)-h, sp(20+h) >RV
1290 FOR t=1 TO 50:NEXT:CALL &A00D, FN po(x,y)-h, sp(20+h) >AP
)
1300 NEXT:x=x-1:CALL &A00D, FN po(x,y), sp(21) >MV
1310 ac=21:IF je(x,y+1)=0 OR je(x,y+1)>4 THEN 1370 ELSE >KD
EI:GOTO 990
1320 REM :: MONTE SUR UNE CAISSE :: >DL
1330 IF je(x,y-1)=0 AND je(x-1,y-1)=0 THEN 1340 ELSE EI >FG
:GOTO 990
1340 CALL &A00D, FN po(x,y), sp(ac):SOUND 1,300,5,1,1,1 >ZP
5
1350 y=y-1:GOTO 1280 >NF
1360 REM :: IL TOMBE :: >RT
1370 DI:tb=0 >EE
1380 IF je(x,y+1)>4 THEN fin=1:GOTO 990 >FF
1390 tb=tb+1:CALL &A00D, FN po(x,y), sp(ac):SOUND 1,46+(t >KJ
b*10),3,5
1400 CALL &A052, FN po(x,y):FOR h=1 TO 3:CALL &A07B, sp(a >ZJ
c)
1410 FOR t=1 TO 10:NEXT:CALL &A06E, sp(ac):NEXT >PP
1420 CALL &A07B, sp(ac):y=y+1:EI >ZZ
1430 IF je(x,y+1)=0 OR je(x,y+1)>4 THEN 1380 ELSE 990 >TD
1440 REM :::::::::::::::::::: >BQ
1450 REM : : >TB
1460 REM : CREATION CAISSES : >XV
1470 REM : : >TD
1480 REM :::::::::::::::::::: >BV
1490 IF ac=18 THEN 1520 >PM
1500 IF je(x-1,y)=0 THEN 1510 ELSE 990 >CZ
1510 CALL &A00D, FN po(x-1,y), sp(10):je(x-1,y)=2:GOTO 99 >AK
0
1520 IF je(x+1,y)=0 THEN 1530 ELSE 990 >CB
1530 CALL &A00D, FN po(x+1,y), sp(10):je(x+1,y)=2:GOTO 99 >AH
0
1540 REM :: COUPS DE TETE :: >WT
1550 IF je(x,y-1)=2 THEN 1560 ELSE 990 >CM
1560 POKE &A020,0:CALL &A00D, FN po(x,y), sp(24) >NM
1570 CALL &A052, FN po(x,y):FOR h=1 TO 3:CALL &A05C, sp(a >ZR
c)
1580 FOR t=1 TO 50:NEXT:CALL &A06E, sp(24):SOUND 1,600,5 >QA
,12,,10:NEXT
1590 CALL &A05C, sp(ac):POKE &A020,&AE:y=y-1 >LC
1600 je(x,y)=0:GOTO 1170 >TA
1610 REM :: AVEC LES PIEDS :: >XW
1620 IF je(x,y+1)=2 THEN 1630 ELSE 990 >CF
1630 POKE &A020,0:CALL &A00D, FN po(x,y), sp(24) >NK
1640 CALL &A052, FN po(x,y):FOR h=1 TO 3:CALL &A07B, sp(a >ZQ
c)
1650 FOR t=1 TO 50:NEXT:CALL &A06E, sp(24):SOUND 1,800,5 >QY
,12,,10:NEXT
1660 CALL &A07B, sp(ac):POKE &A020,&AE:y=y+1 >KH
1670 je(x,y)=0:GOTO 1170 >TH
1680 REM :: CAISSES LATERALES :: >BZ
1690 IF ac=18 THEN 1720 >PR
1700 IF je(x-1,y+1)=0 THEN 1710 ELSE 990 >EN
1710 CALL &A00D, FN po(x-1,y+1), sp(10):je(x-1,y+1)=2:GOT >EC

```

```

0 990
1720 IF je(x+1,y+1)=0 THEN 1730 ELSE 990 >EQ
1730 CALL &A00D, FN po(x+1,y+1), sp(10):je(x+1,y+1)=2:GOT >EA
0 990
1740 REM :::::::::::::::::::: >BU
1750 REM : : >TE
1760 REM : LES ANIMAUX : >TG
1770 REM : : >TG
1780 REM :::::::::::::::::::: >BY
1790 DI:pa=pa+1:DN pa GOTO 1800,1820 >CW
1800 CALL &A090, FN po(mx,my), sp(13) >CK
1810 IF n=1 THEN CALL &A090, FN po(nx,ny), sp(16):EI:RETU >YD
RN ELSE EI:RETURN
1820 pa=0:CALL &A090, FN po(mx,my), sp(12) >HC
1830 IF n=1 THEN CALL &A090, FN po(nx,ny), sp(17):EI:RETU >YG
RN ELSE EI:RETURN
1840 REM :: DEPLACEMENT DE L'OURS :: >WJ
1850 IF je(mx,my+1)=0 OR je(mx,my+1)=4 THEN sm=3 >QZ
1860 ON sm GOTO 1870,1940,2010 >WP
1870 IF mx+1=x AND my=y THEN fin=1:GOTO 1900 >LV
1880 IF je(mx+1,my)=4 THEN fin=1:gan=1:GOTO 1900 >RR
1890 IF je(mx+1,my)<>0 THEN sm=2:RETURN >HW
1900 DI:CALL &A090, FN po(mx,my), sp(24):je(mx,my)=0 >VT
1910 mx=mx+1:CALL &A090, FN po(mx,my), sp(12):je(mx,my)=5 >BT
1920 EI:RETURN >GE
1930 REM :: SENS 2 :: >PK
1940 IF mx-1=x AND my=y THEN fin=1:GOTO 1970 >LC
1950 IF je(mx-1,my)=4 THEN fin=1:gan=1:GOTO 1970 >RZ
1960 IF je(mx-1,my)<>0 THEN sm=1:RETURN >HV
1970 DI:CALL &A090, FN po(mx,my), sp(24):je(mx,my)=0 >VA
1980 mx=mx-1:CALL &A090, FN po(mx,my), sp(12):je(mx,my)=5 >BC
1990 EI:RETURN >HB
2000 REM :: SENS 3 :: >PZ
2010 IF mx=x AND my+1=y THEN fin=1:RETURN >XJ
2020 IF je(mx,my+1)=4 THEN fin=1:gan=1:GOTO 2040 >QG
2030 IF je(mx,my+1)<>0 THEN sm=INT(RND*2)+1:RETURN >WC
2040 DI:CALL &A090, FN po(mx,my), sp(24):je(mx,my)=0 >VN
2050 my=my+1:CALL &A090, FN po(mx,my), sp(12):je(mx,my)=5 >BQ
2060 EI:RETURN >GA
2070 REM :: DEPLACEMENT DU TAUREAU :: >GJ
2080 DI:pas=pas+1:IF pas=4 THEN pas=0:GOTO 2090 ELSE EI >JY
:RETURN
2090 ON sn GOTO 2100,2160,2210,2260 >AF
2100 IF nx+1=x AND ny=y THEN fin=1:GOTO 2130 >LD
2110 IF je(nx+1,ny)=2 AND ca>0 THEN ca=ca-1:CALL &A090, >CG
FN po(nx+1,ny), sp(24):je(nx+1,ny)=0:SOUND 2,50,0,15,2,2
,31:GOTO 2150
2120 IF je(nx+1,ny)>0 THEN sn=INT(RND*3)+2:EI:RETURN >YG
2130 CALL &A090, FN po(nx,ny), sp(24):je(nx,ny)=0 >QH
2140 nx=nx+1:CALL &A090, FN po(nx,ny), sp(16):je(nx,ny)=6 >QX
:EI:RETURN
2150 sn=INT(RND*4)+1:EI:RETURN >AE
2160 IF nx=x AND ny+1=y THEN fin=1:GOTO 2190 >LV
2170 IF je(nx,ny+1)=2 AND ca>0 THEN ca=ca-1:CALL &A090, >CN
FN po(nx,ny+1), sp(24):je(nx,ny+1)=0:SOUND 2,50,0,15,2,2
,31:GOTO 2150
2180 IF je(nx,ny+1)>0 THEN sn=INT(RND*4)+1:EI:RETURN >YN
2190 CALL &A090, FN po(nx,ny), sp(24):je(nx,ny)=0 >RF
2200 ny=ny+1:CALL &A090, FN po(nx,ny), sp(16):je(nx,ny)=6 >QW

```



```

:EI:RETURN
221# IF nx-1=x AND ny=y THEN fin=1:GOTO 224# >LK
222# IF je(nx-1,ny)=2 AND ca># THEN ca=ca-1:CALL &A09# >CQ
FN po(nx-1,ny),sp(24):je(nx-1,ny)=0:SOUND 2,50,0,15,2,2
,31:GOTO 215#
223# IF je(nx-1,ny)># THEN sn=(INT(RND*4)+1):EI:RETURN >YL
224# CALL &A09#,FN po(nx,ny),sp(24):je(nx,ny)=# >RB
225# nx=nx-1:CALL &A09#,FN po(nx,ny),sp(16):je(nx,ny)=6 >QB
:EI:RETURN
226# IF nx=x AND ny-1=y THEN fin=1:GOTO 229# >LA
227# IF je(nx,ny-1)=2 AND ca># THEN ca=ca-1:CALL &A09# >CW
FN po(nx,ny-1),sp(24):je(nx,ny-1)=0:SOUND 2,50,0,15,2,2
,31:GOTO 215#
228# IF je(nx,ny-1)># THEN sn=(INT(RND*3)+1):EI:RETURN >YQ
229# CALL &A09#,FN po(nx,ny),sp(24):je(nx,ny)=# >RG
230# ny=ny-1:CALL &A09#,FN po(nx,ny),sp(16):je(nx,ny)=6 >QZ
:EI:RETURN
231# REM :::::::::::::::::::: >BM
232# REM : : >RJ
233# REM : GESTION COMPTEURS : >ZR
234# REM : : >TA
235# REM :::::::::::::::::::: >BR
236# tp=tp-1:a$=STR$(tp):A$=RIGHT$(A$,LEN(A$)-1):IF tp= >HK
# THEN fin=1:tps=1:ca=2#
237# IF tp>9 THEN z1=2#y1=25:cer=2:GOSUB 42#:RETURN >YJ
238# a$=#*a$:GOSUB 42#:RETURN >ZE
239# a$=STR$(ta):A$=RIGHT$(A$,LEN(A$)-1) >HR
240# IF ta>9 THEN ZL=2# ELSE ZL=21 >AN
241# YL=1:GOSUB 42#:RETURN >WM
242# IF sc=# THEN a$="#0000#":ZL=7:YL=1:GOSUB 42#:RETURN >AG
243# a$=STR$(sc):A$=RIGHT$(A$,LEN(A$)-1) >HM
244# IF sc<1# THEN ZL=11:YL=1:GOSUB 42#:RETURN >QW
245# IF sc<10# THEN ZL=1#YL=1:GOSUB 42#:RETURN >RM
246# IF sc<100# THEN ZL=9:YL=1:GOSUB 42#:RETURN >RV
247# ZL=#YL=1:GOSUB 42#:RETURN >BY
248# IF rec=# THEN a$="#0000#":ZL=3#:YL=1:GOSUB 42#:RETI >CL
RN
249# a$=STR$(rec):A$=RIGHT$(A$,LEN(A$)-1) >JD
250# IF rec<1# THEN ZL=34:YL=1:GOSUB 42#:RETURN >RL
251# IF rec<10# THEN ZL=33:YL=1:GOSUB 42#:RETURN >TD
252# IF rec<100# THEN ZL=32:YL=1:GOSUB 42#:RETURN >UX
253# ZL=31:YL=1:GOSUB 42#:RETURN >BB
254# REM :::::::::::::::::::: >BT
255# REM : : >TD
256# REM : FIN DE PARTIE : >TM
257# REM : : >TF
258# REM :::::::::::::::::::: >BX
259# IF tps=1 THEN tps=0:fin=#:mu=REMAIN(#):EVERY 9 GOS >WB
UB 185#:GOTO 9#
260# FOR h=# TO 3:mu=REMAIN(h):NEXT:IF gan=# THEN 27# >YU
261# REM :: GAGNE :: >PR
262# FOR t=1 TO 100#:NEXT >TW
263# ta=ta+1:BORDER 26:FOR h=1 TO 2#:CALL &A09#,FN po(m >NJ
x,my),sp(12)
264# sc=sc+5:GOSUB 242#:SOUND 1,h*10,10,14:BORDER h >VW
265# FOR t=1 TO 2#:NEXT:CALL &A09#,FN po(mx,my),sp(11) >YM
266# FOR t=1 TO 2#:NEXT t,h >UM
267# IF sc>rec THEN rec=sc:GOSUB 248# >EP
268# BORDER 0:GOTO 5# >QT

```

```

269# REM :: PERDU :: >PH
270# SOUND 4,1200,0,0,14,14:SOUND 4,0,20,2,,1# >MF
271# FOR h=100# TO 150# STEP 10#:SOUND 4,h,30,0,15,15:N >AF
EXT
272# FOR h=1 TO 200#:NEXT :vie=vie-1 >DT
273# IF vie=# THEN 274# ELSE 5# >XY
274# CLS #1:a$="PERDU @ @ @":ZL=15:YL=12:GOSUB 42# >UF
275# RESTORE 276# :FOR h=1 TO 1#:READ a,b:SOUND 1,425+( >MR
a*25),25*b,15:SOUND 1,0,2:NEXT
276# DATA 3,4,3,3,3,1,0,3,1,1,1,3,3,1,3,3,4,1,3,4 >NY
277# a$="UNE AUTRE ? : 0=N :":ZL=11:YL=14:GOSUB 42# >WZ
278# A$=INKEY$:IF A$="" THEN 278# >ZV
279# A$=UPPER$(A$):IF A$="N" THEN END >DD
280# IF A$="O" THEN 281# ELSE 278# >YN
281# A$=" # " :ZL=19:YL=1:cer=2:GOSUB 42# >JQ
282# ta=1:sc=#:GOSUB 242#:vie=4:GOSUB 239#:GOTO 5# >WD
283# REM :::::::::::::::::::: >BV
284# REM : : >TF
285# REM : DATA 12 TABLEUX : >WB
286# REM : : >TH
287# REM :::::::::::::::::::: >BZ
288# DATA 0,0,511,0,0,0,384,48,16,18,0,0,0,7,0,0,0,0, >WQ
#
289# DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 >MH
290# DATA 4,3,3,11,0,18,4 >RT
291# REM :: TAB 2 :: >NN
292# DATA 0,0,256,244,0,0,60,0,4,4,0,56,64,384,6,0,0,0, >YY
0,0
293# DATA 0,0,0,0,12,32,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 >TR
294# DATA 2,3,2,11,1,15,10,11,4 >WA
295# REM :: TAB 3 :: >NU
296# DATA 0,256,0,192,0,511,0,0,0,64,0,0,0,385,0,0,121, >AA
0,0,0
297# DATA 0,0,0,12,0,0,0,0,16,0,0,0,24,0,0,0,64,64,64 >VK
298# DATA 2,2,2,0,1,16,9,14,11 >WK
299# REM :: TAB 4 :: >NZ
300# DATA 8,456,0,48,16,16,4,4,0,16,0,3,0,4,48,0,260,12 >ET
,128,128
301# DATA 0,0,1,2,1,0,128,96,0,0,0,64,0,264,0,0,0,0,0,0 >VB
302# DATA 19,2,2,5,1,10,5,19,11 >WF
303# REM :: TAB 5 :: >NK
304# DATA 4,4,4,413,1,97,0,256,0,0,0,128,152,0,0,0,0,0, >Z2
64,0
305# DATA 392,504,504,96,0,0,0,0,0,0,0,0,0,16,0,0,0,96, >YG
0,0
306# DATA 4,2,19,0,1,15,3,18,11 >WT
307# REM :: TAB 6 :: >NQ
308# DATA 4,0,0,511,0,0,34,34,34,34,0,0,0,505,0,0,64,64 >CZ
,64,64
309# DATA 11,511,511,0,0,0,0,0,0,0,0,504,504,504,0,0,0,0, >AZ
0,0,0
310# DATA 2,2,2,11,1,0,2,7,11 >VE
311# REM :: TAB 7 :: >NL
312# DATA 0,367,0,3,4,20,60,64,256,0,0,256,0,256,128,64 >HB
,32,16,8,4
313# DATA 0,0,0,0,3,3,0,0,0,0,0,0,0,0,256,384,0,64,192, >YJ
448
314# DATA 10,2,16,11,0,2,2 >RK
315# REM :: TAB 8 :: >NR

```





# CIR·BAS

```

5 MEMORY &9700:CLS
7 LOCATE 1,10:PRINT " DATAS DE LA LIGNE      CORRECTS*:PEN 2
10 A=&9720:F=&A0C0:L=100:WHILE A<=F:FOR A=A TO A+15:READ C*:K=
VAL("&"+C*):S=S+K+65536*(S+K)32767):IF A<=F THEN POKE A,K
20 NEXT:READ D*:T=VAL("&"+D*):IF T<>S THEN PRINT CHR*(7);"Err
eur ligne":L:END ELSE L=L+5
25 LOCATE 20,10:PRINT L-5:WEND
100 DATA 00,44,09,00,00,53,08,02,01,01,01,02,01,04,8C,02,0142
105 DATA 04,00,06,00,04,01,00,02,44,04,89,02,44,01,06,02,03B3
110 DATA 51,02,06,01,44,23,00,06,01,99,E6,09,00,03,03,02,060B
115 DATA 01,00,00,44,01,06,09,73,01,1C,E9,66,00,06,09,22,0A10
120 DATA 00,06,88,00,01,04,A3,00,01,02,02,02,01,4C,08,02,0BA4
125 DATA 00,59,00,00,01,00,02,08,01,46,08,88,01,59,02,88,0DCB
130 DATA 02,09,01,A2,09,00,13,88,06,09,66,02,01,03,03,00,106B
135 DATA 88,00,00,02,03,E6,09,02,99,D6,2C,02,11,E6,09,00,153B
140 DATA 11,E6,09,00,99,D6,2C,02,03,E6,09,02,88,00,00,02,1A01
145 DATA 01,03,03,00,06,09,66,02,09,00,13,88,02,09,01,A2,1CA1
150 DATA 01,59,02,88,01,46,08,88,01,00,02,08,00,59,00,08,1ECB
155 DATA 01,4C,08,02,01,02,02,02,01,04,A3,00,00,06,88,00,205C
160 DATA 00,06,09,22,01,1C,E9,66,01,06,09,73,01,00,00,44,2461
165 DATA 00,03,03,02,01,99,E6,09,44,23,00,06,51,02,06,01,26B9
170 DATA 44,01,06,02,44,04,89,02,04,01,00,02,04,00,06,00,292A
175 DATA 01,04,8C,02,01,01,01,02,00,53,08,02,00,44,09,00,2A6C
180 DATA 73,E6,03,88,73,E6,03,88,00,00,00,11,00,00,00,11,2E56
185 DATA 91,33,63,51,91,33,63,51,28,28,00,44,14,00,28,44,325A
190 DATA 8C,3C,00,04,8C,28,28,2C,FC,BC,1C,01,FC,BC,1C,09,3840
195 DATA 8C,C0,03,02,4C,C0,03,02,00,00,00,00,00,00,00,3AA2
200 DATA 44,00,00,00,04,FC,39,00,44,8C,11,00,04,BC,39,00,3E29
205 DATA 04,BC,11,00,01,28,39,00,04,1C,11,00,40,28,11,00,4006
210 DATA 40,1C,11,00,40,28,11,00,40,28,11,00,01,00,11,00,4177
215 DATA 01,28,41,00,01,00,00,00,01,0C,D9,00,00,06,D9,00,43A7
220 DATA 00,00,00,00,00,00,00,00,01,03,C0,8C,01,03,C0,4C,4607
225 DATA 06,2C,7C,FC,02,2C,7C,FC,1C,14,14,7C,08,00,3C,7C,4AD7
230 DATA 88,14,00,28,88,00,14,14,A2,93,33,62,A2,93,33,62,4FDF
235 DATA 22,00,00,00,22,00,00,00,44,03,D9,83,44,03,D9,83,53C9
240 DATA 00,E6,09,00,00,E6,0C,02,00,00,00,02,00,82,14,02,5646
245 DATA 00,22,00,02,00,22,14,88,00,22,14,88,00,22,2C,88,58A4
250 DATA 00,22,14,88,00,22,2C,08,00,36,14,02,00,22,7C,08,5AA2
255 DATA 00,36,7C,08,00,22,7C,88,00,36,FC,08,00,00,00,88,5E44
260 DATA 79,CF,79,CF,F9,CF,F9,CF,79,F3,79,E7,3C,3C,3C,3C,6820
265 DATA E7,9E,E7,9E,E7,DE,E7,DE,F3,9E,F3,86,3C,3C,3C,3C,72DE
270 DATA 79,CF,79,CF,F9,CF,F9,CF,79,F3,79,E7,3C,3C,3C,3C,7CBA
275 DATA E7,9E,E7,9E,E7,DE,E7,DE,F3,9E,F3,86,00,00,00,00,8688
280 DATA 1F,3B,33,2F,85,F0,F0,2F,84,2D,3C,0F,2F,2D,0F,0F,8BAE
285 DATA 2F,5A,0F,0F,2F,85,F0,2F,0F,3F,85,2F,2F,3F,3F,84,9217
290 DATA 7A,7A,3F,0F,7A,3F,0F,0F,3F,7A,F0,1E,0F,0F,0F,0F,9ABC
295 DATA 2F,7A,0F,0F,2F,84,2D,0F,2D,5A,1E,1E,0F,0F,0F,0F,9FB1
300 DATA CC,CC,CC,CC,88,A2,51,44,88,A2,51,44,88,A2,51,44,A7CE
305 DATA 88,8A,45,44,CF,E7,DB,CF,88,A2,51,44,88,A2,51,44,8047

```

```

310 DATA 88,A2,51,44,88,A2,51,44,88,8A,45,44,CF,E7,DB,CF,88C0
315 DATA 88,A2,51,44,88,A2,51,44,88,A2,51,44,CC,CC,CC,CC,C12D
320 DATA 53,02,01,A3,F3,A3,53,F3,A3,A3,53,53,F2,53,A3,F1,C9C7
325 DATA 53,73,73,A3,13,13,13,A3,53,13,53,A3,53,F3,F3,A3,D0B7
330 DATA F3,03,53,F3,F3,03,53,F3,F3,A3,F3,F3,A3,F3,A3,F3,D8D7
335 DATA A3,53,03,F3,53,03,53,A3,53,F3,F3,A3,01,F3,F3,02,E3D4
340 DATA 53,02,01,A3,F3,A3,53,F3,A3,A3,53,53,F2,53,A3,F1,EC6E
345 DATA 53,73,73,A3,13,13,13,A3,53,83,F3,A3,53,F3,F3,A3,F49E
350 DATA F3,03,53,F3,F3,03,53,F3,F3,A3,F3,F3,A3,F3,A3,F3,FFBE
355 DATA A3,03,03,F3,53,07,03,A3,53,A3,53,A3,01,F3,F3,02,062F
360 DATA 00,47,02,00,00,CF,02,00,00,47,03,03,00,47,CF,67,0913
365 DATA 00,47,CF,67,00,47,CF,67,00,47,47,03,00,47,47,02,0D2E
370 DATA 00,47,83,00,00,47,83,00,00,47,83,00,00,01,C3,02,1052
375 DATA 00,01,C3,02,00,01,C3,02,00,01,C3,02,00,01,C3,02,136A
380 DATA 00,43,83,00,00,43,83,00,00,17,2B,00,00,17,AB,00,15FA
385 DATA 00,17,AB,00,00,17,23,00,01,3F,37,02,01,7F,FF,02,18F0
390 DATA 01,7F,88,02,01,D7,FF,02,01,C3,FF,02,01,C3,FF,02,1F90
395 DATA 01,C3,FF,02,01,6B,BF,02,01,7F,BF,02,01,17,2B,02,2408
400 DATA 23,01,02,13,23,12,21,13,9B,30,30,67,47,30,82,88,27C0
405 DATA 12,53,A3,21,12,53,A3,21,12,53,A3,21,30,A3,53,30,2C91
410 DATA 21,43,83,12,43,03,63,83,63,43,83,43,83,43,83,329E
415 DATA 83,C3,C3,43,43,C3,C3,83,01,03,03,02,01,F3,F3,02,3928
420 DATA 23,01,02,13,23,12,21,13,9B,30,30,67,47,30,82,88,3CE0
425 DATA 12,12,A3,21,12,03,A3,21,12,53,A3,21,30,A3,53,30,4120
430 DATA 21,43,83,12,43,C3,63,63,83,43,83,43,83,43,83,471D
435 DATA 83,43,83,43,43,C3,C3,83,01,F3,F3,02,00,00,00,00,4CDE
440 DATA 00,53,F3,02,01,F3,F3,02,01,F3,A3,03,53,F3,43,43,5375
445 DATA 53,F3,43,83,53,A3,C3,83,53,A3,C3,02,53,56,03,02,5A26
450 DATA 53,56,56,A9,F3,03,FC,A9,53,43,56,02,01,AB,A9,00,60AC
455 DATA 01,FF,AB,00,01,FF,02,00,01,FF,03,00,01,0F,0F,02,647D
460 DATA 00,53,F3,02,01,F3,F3,02,01,F3,A3,03,53,F3,43,43,6B14
465 DATA 53,F3,43,83,53,A3,C3,83,53,A3,C3,02,53,56,03,00,71C3
470 DATA 53,56,56,02,53,03,FC,A9,A3,83,FC,A9,03,56,FC,02,78E1
475 DATA 17,57,AB,00,3F,57,AB,01,1F,03,FF,07,07,08,0F,08,7C90
480 DATA 00,53,F3,02,01,F3,F3,02,01,F3,A3,03,53,F3,43,43,8327
485 DATA 53,F3,43,83,53,A3,C3,83,53,A3,C3,02,53,56,03,00,89D6
490 DATA 53,56,56,02,53,03,56,A9,F3,A9,83,A9,03,FE,56,02,904D
495 DATA 57,FF,AB,00,FF,AB,17,03,5F,03,3F,2F,07,0F,07,0B,950A
500 DATA 01,F3,A3,00,01,F3,F3,02,03,53,F3,02,83,83,F3,A3,9C71
505 DATA 43,83,F3,A3,43,C3,53,A3,01,C3,53,A3,01,03,A9,A3,A3D3
510 DATA 56,A9,A9,A3,56,FC,03,F3,01,A9,83,A3,00,56,57,02,AAE5
515 DATA 00,57,FF,02,00,01,FF,02,00,03,FF,02,01,0F,0F,02,AE64
520 DATA 01,F3,A3,00,01,F3,F3,02,03,53,F3,02,83,83,F3,A3,85CB
525 DATA 43,83,F3,A3,43,C3,53,A3,01,C3,53,A3,00,03,A9,A3,8D2C
530 DATA 01,A9,A9,A3,56,FC,03,A3,56,FC,43,53,01,FC,A9,03,C4AB
535 DATA 00,57,AB,2B,02,57,AB,3F,08,FF,03,2F,07,0F,07,0B,C87F
540 DATA 01,F3,A3,00,01,F3,F3,02,03,53,F3,02,83,83,F3,A3,CFE6
545 DATA 43,83,F3,A3,43,C3,53,A3,01,C3,53,A3,00,03,A9,A3,D747
550 DATA 01,A9,A9,A3,56,A9,03,A3,56,43,56,F3,01,A9,FD,03,DE6E
555 DATA 00,57,FF,AB,03,2B,57,FF,1F,3F,03,AF,07,08,0F,0B,E32F
560 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,E32F
565 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,E32F
570 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,E32F
575 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,E32F
580 DATA DD,5E,00,DD,56,01,DD,6E,02,DD,66,03,06,08,C5,E5,E9E9
585 DATA 06,02,1A,77,23,13,10,FA,E1,CD,26,BC,C1,10,EF,C9,F0DB
590 DATA 11,33,11,33,51,51,51,51,15,15,15,15,55,FF,00,00,F44F
595 DATA 00,22,11,22,00,A2,00,A2,00,2A,00,2A,55,FF,00,00,F790
600 DATA 11,33,11,11,00,51,00,51,00,2A,15,00,55,FF,00,00,FA2B

```



```

605 DATA 11,33,11,11,00,51,00,51,00,3F,00,15,55,FF,00,00,FCDB
610 DATA 00,11,00,33,00,F3,51,51,15,3F,00,15,00,55,00,00,FF72
615 DATA 11,33,11,00,51,00,00,F3,00,15,15,15,55,FF,00,00,029E
620 DATA 11,33,11,11,51,00,51,F3,15,15,15,15,55,FF,00,00,0641
625 DATA 11,33,00,11,00,51,00,A2,00,2A,15,00,55,00,00,00,081D
630 DATA 11,33,11,11,51,51,51,F3,15,15,15,15,55,FF,00,00,0C11
635 DATA 11,33,11,11,51,51,51,F3,00,15,00,15,55,FF,00,00,0FDB
640 DATA 00,00,00,00,00,00,00,00,00,A2,00,00,00,22,00,00,109F
645 DATA 00,00,00,00,00,00,00,00,00,A2,00,00,00,22,11,00,1174
650 DATA 00,00,00,00,00,00,00,00,00,00,A2,00,22,00,00,1238
655 DATA 00,00,00,00,00,00,00,00,00,00,A2,00,22,11,00,130D
660 DATA 00,00,00,00,44,00,00,00,00,00,00,00,00,00,00,1361
665 DATA 00,00,04,04,44,44,00,44,00,A2,00,00,00,22,00,00,1501
670 DATA 00,00,00,00,00,00,00,00,00,A2,00,00,00,22,00,00,16E5
675 DATA 04,0C,04,0C,44,44,44,44,51,F3,51,51,11,11,00,00,1A1D
680 DATA 04,0C,04,0C,44,44,44,44,51,A2,51,51,11,33,00,00,1D26
685 DATA 04,0C,04,0C,44,00,44,00,51,00,51,00,11,33,00,00,1EB4
690 DATA 04,06,04,0C,44,44,44,44,51,51,51,51,11,33,00,00,2168
695 DATA 04,0C,04,0C,44,00,44,00,51,A2,51,00,11,33,00,00,2398
700 DATA 04,0C,04,0C,44,00,44,00,51,A2,51,00,11,00,00,00,2595
705 DATA 04,0C,04,0C,44,44,44,00,51,00,51,51,11,33,00,00,2788
710 DATA 04,04,04,04,44,44,44,44,51,F3,51,51,11,11,00,00,2AE0
715 DATA 04,0C,00,00,00,00,00,00,A2,00,A2,11,33,00,00,2D90
720 DATA 04,0C,00,00,00,00,00,00,A2,00,A2,11,22,00,00,302F
725 DATA 04,04,04,04,44,44,44,00,51,A2,51,51,11,11,00,00,334A
730 DATA 04,00,04,00,44,00,44,00,51,00,51,00,11,33,00,00,34C8
735 DATA 04,04,04,0C,44,CC,44,44,51,51,51,51,11,11,00,00,37D6
740 DATA 04,04,04,04,44,CC,44,CC,51,F3,51,51,11,11,00,00,3C0E
745 DATA 04,0C,04,04,44,44,44,44,51,51,51,51,11,33,00,00,3EBE
750 DATA 04,0C,04,0C,44,44,44,44,51,F3,51,00,11,00,00,00,4194

```

```

755 DATA 04,0C,04,0C,44,44,44,44,51,51,51,51,11,33,00,22,446E
760 DATA 04,0C,04,0C,44,44,44,44,51,A2,51,51,11,11,00,00,4755
765 DATA 04,0C,04,0C,44,00,44,00,51,F3,00,51,11,33,00,00,49D6
770 DATA 04,0C,04,0C,00,00,00,00,A2,00,A2,00,22,00,00,4C6C
775 DATA 04,04,04,04,44,44,44,44,51,51,51,51,11,33,00,00,4F14
780 DATA 04,04,04,04,44,44,44,44,51,51,51,51,00,22,00,00,519A
785 DATA 04,04,04,04,44,44,44,44,51,F3,51,F3,11,11,00,00,5564
790 DATA 04,04,04,04,44,44,00,00,00,A2,51,51,11,11,00,00,57EA
795 DATA 04,04,04,04,44,44,00,00,00,A2,00,A2,00,22,00,00,5A70
800 DATA 04,0C,00,04,00,44,00,00,00,A2,51,00,11,33,00,00,5C87
805 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,5C87
810 DATA 7C,C6,06,67,D0,D5,11,50,C0,19,D1,C9,00,DD,5E,00,63EC
815 DATA DD,56,01,DD,6E,02,DD,66,03,06,10,C5,E5,06,04,1A,6997
820 DATA AE,77,13,23,10,F9,E1,CD,00,A0,C1,10,EE,C9,00,00,70D1
825 DATA 00,00,DD,6E,00,DD,66,01,06,10,AF,77,CD,00,A0,10,7619
830 DATA F9,C9,7C,D6,00,67,CB,74,C0,D5,11,00,3F,19,D1,C9,7F23
835 DATA CC,C6,DD,6E,00,DD,66,01,22,50,A0,C9,2A,50,A0,CD,8706
840 DATA 42,A0,CD,42,A0,CD,42,A0,CD,42,A0,22,50,A0,2A,50,8EB1
845 DATA A0,DD,5E,00,DD,56,01,CD,19,A0,C9,2A,50,A0,CD,00,95C6
850 DATA A0,CD,00,A0,CD,00,A0,CD,00,A0,C3,6B,A0,00,00,9C7B
855 DATA DD,5E,00,DD,56,01,DD,6E,02,DD,66,03,06,10,C5,E5,A33D
860 DATA 06,04,1A,00,77,13,23,10,F9,E1,CD,00,A0,C1,10,EE,A924
865 DATA C9,00,00,00,00,00,00,00,00,00,00,00,00,00,00,AGED
870 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,AGED
900 CLS:PRINT " FICHER LANGAGE MACHINE CORRECT..."
910 PRINT:PRINT:PRINT
920 PRINT "UNE TOUCHE POUR LA SAUVEGARDE..."
930 CALL @BB10
940 SAVE "cirbin",b,&9720,&9A0

```

# 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.

BANQUE		SELECTION	MODIFIER	RELEVES	BILAN		
BANQUE		SUPPORT/SORTIE		CRITERE			
SUPPORT / SORTIE							
C	DATE	TYPE	No	ORDRE/OBJET	VERIT	CREDIT	SOLDE
*	09/05/87	REN. CHEQ		VERSEMENT INITIAL		10000.00	10000.00
*	20/05/87	VIREMENT		SALAIRE MOIS DE MAI		12000.00	22000.00
*	16/06/87	CHIQUE	10000	AMSTRAD CRC 0120	2000.00		20000.00
*	27/06/87	CHIQUE	10001	CONSULTATION MEDICIN	110.00		21900.00
*	08/07/87	CHIQUE		RENTRE	250.25		21649.75
*	20/08/87	VIREMENT		REMBOURSEMENT S.S.		187.00	21836.75
*	11/09/87	PRELEVEM		TELEPHONE	874.25		20962.50
*	01/09/87	RET. CB/C			1000.00		19962.50
*	02/09/87	PRELEVEM		ELECTRICITE	3033.45		16929.05
*	02/09/87	REN. LIGN		RETRAIT EPARGNE		200.00	17129.05
*	02/09/87	CHIQUE	10002	SUPERMARCH	503.47		16625.58
*	03/09/87	RET. CARTE		POINT ARGENT	400.00		16225.58
*	04/09/87	CHIQUE	10003	RESTORANT AUTO	350.00		15875.58
*	05/09/87	CHIQUE		ESSENCE	220.12		15655.46
15 06		BANQUE TEST		No 12345678901	Occur. 1 K	Solde r	15655.46

**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 .....  
 Signature ..... FIN .....

MANDAT  CONTRE RBT (+ 35 Frs)

# PRATIQUE DE L'ASSEMBLEUR

Emmanuel GUILLARD

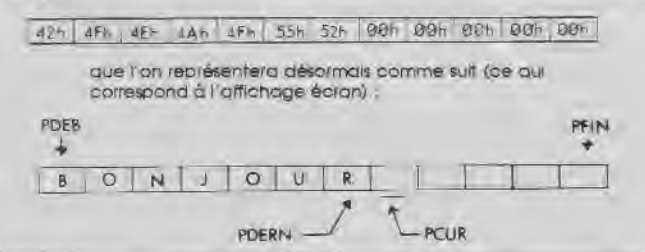
Ce mois-ci, nous allons réaliser en assembleur une fonction que l'on pourrait tout de suite qualifier d'indispensable. Et il est vrai qu'on ne peut passer à côté de ce qui est utilisé dans 90 % des programmes existant : un éditeur ligne, qui sert à saisir du texte, votre nom à la fin d'un jeu, etc...

Tous les éditeurs ne sont pas, hélas, performants. Certains ont des possibilités que d'autres n'ont pas. Tous, en fait, se complètent. Celui que nous allons réaliser ce mois-ci possède les principales fonctions de tout éditeur ligne qui se respecte : saisie (évidemment, il vaut mieux commencer par là !), déplacement dans la ligne avec les flèches latérales, fonctions BACKSPACE, DELETE, INSERTION/RECouvreMENT, et validation par ENTER.

## LE PRINCIPE

Supposons que l'on veuille saisir une ligne de 12 caractères. Nous allons commencer par définir, en mémoire, un buffer (= une zone) de 12 octets Initialisés à 0. Supposons que l'on saisisse le mot BONJOUR. Après la saisie supposée correcte (c'est-à-dire sans avoir eu recours à des fonctions d'effacement, de modification, pour les fontes de frappe...) notre buffer devra être affecté de la manière suivante :

Fig. 1



PDEB : pointeur de début du buffer  
 PFIN : pointeur de fin du buffer  
 PDERN : pointeur sur le dernier caractère de la chaîne  
 PCUR : pointeur de position du curseur

Par l'intermédiaire de schémas, je vais maintenant détailler le fonctionnement de chacune des fonctions de l'éditeur :

- **Appui sur la flèche gauche** : le curseur se déplace vers la gauche

Fig. 2



Après le premier appui, PCUR va se trouver égal à PDERN, mais en fait, peu importe. Ce qui est important, c'est de comparer PCUR à PDEB afin de ne pas sortir de la zone du buffer délimitée par PDEB et PFIN (et par la suite, de ne pas faire de saletés à l'écran !).

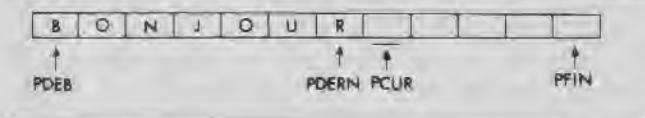
• **Appui sur la flèche de droite :**

Cette fois le curseur se déplacera vers la droite, et PCUR ne devra pas aller au-delà de PDERN+1, soit après le dernier le caractère de la chaîne saisie.

• **Fonction DELETE (touche CLR)**

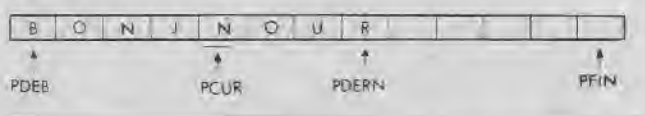
Cette fonction efface le caractère situé sous le curseur. Supposons que nous ayons saisi la chaîne BONJOUR. Il faut corriger le N frappé par mégarde. On a le buffer suivant :

Fig. 3



Il va falloir appuyer 4 fois sur la flèche gauche pour déplacer le curseur sous le "N" :

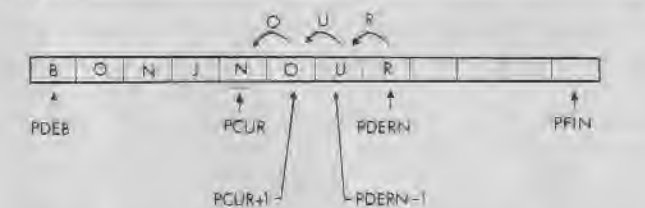
Fig. 4



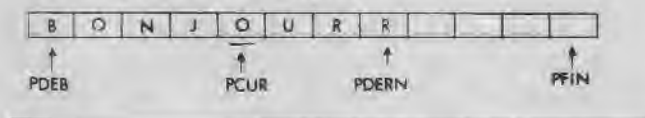
Voici maintenant ce qui va se passer lors de l'appui sur la touche DELETE :

1. Recopie de la zone située entre PCUR+1 et PDERN dans la zone située entre PCUR et PDERN-1, ce qui va "écraser" le N frappé par mégarde :

Fig. 5



on obtient le buffer suivant : il y a un R de trop | on l'annulera en écrivant la valeur 0 à l'adresse PDERN



2. Il reste alors à décrémenter PDERN et le buffer et ses pointeurs associés seront correctement positionnés, ceci après avoir effacé le R en trop.

Fig. 6



• **Fonction BACKSPACE (touche DEL) :**

Cette fonction est quasiment similaire à la fonction DELETE. Lors d'un appui sur la touche BACKSPACE, le caractère situé à gauche du curseur est effacé, et les caractères de droite sont décalés à gauche. Donc, appuyer sur la touche DEL revient à appuyer sur la

flèche gauche, puis sur la touche CLR. Donc, lorsqu'un appui sur la touche DEL aura été détecté, on exécutera dans cet ordre : le module d'appui sur la flèche gauche, puis le module d'appui sur la touche CLR. Et voilà comment on économise de la mémoire sans se fatiguer !

#### • Mode INS/REC :

INS/REC signifie INSERTION/RECOUVREMENT, à savoir :

- En mode INSERTION, les caractères tapés en milieu de ligne décaleront les caractères situés à leur droite, afin de ne pas les écraser.
- En mode RECOUVREMENT, les caractères tapés écraseront ceux qui étaient précédemment à leur place.

#### Mode INSERTION :

ex. : on a tapé BONJUR

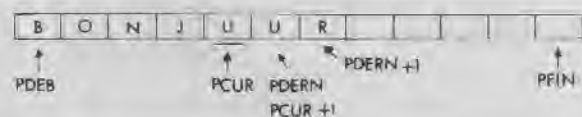
Fig. 7



Une fois le curseur positionné correctement, comme ci-dessus, on va taper le "O" qui manque :

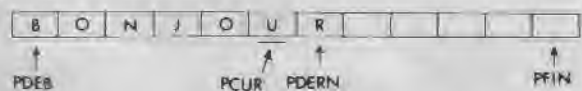
1. Décalage des caractères de droite : copie des caractères de la zone PCUR à PDERN dans la zone PCUR+1 à PDERN+1 (si PCUR ≤ PDERN 1).

Fig. 8



2. Ecriture du O à la position PCUR :

Fig. 9



#### Mode RECOUVREMENT :

Dans ce mode, le caractère tapé est simplement écrit à la position du curseur, et le pointeur PCUR est incrémenté, ainsi que PDERN si on se trouvait en fin de chaîne.

#### Remarque :

Ce mode d'insertion/recouvrement n'est pas toujours utilisé. En fait, tout dépend du contexte d'utilisation de l'éditeur. Par exemple, pour entrer votre nom dans un jeu, ou autre, il ne sera pas forcément nécessaire d'avoir recours à l'insertion, le mode recouvrement sera amplement suffisant.

Par contre, dans un éditeur pleine page (style traitement de texte, etc...), il sera intéressant de passer d'un mode à l'autre.

Le module principal donné en exemple, vous permet donc d'utiliser ces deux modes, commutables par la touche (COPY). La ligne tapée sera validée par la touche (RETURN). Le mode Insertion/Recouvrement, et c'est là le défaut majeur qui empêche de l'utiliser partout, devra être visualisé. Plusieurs méthodes sont possibles, j'ai retenu la suivante :

sur la première ligne d'écran sera affiché en permanence le mode, et à chaque appui sur (COPY), le sous-programme ira changer l'affichage en première ligne.

Cependant, pour une utilisation plus spécifique de l'éditeur, n'utilisant pas les 2 modes, il faudra d'office, choisir si l'on préfère l'insertion au recouvrement, et initialiser, positionner d'office l'éditeur dans le mode choisi, en ne programmant évidemment que le nécessaire.

## LE CURSEUR

Lorsque vous êtes en BASIC, et que vous saisissez un programme, ou autre, vous voyez le curseur se déplacer (tiens en passant, l'éditeur LOCOMOTIVE est positionné en mode INSERTION, vous aviez vu ?) suivant votre frappe.

Par contre, en assembleur, notre éditeur va utiliser la routine BBØ6, qui attend "bêtement" la frappe d'un caractère sans rien faire de plus, et la routine BB5A qui affiche, "bêtement" aussi, le caractère qu'on lui envoie.

On va donc avoir à "gérer" le curseur. En fait, "gérer" est un bien grand mot car en fait il y aura juste à afficher un pavé à la position du curseur qui est en fait géré (!) automatiquement par l'ordinateur (respirez !...).

Que l'on soit sous l'éditeur BASIC, ou à afficher un caractère en assembleur avec une routine quelconque, après l'affichage, le curseur est TOUJOURS déplacé vers la droite, qu'il soit visible (en BASIC) ou non.

Vous devez donc avoir compris qu'il va nous falloir afficher nous-même ce curseur. Soit. La première pensée qui vient à l'esprit serait d'afficher bêtement un caractère ASCII, le n°143, par exemple, à la position du curseur. Soit, mais avez-vous pensé au fait que le curseur pouvait se trouver en plein milieu de ligne ?

## L'INVERSION VIDEO

En BASIC, lorsque c'est le cas, soit que vous déplacez votre curseur sur une ligne de texte, le caractère situé sous le curseur (qui lorsqu'il ne l'était pas était affiché par exemple en blanc sur fond noir) va se trouver affiché en noir sur fond blanc. C'est ce que l'on appelle l'inversion vidéo.

Donc, en ce qui nous concerne, nous n'allons pas utiliser le caractère n°143, mais bel et bien l'inversion vidéo (un bien grand mot !).

Sa mise en œuvre est très simple : pour afficher un caractère en inverse vidéo, il suffit d'activer le mode



d'inversion vidéo, d'afficher le caractère en question, et de désactiver le mode d'inversion. Bien sûr, on peut très bien ne pas le désactiver ; tous les caractères envoyés à l'écran seront donc affichés en Inverse Vidéo.

## COMMENT L'ACTIVER ET LE DESACTIVER ?

A question simple, réponse on ne peut plus simple : Vous savez que l'AMSTRAD possède un jeu de 256 caractères, numérotés de 0 à 255. Vous savez aussi que seuls les caractères 32 à 255 sont affichables, et que les 32 autres (n°0 à 31) sont appelés caractères de contrôle, et qu'ils permettent des opérations spéciales et diverses.

Parmi les fonctions disponibles, on retrouve avec le caractère n°24, l'échange des couleurs PAPER et PEN\* : le noir sur fond blanc devient le blanc sur fond noir : c'est ça l'inversion vidéo !

Pour l'activer, il suffit d'envoyer à l'écran le caractère n°24.

Par exemple, sous BASIC, faites :

PRINT CHR\$(24), puis ENTER

et vous voyez le mot READY s'afficher en inverse vidéo, ainsi que tout ce que vous taperez ensuite. Justement, tapez maintenant quelque chose à l'écran, ce qui vous passe par la tête. Maintenant, déplacez le curseur à l'intérieur de la ligne. Vous voyez votre caractère s'afficher en... inverse vidéo inversée, si je puis me permettre l'expression !

Vous aurez donc facilement compris que pour désactiver le mode Inversion vidéo, il suffit d'envoyer à nouveau ce même caractère n°24 à l'écran.

Tapez donc : PRINT CHR\$(24), puis ENTER et vous voilà revenu en mode normal.

Donc, si maintenant vous tapez :

PRINT CHR\$(24); "HELLO"; CHR\$(24),

vous devriez voir le mot HELLO en inverse vidéo, et votre frappe revenue en mode normal. Plgé ?

En revenant à notre curseur, pour l'afficher, il faudra en fait afficher le caractère correspondant du buffer en Inverse vidéo, A CONDITION QUE L'ON SE TROUVE ENTRE LES POINTEURS PDEB et PDERN INCLUS. Car si l'on se trouve en fin de ligne, donc en PDERN+1, le contenu du buffer à cet endroit sera la valeur 0, qui affichée ou non en Inverse vidéo n'aura aucun effet visuel à l'écran. Il faudra donc, si on lit une valeur nulle dans notre buffer, afficher un espace (caractère n°32) en Inverse vidéo, afin de retrouver le pavé de notre bon vieux LOCOMOTIVE !

Néanmoins, on n'en a pas pour autant terminé avec ce damné curseur. Supposons le schéma de la figure 4, où notre curseur pointe sur le N du mot BONJOUR. Ce N va donc être affiché par la séquence suivante :

- Afficher (caractère 24) : activation Inv. vidéo
- Afficher "N" : affichage du n
- Afficher (caractère 24) : désactivation Inv. vidéo

Ici un problème va se poser : notre pointeur de CURSEUR, PCUR, lui n'a pas bougé, et est toujours situé

sous le N. Par contre, le pointeur réel de l'écran après l'affichage du N s'est déplacé d'un caractère vers la droite :



Vous voyez que cela risque d'entraîner de sérieux problèmes lors d'un futur déplacement de curseur, frappe de caractère, ou autre. Heureusement, il existe un caractère de contrôle fort utile, le caractère n°8 qui déplace le curseur écran réel d'un caractère vers la gauche, ce qui remettra tout dans l'ordre (tiens en passant, retenir aussi le n°9 qui lui nous déplace vers la droite !).

Donc le module curseur se résumera à ces quelques actions :

1. Lire (PCUR) caractère
2. Si caractère = 0 alors caractère ← 32
3. Ecrire (ECRAN) caractère n°24
4. Ecrire (ECRAN) caractère
5. Ecrire (ECRAN) caractère n°24
6. Ecrire (ECRAN) caractère n°8

C'est tout ! Remarquez que c'est très simple : une lecture et quatre écritures, pas de modification de pointeur ou autre, la simplicité même !

Nous venons ensemble de réaliser une analyse conceptuelle d'un problème qui semblait au départ assez simple, non ? En fait, l'analyse conceptuelle, c'est tout ce qui existe entre l'idée d'un projet et l'écriture des algorithmes. C'est le travail, indispensable (et j'espère que cet exemple vous l'aura fait remarquer) sur le papier, où l'on envisage tous les cas de problèmes possibles pouvant survenir lors d'un traitement, et dont on déduit les différentes solutions. Reste maintenant à passer aux algorithmes, puis à la programmation, qui, si l'analyse que nous avons faite est parfaite, d'une part couleront de source, et d'autre part donneront un fonctionnement excellent à 100 %, sans qu'on ait à se poser de question sur le fonctionnement ou non d'un détail quelconque, du fait que tout a été envisagé lors de l'analyse. Je vous laisse donc cogiter sur le reste : les algorithmes et la programmation, à moins que vous ne préféreriez attendre un mois sans vous fatiguer pour découvrir la solution du problème posé. Néanmoins, je termine en vous donnant la liste des touches utilisées pour l'éditeur, et le code ASCII correspondant qu'elles renvoient. Bon amusement ! ■

Touche	Code	Fonction
←	242	Déplacement vers la gauche
→	243	Déplacement vers la droite
CLR	16	Effacement du caractère situé sous le curseur
DEL	127	Effacement du caractère précédent le curseur
COPY	224	Commutation mode Insertion/Recouvrement
ENTER	13	Validation et sortie du programme

## LES OPERATEURS LOGIQUES AND, OR, XOR ET NOT

G. DUBUS

Vous êtes débutant et aimeriez apprendre quand et comment manipuler ces entités mystérieuses ?

Vous programmez et pensez qu'il n'y a rien de plus à tirer des opérateurs logiques que ce qu'en dit le manuel Amstrad ? Tout le monde pourra trouver son profit dans ce qui va suivre, car ces opérateurs se prêtent à maintes applications, de la plus simple à la plus complexe. Encore faut-il comprendre l'essence même des opérations effectuées.

### APPROCHE INTUITIVE

**A**ND, OR et NOT signifient respectivement ET, OU et NON, ce qui veut dire exactement ce que ça a l'air de vouloir dire. Le XOR est un OU exclusif, nous l'examinerons plus loin.

Dans leurs applications les plus faciles à appréhender intuitivement, ces opérateurs sont associés à des tests de condition (IF..THEN..ELSE) ou à des boucles fondées sur un test (WHILE..WEND). Prenons un exemple simple :

SI NOUS SOMMES SAMEDI OU DIMANCHE ALORS REPOS  
SINON AU BOULOT

et traduisons-le en Basic (A) :

```
10 IF JOUR$="samedi" OR JOUR$="dimanche" THEN
PRINT"REPOS" ELSE PRINT"AU BOULOT !"
```

Tapez ensuite en mode direct : JOUR\$="samedi";GOTO 10 et RETURN. Pourquoi GOTO et pas RUN ? Ah que voilà une question qu'elle est bonne ; parce que RUN efface d'abord les variables en mémoire, donc JOURS et la valeur qu'on vient de lui affecter.

Le programme affiche «REPOS», OK. Exemple suivant :

SI NOUS SOMMES LUNDI ET SI JE SUIS MALADE ALORS  
REPOS SINON AU BOULOT

Traduit en Basic (B) :

```
10 IF JOUR$="lundi" AND MALADE$="oui" THEN
PRINT"REPOS" ELSE PRINT"AU BOULOT !"
```

Tapez maintenant JOUR\$="lundi";MALADE\$="oui";GOTO 10. Réponse : REPOS. Essayez MALADE\$="non";GOTO 10. Réponse : AU BOULOT ! Passons à autre chose :

SI NOUS NE SOMMES PAS SAMEDI OU DIMANCHE ALORS  
AU BOULOT SINON REPOS

Essayons de traduire cela en Basic (C) :

```
10 IF NOT JOUR$="samedi" OR NOT JOUR$="dimanche"
THEN PRINT"AU BOULOT !" ELSE PRINT"REPOS"
```



Tapez `JOUR$="samedi";GOTO 10`. Réponse : AU BOULOT ! Ce n'est pas vraiment ce qu'on attendait.

Si l'exécution n'est pas correcte, c'est qu'il y a une erreur de logique dans l'expression de la condition globale. Il aurait fallu poser :

**SI NOUS NE SOMMES PAS (SAMEDI OU DIMANCHE) ALORS AU BOULOT SINON REPOS**

Ou bien encore :

**SI NOUS NE SOMMES PAS SAMEDI ET SI NOUS NE SOMMES PAS DIMANCHE ALORS AU BOULOT SINON REPOS**

Ce qui donne respectivement en Basic (D et E) :

`10 IF NOT(JOUR$="samedi" OR JOUR$="dimanche") THEN PRINT"AU BOULOT !" ELSE PRINT"REPOS"`

`10 IF NOT JOUR$="samedi" AND NOT JOUR$="dimanche" THEN PRINT"AU BOULOT !" ELSE PRINT"REPOS"`

Essayez à nouveau avec `JOUR$="samedi"` : cette fois c'est correct.

## APPROCHE LOGIQUE

Comme leur nom l'indique, les opérateurs logiques relient deux opérands (ou arguments) qui sont des **valeurs logiques** : VRAI ou FAUX. Par exemple, lorsque nous affectons à `JOUR$` la valeur «samedi», la comparaison avec `JOUR$="samedi"`? retournera la valeur logique VRAI. Ce sera la valeur logique FAUX pour la comparaison avec `JOUR$="dimanche"`? Globalement, les choses se passent bien ainsi, et peu importe pour l'instant la manière dont l'interpréteur Basic s'y prend pour effectuer les comparaisons, et la nature exacte des valeurs logiques VRAI et FAUX.

Quelques exemples d'expressions logiques, avec leur résultat :

FAUX ET FAUX = FAUX	FAUX OU FAUX = FAUX	
FAUX ET VRAI = FAUX	FAUX OU VRAI = VRAI	NON VRAI = FAUX
VRAI ET FAUX = FAUX	VRAI OU FAUX = VRAI	NON FAUX = VRAI
VRAI ET VRAI = VRAI	VRAI OU VRAI = VRAI	

Tout ceci est évident ? Je suis heureux de vous l'entendre dire. Alors examinons sous cet angle nos 5 expressions logiques précédentes, repérées par les lettres A à E.

- Expression A

Avec `JOUR$="samedi"`, nous obtenons :

**SI VRAI OU FAUX ALORS REPOS SINON AU BOULOT**  
VRAI OU FAUX = VRAI, donc REPOS (la réponse qui suit ALORS).

- Expression B

Avec `JOUR$="lundi"` et `MALADE$="oui"`, nous avons :

**SI VRAI ET VRAI ALORS REPOS SINON AU BOULOT**  
VRAI ET VRAI = VRAI, donc REPOS.

Avec `MALADE$="non"`, l'expression devient :

**SI VRAI ET FAUX ALORS REPOS SINON AU BOULOT**

VRAI ET FAUX = FAUX, donc AU BOULOT (la réponse qui suit SINON).

- Expression C

Avec `JOUR$="samedi"`, cela donne :

**SI NON VRAI OU NON FAUX ALORS AU BOULOT SINON REPOS**

Autrement dit :

**SI FAUX OU VRAI ALORS AU BOULOT SINON REPOS**

FAUX OU VRAI = VRAI, donc réponse : AU BOULOT (celle qui suit ALORS).

Même si ce n'est pas ce que nous voulions obtenir, c'est parfaitement logique. C'est simplement l'expression qui était mal formulée.

- Expression D

Toujours avec `JOUR$="samedi"` :

**SI NON(VRAI OU FAUX) ALORS AU BOULOT SINON REPOS**

VRAI OU FAUX = VRAI et NON(VRAI) = FAUX, donc réponse : REPOS.

- Expression E

**SI NON VRAI ET NON FAUX ALORS AU BOULOT SINON REPOS**

**SI FAUX ET VRAI ALORS AU BOULOT SINON REPOS**

FAUX ET VRAI = FAUX, donc réponse : REPOS.

Même s'il s'agit du même signe «=», il ne faut pas confondre l'instruction d'affectation (celle qui nous permet d'attribuer une valeur à une variable) et l'opérateur relationnel d'égalité. Pour mieux vous mettre le nez sur le problème, tapez :

`A=5;PRINT A;A=5;A<>5` et RETURN. Réponse : 5 -1 0.

Je vous expliquerai ce -1 et ce 0 en temps utile. Les OPERATEURS RELATIONNELS (=, >, <, >=, <= et <>) font toujours partie d'une condition, annoncée généralement (mais pas obligatoirement) par IF ou WHILE. L'ambiguïté du signe «=» est une source regrettable de confusion, même si l'interpréteur Basic, lui, ne s'y trompe pas. Les autres langages évitent généralement ce piège.

Dans l'évaluation d'une expression conditionnelle, l'interpréteur calcule d'abord le résultat logique des opérations relationnelles s'il y en a (c'est-à-dire compare ces expressions avec la valeur courante des variables concernées), puis effectue sur ces résultats les opérations logiques éventuelles, et enfin retourne un résultat unique. Si c'est VRAI, ce sont les instructions précédées de «alors» (THEN) qui sont exécutées. Dans le cas contraire, l'interpréteur exécute ce qui suit «si-non» (ELSE), ou en son absence les instructions de la ligne suivante.

Comme en algèbre, le résultat des expressions entre parenthèses est calculé en priorité. Au moindre doute sur l'ordre des opérations, usez et abusez des parenthèses. Sans parenthèses, l'ordre de priorité est le suivant : (=,>,<,>=,<=,<>) > NOT > AND > OR > XOR.



Je ne vous ai pas encore expliqué le OU exclusif (XOR). Il retourne VRAI seulement si les 2 opérandes logiques ont des valeurs opposées, en d'autres termes si un seul d'entre eux est VRAI :

FAUX XOR FAUX = FAUX  
 FAUX XOR VRAI = VRAI  
 VRAI XOR FAUX = VRAI  
 VRAI XOR VRAI = FAUX

Au XIX<sup>e</sup> siècle, le mathématicien anglais George BOOLE eut l'idée de rapprocher les mathématiques et la logique. Il formalisa les opérations logiques telles que celles expliquées ici, et créa une algèbre, l'algèbre de Boole, permettant de les représenter. C'est pourquoi, dans certains langages Informatiques, les opérateurs logiques sont également appelés opérateurs booléens, et les valeurs VRAI et FAUX sont dites de type booléen (True et False en Pascal, .T. et .F. en dBase).

En algèbre de Boole, aux valeurs logiques VRAI et FAUX sont respectivement liées les valeurs numériques 1 et 0. Ceci permet de représenter les opérations logiques sous forme de TABLES DE VERITE, d'un aspect tout à fait analogue à nos bonnes vieilles tables de multiplication.

AND	1	0	0	1	0	XOR	1	0	0	1	NOT	1	0
1	1	0	1	1	1	1	0	1	0	1	0	0	1
0	0	0	0	0	0	0	1	0	1	0	1	1	0

On prend un opérande à gauche sur une ligne, l'autre en haut sur une colonne, et on lit le résultat de l'opération logique à l'intersection de la ligne et de la colonne.

L'opérateur NOT ne porte que sur un seul opérande ; il transforme 1 (VRAI) en 0 (FAUX) et vice versa : c'est un **Inverseur logique**.

ATTENTION ; les valeurs 1 et 0 ne sont pas des valeurs décimales, ce sont des VALEURS BINAIRES. L'algèbre de Boole est une algèbre binaire. Avant d'aller plus loin, il est nécessaire d'expliquer (ou de réexpliquer ?) de quoi il retourne.

## LES NOMBRES BINAIRES

Le sujet a déjà été traité maintes fois dans cette revue, dans d'autres contextes. Aussi serai-je le plus bref possible.

Un nombre est représenté au moyen de chiffres : 0 à 9 en décimal, 0 à 1 en binaire. Quel que soit le système employé, chaque chiffre utilisé dans la représentation d'un nombre possède un **poids, qui est fonction de son rang, en partant de la droite**.

Exemple : 123 en décimal représente 1 centaine + 2 dizaines + 3 unités. 3, premier en partant de la droite, possède le poids le plus faible (unités), et 1, au troisième rang (celui des centaines), possède le poids le plus fort. Par ailleurs, vous conviendrez que :  
 $123 = (1 \cdot 10^2) + (2 \cdot 10^1) + (3 \cdot 10^0)$

Autrement dit, la valeur d'un nombre décimal est la somme des produits de chacun des chiffres par son poids. Et le poids d'un chiffre, toujours en décimal, c'est 10 élevé à la puissance n, n étant le rang du dit chiffre dans le nombre, de la droite vers la gauche, **en comptant à partir de zéro**.

Accrochez vos ceintures et écartez vos mégots, on aborde l'étape suivante. Le système décimal est dit en base 10, et le système binaire en base 2 (d'où son nom). Oyez : quel que soit le système employé, le nombre de chiffres différents possibles est égal à la base. Par ailleurs, le poids d'un chiffre dans un nombre est égal à la base, élevée à une puissance qui est le rang du chiffre compte depuis la droite du nombre, le premier rang étant zéro.

Un ordinateur ne sait gérer que le système binaire (2 chiffres possibles : 1 et 0). Pour lui, si le courant passe c'est 1, s'il ne passe pas c'est 0. Simple, non ? Ces 1 et ces 0 sont appelés des bits (pas de ricanements au fond de la classe, SVP), abréviation de **binary digits**, ce qui signifie chiffres binaires.

En informatique, les bits sont regroupés par 8, ce qui donne un octet. Le CPC est un ordinateur 8 bits, car il manipule les données sous forme de nombres binaires de 8 bits. Les compatibles PC, Atari, Amiga, etc... sont des 16 bits, ils traitent 2 octets à la fois.

Examinons un octet, donc 8 bits alignés. Le poids du bit le plus à droite est égal à la base, soit 2, élevée à la puissance du rang, soit 0. Le poids du bit le plus à gauche est 2 élevé à la puissance du rang, donc 7 (0 à 7 font bien 8 rangs, OK ?). Voyons cela de plus près :

Rang	7	6	5	4	3	2	1	0
Poids	27	26	25	24	23	22	21	20
	128	64	32	16	8	4	2	1

Vous voyez en bas les poids traduits en valeurs décimales.

*Première remarque* : chaque poids est égal au précédent (depuis la droite) multiplié par 2, c'est-à-dire par la base. C'est pareil en base 10 (décimal) : le poids du rang des milliers est égal au poids du rang des centaines multiplié par 10.

*Seconde remarque* : chaque poids est égal à la somme de tous les précédents (toujours en partant de la droite) **plus 1**.

Pour calculer la valeur décimale d'un octet, c'est très simple : on multiplie chaque bit par son poids, et on fait la somme.

Exemple : 01101110

$$(0 \cdot 128) + (1 \cdot 64) + (1 \cdot 32) + (0 \cdot 16) + (1 \cdot 8) + (1 \cdot 4) + (1 \cdot 2) + (0 \cdot 1)$$

$$64 + 32 + 8 + 4 + 2 = 110$$

Il y a des raccourcis commodes, dans certains cas particuliers :

$$00000000 = 0$$

$$11111111 = 128 + (128 - 1) = 255$$

$$00011111 = 16 + (16 - 1) = 31$$

Pour les 2 derniers exemples, rappelez-vous la seconde remarque.

Tout ceci pour vous faire bien comprendre la structure des nombres binaires. Je ne vous propose pas d'exercices de conversion binaire-décimal et décimal-binaire, ça ne présente aucun intérêt car l'Amstrad fait cela 1000 fois plus vite que vous. Ce que je vous re-



commande vivement, par contre, c'est d'appeler la fonction **BINS** pour une conversion décimal-binaire, et d'examiner avec soin le résultat, chaque fois que vous avez à résoudre un problème où c'est la représentation binaire des nombres qui est importante. Et c'est bien le cas avec les opérateurs logiques.

Quelques définitions utiles avant de passer à la suite. On dit qu'un bit est **mis** lorsqu'il est à 1. Les entiers occupent 2 octets, soit 16 bits. Le 16ème bit, c'est-à-dire le bit 15, le plus à gauche, est le **bit de signe**; les autres sont les **bits de données**. Si le bit de signe est mis (s'il est égal à 1), l'entier est négatif; sinon l'entier est positif. C'est pourquoi la valeur d'un entier ne peut être comprise qu'entre 32767 et -32768 inclus.

Une telle représentation des **entiers signés** est dite **représentation en complément à 2**. Pour rendre négatif un entier, le système inverse un par un tous les bits (1 devient 0, et 0 devient 1 : complément à 1), puis ajoute 1 (complément à 2).

1 décimal = 00000000 00000001 binaire  
 inversion : 11111111 11111110  
 + 1 = 11111111 11111111 binaire = -1 décimal

Notez que -1 est représenté en binaire par un nombre où tous les bits sont mis. Nous aurons l'occasion d'y revenir.

## BAS LES MASQUES

Imaginez un nombre binaire écrit sur une feuille de papier, et une réglette en carton percée d'une petite fenêtre magique. Vous faites glisser la réglette sur le nombre, jusqu'à encadrer dans la fenêtre le bit choisi, et vous pouvez alors déterminer son état (1 ou 0), l'inverser, ou le forcer au choix à 1 ou à 0, quel que soit son état. C'est ce qu'on appelle un **MASQUE** et c'est exactement ce qu'on peut réaliser avec les opérateurs logiques.

Ils vous donnent en effet accès à des opérations individuelles sur les bits, **directement à partir du Basic**. C'est-y pas fabuleux ?

Précisons tout de suite que les opérations logiques sur les nombres ne fonctionnent que sur les entiers, pas sur les réels (qui sont représentés sur 5 octets au lieu de 2). Examinons quelques exemples (sur un seul octet pour plus de simplicité), (Voir figure 1).

Et sur 2 octets, par exemple avec des nombres négatifs :

NOT -1 = 0 NOT 0 = -1 NOT 1 = -2

-1 : 11111111 11111111      1 : 00000000 00000001  
 0 : 00000000 00000000      -2 : 11111111 11111110

-2 est le complément à 1 de 1 (inversion de chaque bit). Notez que, quel que soit l'entier A, **NOT A = -(A+1)** : c'est une autre manière de représenter le complément à 1. Si on ajoute 1 à -2, on obtient bien -1 (complément à 2 de 1), en décimal comme en binaire. Toujours quel que soit A, complément à 2 de A = complément à 1 de A, plus 1 = -(A+1)+1 = -A. D'accord ? Par ailleurs :

NOT(A XOR -1) = A avec ou sans parenthèses, car (NOT A) XOR -1 = A

En fait, NOT A = A XOR -1, donc NOT(NOT A) = (A XOR -1) XOR -1 = A

Après ce petit hors-d'œuvre, vous avez tous compris comment ça tourne. Mais si, mais si. Nous allons maintenant pouvoir passer en revue ce qu'il est possible de faire concrètement avec les opérateurs logiques.

Soit A un entier signe *quelconque*, A étant représenté sur 2 octets, soit 16 bits, appelons N le bit sur lequel nous voulons travailler. N peut prendre toute valeur de 0 à 14; gardons-nous bien de toucher au bit 15 (le 16ème, c'est-à-dire le bit de signe), sinon gare aux résultats aussi étranges qu'inattendus. Comme nous l'avons vu précédemment, 2<sup>N</sup> représente le poids du bit N, ou sa valeur s'il est mis.

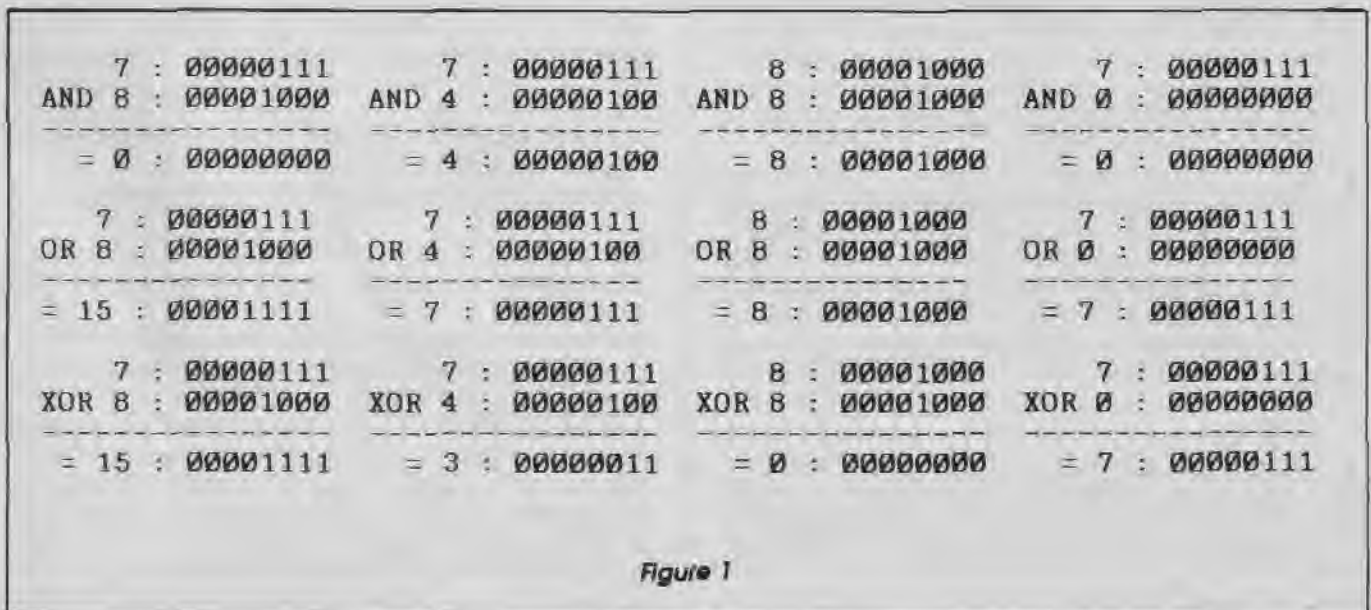
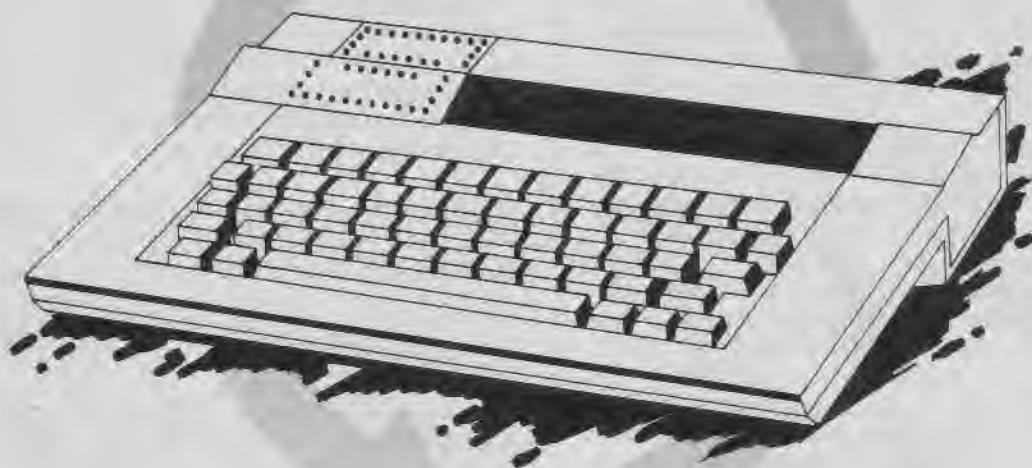


Figure 1



\* **A OR 2<sup>N</sup> force le bit N de A à 1, quel que soit son état antérieur (1 ou 0).** A + 2<sup>N</sup> ne fait la même chose que si le bit N est à 0.

\* **A AND (-1-2<sup>N</sup>) force le bit N de A à 0, quel que soit son état antérieur (1 ou 0).** A - 2<sup>N</sup> ne fait la même chose que si le bit N est à 1.

\* **A AND 2<sup>N</sup> = 2<sup>N</sup> si le bit N est à 1, sinon A AND 2<sup>N</sup> = 0.**

\* **A XOR 2<sup>N</sup> inverse le bit N (et lui seul).** Le bit N est forcé à 1 si son état antérieur est 0. Il est forcé à 0 si son état antérieur est 1.

Soit A une variable entière, initialisée à 0 (ou pas initialisée du tout, ce qui en Basic revient au même).

L'instruction **A = A XOR 1**, chaque fois qu'elle est exécutée, fait passer la valeur de A de 0 à 1, et vice versa. Cela peut être commode pour un flag à 2 valeurs, du genre oui/non, vrai/faux. Un flag (drapeau en anglais) est un *indicateur d'état*, une variable pouvant prendre un petit nombre de valeurs bien définies, généralement 2 : drapeau levé ou drapeau baissé.

Mais l'intérêt principal de cette instruction est de permettre, avec une économie maximum de moyens, la construction d'une **BASCULE**.

Un logiciel ayant 2 modes de fonctionnement utilise une bascule (toggle dans les produits d'Outre-Manche) pour passer d'un mode à l'autre, par exemple le choix entre joystick et clavier dans un jeu. En fait, une vraie bascule permet de changer de mode à tout moment, autant de fois qu'on veut ; exemple : la bascule Insertion/recouvrement dans les traitements de texte.

Comment ça marche ? Une touche ou une combinaison de touches particulières, dans le module de scrutation du clavier, appelle le sous-programme contenant l'instruction **A = A XOR 1**, puis le sous-programme de configuration ad hoc, par exemple par l'instruction :

**ON A+1 GOSUB ...**

En graphisme, XOR a également une application très importante : le mode XOR graphique (voir l'article de Guy Poi dans Amstar & CPC No 27 de novembre-décembre 1988). Ce mode est instauré par **PRINT CHR\$(23); CHR\$(1);:TAG** pour les caractères ASCII, ou en fixant à 1 le 4ème paramètre des instructions graphiques (sur 6128 seulement). Entre autres effets du plus haut intérêt, il permet, dans les animations, d'effacer un objet en le réaffichant simplement à son emplacement courant, et de le faire passer au choix devant ou derrière un autre objet.

## LES MYSTERES DE LA LOGIQUE

Initialisons d'abord 2 variables : **JOUR\$="samedi"** et **A%=5**.

### 1) Expression conditionnelle annoncée par IF ou WHILE

a) Comparaison explicite d'une expression relationnelle avec la valeur de la ou des variables concernées.

Exemples :

```
IF JOUR$="samedi" THEN PRINT"VRAI" ELSE PRINT"FAUX"
IF A%=5 THEN PRINT"VRAI" ELSE PRINT"FAUX"
```

La comparaison retourne l'entier -1, c'est-à-dire VRAI.

Exemples :

```
IF JOUR$="dimanche" THEN PRINT"VRAI" ELSE
PRINT"FAUX"
IF A%=1 THEN PRINT"VRAI" ELSE PRINT"FAUX"
```

La comparaison retourne l'entier 0, c'est-à-dire FAUX.

VRAI est associé à 1 et FAUX à 0, mais ce 1 et ce 0 sont des **bits**. L'entier 0 est un nombre binaire dont tous les bits sont à 0. L'entier -1 est un nombre binaire dont tous les bits sont à 1 ; plus vrai que ça tu meurs ! C'est ainsi que le système (qui ne sait manipuler que des 1 et des 0, enfonçons le clavier...) représente le résultat logique d'une expression conditionnelle, en mettant



tous les bits d'une variable interne à 1 ou à 0, selon que ce résultat est VRAI ou FAUX.

Dans une expression conditionnelle complexe comprenant des expressions relationnelles liées par AND, OR ou XOR (avec ou sans NOT, avec ou sans parenthèses), ceux-ci effectuent tout simplement des opérations logiques sur les **résultats numériques** (-1 ou 0) des expressions relationnelles. Le résultat final unique est soit -1, soit 0, ce qui détermine l'orientation du programme consécutive au test.

**b) Comparaison implicite** (sans opérateur relationnel) de chaque bit de la valeur d'une variable ou du résultat d'une expression algébrique (Ex.  $A+B-3$ ) ou logique (Ex.  $A \text{ AND } B$ ), avec 1.

Exemple :

**IF A%=5 THEN PRINT «VRAI» ELSE PRINT «FAUX»**

- Si le résultat de l'expression est égal à 0 (tous les bits à 0), le résultat de la comparaison est 0 (FAUX). C'est ce que nous obtenons puisque  $A\%=5$ .

- Si le résultat de l'expression n'est pas nul (au moins 1 bit à 1), le résultat de la comparaison est -1 (VRAI).

Exemple :

**IF NOT A% THEN PRINT «VRAI» ELSE PRINT «FAUX»**

- Si  $A\%=-1$ ,  $\text{NOT } A\%=0$ , donc le résultat de la comparaison est 0 (FAUX).

- Pour toute valeur de  $A\%$  différente de -1,  $\text{NOT } A\%$  est différent de 0, donc le résultat est -1 (VRAI).

Cette instruction n'affichera FAUX que si  $A\%=-1$ , et VRAI dans tous les autres cas.

Pour un fonctionnement sans bavures de ce genre de test, il vaut mieux que  $A\%$  soit un flag à 2 valeurs possibles, -1 ou 0. Initialisé à -1, le flag est «mis», à l'endroit ad hoc du programme, par l'instruction  $A\%=\text{NOT } A\%$  : c'est une autre forme de BASCULE.

Vous l'avez compris, la comparaison implicite est :

**IF (expression/variable) <> 0 THEN...**

**ATTENTION : NE JAMAIS CONFONDRE LE RESULTAT NUMERIQUE DE L'EXPRESSION ET LE RESULTAT DE LA COMPARAISON.**

## 2) Expression conditionnelle sans IF ni WHILE, mais avec opérateur(s) relationnel(s)

Soit  $A\%=5$ .  $\text{PRINT } A\%$  affichera 5, OK. Mais nous avons vu plus haut que  $\text{PRINT } A\%=5$  affiche -1, et que  $\text{PRINT } A\%<>5$  affiche 0. Dans ce cas, les règles de la syntaxe indiquent à l'interpréteur que le signe « $=$ » n'est pas l'opérateur d'affectation, mais un opérateur relationnel faisant partie d'une expression dont il doit calculer le résultat logique, par comparaison avec la valeur de la variable contenue dans l'expression. Comme  $A\%$  vaut 5, le résultat est -1 (VRAI) dans le premier cas, et c'est bien ce qu'affiche la commande  $\text{PRINT}$ . Même chose pour l'autre exemple, à ceci près qu'il n'y a aucune ambiguïté sur la nature de l'opérateur. Ces expressions particulières entrent dans la construction d'instructions aussi concises que puissantes, exprimées sous forme de relations «d'allure algébrique», incluant en fait des conditions. Les paramètres en entrée sont de type numérique ou chaîne de caractères ; le résultat

en sortie est toujours numérique. Quand les utilise-t-on ?

- Lorsqu'il est impossible d'employer IF...THEN ou WHILE ;
- dans une formule algébrique,
- dans une fonction utilisateur.
- Lorsqu'on recherche la conclusion dans l'écriture : cas des tests multiples (avec plusieurs IF...THEN...ELSE en cascade).

Exemple : fonction utilisateur testant un bit quelconque d'une variable entière, directement à partir du Basic. En entrée : A est un entier signe, N est le numéro du bit à tester (0 à 15). Résultat en sortie : état du bit N (1 ou 0).

**DEF FNbit(A,N)=-((A AND 2^N)>0)**

Application : **A=132:N=7:PRINT FNbit(A,N) -> 1.**

Explication :  $2^N = 2^7 = 128$

$132 \text{ AND } 128 = 128$

$128 > 0 = \text{VRAI} = -1$ , donc résultat = 1.

Nata : le signe «-» peut être remplacé par la fonction ABS (valeur absolue).

Dans cet exemple (et la plupart du temps), attention aux parenthèses ! Avec les mêmes variables, **-(A AND 2^N>0)** donnera :

$2^N > 0 = \text{VRAI}$ , donc résultat = -1

$A \text{ AND } -1 = 132$

Résultat = -132 !

**-A AND 2^N>0** donnera aussi comme résultat -132.

## APPLICATIONS

Un programme doit parfois déterminer si un nombre entier, a priori inconnu, est pair ou impair. Un nombre pair, divisé par 2, fournit un dividende entier, donc sans reste. Il y a plusieurs solutions :

	Pair	Impair
$A/2-\text{INT}(A/2) =$	0	0,5
$(A/2=\text{INT}(A/2))=$	-1	0
$A \text{ MOD } 2 =$	0	1
$A \text{ AND } 1 =$	0	1

La dernière instruction est la plus rapide : son principe est différent : elle teste simplement le bit 0.

Avant d'envoyer des données à l'imprimante, un programme peut s'assurer qu'elle est prête à les recevoir. Par le port &F532 (en hexadécimal), l'imprimante envoie à l'ordinateur un nombre sur 2 octets qui représente son état. Le bit 6 de ce nombre correspond à la broche BUSY (occupé). Le test est très simple :

**A=INP(&F532):B=A AND 64 (64 = 2^6)**

$B = 64 \rightarrow$  bit 6 à 1  $\rightarrow$  OFF LINE (tampon plein, ou pas de papier)

$B = 0 \rightarrow$  bit 6 à 0  $\rightarrow$  ON LINE (imprimante prête à recevoir les données).

Le seul point noir au tableau, c'est que le bit 6 est aussi à 0 lorsque l'imprimante est éteinte, et qu'il n'existe aucun moyen, pour un programme, de faire la différence avec l'état ON LINE. No comment, Mister SUGAR ?



Le LISTING 1 est une sorte de Shoot'em Up. Principe : apparition, en un point aléatoire de l'écran (ligne 70), d'un «monstre» agité de mouvements imprévisibles : en un temps limité, il faut descendre le bestiau en déplaçant une mire située initialement au centre de l'écran (60). Ce n'est pas PROHIBITION, mais cela montre ce qu'on peut faire rien qu'avec du Basic, en utilisant à bon esclent les opérations logiques, et quelques autres astuces.

En 90, boucle d'attente classique. En 140, boucle principale du programme.

Le mode XOR graphique (nécessaire à l'animation) est instauré en 100 pour l'affichage des caractères (monstre = CHR\$(225)) et en 340 pour celui du viseur. Sur 464, il faut supprimer le 4ème paramètre de la commande MOVE en 340 et en 420, et ajouter POKE &B291,1.

Déplacements du monstre : horizontal en 210, vertical en 220. Affichage en 270, effacement en 200.

Déplacements du viseur : horizontal en 350, vertical en 360. Affichage en 420, effacement en 340. Test de la touche FIRE en 370 et 430 (retourne 0 ou 16) et de la concordance du point visé avec la surface sensible de la cible en 480 (on peut réduire cette surface pour rendre le jeu plus difficile).

Ni le viseur ni le monstre ne peuvent sortir de l'écran (380 à 410, et 230 à 260).

Le principe est très simple. Voyons d'abord l'interprétation des actions sur le joystick pour le déplacement du viseur.

La fonction JOY(0) retourne un nombre sur un octet, dont les bits représentent les actions exercées sur le joystick 1 : pour le joystick 2, ce sera JOY(1). Seuls les 5 premiers bits (0 à 4) sont significatifs.

No du bit	4	3	2	1	0
Poids	16	8	4	2	1
Action	Fire	Droite	Gauche	Bas	Haut

Toutes les combinaisons sont bien sûr possibles. Exemple : JOY(0)=26 -> Fire + Droite + Bas. Comment tester l'état des différents bits ? C'est très facile et très rapide avec AND.

Si nous poussons le joystick à Droite, JOY(0) AND 8 donne 8, et  $8*2=16$ . La position d'affichage est donc décalée de 16 points à droite. Si nous le poussons à Gauche, le déplacement est retranché de la position courante. Fonctionnement analogue pour Haut et Bas. Comme je suis flemmard, j'ai gardé le même principe pour les déplacements du monstre. En 180, le système tire un nombre m au hasard, entre 1 et 10 inclus. J'ai décidé arbitrairement de la correspondance suivante :

No du bit	3	2	1	0
Poids	8	4	2	1
Action	Haut	Bas	Droite	Gauche

Le nombre m est ensuite testé par un AND, comme pour JOY(0). Il y a une valeur de m correspondant à un déplacement nul : c'est 3 (Gauche et Droite à la fois).

Plus qu'une démo, le LISTING 2 est un véritable utilitaire,

il affiche un catalogue déroulant de la disquette, pour l'USER de votre choix, et vous offre le lancement ou le chargement d'un programme Basic ou Binaire, ou l'affichage du contenu d'un fichier ASCII. Bien entendu, les commandes non appropriées au type de fichier sélectionné sont détournées à la source. Si vous changez de disquette, tapez «U» puis le No d'User pour appeler le nouveau catalogue.

Le choix du fichier et le choix de la commande se font à l'aide des mêmes touches (flèches haut et bas). Les touches flèches gauche et droite servent à basculer d'un mode à l'autre.

Au lancement, on est en mode «Choix Commande». En 90, on initialise donc **bascule** à 1 et **x** à 25 (coordonnée d'affichage du mode actif). Si on change de mode (330 et 400-410), **bascule** prend la valeur 0 et **x** la valeur 10. Si on change à nouveau, **bascule** = 1 et **x** =  $10+15 = 25$  (car le test **bascule** = 1 retourne la valeur -1 -> VRAI).

En fonction de la valeur de **bascule**, flèche haut et flèche bas font défiler la barre de choix, soit sur le catalogue (680 à 750), soit sur le menu de commandes (790 à 820).

La boucle de scrutation du clavier (290 à 340) utilise la fonction INKEY() plus rapide que INKEY\$, mais plus chatouilleuse aussi : d'où les CLEAR INPUT qu'on trouve un peu partout. Il faut même parfois ralentir (boucle d'attente en 800 et surtout en 400, Indispensable). INKEY() retourne -1 tant que la touche correspondante n'est pas pressée, 0 si elle est pressée seule, 32 avec SHIFT, 128 avec CONTROL, et 160 avec SHIFT et CONTROL. NOT INKEY() sera donc FAUX si INKEY() retourne -1, et VRAI dans tous les autres cas : l'instruction précédée de THEN sera alors exécutée.

Dès qu'on appuie sur ENTER ou RETURN (290), on sort de la boucle de scrutation, et la ligne 350 est exécutée. Après exécution de la commande «Afficher ASCII», ou après une erreur sur le type de fichier, on retourne à la boucle sans fin. La comparaison sous-entendue dans l'instruction **WHILE sansfin** retourne toujours vrai, puisque **sansfin** n'est jamais annulée. Ceci évite un GOTO.

La variable lop (indice option) sert à la fois :

- à désigner la coordonnée verticale d'affichage de l'option courante du menu et son libellé (790),
- à calculer la nouvelle coordonnée et le nouveau libellé après action sur flèche haut ou flèche bas (810-820),
- à déterminer ce que doit faire le programme après validation (350 et 880).

La ligne 860 est destinée à vous éviter des ennuis si le programme lancé doit faire des allocations mémoire (SYMBOL AFTER ou MEMORY).

Vous pouvez appeler ce programme «M.BAS» et le sauver dans toutes vos disquettes. Il vous suffira de taper **RUN»M»** pour disposer d'une fonction Catalogue (très) améliorée. Si vos disquettes sont bourrées en vrac, pourquoi ne pas répartir les fichiers par familles, dans les différents Users ?





### LISTING1 -BAS

```

1# 'LISTING 1
2# '
3# BORDER 1:INK #,15:INK 1,#:INK 2,4:INK 3,6:MODE 2
4# DEFINT a-z:RANDOMIZE TIME
5# ENV 1,5,3,8,3#,#,4,5,-3,8:ENV 2,127,5,1:ENT -1,4,-2,
2,4,2,2:ENT 2,6#, -2#,2,3#,2#,2
6# h=32#:v=2#
7# hc=CINT(RND*624):vc=CINT(RND*(399-15))+15
8# PRINT"Vous avez 2# secondes pour abattre le monstre,
sinon c'est lui qui vous détruiraBon courage...":PRINT
:PRINT"Appuyez sur FIRE quand vous etes pret"
9# WHILE (JOY(#) AND 16)=#:WEND:MODE 1
10# PRINT CHR$(23);CHR$(1);:TAG
11# GRAPHICS PEN 2:GOSUB 27#:GOSUB 42#
12# EVERY 5.4,# GOSUB 32#
13# AFTER 1###,1 GOSUB 54#
14# fin=#:WHILE fin=#:GOSUB 18#:WEND:END
15# '
16# ' MONSTRE
17# '
18# DI:=INT(RND*10)+1
19# GRAPHICS PEN 2
20# MOVE hc,vc:PRINT CHR$(225);
21# hc=hc+(# AND 2)*8-(# AND 1)*16
22# vc=vc+(# AND 8)*2-(# AND 4)*4
23# IF hc>624 THEN hc=hc-48
24# IF hc<# THEN hc=hc+48
25# IF vc>399 THEN vc=vc-48
26# IF vc<15 THEN vc=vc+48
27# MOVE hc,vc:PRINT CHR$(225);
28# EI:RETURN
29# '
30# ' JOYSTICK
31# '
32# IF JOY(#)=# THEN FOR i=1 TO 15#:NEXT ELSE GOSUB 34#
33# RETURN
34# MOVE h,v,3,1:MOVER -16,#:DRAWR 32,#:MOVER -16,16:DR
AWR #,-32
35# h=h+(JOY(#) AND 8)*2-(JOY(#) AND 4)*4
36# v=v-(JOY(#) AND 2)*8+(JOY(#) AND 1)*12
37# f=(JOY(#) AND 16)
38# IF h>631 THEN h=631
39# IF h<8 THEN h=8
40# IF v>391 THEN v=391
41# IF v<8 THEN v=8
42# MOVE h,v,3:MOVER -16,#:DRAWR 32,#:MOVER -16,16:DR
AWR #,-32

```

```

43# IF f THEN MOVE h-6,v+6:GRAPHICS PEN 1:PRINT CHR$(42 >XM
);:SOUND 1,35##,1#,15,,18:GOSUB 48#
44# RETURN
45# '
46# ' GAGNE ?
47# '
48# IF NOT(h)hc+2 AND h<hc+14 AND v<vc-2 AND v>vc-14) T >JY
HEN RETURN
49# z=REMAIN(1):DI:SOUND 129,#:FOR i=1 TO 10#:NEXT:SOUM >WM
D 2,2###,#,#,1,2:SOUND 1,16,#,#,2,1
50# GOSUB 56#:PAPER 3:INK 3,6,16:CLS:PRINT"BRAVO. Il vo >ZP
us restait";ROUND(z/5#,i);"secondes":CLEAR INPUT:PAPER
#:fin=1:RETURN
51# '
52# ' PERDU !
53# '
54# DI:SOUND 129,#:FOR i=1 TO 4:SOUND 1,1##,2#,15:SOUND >YP
1,134,2#,15:SOUND 1,1#,1,4#,15:FOR j=1 TO 10#:NEXT j,i
55# GOSUB 56#:PAPER 1:PEN 3:CLS:PRINT"TROP TARD, LE NON >QV
STRE VOUS A EU !":CLEAR INPUT:PAPER #:PEN 1:fin=1:RETUR
N
56# TAGOFF:PRINT CHR$(23);CHR$(#);:RETURN

```

### LISTING2 -BAS

```

1# ' LISTING 2
2# '
3# ' Initialisation
4# '
5# OPENOUT"bidon":MEMORY &8E7#:CLOSEOUT
6# FOR a=&8E71 TO &8E7A:READ a#:k=VAL("&"+a#):POKE a,k: >DP
NEXT
7# DATA DD,SE,##,DD,56,##,CD,9B,BC,C9
8# DEFINT a-z
9# DIM fic$(64),choix$(1):user=#:bascule=1:x=25:sansfin >PB
=1:iop=1
10# tit#="SMARTCAT (C) G. DUBUS & CPC Infos"
11# choix$(#)="Choix Fichier ":choix$(1)="Choix Command >HC
e":mes1#="CHR$(242)+ " ou "+CHR$(243)+ " : "+choix$(#)+"/"
+choix$(1):mes2#="CHR$(24#)+ " ou "+CHR$(241)+ " : Silecti
on":mes3#="ENTER ou RETURN : Validation":mes4#="U : Dif
inir USER courant"
12# mes5#=" Appuyez sur COPY pour abandonner":mes6#=" >TT
"Appuyez sur une touche pour la suite...":mes7#=" Retou
r au MENU : une touche quelconque"
13# l$(1)="LOAD *l$(2)="RUN *l$(3) >UC
="Afficher ASCII"
14# MODE 1:BORDER 15:INK #,15:INK 1,#:INK 2,4:INK 3,2 >VJ
15# WINDOW#1,2,18,8,17:PAPER#1,#:PEN#1,1 >HT
16# WINDOW#2,24,39,6,12:PAPER#2,#:PEN#2,3 >JH
17# WINDOW#3,2,39,2,21:PAPER#3,#:PEN#3,1 >HT
18# '

```



```
19# ' Programme principal >TA :DRAW gx1,gy1:RETURN
20# ' >RC 65# ' >TB
21# WHILE sansfin >MB 66# ' D'ifilement catalogue >TC
22# GRAPHICS PEN 3:gx1=32:gx2=59#:gy1=399:gy2=353:GOSUB >PK 67# ' >TD
64#:gx1=gx1+6:gy1=gy1-6:gx2=gx2-6:gy2=gy2+6:GOSUB 64#: 68# IF ific<nfic THEN IF ibar<10 THEN GOSUB 72#:ibar=ib
LOCATE 4,2:PEN 2:PRINT tit#: ar+1:ifc=ifc+i:PAPER#1,3:GOSUB 72#:PAPER#1,0 ELSE ifi
23# GRAPHICS PEN 2:gx1=4:gx2=299:gy1=331:gy2=84:GOSUB 6 >FR c=ifc+1:PRINT#1,CHR(30);:FOR l=ifc-9 TO ifc-1:PRINT
4#:gx1=gx1+6:gx2=gx2-6:gy1=gy1-6:gy2=298:GOSUB 64#:gy1= #1,fic#(i);:NEXT:PAPER#1,3:GOSUB 72#:PAPER#1,0
292:gy2=122:GOSUB 64#:gy1=116:gy2=90:GOSUB 64# 69# RETURN >AA
24# LOCATE 1,22:PRINT mes1#:LOCATE 1,23:PRINT mes2#:LOC >GY 70# IF ific>l THEN IF ibar>l THEN GOSUB 72#:ibar=ibar-1
ATE 1,24:PRINT mes3#:LOCATE 1,25:PRINT mes4#:GOSUB 40# :ifc=ifc-1:PAPER#1,3:GOSUB 72#:PAPER#1,0 ELSE ifc=if
25# GRAPHICS PEN 2:gx1=352:gx2=638:gy1=334:gy2=192:GOSU >LF ic-1:PAPER#1,3:GOSUB 72#:PAPER#1,0:FOR l=ifc+1 TO MIN
B 64#:gx1=gx1+6:gx2=gx2-6:gy1=gy1-6:gy2=gy2+6:GOSUB 64# (ifc+9,nfic):PRINT#1,fic#(i);:NEXT
26# FOR i=1 TO 3:LOCATE#2,2,2+(i-1)*2 >CG 71# RETURN >ZD
27# PRINT#2,1*(i);:NEXT:GOSUB 80# >CC 72# LOCATE#1,1,ibar:PRINT#1,fic#(ifc);:RETURN >TY
28# GOSUB 45#:PAPER#1,3:GOSUB 72#:PAPER#1,0 >HM 73# CLEAR INPUT:LOCATE 22,2#:PAPER 2:PEN 0:INPUT"USER " >AK
29# WHILE INKEY(6)=-1 AND INKEY(18)=-1:CLEAR INPUT >VE ;user:PAPER 0:PEN 1:LOCATE 22,2#:PRINT SPACE*(10);
30# IF NOT INKEY(2) THEN ON bascule+1 GOSUB 68#,82# >TG 74# GOSUB 45#:IF nfc)# THEN PAPER#1,3:GOSUB 72#:PAPER#
31# IF NOT INKEY(8) THEN ON bascule+1 GOSUB 70#,81# >TX 1,0 >DP
32# IF NOT INKEY(42) THEN GOSUB 73# >BB 75# RETURN >ZH
33# IF NOT INKEY(1) OR NOT INKEY(6) THEN GOSUB 40# >RW 76# ' >TD
34# WEND >CF 77# ' Fonctionnement menu >TE
35# file#=LEFT$(fic#(ifc),12):CLEAR INPUT:ON iop GOSUB >QE 78# ' >TF
86#,86#,89# 79# LOCATE#2,2,2+(iop-1)*2:PRINT#2,1*(iop);:RETURN >WA
36# WEND:END >MC 80# PAPER#2,2:PEN#2,1:GOSUB 79#:PAPER#2,0:PEN#2,3:FOR i >XD
37# ' >TA =# TO 20#:NEXT:CLEAR INPUT:RETURN
38# ' Bascule >TB 81# GOSUB 79#:iop=iop-3*(iop=1)-1:GOSUB 80#:RETURN >WV
39# ' >TC 82# GOSUB 79#:iop=iop+iop*(iop=3)+1:GOSUB 80#:RETURN >ZA
40# GOSUB 41#:PEN 3:bascule=bascule XOR 1:x=10-15*(basc >UV 83# ' >TB
ule=1):GOSUB 41#:PEN 2:FOR i=# TO 5#:NEXT:CLEAR INPUT: 84# ' Exicution commande >TC
RETURN 85# ' >TD
41# LOCATE x,22:PRINT choix$(bascule);:RETURN >RK 86# MEMORY PEEK(&B736)+256*PEEK(&B737)-1:SYMBOL AFTER 2 >CM
42# ' >RG 56
43# ' Affichage catalogue >RH 87# ' Pour le 464 : MEMORY PEEK(&B296)+256*PEEK(&B297)- >TF
44# ' >RJ 1:SYMBOL AFTER 256
45# CLS#1:LOCATE 6,6:PEN 1:PRINT USING"USER #";user; >ZB 88# IF RIGHT$(file$,3)="BAS" OR RIGHT$(file$,3)="BIN" T >RD
46# (USER,user:GOSUB 51#:LOCATE 4,19:PRINT USING"## Fic >ZP HEN ON iop GOSUB 101#,102# ELSE GOSUB 103#:RETURN
hier(s)";nfic; 89# IF file$="" OR RIGHT$(file$,3)="BAS" OR RIGHT$(file >TE
47# FOR j=1 TO MIN(10,nfic):PRINT#1,fic#(j);:NEXT:ibar= >FU $,3)="BIN" THEN GOSUB 103#:RETURN
1:ifc=1:PEN 2:RETURN
48# ' >TC 90# CLS:GRAPHICS PEN 3:gx1=#:gx2=639:gy1=399:gy2=48:GOS >MY
49# ' Appel catalogue >TD UB 64#:gx1=gx1+6:gx2=gx2-6:gy1=gy1-6:gy2=gy2+6:GOSUB 64
50# ' >RF #:PEN 3:LOCATE 20-LEN(file$)/2,1:PRINT file$:PEN 2
51# ERASE fic#:DIM fic$(64) >VY 91# CLEAR INPUT:OPENIN file$ >YY
52# CALL &BB57:CALL &BE71,&BE7B:CALL &BB54 >JA 92# LOCATE 1,23:PRINT mes5#:CLS#3:boucle=1 >NL
53# nfc=#:FOR l=#E7B TO #E7B STEP 14 >ED 93# WHILE boucle >LF
54# IF PEEK(1)>255 THEN 60# >VB 94# LINE INPUT#9,a# >NY
55# nfc=nfc+1:fic#="" :FOR j=1 TO 8:fic#=fic#+CHR$(PEE >TU T(LEN(a#)/38):IF III># AND III>=1 THEN PRINT#3,a# ELSE >LM
K(i+j)):NEXT j LOCATE 1,25:PRINT mes6#:CALL &BB06:CLS#3:PRINT#3,a#
56# fic#=fic#+". " >LQ 97# IF EOF THEN LOCATE 1,25:PRINT mes7#:CALL &BB06:bou >GG
57# FOR k=# TO 2:fic#=fic#+CHR$(PEEK(i+j+k) AND 127):NE >CL cle=#
IT k 98# IF NOT INKEY(9) THEN boucle=# >BR
58# fic#=fic#+". " +DEC$(PEEK(i+12),"###")+".K" >JD 99# WEND:CLOSEIN >ML
59# fic#(nfic)=fic# >NY 100# CLS:RETURN >LD
60# NEXT i >NE 101# LOAD file$:RETURN >RP
61# ' >RH 102# RUN file$:RETURN >RW
62# ' Tracé fenetres >RJ 103# LOCATE 1,25:PEN 3:PRINT"ERREUR DE FICHER Appuyez >BZ
63# ' >RK sur une touche";CALL &BB06:PEN 2:CLS:RETURN#
64# PLOT gx1,gy1:DRAW gx2,gy1:DRAW gx2,gy2:DRAW gx1,gy2 >FJ
```

# BON DE COMMANDE

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

DESIGNATION	Prix unitaire	Quantité	Port	Montant
<b>Pour chaque article, entourez le ou les n° choisis.</b>				
<b>ANCIENS Nos AMSTAR &amp; CPC</b> 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	
<b>DISQUETTES AMSTAR &amp; CPC</b> N° 20 - 21 - 22 - 23 - 24 - 25 - 26 - 27 - 28 - 29 - 30 (une disquette réunit 2 numéros consécutifs d'Amstar & CPC)	Abonné : 110 F (unité) Non abonné : 140 F (unité)		Franco Franco	
<b>ANCIEN No CPC Infos</b> N° 25 - 26 - 27 - 28 - 29	26 F (unité)		Franco Franco Franco	
<b>DISQUETTES CPC Infos</b> - 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	
<b>ANCIENS Nos CPC</b> N° 11 - 13 - 16 - 18 - 23 - 24 - 25 - 26 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 36 - 37	25 F (unité)		Franco Franco	
<b>ANCIENS Nos HORS-SERIE CPC</b> 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	
<b>DISQUETTES CPC ET DISQUETTES HORS-SERIE CPC</b> 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 Franco	
<b>ABONNEMENT 6 DISQUETTES (non rétroactif)</b>	600 F		Franco	
<b>DISQUETTES "ARCADES" Téléchargement AMSTRAD CPC</b>	60 F		5 F	
<b>CABLE pour téléchargement</b> (Téléchargement non compatible avec 6128+ et 464+)	89 F		5 F	
<b>LOGICIELS EDUCATIFS - LIVRES - DIVERS</b>				
<b>DESIGNATION</b>	<b>REFERENCE</b>			
<b>Forfait port + 10 F par logiciel et 10 % pour livres</b>				
<input type="checkbox"/> Pour tout envoi par avion : prendre contact avec le service commercial			<input type="checkbox"/> Facultatif : recommandé + 10 F par commande	
			<b>MONTANT GLOBAL</b>	
			10 F	

Je joins mon règlement :  chèque bancaire  chèque postal  mandat

CPC Infos 30

**OC** **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 précisions est sous la responsabilité de l'acheteur. La vente est conclue dès acceptation du bon de commande par notre société, sur les articles disponibles uniquement.

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

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

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

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

# VARIABLES LOCALES ET RECURSIVITE

Yannick GOUR

Valable pour CPC 464 - 664 - 6128

(suite)

**Cet article fait suite à celui paru dans CPC Infos de Janvier. Il propose deux exemples d'utilisation de variables locales et de récursivité, programmes à l'appui.**

Avant de commencer, et en toute généralité, si la récursion peut être gourmande en mémoire, et si elle n'est pas plus rapide que d'autres méthodes, toutefois elle permet d'écrire des algorithmes relativement compacts et de compréhension aisée. Que ceux qui n'en sont pas convaincus réécrivent les programmes présentés sans. Bon courage !

## QUICKSORT

En conclusion de l'article précédent était évoqué le tri rapide "Quicksort". C'est un procédé de tri dû à un monsieur C.A.R. Hoare et inventé en 1962. On se demande pourquoi il a fallu attendre tout ce temps pour un truc simple comme bonjour (quelque parfois...), enfin ce doit être le premier Homme à en avoir donné l'algorithme à des fins informatiques. C'est vraiment très simple :

- supposons que le critère de tri soit du genre "inférieur" ou "supérieur ou égal". On s'y ramène toujours de toutes manières.

1) On met devant soi le tas d'éléments à trier.

2) On pioche un élément.

3) On met à gauche les éléments inférieurs et à droite les éléments supérieurs.

4) On applique ce procédé à chacun des tas ainsi constitués en s'arrêtant naturellement pour un tas lorsqu'il ne contient plus qu'un seul élément ou zéro (mais si, mais si !).

Programmé, voici ce que cela donne (une possibilité parmi tant d'autres) :

- On indique les bornes de gauche et droite dans le tableau à trier.

- On choisit comme élément de partage celui du milieu, que l'on échange avec l'élément de gauche (point 2).

- On examine la partie du tableau à trier de droite à gauche. Si un élément est supérieur ou égal à celui de partage, on le laisse en place, sinon on le place sur la gauche du tas à trier, en l'échangeant avec l'élément à l'extrême gauche de ce tas. Il y a là un index utilisé pour connaître l'avancement du tas d'"inférieurs". Dans ce cas on poursuit l'examen à partir de l'élément échangé.

- Le tri fini, on rééchange l'élément frontière entre les deux tas de gauche) avec l'élément de partage, qu'on avait placé à gauche (point 3).

- On trie le tas de gauche et le tas de droite de la même manière (point 4).

Plus un tas est déjà trié, plus on risque d'atteindre vite la limite de la pile BASIC (80 niveaux environ). Pour éviter cela, il suffit de mettre du désordre dans le tableau à trier (qui l'eût cru ?). Par exemple avec une boucle du genre :

```
FOR i=1 to n STEP 2
```

```
  j=1+INT(RND*n)
```

```
  ÛGIVE,5,àa(i),5àa(j):ÛGET,5,àa(j),5,àa(i)
```

```
NEXT
```

Deux programmes sont proposés, qui sont très semblables. L'un trie des nombres (QUICK-R) et l'autre des chaînes (QUICK-S). Les explications précédentes et les commentaires inclus dans les listings doivent permettre leur compréhension sans trop de problèmes. A noter l'usage du couple ÛGIVE, ÛGET pour l'échange de variables. Pour échanger deux variables, c'est plus élégant qu'économique (en temps), mais au-delà on gagne très vite beaucoup de temps. Ainsi est réalisé dans QUICK-R un rapide transfert de valeurs d'un tableau quelconque au tableau spécifique de la routine de tri.

## TRI BINAIRE

Encore un procédé de tri. Mais autant ne pas voir deux fois la même chose. Revenons sur le Quicksort, ou "rapide tri" en anglais. On l'applique à un ensemble d'éléments, dont on dispose déjà avant de commencer à trier. Seulement bien souvent on a un paquet d'éléments à insérer dans un tas préexistant. Il faut donc faire de l'insertion au coup par coup. Structurer les éléments en un "arbre binaire" est une manière très pratique de réaliser ceci.

L'arbre binaire : "Dessine-moi un arbre binaire". "C'est très simple, petit Prince. Prends un crayon et une feuille



de papier. Dessine en haut un point, puis deux branches en partant. Et fais de même pour chaque point". Voilà le principe d'un arbre binaire, de manière imagée. Plus concrètement, les points sont appelés "nœuds". Les 2 nœuds reliés à un même nœud sont appelés nœuds "fils". On parle de nœud fils gauche ou droite selon leur classement (voir plus loi). Un nœud peut n'avoir qu'un fils ou aucun. Un nœud est privilégié par rapport aux autres : c'est le premier, celui par lequel on "entre" dans l'arbre. On l'appelle "racine". Un nœud est une entité complexe. A tout élément trié correspond un nœud. Un nœud est ainsi constitué :

- pointeur sur l'élément ;
- compteur d'occurrences de l'élément ;
- pointeur sur le fils gauche ;
- pointeur sur le fils droit.

Qu'est-ce qu'un pointeur ? C'est tout bêtement une adresse. Cela indique à quel endroit chercher ou mettre de l'information. Ici par exemple le "pointeur de l'élément" est le numéro de l'élément dans le tableau des éléments ("mottab" ou "xtab" dans les programmes). Les pointeurs sur les fils sont simplement les numéros des nœuds fils dans le tableau des nœuds. Comment classe-t-on dans un arbre (si vous ne l'avez déjà deviné) ? Enfantin ! On se ramène avec l'élément à insérer à la racine. De là c'est toujours le même processus :

- 1) Si le nœud est vide, c'est-à-dire ne pointe sur aucun élément, alors on lui attribue le nouvel élément et on met le compteur d'occurrences à 1. C'est fini ici.
- 2) Si l'élément est égal à celui du nœud, alors on incrémente le compteur d'occurrences et c'est fini ici.
- 3) Si l'élément est inférieur à celui du nœud, alors on se déplace au fils de gauche et on reprend en 1).
- 4) Sinon c'est qu'il est supérieur et alors on se déplace au fils droit. On reprend en 1).

L'insertion se fait donc très simplement. On obtient en fin de compte un arbre avec l'élément le plus petit le plus à gauche (on prend toujours à gauche mais là on ne tourne pas en rond) et le plus grand le plus à droite. Mais comment débrouiller le méll-mélo entre les deux ? L'enfance de l'art, grâce encore une fois à la récursivité. On considère la procédure "afficher nœud". Voilà comment elle se décompose :

- 1) Si nœud vide, alors retour de procédure.
- 2) "afficher fils de gauche".
- 3) Faire apparaître à l'écran ce qui nous intéresse sur le nœud. Souvent il s'agit de l'élément pointé et de son nombre d'occurrences.
- 4) "afficher fils de droite".
- 5) Retour de procédure.

Cela semble magique, mais pour parodier une célèbre citation : "et pourtant ça marche". On voit défiler à l'écran en un temps record tous les éléments triés.

Comme pour Quicksort, des éléments arrivant déjà ordonnés conduisent à un déséquilibre de l'arbre et on a vite fait d'arriver à une saturation de la pile BASIC. Pour des éléments venant au hasard par contre, on a beaucoup plus de chances d'aboutir d'abord à un manque de place pour les tableaux de données.

Encore une fois les deux programmes proposés sont très proches : l'un trie des réels (SHELL-R) et l'autre des chaînes numériques (SHELL-S). Dans ce dernier cas un fichier ASCII préexistant est lu. Comme souvent il s'agit d'un fichier de texte, on lit des lignes entières à chaque "INPUT". Aussi chaque "ligne" est-elle retravaillée pour

supprimer les espaces superflus et en extraire ensuite aisément le contenu. On peut plus facilement taper une série de mots lus par un INPUT, que l'on envoie à chaque fois pour insertion dans l'arbre (c'est toute l'utilité de la chose ! Et cela va très vite, c'est du temps réel). En sauvegardant en même temps les mots d'un fichier, on peut faire ultérieurement l'économie d'une nouvelle entrée de données. Si on veut épater la galerie, on peut même en rajouter soi-même, soit par programme, soit carrément "à la main", en "Gl-Vant" la racine et en appelant la procédure d'insertion.

Au possesseur de 464 : la fonction DEC\$ est buggée sur le 464, mais on peut l'utiliser en ouvrant 2 parenthèses et en n'en fermant qu'une : "DEC\$((...)" . Elle est utilisée dans l'affichage de l'arbre.

## CONCLUSION

**Maintenant il n'y a plus qu'à espérer que quelque lecteur soit inspiré et nous offre bientôt dans ces colonnes un programme dans ce bon vieux BASIC du CPC (gonflé des RSX que vous savez), qu'on n'aurait pas pu écrire sans la récursion (ou pas sans ENORMES difficultés). Je souhaite y avoir donné goût (je trouve très agréable le côté naturel des algorithmes et les deux exemples traités doivent le mettre en valeur, sinon tant pis... y'a plus rien à faire).**

NOTE : Pour utiliser les programmes ci-dessous, il faut bien sûr avoir initialisé les RSX auparavant. Reportez-vous au numéro 28 de CPC Infos pour cela.

## RECTIFICATIF

Dans le numéro 28 de CPC Infos, le texte de l'article "Variables locales et récursivité" a été tronqué, voici ce qu'il fallait lire :

### LECTURE DES ARGUMENTS D'ENTREE

Commande : IGET, typ, @var, typ2, @var2,...

Idem qu'avant. Il doit y avoir identité entre les types des arguments transmis en entrée et ces derniers. Même si le type d'une variable n'est indiqué que par "typ", il faut aussi que les variables concernées soient de même type pour que la transmission et la lecture de l'argument aient un sens.

### INDICATION ET LECTURE DES ARGUMENTS DE SORTIE

Celui qui a bien suivi aura compris que cela se fait respectivement avec les commandes IGIVE et IGET.

### RESTAURATION DES VARIABLES LOCALES

Commande : IENDLOCAL, typ, @var, typ2, @var2,...

Il s'agit de l'injection dans les variables locales de leurs valeurs avant appel de la procédure. La liste des arguments doit être identique à celle de la commande ILOCAL. Aussi le plus simple est de recopier grâce à la touche COPY cette liste.



# QUICK-R

```

100 MODE 2: /STACK, &7000, &1000 >XZ
110 DEFINT b-s: DEFREAL a, t, x: n=100: DIM a(n), tri(n): b=b: >RJ
d=d: g=g: f=f
120 ' >RD
130 'remplissage d'un tableau de nombres au hasard >RE
140 FOR i=1 TO n: a(i)=RND*n: NEXT >AL
150 ' >RG
160 'transmission au tableau de tri >RH
170 /GIVE, n*5, @a(1): /GET, n*5, @tri(1) >FM
180 ' >RK
190 'indication des bornes Gauche et Droite >TA
200 g=1: d=n: /GIVE, 2, @g, 2, @d >WY
210 ' >RD
220 'tri du tableau >RE
230 z=TIME: GOSUB 370: z=(TIME-z)/300 >EH
240 ' >RG
250 'recuperation du tableau trie >RH
260 /GIVE, n*5, @tri(1): /GET, n*5, @a(1) >FM
270 ' >RK
280 'affichage du tableau trie et du temps de tri >TA
290 FOR i=1 TO n: PRINT a(i): NEXT >AE
300 PRINT z >ZG
310 END >RH
320 ' >RF
330 '----- >RG
340 'Routine "QUICKSORT" d'apres C.A.R. Hoare >RH
350 ' >RJ
360 'definition des variables locales >RK
370 /LOCAL, 2, @g, 2, @d, 2, @f >UJ
380 'recuperation des bornes de tri >TB
390 /GET, 2, @g, 2, @d >WQ
400 ' >RE
410 'tri fini si 1 seul element >RF
420 IF g>d THEN 700 >ML
430 ' >RH
440 'tri par rapport a l'element central, echange avec >RJ
l'element de gauche
450 i=(g+d)/2: x=tri(i): tri(i)=tri(g) >FN
460 ' >TA
470 'on tri de droite a gauche >TB
480 f=g: i=d >ZA
490 ' >TD
500 'trier le tas >RF
510 WHILE i>f >MK
520 ' >RH
530 ' placer a gauche tout element inferieur >RJ
540 IF tri(i)<x THEN f=f+1: /GIVE, 5, @tri(i), 5, @tri(f): >YW
/GIVE, 5, @tri(f), 5, @tri(i): GOTO 580
550 ' passer a l'element suivant >TA
560 i=i-1 >EH
570 ' >TC
580 WEND >DB
590 ' >TE
600 'placer element de partage entre les tas >RG
610 tri(g)=tri(f): tri(f)=x >WQ
620 ' >RJ

```

```

630 'trier le tas de gauche >RK
640 b=f-1: /GIVE, 2, @g, 2, @b: GOSUB 370 >DN
650 ' >TB
660 'trier le tas de droite >TC
670 b=f+1: /GIVE, 2, @b, 2, @d: GOSUB 370 >DK
680 ' >TE
690 'restauration des variables locales >TF
700 /ENDLOCAL, 2, @g, 2, @d, 2, @f >XB
710 ' >RJ
720 RETURN >ZE

```

# QUICK-S

```

100 MODE 2: /STACK, &7000, &1000 >XZ
110 DEFINT b-w: DEFSTR a, t, x: n=100: DIM a(n): b=b: d=d: g=g: >FG
f=f
120 ' >RD
130 'remplissage d'un tableau de chaines >RE
140 PRINT "cree chaines..." >ZJ
150 FOR i=1 TO n: FOR j=1 TO (i+RND*6): a(i)=a(i)+CHR$(65 >TC
+RND*7): NEXT j, i
160 ' >RH
170 'transmission au tableau de tri >RJ
180 DIM tri(n): FOR i=1 TO n: tri(i)=a(i): NEXT i: b=FRE("") >ZE
190 ' >TA
200 'indication des bornes Gauche et Droite >RC
210 g=1: d=n: /GIVE, 2, @g, 2, @d >WZ
220 ' >RE
230 'tri du tableau >RF
240 PRINT "trie...", z=TIME: GOSUB 380: z=(TIME-z)/300 >YP
250 ' >RH
260 'recuperation du tableau trie >RJ
270 FOR i=1 TO n: a(i)=tri(i): NEXT i: ERASE tri: b=FRE("") >YW
280 ' >TA
290 'affichage du tableau trie et du temps de tri >TB
300 FOR i=1 TO n: PRINT a(i): NEXT >AW
310 PRINT z >ZH
320 END >RJ
330 ' >RG
340 '----- >RH
350 'Routine "QUICKSORT" d'apres C.A.R. Hoare >RJ
360 ' >RK
370 'definition des variables locales >TA
380 /LOCAL, 2, @g, 2, @d, 2, @f >UK
390 'recuperation des bornes de tri >TC
400 /GET, 2, @g, 2, @d >NG
410 ' >RF
420 'tri fini si 1 seul element >RG
430 IF g>d THEN 710 >MN
440 ' >RJ
450 'tri par rapport a l'element central, echange avec >RK
l'element de gauche
460 i=(g+d)/2: x=tri(i): tri(i)=tri(g) >FP
470 ' >TB
480 'on tri de droite a gauche >TC
490 f=g: i=d >ZB
500 ' >RF

```



```

510 'trier le tas >RG ● d)=x:noeud(ptr,occ)=1:GOTO 490
520 WHILE i>f >NA ● 390 ' >TC
530 ' >RJ ● 400 'identite entre valeur X et celle du noeud >BE
540 ' placer a gauche tout element inferieur >RK ● 410 IF x=xtab(noeud(ptr,ptrx)) THEN noeud(ptr,occ)=noeu >GC
550 IF tri(i)<x THEN f=f+1:GIVE,3,@tri(i),3,@tri(f): >YP
:GET,3,@tri(f),3,@tri(i): GOTO 590 d(ptr,occ)+1:GOTO 490
560 ' passer a l'element suivant. >TB ● 420 ' >RG
570 i=i-1 >EJ ● 430 'aller a gauche >RH
580 ' >TD ● 440 IF x<xtab(noeud(ptr,ptrx)) THEN :GIVE,2,@ptr,2,@fil >TT
590 WEND >DC ● 90 sg:GOSUB 740:GIVE,2,@noeud(ptr,filsg):GOSUB 350:GOTO 4
600 ' >RG ● 450 ' >RK
610 'placer element de partage entre les tas >RH ● 460 'aller a droite >TA
620 tri(g)=tri(f):tri(f)=x >WR ● 470 :GIVE,2,@ptr,2,@filsg:GOSUB 740:GIVE,2,@noeud(ptr, >BY
630 ' >RK ● filsg):GOSUB 350
640 'trier le tas de gauche >TA ● 480 ' >TC
650 b=f-1:GIVE,2,@g,2,@b:GOSUB 380 >DP ● 490 RETURN >ZJ
660 ' >TC ● 500 ' >BF
670 'trier le tas de droite >TD ● 510 '----- >RG
680 b=f+1:GIVE,2,@b,2,@d:GOSUB 380 >DM ● 520 'Affichage d'arbre >RH
690 ' >TF ● 530 ' >RJ
700 'restauration des variables locales >RH ● 540 :LOCAL,2,@ptr >NQ
710 :ENDLOCAL,2,@g,2,@d,2,@f >XC ● 550 ' >TA
720 ' >RK ● 560 :GET,2,@ptr >LY
730 RETURN >ZF ● 570 ' >TC
● 580 IF noeud(ptr,occ)=0 THEN 670 >ZE
● 590 ' >TE
● 600 'affiche fils gauche >RG
● 610 :GIVE,2,@noeud(ptr,filsg):GOSUB 540 >KM
● 620 PRINT DEC$(noeud(ptr,occ),"###");" => ";xtab(noeud >RA
(ptr,ptrx))
● 630 ' >RK
● 640 'affiche fils droit >TA
● 650 :GIVE,2,@noeud(ptr,filsg):GOSUB 540 >KN
● 660 ' >TC
● 670 :ENDLOCAL,2,@ptr >RP
● 680 ' >TE
● 690 RETURN >AA
● 700 ' >RH
● 710 '----- >RJ
● 720 'Allocation de fils a un noeud "n" >RK
● 730 ' >TA
● 740 :LOCAL,2,@n,2,@fils >TG
● 750 ' >TC
● 760 :GET,2,@n,2,@fils >RT
● 770 ' >TE
● 780 IF noeud(n,fils)=0 THEN id=id+1:noeud(n,fils)=id >YE
● 790 ' >TG
● 800 :ENDLOCAL,2,@n,2,@fils >WT
● 810 ' >RK
● 820 RETURN >ZF

```

## SHELL-R

```

100 MEMORY &0F00 >HK ●
110 MODE 2:STACK,&9000,&1000 >XC ●
120 DEFINT a-s:DEFREAL t-z:n=400:DIM xtab(n),noeud(n,3) >AG
:fils=0:n=0:ptr=0
130 ' >RE ●
140 'pointeurs du noeud >RF ●
150 ptrx=0:occ=1:filsg=2:filsd=3 >CM ●
160 ' >RH ●
170 'initialisation de l'arbre >RJ ●
180 racine=1:id=1 >MK ●
190 ' >TA ●
200 'remplissage d'arbre >RC ●
210 PRINT"insere dans arbre..." >FG ●
220 FOR i=0 TO 100 >LB ●
230 x=ROUND(100*NRND,0):GIVE,2,@racine:GOSUB 350 >VP ●
240 NEXT >EB ●
250 ' >RH ●
260 'affichage de l'arbre >RJ ●
270 :GIVE,2,@racine:GOSUB 540 >ZN ●
280 ' >TA ●
290 END >TE ●
300 ' >RD ●
310 '----- >RE ●
320 'Routine de tri binaire >RF ●
330 ' >RG ●
340 'recuperation du pointeur de noeud >RH ●
350 :GET,2,@ptr >LV ●
360 ' >RK ●
370 'si noeud vide alors ajouter valeur >TA ●
380 IF noeud(ptr,ptrx)=0 THEN noeud(ptr,ptrx)=id:xtab(1 >XK

```

## SHELL-S

```

100 MEMORY &0F00:OPENOUT"x":MEMORY HIMEM-1:CLOSEOUT >AE
110 MODE 2:STACK,&9000,&1000 >XC
120 DEFINT a-l,n-z:DEFSTR m:n=600:DIM mottab(n),noeud(n >FN
,3):fils=0:n=0:ptr=0

```



```

130 '
140 'pointeurs du noeud
150 ptrroot=0:occ=1:filsg=2:filsd=3
160 '
170 'initialisation de l'arbre
180 racine=1:id=1
190 '
200 'remplissage d'arbre
210 OPENIN"article3":WHILE NOT EOF
220 'lire et formater ligne
230 INPUT#9,ligne$:GOSUB 930:i=INSTR(ligne$," ")
240 ' trier mot
250 WHILE i<>0
260 mot=LOWER$(MID$(ligne$,1,i-1)):ligne$=MID$(ligne$,
,i+1)
270 IF mot<>" THEN PRINT"insere ";mot:GIVE,2,@racine
:GOSUB 410
280 i=INSTR(ligne$," ")
290 WEND
300 WEND:CLOSEIN
310 '
320 'affichage de l'arbre
330 :GIVE,2,@racine:GOSUB 600
340 '
350 END
360 '
370 '-----
380 'Routine de tri binaire
390 '
400 'recuperation du pointeur de noeud
410 :GET,2,@ptr
420 '
430 'si noeud vide alors ajouter mot
440 IF mottab(noead(ptr,ptrmot))="" THEN noead(ptr,ptrm
ot)=id:mottab(id)=mot:noead(ptr,occ)=1:GOTO 550
450 '
460 'identite entre mot et mot du noeud
470 IF mot=mottab(noead(ptr,ptrmot)) THEN noead(ptr,occ
)=noead(ptr,occ)+1:GOTO 550
480 '
490 'aller a gauche
500 IF mot<mottab(noead(ptr,ptrmot)) THEN :GIVE,2,@ptr,
2,@filsg:GOSUB 600:GIVE,2,@noead(ptr,filsg):GOSUB 410:
GOTO 550
510 '
520 'aller a droite
530 :GIVE,2,@ptr,2,@filsd:GOSUB 800:GIVE,2,@noead(ptr,
filsd):GOSUB 410
540 '
550 RETURN
560 '
570 '-----
580 'Affichage d'arbre
590 '
600 :LOCAL,2,@ptr
610 '
620 :GET,2,@ptr
630 '
640 IF noead(ptr,occ)=0 THEN 730
>RE ● 650 ' >TB
>RF 660 'affiche fils gauche >TC
>EH 670 :GIVE,2,@noead(ptr,filsg):GOSUB 600 >KQ
>RH ● 680 PRINT DEC$(noead(ptr,occ),"###");" => ";mottab(noead(ptr,ptrroot)) >YH
>RJ 690 '
>MK 700 'affiche fils droit >TF
>TA ● 710 :GIVE,2,@noead(ptr,filsd):GOSUB 600 >RH
>RC 720 ' >KG
>FY 730 :ENDLOCAL,2,@ptr >RL
>RE ● 740 ' >TB
>UE 750 RETURN >ZH
>RG ● 760 ' >TD
>VB 770 '----- >TE
>GX ● 780 'Allocation de fils a un noeud "n" >TF
● 790 ' >TC
800 :LOCAL,2,@n,2,@fils >TD
>TD ● 810 ' >RK
>DD ● 820 :GET,2,@n,2,@fils >RP
>MV 830 ' >TB
>RE 840 IF noead(n,fils)=0 THEN id=id+1:noead(n,fils)=id >YY
>RF ● 850 ' >TD
>ZG 860 :ENDLOCAL,2,@n,2,@fils >WZ
>RH 870 ' >TF
>TB ● 880 RETURN >AB
>RK 890 '----- >TH
>TA 900 'suppression espaces superflus >RK
>TB ● 910 ' >TA
>TC 920 'supprime signes de ponctuation & co >TB
>RE 930 ligne$=" "+ligne$+" ":s$="!?!%&'()*_+=+.,/:;|@[]" >WD
>LR ● CHR$(34):FOR i=2 TO LEN(ligne$)-1:j=INSTR(s$,MID$(ligne$,i,1)):IF j<>0 THEN ligne$=LEFT$(ligne$,i-1)+" "+MID$(
ligne$,i+1)
>RG ● 940 NEXT >EJ
>RH 950 ' >TE
>BD ● 960 'supprime espaces a gauche >TF
>RK ● 970 i=1:k$=" ":WHILE k$=" ":k$=LEFT$(ligne$,1):IF k$=" >VA
" AND ligne$=" " THEN ligne$="":RETURN ELSE IF k$=" "THE
N ligne$=MID$(ligne$,2)
>TN ● 980 WEND >DF
>TC ● 990 ' >TJ
>EA 1000 'supprime espaces a droite >XD
● 1010 i=LEN(ligne$):k$=" ":WHILE k$=" ":k$=RIGHT$(ligne$,1):IF k$=" "AND ligne$=" " THEN ligne$="":RETURN ELSE I
F k$=" " THEN ligne$=LEFT$(ligne$,LEN(ligne$)-1)
>RG ● 1020 WEND >HH
>RH ● 1030 ' >XG
>BN 1040 'supprime autres espaces superflus >XH
● 1050 i=INSTR(ligne$," "):WHILE i<>0:j=i:WHILE MID$(lign
e$,j,1)=" ":j=j+1:WEND:IF j>i+1 THEN ligne$=LEFT$(ligne
$,i)+MID$(ligne$,j)
>RK ● 1060 j=i:i=i+INSTR(MID$(ligne$,i+1)," "):IF i=j THEN i= >AE
>ZF 0
>TB ● 1070 WEND >JC
>TC ● 1080 ' >YB
>TD ● 1090 'ajoute un espace en fin >YC
>TE ● 1100 ligne$=ligne$+" " >QG
>TN ● 1110 ' >XF
>TB ● 1120 RETURN >EH

```





CPC  
jeu



T H E

**Fabien FESSARD  
et  
Stéphane SAINT-MARTIN**

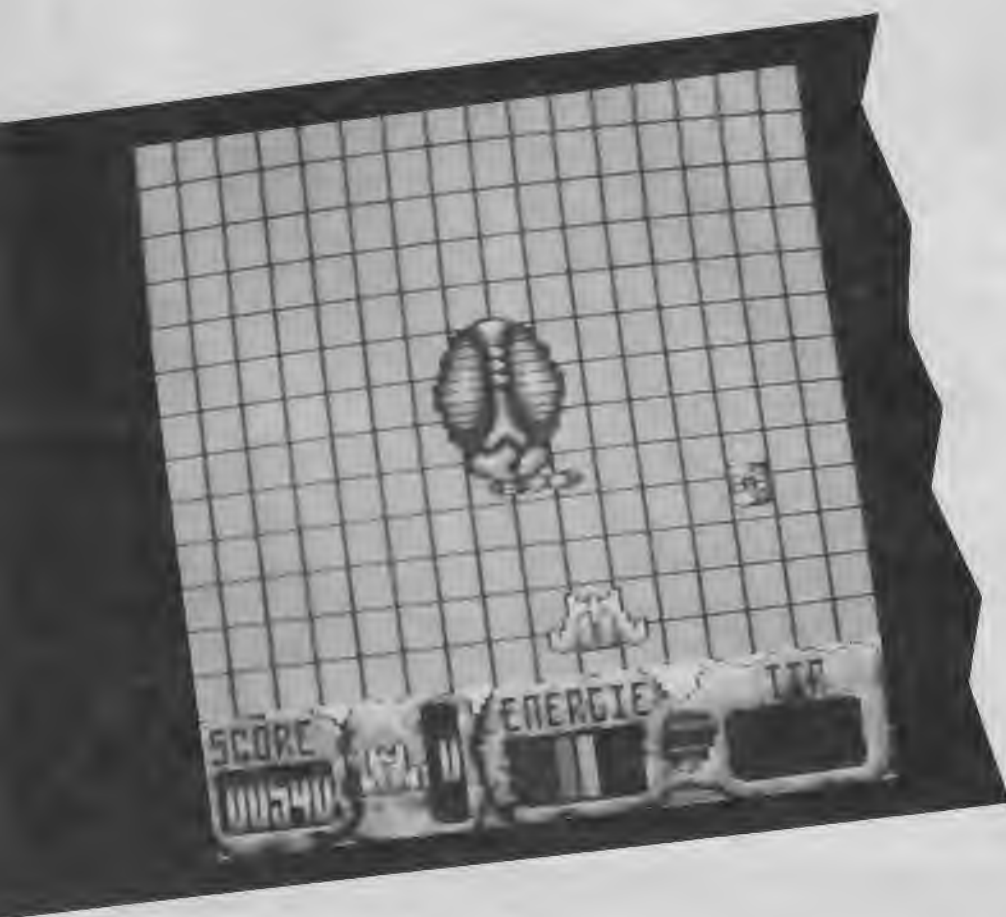
Valable pour CPC  
6128  
464 + ext. 64 ko

Les marathoniens du listing peuvent continuer leur épreuve : la suite d'Axys s'étend sous leurs doigts agiles.

N'oubliez pas d'utiliser **DATALOAD** pour vous faciliter la tâche. De toutes façons, le résultat en vaut la peine.

Ce mois-ci les listings sont intitulés **DATAS2B.BAS** et **DATAS2C.BAS**.

A suivre...



# XXYS

L A S T . B A T T L E

DATAS2B.BAS



10 ' DATAS2B AXYS

```
20 '  
30 MODE 2:AD=4C000:NL=90  
40 FOR A=1 TO 667: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"INTROB",B,&C000,&2710:CLS:END  
80 CLS:PRINT"LIGNE"NL;" INCORRECTE":END  
90 DATA 76,03,08,43,C3,96,0F,33,CB,C7,76,CF,04,4F,0F,598  
100 DATA 0F,1B,1B,76,33,05,0F,4F,CF,CB,C3,03,03,76,3C,466  
110 DATA 00,00,15,3F,A2,00,03,42,14,2D,0F,0F,CF,03,42,289  
120 DATA 0E,3C,3C,78,3C,03,42,6C,1B,B0,0C,48,C0,60,84,528  
130 DATA F0,8C,4C,D8,B4,3C,38,B4,76,3C,05,B0,64,0C,C0,713  
140 DATA 80,00,21,01,03,43,C3,C3,C7,CF,03,42,76,0F,06,4D4  
150 DATA 1B,33,03,42,76,76,01,76,FC,04,33,33,67,CF,C7,559  
160 DATA C3,83,03,76,3C,07,00,F3,F3,15,00,B7,F3,A2,00,849  
170 DATA 03,42,3E,39,0F,0F,4F,CF,03,42,8E,DA,78,4C,58,541  
180 DATA 76,3C,04,F4,8C,C0,58,78,B4,3C,F0,34,3C,C8,90,76E  
190 DATA D8,3C,24,98,8C,8C,48,C4,CC,CC,8C,48,C0,D8,38,828  
200 DATA 3C,B4,0F,0F,0F,76,33,06,76,76,01,76,FC,07,7C,4FE  
210 DATA FC,FC,33,33,4F,CF,CF,C3,83,03,76,3C,06,AB,51,745  
220 DATA A2,00,3F,51,F3,F3,A2,00,03,42,FE,A2,E7,76,CF,7CB  
230 DATA 05,0F,58,76,3C,06,B4,D8,38,B4,B4,3C,B4,98,F0,748  
240 DATA B0,F0,F0,E0,84,70,64,76,0C,05,CC,98,CC,CC,0C,857  
250 DATA CC,CC,30,F0,B1,05,33,03,42,76,FC,05,76,3C,07,696  
260 DATA 7C,FC,FC,33,27,0F,CB,C3,C3,03,03,76,3C,06,AB,714  
270 DATA 51,F3,A2,3F,00,F3,F3,A2,00,03,42,3E,2A,55,76,625  
280 DATA CF,04,4D,8E,B4,76,3C,06,F0,6C,34,3C,F0,03,42,619  
290 DATA 0C,D8,24,CC,30,24,80,8C,4C,98,30,64,CC,8C,76,67A  
300 DATA 0C,04,CC,76,0C,04,18,28,1B,33,33,76,76,01,76,37E  
310 DATA FC,04,8C,8C,76,3C,05,7C,FC,FC,33,27,0F,CB,C3,79A  
320 DATA CB,03,03,76,3C,05,7D,00,00,51,51,FF,51,3F,3F,4F5  
330 DATA A2,00,03,42,7E,2A,15,9F,48,0C,76,3C,04,38,3C,389  
340 DATA 3C,64,70,64,64,24,4C,30,CC,CC,8C,84,CC,C8,0C,8C0  
350 DATA 8C,48,40,8C,C0,84,C0,D8,38,0C,C0,84,84,48,C0,690  
360 DATA 03,42,80,C0,00,C4,B4,3C,78,64,8C,48,C0,00,18,5C1  
370 DATA 33,33,FC,03,42,8C,3C,FC,76,76,01,33,33,0F,4F,54C  
380 DATA CF,C3,C3,83,3C,03,42,39,76,00,06,15,87,3F,3F,558  
390 DATA A2,00,03,42,7E,AB,00,A7,4F,CF,CF,8E,0C,0C,C0,687  
400 DATA 18,8C,C0,84,C0,C0,48,C0,03,42,84,CB,C4,0C,84,755  
410 DATA 48,84,80,04,40,80,00,40,04,84,48,C0,48,80,00,498  
420 DATA 00,48,58,B4,64,F0,80,B4,40,80,76,00,07,80,33,5CC  
430 DATA 89,33,76,76,01,FC,FC,B9,33,27,0F,CF,CF,C3,CF,823  
440 DATA 83,3C,3C,79,76,00,08,B7,7F,FF,AA,00,03,42,7E,584  
450 DATA 28,55,27,4F,0F,4F,CF,0F,6D,CF,8F,4F,CF,CF,8E,6F5  
460 DATA 48,84,C0,48,76,C0,05,80,76,00,13,0D,80,76,00,513  
470 DATA 04,40,C0,03,42,00,00,40,85,33,33,76,76,01,33,394  
480 DATA 89,33,33,76,76,01,27,4F,CF,CB,C3,C3,03,3C,28,609  
490 DATA 51,A2,76,00,0E,76,76,01,A8,00,05,1B,0F,03,42,380  
500 DATA CF,CF,8F,0F,0F,4F,CF,03,42,C3,C7,CF,CB,C7,CB,864  
510 DATA C0,40,00,C0,C3,C3,CB,CB,C3,CB,76,C3,05,CF,CF,946  
520 DATA C3,CB,C7,CF,03,42,8F,76,0F,00,33,33,0F,0F,33,53C  
530 DATA 1B,0F,0F,33,27,33,03,42,27,0F,CF,C7,C3,83,76,493  
540 DATA 00,0F,FE,8C,B9,FC,F0,00,F3,3B,33,38,3C,0F,27,680  
550 DATA 76,0F,07,76,CF,05,CB,CF,76,C3,0D,C7,C3,CF,C3,7D2  
560 DATA 76,CF,0C,0F,4F,4F,CF,4F,4F,0F,8F,76,0F,05,1B,4AE  
570 DATA 33,0F,4F,4F,CB,C3,83,A2,76,00,10,54,A8,00,00,515  
580 DATA 51,00,55,FC,36,B9,76,33,05,1B,27,0F,03,42,FC,4D1  
590 DATA 27,76,CF,04,CB,CB,C7,C7,7C,76,C3,05,C7,CF,CF,883  
600 DATA 0F,33,4F,CF,C3,CF,C3,C3,C7,CF,C7,C7,76,CF,0E,8EF  
610 DATA 0F,0F,0F,1B,33,0F,4F,C3,3C,FF,76,00,06,8B,FC,60A  
620 DATA 3C,00,00,7F,FF,3F,00,03,42,FE,AA,00,03,42,A2,4CD  
630 DATA 00,00,76,76,01,8C,76,FC,05,89,33,03,42,1B,27,493  
640 DATA 1B,36,3C,27,4F,76,CF,04,C3,CF,C3,C3,CB,36,27,68C  
650 DATA 4F,76,CF,05,CB,C3,CB,CB,76,C3,04,76,CF,07,87,7CD  
660 DATA 4B,CB,C3,03,42,CF,CF,8F,76,0F,04,CF,3C,B7,FD,793  
670 DATA 76,00,8A,51,A2,76,00,05,A2,00,54,FC,FC,7D,76,5CF  
680 DATA 00,07,51,3B,8C,FC,03,42,76,33,04,0F,1B,8C,39,45C  
690 DATA 27,0F,76,CF,07,8F,0F,03,42,76,CF,09,C7,33,27,4D4  
700 DATA C7,76,CF,07,C7,03,42,C3,CF,03,42,0F,0F,CF,DF,6C2  
710 DATA FC,7C,BD,A2,76,00,11,2A,00,55,AA,00,8B,33,FC,671  
720 DATA FC,77,8B,FC,33,33,77,FB,A2,76,00,06,51,7F,76,766
```



730 DATA 76, 01, 3C, 39, AA, 76, 00, 10, 51, AF, 0F, 5F, A2, 00, 03, 42F  
740 DATA 42, 15, 7F, BF, 3F, 03, 42, FF, 33, 33, 37, 2A, 00, 03, 42, 424  
750 DATA 7F, FF, FD, 33, BF, 3B, A2, 15, 76, 00, 00, 83, 76, 33, 0C, 645  
760 DATA 77, FF, 33, 76, 76, 01, FC, 37, A2, 00, 51, 3F, 76, 33, 05, 5A9  
770 DATA BB, BB, 33, 76, 76, 01, 33, 03, 42, BB, 37, F3, FF, FF, 33, 724  
780 DATA 76, 76, 01, FC, B9, 33, 03, 42, 77, FD, 7C, FC, FF, FF, BB, BBF  
790 DATA 33, 33, 77, 76, 33, 07, FF, FF, 77, BB, 33, FF, FC, FC, BC, 8A3  
800 DATA FF, FF, 77, FF, FF, 00, 7F, BB, BF, 00, 7B, 11, 33, 33, FF, 85D  
810 DATA FF, 76, 3F, 05, BF, FF, FB, B7, BB, BF, B7, BB, 37, 7B, 00, 8C7  
820 DATA 15, 33, 33, AA, 00, 03, 42, 33, 76, 76, 01, B9, 77, FC, BC, 572  
830 DATA 3C, FC, 33, 22, 00, 00, 51, FF, 33, 76, 76, 01, BC, 3C, FC, 5F1  
840 DATA FD, FF, FF, 77, FF, 03, 42, 76, 33, 09, FF, FC, 03, 42, FF, 8A7  
850 DATA FF, 3F, 3F, 7F, 76, 33, 05, 77, FC, FC, BB, BB, 33, 77, FF, 838  
860 DATA FF, 77, BF, 7B, 76, 00, 09, 3B, FF, B7, 77, 00, F3, 3F, FF, 7C0  
870 DATA AA, 76, 00, 00, 3F, BF, FF, 33, BB, 33, 3F, F3, 7F, BB, A2, 754  
880 DATA 00, 51, 33, FF, FF, BB, BB, 33, 33, 7F, FB, B7, BB, 33, FE, 87B  
890 DATA FC, FC, FF, FF, 3F, 2A, 76, 00, 06, B7, FE, FC, BC, 3C, FF, 983  
900 DATA 76, 33, 05, 3F, BB, 76, 33, 0A, 77, 3B, FF, BF, 7F, BB, 33, 638  
910 DATA 7B, 76, 3C, 00, 76, 3C, 00, 76, 3C, 00, 76, 3C, 5A, 00, 00, 39D  
920 DATA A2, 2A, 3F, 76, 00, 07, 7E, 2A, 15, CF, 03, 42, 8E, 8C, 5B, 4CB  
930 DATA 3C, 3C, 04, 3C, 03, 42, 04, 76, F0, 04, F0, 7B, 6A, CC, 0C, 719  
940 DATA 4C, 2A, CC, CC, 9B, C0, 0C, C0, 03, 42, 0C, 4C, 4C, CC, 6A, 6C5  
950 DATA CC, 03, 42, 0C, 48, 48, 04, 08, 08, 1B, 33, 03, 42, 76, FC, 546  
960 DATA 05, BC, 76, 3C, 05, 7C, FC, FC, B9, 27, 0F, CF, C3, C3, 83, 7B3  
970 DATA 03, 76, 3C, 05, 7D, 00, 00, 51, 51, FF, 15, 3F, 3F, A2, 00, 40D  
980 DATA 03, 42, 54, AA, 15, 67, 4F, C0, 04, 34, 3C, E4, B4, 3C, A4, 66A  
990 DATA 9B, 64, 9B, 1B, 24, 4C, 9B, 4C, CC, 0C, 84, CC, 48, 4C, 0C, 5C8  
1000 DATA C0, 40, 4C, 4C, 4B, 84, 0C, B0, 84, 0C, 0C, 80, C0, C0, 40, 5FC  
1010 DATA C0, 00, 40, 34, 3C, 78, 24, C0, CC, 30, F0, 3C, A4, 40, 33, 60B  
1020 DATA 33, FC, FC, 7C, BC, BC, FC, FC, 33, 33, 0F, 0F, CF, CF, 03, 8BC  
1030 DATA 83, 3C, 03, 42, 7D, 76, 00, 06, 15, B7, 3F, 3F, 2A, 00, 03, 374  
1040 DATA 42, 7E, A8, 00, A7, 4F, CF, 0F, CF, 0C, 0C, 84, 4B, C0, 84, 6B3  
1050 DATA 0C, C0, 84, C0, 03, 42, 00, 40, C0, 4C, 0C, 84, 4B, C0, 80, 689  
1060 DATA 04, 84, 0C, 80, C0, C0, 76, 00, 00, 84, 70, 50, F0, 60, 76, 61C  
1070 DATA 80, 04, 76, 00, 07, 76, 76, 01, B9, 33, 76, 76, 01, FC, FC, 5BF  
1080 DATA 33, 33, 27, 0F, CF, CF, C7, C8, 83, 3C, 3C, 79, 76, 00, 00, 5BE  
1090 DATA 15, 7F, FF, AA, 00, 03, 42, 7E, 20, 15, 2F, 0F, 0F, 4F, CF, 4A0  
1100 DATA 9B, 27, 2D, 8F, 4F, CF, CF, 9B, AC, 80, C0, 0C, C0, 40, C0, 7BE  
1110 DATA C0, 00, 76, 00, 13, 04, 1B, 27, 0F, 27, 0F, BC, C0, 00, C0, 460  
1120 DATA 84, CC, 0D, 33, 76, 76, 01, 33, 76, 76, 01, FC, 76, 33, 04, 5C6  
1130 DATA 76, 76, 01, 27, 4F, CF, CF, CB, 83, 83, 3C, 79, F3, A2, 76, 792  
1140 DATA 00, 0D, 51, BC, 28, 00, A7, 33, 27, 0F, 03, 42, 4F, CF, 8F, 444  
1150 DATA 76, CF, 05, C3, C7, 76, CF, 05, C7, C3, CB, 76, C3, 04, 83, 833  
1160 DATA E9, 76, C3, 04, C7, CF, C7, C7, CB, C7, CF, CF, 8F, 0F, 0F, 927  
1170 DATA 8F, 76, 0F, 0C, 27, 0F, 1B, 27, 33, 03, 42, 27, 4F, CF, CB, 420  
1180 DATA C3, 03, 00, 03, 42, A2, 00, 03, 42, F3, 00, 51, 76, 00, 06, 3B2  
1190 DATA 11, FC, 33, F9, 00, A2, 11, 33, 36, 3C, 0F, 1B, 76, 0F, 07, 447  
1200 DATA 76, CF, 07, 76, C3, 0E, CB, CF, C3, C7, 76, CF, 0D, 4F, CF, 827  
1210 DATA CF, 8F, 0F, 4F, CF, 8F, 8F, 0F, 03, 42, 33, 0F, 4F, 8F, CF, 5EC  
1220 DATA CB, 83, A2, 76, 00, 00, A2, A2, 00, 03, 42, 54, 22, 00, 00, 470  
1230 DATA 51, 2A, 00, FC, 33, B9, 76, 33, 06, 76, 0F, 04, 36, 39, 0F, 419  
1240 DATA 76, CF, 04, C3, CF, CB, B9, C3, 03, 42, CF, C3, CF, CB, 8F, 922  
1250 DATA 5E, 27, CF, CF, C7, CF, CF, C3, C3, C7, 03, 42, 76, CF, 0A, 869  
1260 DATA CB, 76, CF, 04, 8F, 0F, 0F, 33, 0F, 4F, DF, 7C, FD, 76, 00, 620  
1270 DATA 05, 76, 33, 04, 00, B7, 3F, F3, 3F, A2, 00, 51, BC, AB, 76, 5A7  
1280 DATA 00, 06, 7F, 7C, 7C, BC, B9, FC, FC, 76, 33, 05, 1B, 0F, 1B, 5DD  
1290 DATA 3C, B9, 0F, 76, CF, 04, C7, CB, C7, C7, CF, 8F, 4F, CF, CF, 8B0  
1300 DATA 8F, CF, 03, 42, C7, CB, CF, C7, CF, C3, 03, 42, CF, 03, 42, 7B6  
1310 DATA CB, CF, 03, 42, C7, 4F, 76, C3, 04, C7, CF, CF, 76, 0F, 04, 720

1320 DATA 4F, 7C, B7, FF, 76, 00, 11, 2A, 00, B3, FC, 03, 42, BF, 76, 65B  
1330 DATA 00, 00, 51, 76, 76, 01, FC, 03, 42, 76, 33, 04, 27, 76, 76, 447  
1340 DATA 01, 7C, 33, 0F, 0F, CF, CF, 0F, CF, 4F, CF, 8F, 0F, 03, 42, 5CB  
1350 DATA 4F, 76, CF, 00, C7, 76, 76, 01, 27, CF, 03, 42, C7, CF, CF, 6F0  
1360 DATA C7, CF, CF, C3, CF, C7, CB, CF, CF, 0F, 5F, FF, 3C, B7, FC, A83  
1370 DATA BF, A2, 76, 00, 10, 51, 7B, 00, 55, AA, 00, BB, 33, 76, 76, 58C  
1380 DATA 01, FC, 37, 3B, 76, 76, 01, 3C, FC, 76, 33, 04, 77, FF, F3, 6AA  
1390 DATA 76, 00, 0A, 76, FF, 04, 76, 00, 00, BF, 3F, 76, 00, 05, 3F, 42F  
1400 DATA 7F, FF, FF, BB, 33, 33, BF, 7F, FF, 77, BF, 76, 00, 04, 3F, 7CA  
1410 DATA 2A, 00, FF, BB, 77, BF, FB, 76, 00, 00, 15, FC, 76, 33, 00, 655  
1420 DATA B9, 33, FF, 77, 77, 7B, B7, 33, 33, FD, 3B, 00, 03, 42, 11, 5FF  
1430 DATA 33, 03, 42, 77, FF, FE, 33, FC, 77, FF, BB, BF, A2, 00, 00, 7AD  
1440 DATA F7, FF, 76, 33, 06, 77, FD, 7C, FF, FF, 33, 77, BB, 33, BB, 8E6  
1450 DATA 33, 37, FF, 33, 03, 42, 77, F3, F3, 33, FF, 33, 77, FF, FF, 818  
1460 DATA FC, BB, 33, 00, 55, F3, A2, 15, 7F, 33, 33, 00, 77, FF, FF, 743  
1470 DATA BF, 76, 3F, 04, F3, 3F, 03, 42, 33, 03, 42, 77, 3F, 2A, 76, 4BD  
1480 DATA 00, 04, 3B, FF, 33, 51, 00, 00, 76, 33, 06, BE, FC, FC, 00, 527  
1490 DATA 00, F3, F7, 33, 33, FC, BC, 33, FF, FD, FC, FC, FF, FF, FD, 82A  
1500 DATA FC, BB, BB, 76, 33, 06, FC, FC, 3C, FF, 03, 42, 7F, FF, 03, 81A  
1510 DATA 42, 77, 33, FF, FC, 3C, FC, FF, FF, 76, 33, 04, FF, BB, 37, 8BB  
1520 DATA 3F, FB, F3, 00, 00, A2, 76, 00, 05, 55, 3F, 3F, 76, 76, 01, 50A  
1530 DATA 00, 51, B7, 3F, AA, 76, 00, 07, 3B, 33, 03, 42, 3F, B3, 33, 446  
1540 DATA 33, 77, FF, FC, 37, 00, 51, 3B, BB, 33, 33, 77, FF, 3F, 7B, 689  
1550 DATA F3, B7, F7, 33, 77, FE, FF, BB, 33, FF, BF, 7B, F3, 00, 00, 962  
1560 DATA 3F, 7F, FF, FF, BE, 7C, 7C, BB, 77, BB, FF, BF, 3F, 76, FF, 901  
1570 DATA 04, 76, 33, 00, 77, BB, BF, 2A, 00, 11, 33, 7B, 76, 3C, 35, 476  
1580 DATA FC, B9, 33, 76, 76, 01, FC, 7C, FC, 33, 1B, 76, 76, 01, FC, 780  
1590 DATA 0C, 3C, 7C, 33, 76, 3C, 4D, FC, B9, 33, 33, FC, 3C, FC, 33, 728  
1600 DATA 1B, 76, 76, 01, FC, BC, 3C, 7C, 33, 3C, 03, 42, 76, 0C, 07, 4B5  
1610 DATA 3C, 3C, 0C, 0C, 76, 3C, 04, 0C, 0C, 3C, 0C, 0C, 76, 3C, 04, 268  
1620 DATA 0C, 0C, 3C, 3C, 76, 0C, 06, 76, 3C, 28, 7C, B9, 33, 33, 76, 403  
1630 DATA 76, 01, FC, FC, B9, 0F, 76, 76, 01, FC, BC, 3C, 7C, 33, 3C, 703  
1640 DATA 03, 42, 76, 8C, 07, 9C, 3C, BC, BC, 9C, 3C, 3C, 2C, 8C, 8C, 596  
1650 DATA 3C, BC, 8C, 76, 3C, 04, 8C, BC, 3C, 2C, 76, 8C, 06, 9C, 76, 5A4  
1660 DATA 3C, 27, 7C, FC, 33, 03, 42, FC, 03, 42, 27, 33, FC, BC, 3C, 5E2  
1670 DATA 7C, 33, 3C, 03, 42, 76, 4C, 00, 3C, 4C, 03, 42, 3C, 3C, 4C, 38B  
1680 DATA 03, 42, 3C, 4C, 4C, 76, 3C, 04, 4C, 4C, 3C, 76, 4C, 00, 76, 3E3  
1690 DATA 3C, 28, FC, 33, 03, 42, 76, 76, 01, FC, B9, 27, 1B, 76, 76, 5A0  
1700 DATA 01, FC, 3C, FC, 33, 3C, 03, 42, 76, 90, 00, 3C, 90, 03, 42, 518  
1710 DATA 9C, 38, 90, 03, 42, 3C, 90, 98, 76, 3C, 04, 98, 98, 3C, 76, 585  
1720 DATA 98, 00, 76, 3C, 19, BC, 76, 3C, 0D, 7C, FC, 33, 03, 42, 76, 54C  
1730 DATA 76, 01, BC, FC, 27, 1B, 76, 76, 01, FC, BC, FC, 27, 76, 3C, 6EB  
1740 DATA 09, 30, 30, 3C, 38, 76, 30, 06, 34, 3C, 30, 30, 76, 3C, 04, 30F  
1750 DATA 30, 30, 3C, 38, 30, 03, 42, 3C, 3C, 30, 30, 76, 3C, 19, DC, 3C8  
1760 DATA 3C, 7C, FC, FC, BC, BC, 76, 3C, 07, 7C, FC, B9, 33, 03, 42, 78A  
1770 DATA FC, FC, 33, 0F, 76, 76, 01, FC, BC, FC, 27, 76, 3C, 09, 32, 6EF  
1780 DATA 32, 3C, 3C, 76, 32, 06, 3C, 3C, 32, 32, 76, 3C, 04, 32, 32, 34E  
1790 DATA 3C, 3C, 32, 03, 42, 36, 3C, 32, 32, 76, 3C, 19, 5C, BC, 76, 41E  
1800 DATA FC, 06, BC, 3C, 03, 42, 7C, 3C, 7C, FC, 76, 33, 04, 76, 76, 60B  
1810 DATA 01, FC, 33, 0F, 76, 76, 01, FC, FC, B9, 27, 3C, 03, 42, 31, 586  
1820 DATA 31, 76, 3C, 04, 31, 31, 3C, 3C, 39, 76, 31, 04, 34, 3C, 3C, 351  
1830 DATA 76, 31, 00, 3C, 3C, 39, 31, 03, 42, 76, 3C, 1B, 7C, 5C, FC, 477  
1840 DATA FC, 33, FC, 76, 76, 01, FC, 03, 42, 7C, 3C, 3C, 7C, 3C, 3C, 641  
1850 DATA FC, 76, 33, 04, 76, 76, 01, FC, 33, 0F, 33, FC, FC, B9, 27, 6DF  
1860 DATA 3C, 03, 42, 33, 33, 76, 3C, 04, 33, 33, 3C, 03, 42, 76, 33, 32D  
1870 DATA 04, 3C, 03, 42, 76, 33, 00, 3C, 03, 42, 33, 03, 42, 36, 76, 2DB  
1880 DATA 3C, 12, 00, 03, 42, 2A, 15, 76, 00, 07, F6, AA, 51, CF, CF, 4DE  
1890 DATA 8E, 34, 38, B4, 3C, 78, 3C, 03, 42, 78, 78, 3C, 78, 58, 00, 58F  
1900 DATA CC, 30, 30, 8C, 70, 64, 98, 30, 98, 80, 4C, 98, 8C, 30, 30, 63C



1910 DATA 64, CC, 98, 30, CC, CC, 30, 84, 98, 30, CC, 00, EC, 5E, 89, 7C3  
1920 DATA 76, 76, 02, 76, FC, 06, 3C, 03, 42, BC, 3C, 7C, FC, FC, B9, 70C  
1930 DATA 27, 0F, CF, C3, C3, 03, 03, 76, 3C, 05, 37, 00, 00, B7, F3, 529  
1940 DATA FF, 51, 3F, 3F, A2, 00, 03, 42, 55, A8, 00, 2F, 4F, CA, 94, 50E  
1950 DATA 70, 78, 1C, 3C, EA, 98, 98, 24, 30, 98, 24, C4, CC, 4C, 0C, 6CC  
1960 DATA CC, 84, CC, 0C, 4C, 0C, 40, 40, 0C, 4C, C0, 00, 40, 90, 40, 530  
1970 DATA 84, 0C, 18, 30, 40, C0, 8C, 48, D0, 3C, B0, 0C, 0C, F0, F0, 668  
1980 DATA 64, 40, C0, C0, 00, 91, B9, FC, FC, 7C, FC, 8C, FC, FC, B9, A58  
1990 DATA 33, 0F, 0F, CF, CF, C3, 83, 3C, 03, 42, 39, 76, 00, 06, 15, 400  
2000 DATA 00, F3, 7B, A2, 00, 03, 42, F6, A8, 51, 27, 0F, CF, 03, 42, 50E  
2010 DATA 65, 0C, 84, 8C, 0C, 40, 0C, 84, 84, C0, 40, C0, 03, 42, 00, 4F6  
2020 DATA 40, 0C, 84, 40, 0C, 80, 04, 0C, 0C, 04, 00, 76, 00, 09, 48, 293  
2030 DATA 84, 54, B0, 00, C0, C0, 76, 00, 07, C0, C0, FC, 33, 33, FC, 773  
2040 DATA 03, 42, B9, 33, 33, 0F, CF, C0, C7, C0, 03, 3C, 3C, 28, 76, 50B  
2050 DATA 00, 09, 7F, FF, AA, 00, 03, 42, FE, 28, 00, A7, 0F, 03, 42, 497  
2060 DATA 8F, CF, 87, 6D, 70, CF, 04, 9A, 39, CB, C0, C3, 76, C0, 04, 7F6  
2070 DATA 76, 00, 04, 40, 76, 00, 0F, 85, 76, 0F, 04, 78, 33, 13, 4F, 350  
2080 DATA CF, CF, C0, C3, 83, 3C, 22, F3, 76, 00, 0E, 51, FC, 28, 00, 6F9  
2090 DATA 83, 33, 27, 76, 0F, 05, 8F, 4F, 4F, CF, 03, 42, CB, CF, 03, 575  
2100 DATA 42, C7, 76, C3, 0F, CF, 03, 42, CB, CB, 76, CF, 05, 8F, 8F, 763  
2110 DATA 76, 0F, 0F, 76, 33, 04, 27, 4F, CF, CB, C3, 03, 00, 00, F3, 50A  
2120 DATA A2, 00, 03, 42, F3, 76, 00, 08, 55, FC, B9, A8, 00, A2, 11, 50D  
2130 DATA 00, 1E, 39, 1B, 27, 76, 0F, 06, 1B, 4F, 76, CF, 06, CB, C3, 520  
2140 DATA 03, 42, CB, CB, 76, C3, 06, C7, C7, 76, CF, 0E, 8F, CF, 03, 75C  
2150 DATA 42, 0F, CF, 8F, 4F, CF, 03, 42, 8F, 0F, 03, 42, 33, 27, 0F, 45E  
2160 DATA 8F, CF, CB, C3, A2, 76, 00, 0B, F3, 3F, 00, 03, 42, 54, 22, 5FC  
2170 DATA 00, 00, 51, 2A, 00, FC, FC, B9, 76, 33, 07, 0F, 03, 42, 76, 4A6  
2180 DATA 76, 01, 7C, 0F, 76, CF, 08, C3, 03, 42, CB, C7, CF, CF, 1B, 6A2  
2190 DATA 33, 4F, CB, C7, 76, C3, 06, C7, C3, 76, CF, 09, 67, CB, C3, 820  
2200 DATA CF, 03, 42, 0F, 03, 42, 33, 0F, 4F, DF, 7C, FF, 76, 00, 04, 4CD  
2210 DATA 15, 33, 03, 42, 77, 00, 3F, F3, 00, B7, A2, 00, 00, 36, 7D, 442  
2220 DATA 76, 00, 06, 51, 76, 76, 01, BC, BC, 7C, FC, 03, 42, 76, 33, 598  
2230 DATA 05, 0F, 1B, BC, 7C, 0F, CF, 03, 42, C7, CF, 03, 42, C7, 76, 5A2  
2240 DATA CF, 07, CB, C7, 03, 42, CF, C3, CB, C3, CB, C3, 76, CF, 07, 8A7  
2250 DATA C7, CF, CB, C3, C3, C7, CF, CF, 8F, 76, 0F, 04, 5F, 7C, B7, 8F6  
2260 DATA 3F, 76, 00, 11, 2A, 00, 55, BB, 76, 76, 01, B9, 76, 76, 01, 493  
2270 DATA FC, 73, 76, 00, 07, 51, BB, 7C, FC, FC, 76, 33, 04, 1B, BC, 6F0  
2280 DATA FC, 0F, 03, 42, 8F, 0F, 0F, 4F, 76, 0F, 07, 4F, 76, CF, 07, 473  
2290 DATA 1E, 27, 76, CF, 09, C7, 76, CF, 05, DF, BE, 3C, 3C, F9, 7E, 730  
2300 DATA FB, F3, 76, 00, 14, AA, 3B, 76, 33, 04, B9, BB, FC, FC, 33, 7A9  
2310 DATA 89, 76, 76, 01, 39, 76, 33, 05, FF, F3, 76, 00, 05, FF, BB, 6B4  
2320 DATA 33, 03, 42, FF, FF, AA, 76, 00, 05, FF, BB, FF, BF, 3F, 03, 755  
2330 DATA 42, 8F, FF, FF, 76, 33, 04, FF, BF, 3F, 3B, 37, 76, 00, 04, 695  
2340 DATA F3, 7F, FF, 00, 00, 33, 03, 42, 7B, 76, 00, 07, 76, 76, 01, 4CE  
2350 DATA 76, 33, 0A, 37, FF, 33, BF, 00, 03, 42, 7F, 33, 03, 42, 73, 40A  
2360 DATA 00, 00, 33, 33, 77, FF, BF, 77, 33, FC, 33, 03, 42, 73, A2, 5CE  
2370 DATA 76, 00, 04, FF, FF, BF, 3F, 7F, 33, 77, FF, FF, FD, 33, 77, 844  
2380 DATA 76, 33, 06, BF, 3F, 7F, FF, BF, A2, 00, 00, 15, 7B, BB, 77, 64E  
2390 DATA FF, FC, FC, B9, 33, AA, 76, 00, 04, 7F, 33, BB, AA, BF, 7F, 85C  
2400 DATA FF, BF, FF, 33, 77, 3F, F3, 3F, 03, 42, BB, 33, 33, FF, 7B, 7B0  
2410 DATA 76, 00, 05, 11, FF, FF, 7F, FB, 83, 33, 33, 37, F3, 3F, 3B, 6C1  
2420 DATA 77, FF, 33, 3B, B7, 3B, BB, 33, 03, 42, FC, B9, 33, FC, 3C, 729  
2430 DATA 3C, FF, FF, BC, FF, BB, 76, 33, 07, FF, FE, BC, 7C, BF, 3F, 993  
2440 DATA 7F, FF, FF, BB, 33, 33, FF, BE, FC, FC, 76, 33, 04, 77, BB, 932  
2450 DATA 33, 33, BF, 7F, FF, 3F, F3, 00, 51, 76, 00, 05, 15, A2, 7B, 5D3  
2460 DATA 33, 00, 51, B7, 3F, 2A, 76, 00, 06, 3B, 76, 33, 04, 77, 11, 390  
2470 DATA 77, FF, 03, 42, 33, 33, 3F, 7F, FF, BB, 33, 03, 42, FF, 03, 66B  
2480 DATA 42, 3B, F7, F7, 33, 77, FC, FF, 33, 03, 42, 77, FF, FF, 3F, 83C  
2490 DATA F3, 00, FF, FD, FF, BE, FC, FC, 33, 03, 42, BF, 3F, 3B, 77, BCC

2500 DATA FF, FD, FE, FE, FD, 76, 33, 00, 00, 03, 42, 15, 3B, FB, FC, 832  
2510 DATA FC, 76, 33, 06, 0F, 33, 27, 0F, 0F, 14, 00, 40, 05, 76, 0F, 3C8  
2520 DATA 05, CF, 8F, CF, CF, 0F, 1B, 4F, 8F, 0F, 0F, 76, 33, 04, 27, 4FB  
2530 DATA 0F, 1B, 1B, 27, 0F, 4F, CF, 3C, 0C, 03, 42, 1C, 0C, 3C, 3C, 2C6  
2540 DATA 0C, 2C, 0C, 03, 42, 1C, 76, 3C, 04, 2C, 0C, 3C, 03, 42, 0C, 220  
2550 DATA 03, 42, 1C, 3C, 0C, 03, 42, 3C, 2C, 0C, 03, 42, 76, 3C, 0A, 263  
2560 DATA 7C, FC, FC, 33, 33, 27, 33, 33, 0F, 03, 42, 1E, CC, CC, 4A, 5BB  
2570 DATA 94, CC, 00, 45, 8F, 4F, 8F, CF, 4F, CF, 03, 42, 8F, 1B, 27, 695  
2580 DATA 4F, CF, 0F, 0F, 1B, 33, 33, 27, 27, 0F, 1B, 1B, 0F, 0F, CF, 33D  
2590 DATA CF, 3C, 8C, 03, 42, 9C, 8C, 3C, 3C, 8C, 2C, 8C, 03, 42, 9C, 5A1  
2600 DATA 76, 3C, 04, 2C, 8C, 3C, 03, 42, 76, 8C, 04, 2C, 8C, 03, 42, 3F2  
2610 DATA 9C, 2C, 8C, 03, 42, 76, 3C, 0A, BC, FC, B9, 33, 33, 8D, 33, 5EC  
2620 DATA 33, 0F, 03, 42, CC, 85, 4A, 88, 34, B0, 80, C5, 8F, 8F, 0F, 600  
2630 DATA 0F, 76, CF, 04, 8F, 1B, 27, CF, CF, 8F, 0F, 1B, 76, 33, 04, 52D  
2640 DATA 0F, 1B, 0F, 0F, 4F, CF, CF, 76, 3C, 05, 4C, 3C, 3C, 4C, 76, 472  
2650 DATA 3C, 09, 6C, 4C, 76, 3C, 06, 4C, 6C, 4C, 1C, 6C, 1C, 76, 3C, 40F  
2660 DATA 0C, BC, 3C, 7C, FC, FC, 33, 4E, 4C, 0F, 03, 42, C0, C0, C4, 6E5  
2670 DATA B4, E4, 80, 34, B0, C0, 05, 87, CF, 8F, 0F, CF, 4F, CF, CF, 871  
2680 DATA 8F, 33, 27, CF, 03, 42, 0F, 0F, 33, 03, 42, 1B, 4F, 1B, 0F, 327  
2690 DATA 0F, 4F, CF, CF, 3C, 6C, CC, 3C, 3C, CC, 6C, CC, CC, 6C, CC, 7F0  
2700 DATA 03, 42, 9C, 76, 3C, 04, 6C, CC, 3C, 03, 42, CC, 3C, 3C, CC, 560  
2710 DATA 3C, CC, CC, 3C, 03, 42, CC, 9C, 76, 3C, 09, 76, FC, 04, 33, 621  
2720 DATA 0F, 48, C0, 0D, 0F, 0E, D4, C0, 3C, 3C, 38, C8, 34, B0, C0, 5F1  
2730 DATA 45, 87, CF, 8F, 76, CF, 05, 1B, B9, 4F, CB, CF, 03, 42, 0F, 685  
2740 DATA 1B, 33, 03, 42, 4F, 9B, 0F, 0F, CF, CF, CB, 3C, 38, 98, 3C, 54C  
2750 DATA 3C, 98, 38, 98, 98, 38, 98, 03, 42, 9C, 76, 3C, 04, 38, 98, 573  
2760 DATA 3C, 03, 42, 76, 98, 04, 3C, 3C, 98, 98, 3C, 3C, 98, 9C, 76, 55D  
2770 DATA 3C, 08, 7C, FC, 03, 42, BB, 0C, C4, 58, 1B, 0C, 84, C0, F8, 6A1  
2780 DATA 1C, 38, 2C, 0C, C0, 34, E4, C0, 45, C7, 76, CF, 07, 5E, 39, 613  
2790 DATA CF, 03, 42, C7, CF, 0F, 1B, 33, 33, 1B, 0F, 8F, 8F, CF, 03, 40A  
2800 DATA 42, CB, 3C, 39, 31, 3C, 3C, 31, 3C, 3C, 31, 39, 31, 76, 3C, 421  
2810 DATA 07, 39, 31, 3C, 03, 42, 76, 31, 04, 39, 34, 39, 31, 34, 3C, 2E4  
2820 DATA 31, 34, 76, 3C, 08, FC, 03, 42, 31, 24, B0, 3C, BC, 3C, 3C, 4DD  
2830 DATA E0, D0, 90, 70, 38, E4, 38, BC, 34, B0, C0, 41, C7, 76, CF, 081  
2840 DATA 07, 1B, 33, CB, C7, CF, C7, CF, 8F, 0F, 33, 33, 1B, 0F, CF, 649  
2850 DATA 0F, 8F, 9B, CF, CB, 3C, 38, 32, 32, 3C, 32, 3C, 3C, 32, 3B, 4FB  
2860 DATA 32, 03, 42, 36, 76, 3C, 04, 38, 32, 03, 42, 36, 32, 3C, 3C, 2F2  
2870 DATA 32, 38, 32, 03, 42, 36, 3C, 32, 32, 36, 76, 3C, 07, FC, 89, 45B  
2880 DATA 66, C4, 9C, 3C, 36, 3C, 3C, 78, 0C, 0C, B0, 98, 78, 38, B0, 5E0  
2890 DATA C8, 9C, E4, 40, C0, C7, 76, CF, 07, 1B, 27, CB, C7, CB, CB, 8CD  
2900 DATA C7, CF, 0F, 1B, 33, 1B, 0F, 4F, 0F, CF, DE, 67, CB, 3C, 3C, 5D2  
2910 DATA 33, 33, 3C, 33, 3C, 3C, 33, 39, 33, 03, 42, 36, 76, 3C, 04, 31D  
2920 DATA 39, 33, 03, 42, 36, 33, 3C, 3C, 33, 3C, 33, 03, 42, 3C, 3C, 2F1  
2930 DATA 39, 33, 36, 76, 3C, 07, FC, B9, AC, B4, 3C, 78, 3C, 03, 42, 5A5  
2940 DATA A4, C0, 0C, 58, 64, 98, 78, F0, 20, 1C, 24, 40, C0, C5, 76, 6CF  
2950 DATA CF, 07, 1B, 0F, CB, C7, CB, 03, 42, CF, 8F, 0F, 33, 0F, 27, 578  
2960 DATA 4F, 0F, 4F, 9E, 67, CB, 78, 3C, 28, 7C, 3C, 3C, FC, 76, 76, 633  
2970 DATA 01, C8, B4, 3C, B4, 34, 3C, 6C, 80, 80, 84, 40, 64, 98, 38, 641  
2980 DATA B0, 88, 9C, CC, 48, 80, C7, C7, 76, CF, 05, 76, 3C, 06, 00, 6F8  
2990 DATA 51, 00, 00, A2, 00, F3, 76, 00, 05, 54, A2, 55, CF, 9A, DA, 5EF  
3000 DATA 3C, 78, 3C, 3C, B4, 3C, 3C, 78, F0, B4, 3C, A4, C4, F0, 90, 738  
3010 DATA 30, 98, 0C, 98, BC, 4C, CC, CC, 80, 0C, CC, 8C, 30, 30, 80, 828  
3020 DATA 00, 76, C0, 04, 84, 0C, CC, 30, CC, C8, 22, 1B, 76, 76, 01, 584  
3030 DATA 33, 33, FC, 76, 76, 01, 33, FC, 03, 42, BC, 3C, 3C, BC, 3C, 5EF  
3040 DATA 7C, FC, FC, 33, 27, 0F, CF, CB, C3, 83, 03, 76, 3C, 05, 73, 6EA  
3050 DATA 00, 00, F3, 51, FF, 51, 3F, 7B, A2, 00, 03, 42, 11, A8, 00, 4EE  
3060 DATA 45, CF, CF, C0, D0, A4, 3C, 6C, 98, B0, CC, 4C, 64, 38, 0C, 78F  
3070 DATA C4, CC, 4C, 8C, BC, 84, CC, 0C, 0C, 88, C0, 04, 0C, 03, 42, 5F9  
3080 DATA 80, 40, 4C, 40, C0, C4, B4, 78, 98, 34, 3C, 70, 78, 60, 04, 650



3090 DATA F0, 78, BC, B6, 00, 00, 40, C0, 80, 10, 76, 76, 01, B9, FC, 6A6  
 3100 DATA 03, 42, BC, FC, 03, 42, 33, 0F, 4F, 4F, C3, C3, 03, 3C, 03, 56A  
 3110 DATA 42, 3D, 76, 00, 06, 51, 76, 00, 07, 54, AB, 15, AD, 0F, 4F, 3E5  
 3120 DATA 76, CF, 04, 48, 84, 0C, 0C, 84, 84, 0C, 76, C0, 04, 80, 00, 4FB  
 3130 DATA 00, 40, 44, C0, 0C, 00, 40, 84, 0C, 84, C0, 76, 00, 09, 40, 4AB  
 3140 DATA C0, 0C, 64, 40, 48, C0, 00, 03, 42, 40, C0, C0, 00, 80, 04, 501  
 3150 DATA FC, 33, 33, FC, 03, 42, 89, 33, 33, 0F, CF, CB, C7, C3, 03, 6FB  
 3160 DATA 3C, 3C, 28, 00, A2, 76, 00, 07, 15, 7F, 2A, 00, 03, 42, 7E, 340  
 3170 DATA AB, 00, A7, 0F, 4F, 0F, 0F, CF, CF, 4F, 8F, 4F, CF, 03, 42, 5AA  
 3180 DATA 38, C7, CB, C0, 84, 0C, C0, C0, 76, 00, 13, 04, 76, 0F, 06, 5B2  
 3190 DATA 18, 33, 27, 27, 76, 33, 0E, 0F, CF, 03, 42, C3, 03, 79, 76, 42B  
 3200 DATA 00, 10, 15, FC, 8D, 00, 11, 27, 33, 76, 0F, 00, CF, 03, 42, 3EA  
 3210 DATA C7, CF, 03, 42, C7, 76, C3, 11, CF, CF, CB, 8F, 4F, CF, 03, 605  
 3220 DATA 42, 0F, CF, 8F, 76, 0F, 10, 33, 33, 0F, 4F, CF, CF, C3, 03, 5EC  
 3230 DATA 00, 51, F3, A2, 76, 00, 0D, 7E, B9, 22, 00, 00, 11, B9, 76, 502  
 3240 DATA 76, 01, B9, 33, 33, 27, 0F, 10, 0F, 03, 42, 9E, 4F, 76, CF, 46D  
 3250 DATA 06, C3, 03, 42, C7, 13, 63, 76, C3, 06, C7, C7, C3, CF, CB, 775  
 3260 DATA C7, CF, 0F, 76, CF, 07, 9B, 76, CF, 09, 0F, CF, CF, 8F, 0F, 025  
 3270 DATA 0F, 33, 27, 0F, 8F, CF, CB, D7, F3, 2A, A2, 76, 00, 8A, FF, 686  
 3280 DATA 2A, 00, 00, 54, AA, 78, 00, 05, 3B, FC, FC, 76, 76, 01, 76, 539  
 3290 DATA 33, 06, 0F, 03, 42, 5E, 3C, 27, 4F, 76, CF, 04, 8F, 4F, CB, 48F  
 3300 DATA CB, C7, C3, CB, CF, 0F, 0F, 33, 89, 4F, C7, C7, C3, 03, 42, 7DE  
 3310 DATA C7, 03, 42, CF, C7, 76, CF, 00, CB, 87, CF, CF, C7, CF, CF, 924  
 3320 DATA 0F, 0F, 0F, 33, 0F, 4F, DF, 7D, FF, 76, 00, 04, 7F, FF, BB, 64C  
 3330 DATA 33, 73, 00, 3F, 00, 00, 51, A2, 00, 51, FC, BC, AA, 76, 00, 501  
 3340 DATA 06, BB, 3C, FC, 7C, FC, 03, 42, 76, 33, 05, 0F, 0F, 36, 3C, 4F4  
 3350 DATA 27, 4F, 76, CF, 05, C7, 76, CF, 05, 8F, 76, CF, 05, CB, CF, 744  
 3360 DATA C7, CF, C3, C3, C7, CF, CB, 76, CF, 04, C7, CF, CF, 76, C3, A64  
 3370 DATA 04, CF, CF, 8F, 76, 0F, 04, 9E, 7C, FB, 7F, 76, 00, 11, A2, 677  
 3380 DATA 00, 51, BF, 0B, 89, 76, 76, 01, 33, FC, 76, 00, 09, 3B, BC, 616  
 3390 DATA FC, B9, 76, 33, 04, 86, FC, 33, 76, 0F, 0D, 4F, CF, 4F, 76, 63C  
 3400 DATA CF, 05, 5E, 27, 76, CF, 06, CB, 76, CF, 06, FF, 3C, 3C, 7C, 6AD  
 3410 DATA FC, FC, F9, F7, BF, F3, 76, 00, 14, FF, 76, 76, 01, B9, 33, 0FC  
 3420 DATA 76, 76, 01, 33, 33, B9, 76, 33, 05, 73, 83, 76, 76, 01, FC, 5C9  
 3430 DATA 3C, 39, 39, 03, 42, 77, FB, A2, 00, FF, 33, 76, 76, 01, 33, 553  
 3440 DATA 03, 42, FC, FC, FF, 2A, 00, 15, F7, FF, 33, 03, 42, 7F, FF, 767  
 3450 DATA FF, BF, BB, 33, 33, 3C, 3C, FC, FC, BF, 7F, 33, FF, BF, 00, 87E  
 3460 DATA 15, F3, FF, 00, 33, 77, AA, 00, BF, 3B, FB, 76, 00, 06, 55, 6DC  
 3470 DATA 76, 33, 06, FF, 33, 77, BB, 33, 37, FF, 00, 73, 00, 03, 42, 5EF  
 3480 DATA F7, 33, 33, B9, 33, 7B, 03, 33, FF, BF, 87, 3F, FF, 76, 33, 006  
 3490 DATA 04, 22, 76, 00, 07, 33, 37, 3F, 3B, 33, 77, FC, FC, FD, 33, 559  
 3500 DATA 03, 42, 77, 76, 33, 05, FF, 7B, F3, 76, 00, 06, 11, 33, 77, 50E  
 3510 DATA FC, FE, FC, BB, FF, BF, 2A, 00, 00, 15, 3F, 3B, FF, 55, FF, 87B  
 3520 DATA BF, 3F, 3F, FF, FF, 33, 7B, 00, F3, 3F, FF, 33, 33, 77, 76, 76D  
 3530 DATA 00, 06, 55, BF, 7F, BF, F3, BB, 37, 7B, A2, 00, F3, F3, 33, 773  
 3540 DATA 77, BB, FF, BF, FF, BB, 33, 76, 76, 01, 33, 03, 42, 77, FD, 7B6  
 3550 DATA FF, BC, FC, FC, BC, FC, FD, FF, FF, BB, 33, 03, 42, BB, 33, A07  
 3560 DATA FF, FF, FE, FC, BF, 3F, FF, 76, 33, 04, FC, FC, FD, FF, FF, B95  
 3570 DATA AA, 55, BB, FE, BC, FF, FF, 3F, 7F, FF, FF, 7B, A2, 76, 00, 9C1  
 3580 DATA 06, 51, 00, BB, 76, 76, 01, 00, 51, F3, F3, B7, 22, 00, 03, 512  
 3590 DATA 42, 51, 33, FF, BF, 3F, F3, 7F, 33, 37, 3B, 33, 33, 77, FF, 686  
 3600 DATA FF, BB, FF, 76, 33, 05, F3, 3F, 77, FF, 3F, F3, FF, 33, FF, 972  
 3610 DATA BB, 78, 33, 06, 76, FF, 05, FC, 03, 42, FD, BB, 33, BF, 3F, 70E  
 3620 DATA FF, FF, 77, FC, FC, BC, FF, FF, 76, 33, 06, 77, 3F, 76, 00, 902  
 3630 DATA 04, 15, 7B, C3, CF, CF, 4F, C3, C3, C7, CB, C3, C3, C7, 8F, 938  
 3640 DATA 0F, 27, 4F, CF, C3, 03, 42, 03, 69, 3C, 3C, 69, 3C, 3C, 96, 487  
 3650 DATA 69, C3, 69, C3, 69, C3, 76, 3C, 00, 7C, FC, 89, FC, 76, 76, 857  
 3660 DATA 01, FC, BC, B9, 33, 76, 76, 01, FC, 34, 3C, 74, 3C, 3C, 78, 662  
 3670 DATA 76, 3C, 05, F0, 80, 9C, 60, 40, C0, 80, 00, 00, 40, 80, 20, 83B

3680 DATA 00, 00, 00, 04, 40, B4, 98, B4, 0C, 80, 00, 40, 80, 04, CC, 480  
 3690 DATA C3, 03, 42, C7, 8F, 4F, 76, C3, 07, 8F, 0F, 27, 4F, CB, C3, 68F  
 3700 DATA C3, 43, 03, 6D, 69, C3, 6D, 3C, 2D, 1E, 76, 2D, 06, 76, 3C, 4F1  
 3710 DATA 09, FC, FC, 33, 33, 76, 76, 01, BC, 33, 03, 42, 31, 3C, F0, 5E5  
 3720 DATA 3C, 03, 42, B4, 7B, 70, B4, 6B, D8, 3C, 78, 1C, 60, C0, 03, 604  
 3730 DATA 42, 00, 00, 40, 80, 6B, 76, 00, 04, 40, 84, 84, C0, C4, 40, 4F0  
 3740 DATA 00, 40, C0, 84, F0, 76, C3, 04, 8F, 0F, 76, C3, 06, C7, CF, 724  
 3750 DATA BF, 33, 4F, CB, C3, C3, 83, 03, 39, 6D, 6D, 39, 3C, 3C, 36, 5E2  
 3760 DATA 76, 39, 06, 76, 3C, 09, 7C, FC, 33, 27, 27, FC, 33, 33, 32, 4FD  
 3770 DATA 3C, 38, 3C, 38, 76, 3C, 05, 78, B4, 3C, 03, 42, 9C, E0, 76, 53E  
 3780 DATA C0, 04, 00, 03, 42, C8, C0, 00, 00, 40, 40, 00, 18, E4, BC, 499  
 3790 DATA 4C, 08, 00, 40, C4, 78, 83, C3, 03, 42, 8F, 27, 76, C3, 07, 551  
 3800 DATA CF, CF, 1B, 4F, CF, C3, C3, 83, 03, 39, 39, 3C, 39, 3C, 3C, 642  
 3810 DATA 36, 39, 33, 39, 33, 39, 39, 76, 3C, 07, BC, 3C, 7C, 33, 03, 3E3  
 3820 DATA 42, 1B, 33, 33, 32, 36, 3C, 34, 3C, 03, 42, 78, 3C, 03, 42, 315  
 3830 DATA 4C, 30, 30, BC, BC, 1C, 20, C0, 40, 40, 00, 00, 40, 00, 88, 408  
 3840 DATA 00, 00, 00, 40, 00, 90, F0, B0, 64, C0, 0C, 00, 84, 04, 6C, 594  
 3850 DATA 03, C3, 03, 42, 9B, B9, C3, C3, CB, 76, C3, 04, CF, 8F, 1B, 766  
 3860 DATA 4F, CB, C3, C3, 03, 03, 76, 6D, 04, 3C, 3C, 1E, 3C, 2D, 3C, 4C8  
 3870 DATA 2D, 03, 42, 76, 3C, 06, 7C, FC, 3C, 3C, B9, 33, 33, 27, 1A, 47A  
 3880 DATA 33, 34, 3C, 3C, 9C, 3C, 3C, 78, 30, 78, F0, F0, 76, 3C, 05, 5AA  
 3890 DATA 70, E0, 40, C0, 40, 40, 00, 40, 00, 88, C0, 80, 00, 80, C4, 624  
 3900 DATA B4, F0, 76, D8, 00, 40, C0, 84, 40, 8C, 03, 03, C3, C3, 8F, 767  
 3910 DATA 39, C3, CF, CF, 76, C3, 04, C7, CF, 1B, 4F, CB, CB, 83, 03, 7F3  
 3920 DATA 03, 69, 69, C3, 69, 3C, 69, C3, 69, C3, 69, C3, 69, C3, 76, 763  
 3930 DATA 3C, 06, 7C, 7C, 3C, 3C, 39, 27, 33, 33, 1B, 32, 3C, 3C, 78, 385  
 3940 DATA 34, E4, 78, B4, F0, F0, 30, A4, 98, 30, CC, 0C, 90, 58, 60, 7E8  
 3950 DATA 08, 80, C0, 80, C0, 40, 80, 20, 80, 00, 40, 00, 84, D8, 80, 634  
 3960 DATA F0, D8, C0, C0, 00, 84, 40, C0, 03, 03, 43, C3, CF, 4F, C3, 7C1  
 3970 DATA CF, 8F, CB, C3, 03, 42, CF, CF, 0F, 4F, C7, CB, 83, 83, 03, 7C8  
 3980 DATA 69, 96, 3C, C3, 76, 3C, 0F, BC, 3C, FC, 3C, 7C, 0F, 33, 27, 5D4  
 3990 DATA 1B, CC, 6C, F0, 03, 42, CC, 30, 30, 98, B0, 8C, E4, 70, 80, 78C  
 4000 DATA F0, E4, 78, 64, 30, B0, 64, BC, 0C, 0C, C0, 40, E0, 80, 48, 740  
 4010 DATA 0C, C0, 18, 3C, 3C, 78, 98, E4, 48, 80, 00, 40, 80, 03, 03, 40E  
 4020 DATA 43, C7, CB, CB, C3, CF, 76, 76, 01, CB, C3, C3, CF, 03, 42, 804  
 4030 DATA 0F, 4F, CF, CB, C3, 83, 03, 3C, C3, C3, 96, 76, 3C, 0F, BC, 716  
 4040 DATA 3C, FC, BC, 7C, 33, 33, 0F, 98, 3C, 6C, F0, 8C, B0, 34, 3C, 6C1  
 4050 DATA B0, B4, 70, E4, 30, B4, 98, CC, 18, 34, E0, C0, C0, 0C, 40, 808  
 4060 DATA 84, 0C, 4C, 98, 20, 00, 84, 98, F0, F0, B0, F0, B4, CC, E4, 894  
 4070 DATA C6, 80, 00, 40, 00, 03, 03, 43, 76, C3, 04, C7, 78, 76, 01, 4C2  
 4080 DATA CB, C3, 03, 42, CF, 8F, 1B, 4F, CF, C3, C3, 83, 03, 76, 3C, 728  
 4090 DATA 15, FC, 76, 76, 01, 3C, 39, 33, 66, 3C, 64, 5C, 3C, F0, 3C, 570  
 4100 DATA B4, 38, BC, B4, 34, 30, 2C, B4, 96, 96, 64, 3C, 78, B0, 48, 6B0  
 4110 DATA 0C, 0C, 40, C0, C0, 40, 00, 00, 84, 40, 84, D8, 3C, 78, 30, 52C  
 4120 DATA 4C, A4, 8C, C0, 00, 03, 42, 03, 03, 43, C3, 03, C3, 03, 42, 498  
 4130 DATA 76, 76, 01, 48, C3, C3, C7, CF, 0F, 1B, 4F, CB, C3, C3, 03, 721  
 4140 DATA 03, 76, 3C, 0A, 78, 00, 04, 3F, 76, 00, 07, 54, 2A, 15, CF, 357  
 4150 DATA 8E, 3C, 3C, 9C, 76, 3C, 05, B4, B4, 3C, 38, 64, 1B, B0, 64, 5C5  
 4160 DATA 30, 98, C0, CC, 8C, 8C, CC, CC, 00, 0C, 4C, 8C, 30, 48, D8, 738  
 4170 DATA F0, 64, 70, F0, F0, 64, 48, 44, 98, 00, 00, 22, 76, 33, 05, 604  
 4180 DATA 76, 76, 02, FC, 03, 42, BC, 3C, 7C, FC, 3C, 3C, FC, FC, 33, 742  
 4190 DATA 27, 27, CF, CF, C3, 83, 83, 76, 3C, 04, 39, 00, 03, 42, 15, 4FE  
 4200 DATA 51, F3, 51, 3F, 7B, A2, 00, 03, 42, BC, A8, 00, 6F, CB, CF, 6A3  
 4210 DATA CF, 48, 70, D8, 58, B0, 24, 0C, 18, 64, 30, 0C, 18, CC, 4C, 57F  
 4220 DATA 4C, CB, 84, CC, 0C, 4C, C0, C0, 04, 0C, 0C, 84, B0, 40, 0C, 5A8  
 4230 DATA C0, 48, 34, B4, 68, B0, 3C, 78, 34, 3C, E4, 70, 7B, CB, 40, 700  
 4240 DATA 40, C0, 03, 42, 0C, 48, 04, 76, 76, 01, 76, FC, 07, 33, 33, 469  
 4250 DATA 8F, CF, 4F, CF, C3, C3, 3C, 03, 42, 79, 76, 00, 0E, 55, 22, 577  
 4260 DATA 15, 27, 0F, 0F, 4F, 76, CF, 04, 0C, 0C, 48, B4, 48, 0C, C0, 3EA



4270 DATA 04, C0, C0, 00, 40, C0, 00, 40, 40, 0C, 00, C0, 0C, 0C, 04, 06C  
 4280 DATA 0C, 76, C0, 00, 00, 00, 04, 04, 00, C0, 03, 42, 00, 00, C0, 517  
 4290 DATA 40, 76, 00, 04, 10, B9, 33, 33, 76, 76, 01, FC, 33, 76, 76, 4F1  
 4300 DATA 01, 33, 33, 0F, CF, C0, C3, C3, 03, 3C, 3C, AA, 51, A2, 76, 624  
 4310 DATA 00, 0D, FE, 20, 00, AF, 76, 0F, 04, CF, 0F, CF, 4F, 76, CF, 02C  
 4320 DATA 04, C0, C3, CF, C0, 04, 40, C0, C0, 76, 00, 0D, 40, 76, 00, 0A6  
 4330 DATA 05, 4D, 76, 0F, 05, 27, 0F, 33, 33, 27, 10, 76, 33, 0D, 0F, 27F  
 4340 DATA CF, CF, C7, C3, 03, A2, 76, 00, 10, 55, FC, F9, 00, 11, 33, 761  
 4350 DATA 33, 27, 76, 0F, 07, CF, 4F, CF, CF, C7, CF, CF, C0, 76, C3, 00B  
 4360 DATA 0F, CF, C0, 03, 42, C7, CF, 4F, 76, CF, 05, 0F, 76, 0F, 0E, 03F  
 4370 DATA 10, 0F, 33, 33, 0F, 4F, CF, CF, 03, 03, 00, 51, F3, A2, 76, 56E  
 4380 DATA 00, 00, A2, 00, F6, B9, 22, 00, 00, 11, 33, 03, 42, B9, 33, 3F3  
 4390 DATA 10, 0F, 0F, 27, 0F, 0F, 0E, 4F, CF, CF, 4F, 76, CF, 04, C3, 564  
 4400 DATA C3, C7, C3, 40, 76, C3, 06, C7, C0, C7, 76, CF, 0C, 0F, 76, 000  
 4410 DATA CF, 0C, 0F, 0F, 0F, 27, 27, 0F, CF, 0F, C0, 3C, 51, F3, A2, 630  
 4420 DATA 76, 00, 09, 15, BF, 3F, 00, 00, 14, 22, 76, 00, 05, 30, FC, 37A  
 4430 DATA 03, 42, B9, 33, 76, 76, 01, 33, 03, 42, 76, 0F, 04, BC, 39, 414  
 4440 DATA 4F, 76, CF, 04, 00, 2D, C3, C0, C3, 03, 42, 0F, 27, 33, 33, 612  
 4450 DATA 27, CF, CF, C3, CF, C3, 03, 42, C7, C0, C3, C7, CF, 03, 42, 00F  
 4460 DATA C0, C7, 76, CF, 05, C3, C7, C7, CF, CF, 0F, 03, 42, 10, 0F, 749  
 4470 DATA CF, DF, 7C, FF, 3F, FF, 77, FF, 03, 42, 00, 33, 2A, 51, 2A, 705  
 4480 DATA 00, 03, 42, A2, 00, 15, 76, 76, 01, F7, 20, 76, 00, 06, 15, 399  
 4490 DATA 36, 3C, BC, 7C, FC, 03, 42, 76, 33, 04, 10, 0F, 33, 3C, B9, 4EA  
 4500 DATA 0F, 76, CF, 00, 0F, 4F, 76, CF, 07, C0, C3, C7, 76, CF, 05, 0A0  
 4510 DATA C7, CF, 03, 42, C7, CF, C0, C3, C7, CF, 03, 42, 76, 0F, 04, 763  
 4520 DATA 9E, FC, F0, F3, 76, 00, 11, 30, A2, 00, 51, 00, 76, 76, 01, 6E5  
 4530 DATA B9, 7F, 00, 33, 76, 00, 09, 03, 3C, FC, 03, 42, 33, 33, 76, 501  
 4540 DATA 76, 01, 3C, B9, 76, 0F, 07, 10, 33, 33, 27, 0F, 03, 42, 0F, 303  
 4550 DATA 0F, CF, 4F, CF, 03, 42, 10, 27, 76, CF, 09, FF, A2, 00, 15, 507  
 4560 DATA 3F, BE, 3C, 3C, 00, FF, 00, F3, 3F, F3, 76, 00, 14, 54, FC, 709

4570 DATA 03, 42, 76, 33, 05, 76, 76, 02, 33, 00, 00, 15, 33, 33, FC, 300  
 4580 DATA 3C, FC, 76, 76, 01, FC, 76, 33, 07, 76, 76, 01, 3C, 3C, 33, 509  
 4590 DATA 33, FF, AA, 15, 70, FF, 76, 33, 04, 7F, 33, 33, FF, FC, FC, 7FA  
 4600 DATA 0C, 7C, FC, 76, FF, 04, 77, FD, FF, 3F, 03, 77, 00, 33, 77, 0EE  
 4610 DATA 33, 3F, 2A, 55, 7F, FB, A2, 76, 00, 05, 76, 33, 05, FF, 33, 560  
 4620 DATA FF, 33, 77, 33, 77, BF, BF, 30, 33, FB, A2, 00, 00, 33, 33, 0FD  
 4630 DATA 77, FF, 00, 37, 33, FB, F3, 07, FF, 00, 76, 33, 04, 37, 2A, 000  
 4640 DATA 76, 00, 04, 15, 7F, 37, 3F, 30, 33, 03, 42, FF, FD, FF, 77, 5A9  
 4650 DATA 33, 33, FD, FF, 00, 00, 33, 33, 76, 00, 0A, 11, 77, FF, 03, 0A0  
 4660 DATA 42, 00, 33, 33, 77, 3F, 00, 03, 42, 15, 33, 15, 7F, FB, 00, 435  
 4670 DATA F3, 70, 3F, 00, 37, A2, B7, B7, FF, 00, 33, 37, 00, 51, A2, 700  
 4680 DATA 00, 03, 42, 55, FB, B7, 77, F3, 30, F3, 76, 00, 05, 15, 33, 5A7  
 4690 DATA 03, 42, 00, 33, 03, 42, FF, FF, 33, 77, 77, FC, 33, 0E, 3C, 600  
 4700 DATA 3C, 7C, FF, FF, BF, 3F, FF, 00, 77, 00, 76, 33, 05, FE, FD, 9A9  
 4710 DATA 00, 33, 77, 33, 77, 77, 76, FF, 04, 00, 03, 42, 00, 33, FE, 030  
 4720 DATA FC, 09, 37, 3F, 00, 77, 70, A2, 76, 00, 05, 30, 33, 00, 51, 507  
 4730 DATA 00, F3, F3, 22, 00, 03, 42, 55, 77, 0F, 2A, 00, 00, 51, 7F, 402  
 4740 DATA 33, 3F, 33, 37, 33, 03, 42, FF, 00, 33, 03, 42, 37, 07, 00, 474  
 4750 DATA 15, 30, 77, BF, 70, F7, 3F, 7F, 33, BF, 00, 76, 33, 06, 77, 009  
 4760 DATA BE, 3C, FC, 76, FF, 04, BF, 00, 00, 15, FF, 00, FF, 33, 76, 7A5  
 4770 DATA 76, 01, FC, 76, 33, 05, 77, 33, FB, 00, 03, 42, B7, 00, 00, 4C2  
 4780 DATA 0F, 69, C3, 69, C3, 3C, 3C, 69, 03, 42, C3, 69, C3, 69, C3, 750  
 4790 DATA 69, C3, 69, C3, FC, 33, 03, 42, 27, 0F, 4F, 4F, 0F, 4F, 9E, 59C  
 4800 DATA 3C, F0, 70, BC, CC, CC, 10, 30, 30, 04, 40, 40, 04, 40, C0, 600  
 4810 DATA C0, 00, 40, 04, 0C, 10, 04, 0C, 00, 00, 04, 30, 4C, 24, 40, 44C  
 4820 DATA C0, 70, 00, 05, C0, 70, 40, 40, 10, 00, 01, 03, 03, 40, C3, 410  
 4830 DATA C7, CF, C3, C3, CF, 76, C3, 06, 03, 2D, 3C, 03, 42, 2D, 3C, 0C4  
 4840 DATA 3C, 1E, 3C, 3C, 2D, 0F, 76, 2D, 04, 3C, 1E, 3C, 1E, 2D, 2D, 2C3  
 4850 DATA B9, 33, 33, 0F, 27, 0F, 0F, 4F, 0F, 0F, 1C, 3C, 2C, 70, 70, 344  
 4860 DATA 00, CC, 0C, CC, 0C, 0C, CC, 0C, 0C, 40, 76, C0, 05, 40, 69F

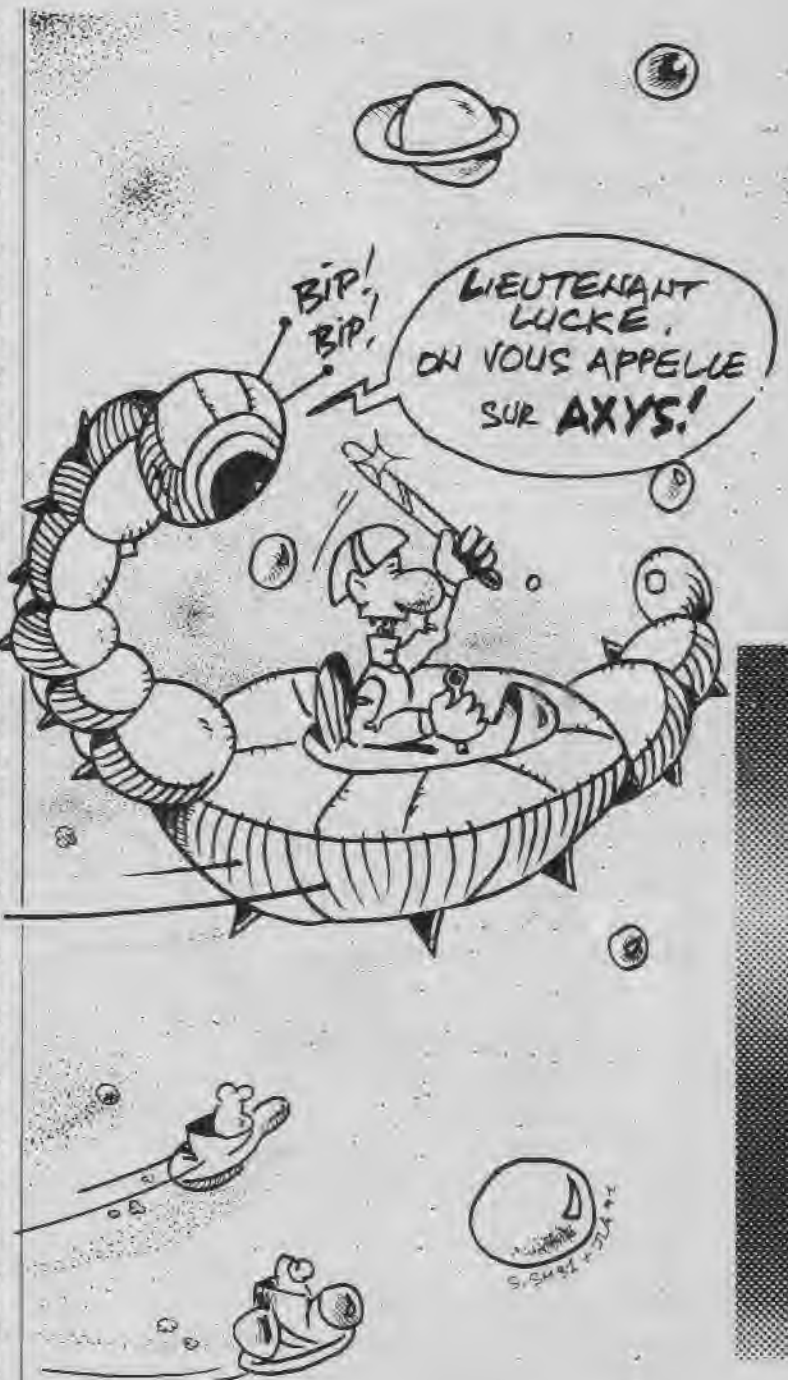


4870 DATA 98, 8C, B0, 8C, 00, 40, 90, CB, C0, 80, 76, 00, 04, 40, C4, 686  
 4880 DATA 64, 8C, 04, 44, 00, 00, 03, 03, 43, C3, C7, CF, C3, CB, CB, 633  
 4890 DATA C3, C3, CB, C3, 03, 42, 83, 39, 33, 3C, 3C, 39, 33, 3C, 36, 59E  
 4900 DATA 3C, 3C, 76, 39, 06, 3C, 36, 3C, 36, 39, 39, FC, 33, 03, 42, 3F7  
 4910 DATA 27, 76, 0F, 04, 4F, 9A, 78, 98, 00, 30, 8C, 8C, 0C, 4C, 8C, 585  
 4920 DATA 4C, CC, 8C, 4C, 48, C0, 0C, C0, C0, 80, 00, 00, C4, C8, 30, 6C0  
 4930 DATA 24, 00, 00, 44, 0C, 80, 00, 00, 00, 40, 00, 00, C4, 18, 60, 2F0  
 4940 DATA 76, 00, 04, 03, 03, C3, 03, 42, 8F, C3, C7, C3, C3, C7, 67, 655  
 4950 DATA C3, C3, 83, 03, 3C, 39, 3C, 03, 42, 39, 3C, 36, 39, 33, 39, 452  
 4960 DATA 03, 42, 33, 39, 36, 3C, 36, 3C, 36, 39, 39, FC, 33, 03, 42, 3B1  
 4970 DATA 27, 1B, 0F, CF, 0F, 4F, CF, CE, 30, 64, CC, 4C, 8C, 0C, 84, 5E3  
 4980 DATA 0C, 4C, C0, C0, 84, 48, C0, 84, 0C, 76, C0, 04, 00, C8, 30, 626  
 4990 DATA CC, 00, 00, 04, 48, 80, C0, 00, 03, 42, 40, 40, C0, 58, 0C, 441  
 5000 DATA 00, 03, 42, 80, 03, 43, C3, C3, C7, 87, CB, C3, 03, 42, C7, 679  
 5010 DATA ED, C3, C3, 83, 03, 3C, 2D, 3C, 1E, 3C, 2D, 3C, 1E, 3C, 3C, 4F7  
 5020 DATA 76, 2D, 06, 3C, 1E, 3C, 1E, 2D, 2D, 7C, 76, 33, 04, 76, 0F, 385  
 5030 DATA 06, 0C, 70, 64, 8C, 0C, 4C, 0C, CC, 8C, 0C, 0C, 48, 76, C0, 4C4  
 5040 DATA 04, 00, C0, 76, 80, 05, CC, 64, 00, 00, 84, 00, C0, C0, 00, 4F3  
 5050 DATA 03, 42, 84, C0, 00, C8, 48, 76, 00, 04, 01, 03, 83, C3, C3, 528  
 5060 DATA 07, CF, C3, 03, 42, C7, CB, C3, C3, 03, 03, 89, C3, 3C, 96, 77A  
 5070 DATA 69, C3, 3C, 96, 3C, 3C, 76, 69, 06, 3C, 96, 69, C3, 89, 69, 62B  
 5080 DATA 39, 33, 27, 33, 27, 0F, 0F, 4F, 0F, CF, 0F, 0E, 18, 64, 4C, 31D  
 5090 DATA 0C, 48, C0, 0C, 0C, C0, 0C, 76, C0, 06, 00, 00, 00, C0, 80, 4F4  
 5100 DATA C0, 44, 18, 00, 00, 84, C0, 03, 42, 00, 03, 42, C0, C0, 00, 46A  
 5110 DATA 04, 00, 00, 03, 42, 01, 03, 03, 42, C3, C3, C7, 0F, CB, 76, 437  
 5120 DATA C3, 06, 43, 03, 76, 3C, 16, 7C, 33, 03, 42, 27, 0F, 0F, 4F, 35F  
 5130 DATA 8F, 4F, 8F, 0F, BE, 8C, CC, 8C, 48, C4, 24, 48, C0, 84, C0, 76A  
 5140 DATA 03, 42, 84, C0, 00, 80, 00, 76, 40, 05, 90, 80, 00, C0, 48, 4DC  
 5150 DATA 40, 80, 00, 00, 48, C0, 48, 00, 04, 76, 00, 05, 03, 03, C3, 350  
 5160 DATA 03, 42, C7, 8F, C7, 76, C3, 05, 83, 43, 83, 76, 3C, 16, FC, 6AD  
 5170 DATA 33, 03, 42, 0F, 03, 42, 4F, 8F, 4F, CF, 0F, 64, 4C, 64, 64, 44F  
 5180 DATA 8C, 84, 0C, 48, 48, 0C, 84, 0C, 48, C0, 80, 76, 00, 05, C0, 50B  
 5190 DATA 04, 48, 04, CC, 48, 48, C8, 80, 80, 00, 00, C0, 40, 0C, 76, 576  
 5200 DATA 00, 07, 03, 03, 43, C3, 03, 42, 8F, 4F, CB, 76, C3, 05, C7, 506  
 5210 DATA 83, 76, 3C, 16, 7C, 33, 03, 42, 1B, 76, 0F, 04, 4F, CF, 8F, 490  
 5220 DATA CE, 24, 98, 30, 64, 30, 48, 84, 48, C0, C0, 84, 48, 84, 48, 67A  
 5230 DATA C0, 76, 00, 05, 48, 0C, 48, 40, 0C, 00, 80, 76, 00, 84, 4C, 369  
 5240 DATA C0, 88, 76, 00, 05, 01, 03, 03, 42, 43, C3, 03, 42, 8F, 0F, 3F5  
 5250 DATA CF, 76, C3, 05, 96, 88, 69, C3, 3C, 3C, 69, C3, 69, C3, 69, 793  
 5260 DATA C3, 69, C3, 69, C3, 69, C3, 69, 96, 3C, 03, 42, 7C, FC, 89, 7F8  
 5270 DATA 33, 1B, 27, 0F, 76, 3C, 06, 00, 00, F3, A2, 76, 00, 88, 7E, 3CD  
 5280 DATA A2, 15, CF, 1C, 3C, 6C, 76, 3C, 05, F0, B4, 3C, 78, 98, 8C, 67D  
 5290 DATA 4C, 30, 24, CC, CC, C0, CC, 8C, 8C, 4C, 4C, 80, 4C, 4C, C8, 6D4  
 5300 DATA 24, 10, F0, F0, 3C, 98, 76, F0, 04, 24, CC, 40, 00, 28, 76, 6A0  
 5310 DATA 76, 01, 33, 32, 30, 31, 76, 76, 01, 33, 33, FC, 03, 42, 3C, 40D  
 5320 DATA 7C, FC, 3C, 3C, 7C, FC, 33, 33, 0F, 4F, C7, C3, 83, 03, 76, 6B2  
 5330 DATA 3C, 04, 39, 00, 03, 42, 15, F3, 00, 51, 3F, 3F, A2, 00, 03, 33A  
 5340 DATA 42, 76, 76, 01, 28, 00, E7, 4F, 0F, CF, 48, 84, 8C, 18, 24, 4FF  
 5350 DATA 4C, 0C, 98, 8C, 18, 0C, C4, CC, 4C, 4C, 8C, 84, 0C, 03, 42, 529  
 5360 DATA 40, C0, 40, 84, 0C, C0, 00, 04, 0C, 40, 0C, 18, 30, C8, 70, 474  
 5370 DATA 3C, 6C, 84, 3C, CC, 3C, A4, 40, 40, C0, C0, 80, 00, 00, 83, 647  
 5380 DATA 42, 19, FC, 76, 76, 01, 76, FC, 05, 33, 27, 0F, 8F, CF, CB, 64D  
 5390 DATA CB, C3, 3C, 03, 42, 73, 76, 00, 00, A2, B7, 76, 00, 04, 51, 524  
 5400 DATA 22, 15, 27, 0F, 03, 42, CF, 03, 42, 8F, 0F, 0F, CF, 48, 0C, 396  
 5410 DATA 03, 42, C0, 03, 42, 00, 40, 80, 00, 80, 00, 40, 80, 40, 84, 40E  
 5420 DATA 0C, 84, 0C, C0, 03, 42, 0C, 03, 42, C0, 76, 00, 04, 0C, 00, 340  
 5430 DATA C0, 03, 42, 76, 00, 00, 45, FC, B9, 33, FC, FC, 76, 33, 04, 655  
 5440 DATA 0F, CF, CF, C3, C3, 03, 3C, 3C, 2A, F3, F3, 76, 00, 0D, 7E, 6BF  
 5450 DATA A8, 00, 2F, 0F, 03, 42, CF, 03, 42, 8F, CF, 8F, CF, 03, 42, 540

5460 DATA CB, C3, C3, CB, C0, 8C, 48, 80, 76, 00, 0F, 84, 0C, CF, CF, 763  
 5470 DATA 4F, 76, 0F, 05, 27, 0F, 1B, 33, 33, 1B, 76, 33, 0D, 0F, 4F, 2BF  
 5480 DATA CF, C7, CB, 83, 76, 00, 11, 11, FC, BD, 00, 11, 33, 33, 0F, 5BB  
 5490 DATA 33, 76, 0F, 06, CF, 4F, CF, CF, C7, CF, 03, 42, 76, C3, 0F, 69D  
 5500 DATA C7, CB, CF, CB, C3, C3, CF, C7, CF, 03, 42, 4F, 8F, 76, 0F, 8BF  
 5510 DATA 05, CF, 76, 0F, 09, 1B, 33, 33, 0F, 0F, CF, CF, 83, 03, 00, 425  
 5520 DATA 00, 51, A2, 51, 76, 00, 0A, A2, 00, 54, 89, 7B, 00, 00, 55, 443  
 5530 DATA 76, 76, 01, 33, 36, 7C, 76, 33, 04, 27, 0F, 03, 42, CF, 0F, 308  
 5540 DATA CF, 4F, CF, 03, 42, C3, C7, CB, C7, 0F, CB, 76, C3, 07, CB, 833  
 5550 DATA CF, 03, 42, 4F, CF, CF, CB, 03, 42, C3, 76, CF, 10, 8F, 0F, 8C7  
 5560 DATA 0F, 1B, 27, 0F, CB, CF, DF, 3C, F3, A2, 76, 00, 0A, 7F, 2A, 5D3  
 5570 DATA 51, 00, 00, BE, AA, 76, 00, 05, 11, FC, 03, 42, 76, 33, 05, 434  
 5580 DATA 89, 76, 0F, 04, 36, 3C, 27, 76, CF, 04, 98, 2D, CF, C3, CF, 64D  
 5590 DATA C3, C3, 5E, 89, 76, 76, 01, 89, 4F, CF, CF, C7, CF, C3, C7, 950  
 5600 DATA CF, C3, C7, CF, CB, 76, CF, 07, C7, CF, CF, C3, CF, C3, CF, AC8  
 5610 DATA CF, 8F, 0F, 0F, 1B, 0F, CF, DF, BE, FF, 7F, FF, BF, BF, 3F, 84C  
 5620 DATA FF, BB, BB, 00, B7, A2, 76, 00, 05, 55, 89, 51, A8, 76, 00, 6C6  
 5630 DATA 07, 77, 3C, FC, 7C, FC, FC, B9, 76, 33, 04, 27, 0F, 0F, BC, 691  
 5640 DATA 39, 0F, 4F, 76, CF, 09, 8F, 0F, 76, CF, 07, CB, C7, CF, C3, 6F3  
 5650 DATA 8F, 4F, C7, CF, CF, C7, 76, CF, 04, CB, C3, 03, 42, CB, CF, 8C0  
 5660 DATA CF, 0F, 03, 42, CF, BE, FC, FF, 7B, 76, 00, 11, 55, 22, 00, 624  
 5670 DATA 00, 55, 76, 76, 01, B9, B7, FF, 76, 76, 01, 76, 00, 8A, B7, 5D5  
 5680 DATA 36, FC, FC, B9, 33, 36, 7C, 39, 76, 0F, 07, 1B, 33, 1B, 27, 521  
 5690 DATA 76, 0F, 06, 76, CF, 04, 0F, 0F, 76, CF, 05, BF, F3, 76, 00, 564  
 5700 DATA 06, 15, 7F, FE, FC, B9, FF, 33, F9, 7F, 7B, 76, 00, 14, F6, 7F2  
 5710 DATA 3C, 7C, FC, 76, 33, 05, 76, 76, 01, FC, B9, 37, 00, 00, 11, 54C  
 5720 DATA FD, FD, FC, FC, 3C, 77, 33, FC, FF, FF, 33, FC, FC, 8C, 7C, 835  
 5730 DATA FC, 89, 33, 33, AA, FF, FF, BB, 33, 77, FF, 33, 33, 77, 33, 837  
 5740 DATA 03, 42, FF, 3C, FC, FF, 76, 33, 04, 77, FF, 03, 42, 33, 03, 619  
 5750 DATA 42, FE, FC, BB, 77, BF, 00, 3F, BF, 76, 00, 05, 33, 77, 33, 683  
 5760 DATA 33, 77, FF, 77, FF, 33, BF, 3F, 3F, 7B, F3, 87, 7F, BB, 33, 821  
 5770 DATA 76, 76, 01, FC, B9, BF, 51, BB, FF, 77, FF, FC, 3F, 7B, FF, 997  
 5780 DATA 33, BB, FF, 76, 33, 05, 3F, B3, 00, 3F, 7F, 33, 03, 42, 37, 4FA  
 5790 DATA FF, 33, BB, 77, FF, B9, 33, 33, FF, FC, 3C, FF, FF, AA, 76, 9D7  
 5800 DATA 00, 09, 15, 3F, BB, 33, FF, FF, BB, 33, 33, 3F, 03, 42, 2A, 518  
 5810 DATA 00, 03, 42, 8F, 2A, 51, 2A, 00, 03, 42, A2, B7, 76, 76, 01, 434  
 5820 DATA 7B, F3, 87, FF, BB, BB, FB, 00, 11, 37, B7, F3, 00, 00, F3, 87A  
 5830 DATA 00, 15, 2A, 76, 00, 00, 83, B9, 33, FF, FF, BB, 33, 03, 42, 58D  
 5840 DATA 77, FF, 76, 76, 01, FC, B9, FF, 3C, FE, FC, FE, FB, 00, F3, A39  
 5850 DATA 7F, FB, FF, BB, 33, 33, 77, FF, FF, 7C, FD, BB, 76, 33, 04, 8F0  
 5860 DATA 77, BB, 33, 33, 77, 2A, 00, 03, 42, FF, 03, 42, BB, 3F, 3B, 4F7  
 5870 DATA 33, 77, BF, F3, 76, 00, 00, 55, 33, 00, 03, 42, 3F, 7B, B7, 518  
 5880 DATA A2, 00, 00, FF, 7B, A2, 76, 00, 04, F7, 3F, 03, 42, 76, 33, 55C  
 5890 DATA 04, 77, FF, BB, 33, 33, A2, 00, 03, 42, F3, B7, 37, BF, 3F, 661  
 5900 DATA 03, 42, 33, 03, 42, 77, BB, 33, 03, 42, 77, FC, 8C, FC, FD, 68F  
 5910 DATA FF, 03, 42, BF, 2A, 00, 03, 42, 15, BB, 76, 33, 05, 77, FF, 586  
 5920 DATA FF, BB, 77, 33, 33, 77, 2A, 00, 51, F3, 00, 3F, 33, 27, CF, 5E4  
 5930 DATA 03, 42, 6D, CB, C3, C7, C3, C3, CB, 96, 88, 43, D6, 3C, 3C, 80A  
 5940 DATA 8C, 30, B0, 48, C0, 03, 42, 01, 76, 03, 07, 83, 76, 03, 07, 43D  
 5950 DATA C7, 8B, 03, 03, C3, 03, 42, 87, C7, C7, CF, CF, 76, 0F, 04, 69C  
 5960 DATA CF, CB, C3, 76, 03, 04, 76, 3C, 11, 7C, 7C, B9, 33, 33, 76, 62A  
 5970 DATA 76, 01, 3C, 7C, CF, 4F, 1B, 33, 27, CF, 03, 42, CB, CB, CF, 63B  
 5980 DATA C3, 03, 42, C7, C3, 83, C3, 84, 3C, 38, 4C, 60, 40, 76, 30, 692  
 5990 DATA 04, 03, 03, 42, 01, 76, 03, 0C, DE, 88, 03, 76, C3, 05, C7, 443  
 6000 DATA CF, CF, 8F, 76, 0F, 04, CF, CB, C3, 76, 03, 04, 76, 3C, 11, 653  
 6010 DATA FC, 03, 42, 33, 33, FC, 3C, B9, CF, 0F, 76, 76, 01, 8C, 27, 646  
 6020 DATA CF, 03, 42, CB, CB, CF, CB, C3, C3, 87, 4F, C3, C6, 3C, 3C, 8A1  
 6030 DATA 6C, 58, 60, 84, 24, CC, 0C, 4C, 02, 03, 00, 76, 03, 0D, 9E, 419  
 6040 DATA E9, 43, 76, C3, 05, C7, CF, CF, 76, 0F, 05, 4F, CB, C3, C3, 7F9



6050 DATA 03,03,42,76,3C,11,FC,BC,FC,33,36,3C,3C,ED,CF,65C  
6060 DATA 0F,76,76,01,3C,67,0F,CF,CF,C3,C3,1B,C3,03,42,675  
6070 DATA 1E,27,C3,96,3C,3C,24,DC,00,C4,88,76,00,04,76,55A  
6080 DATA 03,09,C3,76,03,05,1E,CB,43,76,C3,04,C7,CF,03,54F  
6090 DATA 42,0F,03,42,1B,0F,4F,CB,C3,C3,03,03,42,76,3C,45A  
6100 DATA 10,7C,FC,BC,3C,AD,36,3C,FC,AD,4F,0F,76,76,01,693  
6110 DATA 3C,2D,0F,CF,CF,CB,CB,36,4B,C3,C7,1E,6D,C2,34,78B  
6120 DATA 3C,3C,8C,74,C0,4C,C8,C8,C0,80,02,03,02,00,01,554  
6130 DATA 76,03,05,43,C3,03,42,43,03,93,4B,43,76,C3,04,4ED  
6140 DATA CF,03,42,0F,0F,0F,27,33,0F,0F,CB,C3,C3,83,03,510  
6150 DATA 03,76,3C,0F,7C,FC,FC,3C,3C,2D,76,76,01,3C,33,539  
6160 DATA 4F,CF,CF,36,3C,6D,76,CF,04,CB,BC,CB,C3,C7,BC,8AD  
6170 DATA 69,C3,3C,3C,38,0C,38,C0,18,4B,0C,03,42,09,03,39D  
6180 DATA 02,01,01,76,03,05,43,C7,CF,C3,C7,C3,C7,CB,76,6B0  
6190 DATA C3,04,CB,CF,03,42,0F,0F,27,33,27,33,0F,CF,C3,599  
6200 DATA C3,83,03,03,76,3C,11,7C,3C,3C,27,76,76,01,FC,513  
6210 DATA 33,0F,0F,4F,1E,3C,67,76,CF,04,C3,FC,C3,C7,C3,8B6  
6220 DATA 0C,63,00,3C,3C,EC,5C,AC,04,CC,C0,00,00,84,02,679  
6230 DATA 03,02,02,76,03,07,C3,0F,4F,1B,4B,C7,76,C3,05,493  
6240 DATA CB,CF,03,42,0F,0F,1B,33,27,33,0F,CF,C3,C3,83,58C  
6250 DATA 03,03,76,3C,0F,7C,BC,FC,3C,BC,27,5E,39,33,27,50B  
6260 DATA 4F,CB,9E,3C,4B,76,CF,04,C3,0F,C3,C7,CB,D6,C2,8C7  
6270 DATA 96,3C,3C,24,1C,AA,C4,CC,C0,80,40,84,03,02,02,58D  
6280 DATA 03,01,76,03,05,01,03,C7,0F,1E,ED,76,C3,05,CB,470  
6290 DATA CF,CF,0F,0F,03,42,76,33,04,0F,CF,C3,C3,83,03,618  
6300 DATA 03,76,3C,13,FC,0F,1B,27,36,39,CF,CF,9B,7C,76,5AF  
6310 DATA CF,05,CB,CB,C3,03,42,C7,C0,9C,3C,3C,0C,3C,60,685  
6320 DATA 84,0C,84,80,40,0C,09,76,03,0B,43,0F,36,6D,76,4D0  
6330 DATA C3,04,76,CF,04,0F,03,42,1B,33,03,42,CF,0F,CF,4AA  
6340 DATA CB,C3,C3,03,03,76,3C,0F,FC,37,83,BC,33,27,0F,623  
6350 DATA 0F,76,76,01,7C,CF,03,42,4F,76,CF,07,C7,C3,03,584  
6360 DATA 42,84,EA,84,78,0C,7C,60,84,8C,C0,80,00,76,3C,6EC  
6370 DATA 06,00,00,F3,A2,00,03,42,3F,A2,00,03,42,FE,00,404  
6380 DATA 15,9F,3C,3C,F0,3C,78,3C,3C,38,78,3C,78,98,CC,608  
6390 DATA 24,4C,98,CC,03,42,84,CC,8C,8C,CC,8C,00,0C,0C,5F1  
6400 DATA 84,4B,84,70,F0,84,60,8C,0C,0C,84,C8,00,C0,4B,68C  
6410 DATA 60,80,4B,00,00,04,31,33,33,89,76,76,01,FC,8C,551  
6420 DATA BC,FC,FC,3C,FC,33,03,42,0F,4F,C3,C3,83,03,76,744  
6430 DATA 3C,04,AA,76,00,05,7F,51,3F,BF,A2,00,03,42,76,490  
6440 DATA 76,01,2B,00,AF,4F,0F,CF,8E,0C,C0,C0,1B,8C,24,5DD  
6450 DATA CC,4B,84,84,C4,CC,4C,C4,CB,C4,CC,84,8C,C0,80,964  
6460 DATA 04,84,0C,84,80,04,4C,00,C0,98,70,64,4C,70,F0,5C0  
6470 DATA 3C,7B,1C,3C,6C,C0,4B,C0,76,00,06,11,FC,33,76,572  
6480 DATA FC,05,33,27,0F,4F,CF,CF,CB,83,3C,03,42,22,76,5BE  
6490 DATA 00,0B,FF,FF,AA,00,03,42,11,AB,15,27,0F,76,CF,53E  
6500 DATA 06,4F,CF,CF,4B,84,84,4B,C0,C0,80,C0,C0,4B,80,7CB  
6510 DATA 00,80,00,00,40,4B,4B,40,C0,4B,76,C0,04,76,00,448  
6520 DATA 06,40,76,C0,05,76,00,07,04,76,76,01,33,76,76,40E  
6530 DATA 01,FC,FC,33,03,42,27,4F,CF,CF,C3,03,03,3C,7D,607  
6540 DATA 51,F3,F3,76,00,00,42,76,00,04,55,22,00,A7,76,565  
6550 DATA 0F,04,CF,0F,03,42,CF,4F,CF,03,42,C3,C3,CF,4B,605  
6560 DATA C0,C0,00,03,42,80,76,00,09,40,84,C3,C3,C7,CF,6AA  
6570 DATA 03,42,0F,76,0F,09,1B,27,27,33,27,0F,76,33,00,2E5  
6580 DATA 0F,4F,CF,CB,CB,83,76,00,11,54,7C,BD,00,55,33,5E2  
6590 DATA 03,42,89,27,76,0F,06,0F,4F,76,CF,06,CB,76,C3,5DD  
6600 DATA 0D,76,CF,04,C3,C7,76,CF,07,0F,76,0F,0F,33,33,5B5  
6610 DATA 1B,0F,CF,CF,C3,03,00,00,51,A2,76,00,0A,51,A2,4F4  
6620 DATA 00,54,BD,00,03,42,55,76,76,01,89,36,7C,33,33,469  
6630 DATA 1B,33,33,0F,03,42,CF,0F,0F,4F,76,CF,05,C3,C3,561



6640 DATA CF,0F,CB,76,C3,05,CB,C3,CF,0F,0F,76,CF,07,7BD  
6650 DATA C7,76,CF,0E,0F,03,42,1B,27,0F,4B,C7,D7,7C,76,59A  
6660 DATA 00,0C,BF,00,51,00,00,36,22,76,00,05,11,76,FC,372  
6670 DATA 04,33,33,89,33,33,0F,0F,1B,0F,76,76,01,3C,27,321  
6680 DATA 4F,CF,03,42,DE,6D,C7,C3,CB,C7,0F,3C,89,33,0F,790  
6690 DATA 76,CF,04,76,C3,04,CB,C7,C3,C3,76,CF,05,CB,CB,87E  
6700 DATA 0F,4F,CB,C3,CB,C7,CF,CF,0F,0F,0F,1B,0F,CF,BE,800  
6710 DATA FE,FD,3F,3F,BF,3F,A2,76,00,04,A2,00,F3,76,00,69E  
6720 DATA 04,11,FD,00,3F,B3,39,76,00,05,F7,33,3C,7C,03,49D  
6730 DATA 42,FC,76,76,01,76,33,04,0F,0F,76,76,01,3C,33,452  
6740 DATA 0F,4F,76,CF,00,0F,0F,0F,76,CF,06,C7,CF,CF,C7,6CF  
6750 DATA 1B,0F,CF,03,42,C7,76,CF,05,CB,00,00,00,00,00,41A

# DATAS 2C. BAS



```

10 ' DATAS2C AXYS
20 '
30 MODE 2:AD=&C000:NL=90
40 FOR A=1 TO 666: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"INTROC",B,&C000,&26F9:CLS:END
80 CLS:PRINT"LIGNE"NL;" INCORRECTE":END
90 DATA C3,C7,CF,03,42,0F,0F,4F,CF,BE,FD,FF,A2,76,00,7AC
100 DATA 12,55,AA,00,55,33,FC,00,51,7F,B9,BB,76,00,09,558
110 DATA 51,76,76,01,FC,FC,09,76,76,01,3C,FC,27,27,33,695
120 DATA 33,27,33,0F,33,03,42,1B,33,33,76,0F,06,4F,0F,27E
130 DATA 0F,0F,0F,CF,DF,FB,A2,76,00,09,3F,F6,3C,FE,33,719
140 DATA 8B,73,3F,7B,76,00,0B,B7,FF,7B,11,FC,FC,3C,FC,7DB
150 DATA 89,76,FC,04,89,77,0F,8B,33,33,FC,33,33,A2,51,794
160 DATA 33,FD,FF,FF,8B,FC,8B,76,76,01,89,FD,8B,33,33,964
170 DATA FC,3C,89,33,89,33,03,42,76,76,01,FC,3C,3C,BE,674
180 DATA 7C,FD,33,03,42,8B,FE,FC,03,42,FF,FF,8B,33,33,80A
190 DATA 76,76,01,FD,76,FC,04,FE,FD,FF,33,03,42,22,15,709
200 DATA 7B,A2,00,03,42,15,33,8B,33,8B,33,FF,FF,33,8F,676
210 DATA F3,51,87,2A,51,15,3B,76,33,04,77,00,00,83,77,514
220 DATA FF,76,76,01,FC,03,42,33,76,76,01,FF,8B,73,3F,689
230 DATA 3F,76,33,04,37,FF,76,33,04,0F,FF,76,33,04,8B,5F5
240 DATA 8B,33,FF,FE,FC,3C,3C,FF,0F,3F,76,00,06,B7,55,7E4
250 DATA 76,33,04,8B,8B,33,3F,03,42,FF,F3,A2,00,00,A2,610
260 DATA 76,00,06,2A,15,33,FB,F3,07,FF,03,42,2A,00,83,5B4
270 DATA 0F,F3,A2,76,00,04,87,76,00,09,11,33,77,FE,77,634
280 DATA FF,FF,33,33,77,7F,33,FE,89,76,76,01,8E,FC,FD,6E8
290 DATA FF,00,03,42,15,2A,F3,FF,33,03,42,FC,03,42,76,5A4
300 DATA 33,05,37,76,33,05,FD,3F,00,55,FF,03,42,33,7F,4AA
310 DATA 77,77,33,37,F3,76,00,0B,15,77,00,00,51,8B,FF,560
320 DATA 8B,00,00,55,37,A2,76,00,05,15,76,3F,04,33,03,3A8
330 DATA 42,77,FF,33,03,42,76,00,06,11,77,0F,3F,FF,FF,630
340 DATA 8B,33,03,42,77,76,33,04,FF,FC,3C,FF,3F,F3,7B,73A
350 DATA 76,00,05,15,7F,FF,FF,8B,33,77,FF,3F,FF,33,33,715
360 DATA 77,33,37,7B,00,15,7B,00,30,30,F0,03,42,3C,03,3C0
370 DATA 42,38,C0,C3,CF,C7,CF,0F,03,42,1B,0F,76,33,06,58F
380 DATA 76,76,01,76,FC,05,89,33,33,0F,4F,CF,CB,03,03,601
390 DATA 76,3C,06,7C,3C,2A,00,03,42,F3,3F,A2,00,03,42,3F8
400 DATA 8B,89,0F,0F,4F,4F,8E,90,76,3C,04,C0,0C,58,80,5D3
410 DATA CC,FC,F0,4C,3C,78,84,3C,3C,60,98,8C,0C,30,24,6C8
420 DATA D8,F0,03,42,30,64,CC,0C,0C,98,F0,3C,E1,41,CF,73A
430 DATA 03,42,0F,0F,0F,76,33,06,89,76,FC,06,33,33,0F,449
440 DATA 4F,CF,C3,83,03,76,3C,06,7C,BC,76,00,04,F3,7B,63F
450 DATA A2,00,03,42,86,89,0F,0F,CF,CF,4F,58,DA,84,3C,683
460 DATA 68,84,48,D8,6C,4C,BC,2C,1B,3C,3C,78,3C,3C,38,564
470 DATA C4,48,24,CC,8C,78,80,CC,CC,4C,8C,CC,98,30,8C,838
480 DATA 48,0C,8C,C0,CF,03,42,0F,03,42,76,33,88,76,76,4A5
490 DATA 01,76,FC,05,89,33,33,0F,4F,CB,C3,83,03,76,3C,58B

```

```

500 DATA 07,37,76,00,04,87,F3,A2,00,03,42,3E,77,0F,0F,41C
510 DATA 4F,4F,CF,0E,3C,E5,3C,68,4C,4C,84,3C,03,42,64,5F1
520 DATA 64,70,3C,3C,84,3C,3C,60,84,30,8C,CC,98,70,F0,68C
530 DATA 30,98,CC,4C,76,CC,04,30,60,00,CF,CF,0F,0F,1B,60D
540 DATA 1B,76,33,04,89,76,76,02,89,76,FC,07,33,33,4F,556
550 DATA 4F,CB,C3,83,03,76,3C,06,7C,AA,00,F3,51,00,F3,678
560 DATA F3,A2,00,03,42,3E,77,0F,0F,CF,4F,CF,CF,4D,3C,5F2
570 DATA 65,48,8C,D8,3C,00,3C,78,20,44,98,34,3C,78,3C,5D1
580 DATA 78,F0,0C,58,6C,3C,B4,F0,E0,98,30,64,CC,4C,CC,808
590 DATA 98,70,80,8C,8C,4D,CF,8F,0F,1B,1B,76,33,04,FC,669
600 DATA 03,42,33,76,FC,07,33,33,4F,4F,CB,C3,C3,03,76,58F
610 DATA 3C,07,00,F3,F3,15,00,87,F3,A2,00,03,42,3E,39,546
620 DATA 0F,0F,4F,CF,03,42,8E,DA,78,4C,58,76,3C,04,F4,62F
630 DATA 8C,C0,58,78,84,3C,F0,34,3C,C0,90,D8,3C,24,98,794
640 DATA 8C,8C,48,C4,CC,CC,8C,48,C0,D0,38,3C,64,8F,0F,786
650 DATA 0F,76,33,06,76,76,01,76,FC,07,7C,FC,FC,33,33,5FE
660 DATA 4F,CF,CF,C3,83,03,76,3C,06,7D,00,F3,F3,15,00,666
670 DATA 87,F3,A2,00,03,42,8E,33,CF,4F,CF,4F,CF,8F,8F,7AB
680 DATA 0D,80,8E,76,3C,04,38,9C,30,8C,10,78,F0,3C,18,55D
690 DATA 3C,3C,E0,C4,84,78,6C,F0,F0,70,D8,98,30,30,804
700 DATA F0,30,0C,64,1C,E0,8F,0F,0F,76,33,04,76,76,01,4D3
710 DATA 89,76,76,01,FC,FC,89,FC,03,42,3C,FC,03,42,33,748
720 DATA 33,0F,CF,CF,C3,03,83,76,3C,05,7C,79,00,F3,51,619
730 DATA 15,00,F3,F3,76,00,04,8E,33,0F,4F,CF,0F,CF,CF,640
740 DATA 0F,8E,C0,24,34,7C,7C,3C,03,42,84,E4,08,2C,80,62A
750 DATA 80,38,3C,3C,68,C0,F0,F0,E4,98,30,4C,98,24,38,74C
760 DATA 64,CC,03,42,0C,58,98,68,8D,0F,1B,33,33,76,76,4E2
770 DATA 02,FC,76,76,01,76,FC,07,7C,7C,FC,FC,33,33,27,6E1
780 DATA 4F,CF,C3,43,83,76,3C,06,28,51,F3,51,15,76,00,5A7
790 DATA 07,7E,77,0F,4F,CF,0F,CF,03,42,0F,4F,9A,1C,3C,51C
800 DATA 84,3C,84,3C,3C,84,48,38,64,1C,78,70,84,38,C0,614
810 DATA 98,F0,80,70,F0,E4,48,04,0C,0C,CC,8C,4C,8C,70,780
820 DATA E4,78,4C,0F,1B,33,03,42,76,76,01,33,76,76,01,457
830 DATA 76,FC,06,3C,76,FC,04,33,33,27,4F,CF,C3,C3,83,6DE
840 DATA 76,3C,05,7C,28,51,F3,A2,15,76,00,07,11,22,0F,415
850 DATA CF,CF,4F,CF,03,42,1B,1A,F0,3C,03,42,38,3C,03,51E
860 DATA 42,78,30,3C,80,84,78,70,3C,78,C0,18,48,C0,D0,6D6
870 DATA F0,A4,4C,CC,CC,0C,03,42,C0,00,64,64,F0,C4,0F,714
880 DATA 76,33,04,76,FC,06,76,3C,05,39,00,00,F3,51,FF,558
890 DATA 00,87,3F,A2,00,03,42,76,76,01,00,55,9F,3C,3C,436
900 DATA 84,3C,03,42,78,70,3C,78,CC,E4,64,8C,18,30,4C,605
910 DATA CC,8C,84,CC,CB,8C,0C,48,40,84,0C,84,D8,78,64,750
920 DATA D8,E0,00,00,80,00,00,40,40,84,C0,00,00,40,18,454
930 DATA E4,80,00,04,98,33,33,FC,BC,BC,FC,FC,3C,89,33,7FA
940 DATA 03,42,0F,4F,CB,C3,C3,83,76,3C,04,AA,76,00,05,552
950 DATA F7,51,3F,8F,A2,00,03,42,FE,28,00,2F,4F,8F,CF,62F
960 DATA 8E,0C,0C,84,64,98,CB,48,C0,03,42,00,04,4C,4C,557
970 DATA C8,C4,4C,84,CB,C0,80,04,0C,48,C0,80,48,4C,48,6CB
980 DATA 84,00,00,84,98,68,D8,3C,2C,34,F0,60,08,C0,00,584
990 DATA 00,40,76,00,04,40,FC,89,76,76,02,FC,03,42,33,511
1000 DATA 27,0F,4F,CF,CB,CF,83,3C,3C,39,A2,76,00,07,51,592
1010 DATA FF,FF,AA,00,03,42,3E,AB,55,0F,03,42,CF,4F,CF,669
1020 DATA 4F,CF,CF,4F,CF,CF,48,84,0C,C0,80,40,C0,80,00,772
1030 DATA 40,40,76,00,12,40,00,40,80,00,C0,C0,00,03,42,3C0
1040 DATA 80,C0,84,40,40,FC,33,76,76,01,FC,33,FC,89,33,77F
1050 DATA 27,4F,CF,CB,83,83,03,3C,79,51,A2,76,00,09,A2,5E2
1060 DATA 76,00,04,15,22,00,F6,0F,03,42,8F,8F,CF,4F,8F,4C6
1070 DATA 4F,4F,CF,03,42,CB,C3,C7,CF,CB,00,00,00,C0,76,757
1080 DATA 00,84,C0,C0,76,C3,07,CF,C7,C7,CF,CF,8F,CF,8F,8AC

```



1090 DATA 76,0F,00,27,1B,0F,27,1B,1B,33,27,1B,76,33,04,25D  
1100 DATA 27,0F,CB,C3,CB,83,76,00,0F,3E,7C,33,FC,F9,00,679  
1110 DATA 55,33,03,42,7C,27,27,0F,03,42,CF,0F,0F,76,CF,49D  
1120 DATA 05,C3,CB,76,C3,0F,C7,CF,C3,C7,C3,C7,76,CF,0A,8D4  
1130 DATA 4F,4F,CF,76,0F,09,1B,33,1B,0F,CF,CF,C3,03,76,54D  
1140 DATA 00,11,54,AB,00,00,51,00,3B,89,36,39,76,33,05,36F  
1150 DATA 0F,27,0F,0F,1B,89,0F,4F,76,CF,04,CB,C3,C7,89,5DD  
1160 DATA 76,C3,05,C7,CB,CF,CF,1B,27,4F,C3,CF,C3,76,CF,899  
1170 DATA 13,1B,0F,0F,1B,33,0F,4F,C7,3C,7D,76,00,07,3F,334  
1180 DATA 0F,00,00,87,77,A2,00,03,42,FE,22,76,00,05,51,4C0  
1190 DATA FC,BC,FC,FC,76,76,01,FC,FC,33,33,1B,0F,1B,0F,74F  
1200 DATA 1B,BC,3B,0F,4F,CF,CF,8F,67,C3,C7,CB,C3,CF,36,81F  
1210 DATA 39,27,76,CF,06,CB,CB,76,C3,05,76,CF,07,9B,67,6CD  
1220 DATA CF,C3,CF,C3,CF,03,42,76,0F,04,CF,3C,FE,FD,00,7C7  
1230 DATA 87,3F,76,00,00,7B,76,00,04,55,73,00,06,FC,BC,5DF  
1240 DATA 22,76,00,06,3B,FC,3C,FC,03,42,76,33,04,0F,0F,41D  
1250 DATA 1B,FC,89,0F,0F,76,CF,00,0F,03,42,76,CF,08,C3,59F  
1260 DATA C7,1B,27,C7,76,CF,06,C7,C7,76,C3,04,CF,CF,0F,793  
1270 DATA 0F,4F,CF,3C,FD,FF,76,00,13,51,22,00,15,33,FC,5A5  
1280 DATA 2A,51,8B,33,33,77,A2,76,00,09,76,76,01,FC,FC,619  
1290 DATA BC,3C,FC,77,FB,F3,76,00,04,51,7F,FF,F3,A2,00,837  
1300 DATA F3,7F,8B,76,0F,04,4F,0F,5F,A2,76,00,05,15,3F,564  
1310 DATA 8B,33,77,3F,00,00,51,FE,FF,FC,33,FF,33,A2,3F,734  
1320 DATA 76,00,0A,36,FC,BC,3C,7C,FC,03,42,33,76,76,01,587  
1330 DATA FC,FC,76,33,04,BF,3B,3F,33,03,42,76,76,01,37,57A  
1340 DATA 7E,FC,FF,33,89,33,FD,8B,33,77,FF,33,8B,76,33,890  
1350 DATA 04,89,76,33,04,76,76,01,BE,3C,FC,FC,FD,8B,77,778  
1360 DATA 33,03,42,7F,FF,FF,FD,FC,FD,8B,76,76,01,FF,33,8C5  
1370 DATA FF,FF,BE,3C,3C,FC,33,33,77,FF,FF,33,00,15,F3,846  
1380 DATA 51,33,8B,77,FF,03,42,BF,FF,03,42,8B,2A,00,51,633  
1390 DATA 87,F3,55,33,77,7B,83,33,33,2A,00,00,11,76,76,564  
1400 DATA 01,3C,BC,3C,7D,77,76,33,04,A2,00,00,55,FF,FC,5C8  
1410 DATA 33,33,8B,33,77,8B,33,03,42,77,77,76,33,04,77,510  
1420 DATA 33,FF,FF,FE,FC,FD,FF,FF,8F,7B,51,F3,3F,FF,FF,BE1  
1430 DATA 33,03,42,FF,8B,33,03,42,77,8F,3F,33,8F,A2,76,829  
1440 DATA 00,09,FF,7F,3F,37,00,87,3F,7F,FB,76,00,00,51,53C  
1450 DATA 7B,33,FC,00,73,A2,00,87,F3,00,00,11,33,FF,33,5DF  
1460 DATA FF,8B,33,33,FF,BF,7F,8B,FF,FC,FF,FE,FC,FF,76,881  
1470 DATA 00,07,51,FF,03,42,BC,FE,FD,33,33,77,8F,3B,3F,669  
1480 DATA FF,8B,76,33,04,77,3F,7F,33,03,42,37,3B,33,77,530  
1490 DATA 33,77,7B,76,00,1C,76,3F,05,3B,76,33,04,7B,76,44A  
1500 DATA 00,07,33,77,FF,8B,76,33,09,FF,FF,FC,BC,79,A2,7EE  
1510 DATA 76,00,09,3F,7F,FF,03,42,3F,03,42,FF,FF,33,FF,635  
1520 DATA 33,FF,2A,00,03,42,54,10,AE,A2,82,8B,DB,99,83,6E9  
1530 DATA 8A,82,8B,82,8D,8C,81,8F,83,87,C5,F0,F9,83,89,9F6  
1540 DATA 72,30,3B,8B,53,D6,CF,CD,D7,D3,A2,A2,83,83,54,905  
1550 DATA 58,02,02,89,89,12,9F,EE,93,53,56,8B,83,C1,D0,748  
1560 DATA 10,31,39,80,83,A3,52,FF,DA,FA,92,8F,92,A6,93,8C1  
1570 DATA F3,86,8F,8B,89,FB,FB,A3,A2,C3,C2,AB,E3,E2,E2,BEB  
1580 DATA A0,89,5B,10,4E,09,5F,41,46,00,0C,31,EA,94,4E,50A  
1590 DATA 4F,5F,41,46,31,00,1F,4E,14,4C,79,47,4F,4F,6B,3FC  
1600 DATA 00,21,8D,95,49,4E,00,1F,0C,31,00,AF,0F,25,4F,368  
1610 DATA 85,44,20,00,1F,0D,31,2C,15,49,4E,49,54,5C,4C,343  
1620 DATA 55,53,39,95,50,41,4C,31,00,9F,0C,31,3B,95,50,400  
1630 DATA 41,4C,32,00,31,00,9F,4B,95,41,53,43,49,49,00,308  
1640 DATA 0C,31,5B,95,54,45,50,54,00,31,00,9F,97,95,00,46E  
1650 DATA 9F,0C,01,00,1F,0C,01,00,1D,1C,31,00,1D,4C,31,1DC  
1660 DATA 00,0F,0C,01,00,0F,9C,9C,80,8B,3C,31,80,8B,0C,4B2  
1670 DATA 31,00,9F,0C,31,00,9F,0C,31,00,1F,0C,31,00,1F,254

1680 DATA 0C,31,10,BF,1C,31,10,BF,1C,31,80,BF,0C,31,30,451  
1690 DATA 3F,1C,31,10,BF,1C,31,10,BF,1C,81,10,3F,8D,31,3D1  
1700 DATA 10,1F,1C,31,00,9F,0C,30,00,9F,0C,31,00,9F,0C,2DE  
1710 DATA 31,00,9F,0C,31,00,9F,0C,31,00,9F,0C,31,00,9F,364  
1720 DATA 0C,31,00,9F,0C,31,00,9F,0C,31,00,9F,0C,31,00,2D9  
1730 DATA 9F,0C,31,00,9F,0C,31,00,9F,0C,81,80,9F,8C,81,590  
1740 DATA 00,9D,8C,85,80,8D,7C,71,80,8F,8C,81,80,8F,9F,8B2  
1750 DATA DC,FF,FF,28,33,10,89,28,81,20,AB,2C,E0,00,9F,74A  
1760 DATA 2C,E0,00,99,28,81,20,89,0C,31,00,8E,2C,E0,20,54E  
1770 DATA EE,2C,E0,00,19,28,31,00,19,28,31,00,8E,2C,E0,3F8  
1780 DATA 00,9F,2C,E0,20,89,08,39,20,89,28,81,00,8C,0C,50F  
1790 DATA 80,00,8E,0C,80,10,19,28,81,00,19,28,31,00,8E,31C  
1800 DATA 0C,80,00,9F,AB,E0,30,89,0C,31,02,02,28,81,10,4C8  
1810 DATA 9F,0C,31,10,9F,00,00,00,1D,1B,31,10,19,28,81,2FB  
1820 DATA 00,8E,08,00,10,8C,0C,00,00,1F,1C,31,00,39,59,23C  
1830 DATA 79,20,E8,08,C0,20,EA,AB,E2,FF,FF,3A,3B,82,F9,8F3  
1840 DATA 8B,89,82,AB,BA,A1,10,9F,8E,A1,80,8D,BA,81,80,98D  
1850 DATA 89,18,31,10,3F,BA,A1,82,AD,BA,E1,10,39,3A,31,65A  
1860 DATA 10,39,3B,31,30,3F,BA,A1,12,9F,BA,A1,82,89,18,60B  
1870 DATA 39,80,89,BA,81,92,8D,9C,81,92,8F,9E,81,80,89,8F2  
1880 DATA BA,F1,10,39,3A,79,12,1F,8A,C1,92,9F,AA,E3,82,793  
1890 DATA F9,0C,31,02,02,8B,81,12,9F,0C,31,12,9F,9A,C3,59F  
1900 DATA 80,9D,1A,31,10,39,AA,81,12,3F,1A,33,12,3F,3A,435  
1910 DATA 33,30,3F,8E,85,80,89,FB,F9,82,E9,9A,81,82,E9,9C3  
1920 DATA A2,E2,A0,8F,0C,11,00,1F,0C,31,00,9F,0C,31,00,438  
1930 DATA 9F,0C,31,00,1F,0C,31,00,1F,0C,31,00,9F,0C,81,240  
1940 DATA 00,9F,0C,81,80,9F,0C,11,00,1F,0C,11,00,8F,0C,2BF  
1950 DATA 01,00,8F,0C,00,00,1F,0D,31,00,1F,0C,31,00,9F,1F4  
1960 DATA 0C,30,00,9F,0C,31,00,9F,0C,31,00,9F,0C,31,00,2D0  
1970 DATA 9F,0C,31,00,9F,0C,31,00,9F,0C,31,00,9F,0C,31,370  
1980 DATA 00,9F,0C,31,00,9F,0C,31,00,9F,0C,31,00,9F,0C,33F  
1990 DATA 31,00,9F,0C,00,00,1F,0C,00,00,1D,1C,31,00,1D,18E  
2000 DATA 4C,71,00,8E,0C,00,00,8E,1C,8C,8B,8D,38,31,30,413  
2010 DATA 3F,0C,31,00,BF,0C,31,00,9F,0C,31,00,1F,0C,31,280  
2020 DATA 00,1F,0C,31,10,BF,1C,31,10,BF,1C,31,80,9F,0C,3BF  
2030 DATA 31,00,1F,0C,31,10,BF,1C,11,10,BF,1C,31,10,1F,2D4  
2040 DATA 1D,31,10,1F,0C,31,00,9F,0C,30,00,9F,0C,31,00,271  
2050 DATA 9F,0C,31,00,9F,0C,31,00,9F,0C,31,00,9F,0C,31,370  
2060 DATA 00,9F,0C,31,00,9F,0C,31,00,9F,0C,31,00,9F,0C,33F  
2070 DATA 31,08,9F,0C,31,00,9F,0C,31,00,9F,8E,91,82,9F,4D0  
2080 DATA 8E,91,80,9D,9C,35,90,3D,5C,71,00,0F,2C,81,00,4E3  
2090 DATA 0F,9F,8D,C3,1C,80,C3,4F,80,C3,3E,80,C3,8D,80,72D  
2100 DATA 00,DD,21,C7,81,DD,7E,09,76,00,04,32,0C,80,C9,5AB  
2110 DATA FE,03,D0,87,C8,11,FF,FF,FE,02,20,8A,DD,56,01,78D  
2120 DATA DD,5E,00,DD,23,DD,23,DD,46,01,DD,4E,00,CD,AD,704  
2130 DATA 80,C3,3E,80,21,38,82,11,29,82,01,09,00,ED,80,53A  
2140 DATA 21,29,82,C3,E3,BC,21,29,82,CD,E6,BC,8E,3F,3E,6F4  
2150 DATA 07,C3,43,85,DD,E5,FD,E5,E5,D5,F5,C5,FD,21,26,9EE  
2160 DATA 82,DD,21,C7,81,FD,7E,00,87,C4,3F,82,FD,7E,01,7FB  
2170 DATA DD,21,D7,81,87,C4,3F,82,DD,21,E7,81,FD,7E,02,875  
2180 DATA 87,C4,3F,82,FD,7E,00,FD,86,01,FD,86,02,CC,4F,83B  
2190 DATA 80,3A,3E,82,3D,20,06,CD,68,83,3A,3D,82,32,3E,4FE  
2200 DATA 82,C1,F1,D1,E1,FD,E1,DD,E1,C9,21,41,81,C5,D5,AC8  
2210 DATA 11,C7,81,01,10,00,3E,01,E5,C5,ED,80,C1,E1,32,6C4  
2220 DATA D2,81,E5,C5,11,D7,81,ED,80,C1,E1,3C,32,E2,81,976  
2230 DATA 11,E7,81,ED,80,3C,32,F2,81,3E,01,32,3E,82,FD,725  
2240 DATA 21,26,82,FD,77,00,FD,77,01,FD,77,02,D1,C1,2A,6E4  
2250 DATA 20,82,DD,21,C7,81,DD,73,09,DD,72,8A,D5,C5,CD,801  
2260 DATA 22,81,C1,D1,C5,D5,DD,21,D7,81,2A,22,82,DD,73,843

227# DATA #9, DD, 72, #A, CD, 22, 81, D1, C1, DD, 21, E7, 81, 2A, 24, 71#  
228# DATA 82, DD, 73, #9, DD, 72, #A, 78, B1, 28, 14, C5, 7E, CB, 7F, 72#  
229# DATA 28, #5, CD, 51, #1, 18, F6, 7E, 23, B7, 2#, FB, C1, #B, 18, 631  
230# DATA E8, DD, 75, #7, DD, 74, #8, C9, #B, #B, #1, #B, #1, #3, #1, 469  
231# DATA 76, #B, #5, #1, #1, #B, #B, E6, #7, 3D, 2#, #D, 7E, 23, CB, 34#  
232# DATA 5F, 3E, #B, 28, #1, 3C, DD, 77, #2, C9, 3D, 2#, 12, 23, 7E, 431  
233# DATA DD, 77, #B, DD, 77, #E, 23, 7E, DD, 77, #1, DD, 77, #F, 23, 632  
234# DATA C9, 3D, 2#, #7, 23, 7E, 23, DD, 77, #D, C9, 3D, 2#, #2, 23, 49D  
235# DATA C9, 3D, 2#, 17, 23, 7E, E6, #F, 3C, DD, 77, #3, 7E, CB, 3F, 5EE  
236# DATA CB, 3F, CB, 3F, CB, 3F, 3C, DD, 77, #4, 23, C9, 3D, 2#, #E, 6#9  
237# DATA 7E, E6, 3F, CB, 3F, CB, 3F, CB, 3F, DD, 77, #5, 23, C9, 3D, 743  
238# DATA 23, C#7, 7E, 23, E5, DD, E5, DD, 46, #B, E6, #F, 4F, CD, 2A, 794  
239# DATA 85, DD, E1, E1, C9, 76, #B, #A, #1, 76, #B, #F, #2, 76, #B, 56B  
240# DATA #F, #3, 76, #B, #5, #B, #B, 76, #B, #A, 1#9, #2, 76, #B, 1AE  
241# DATA #9, 2#, #A, #4, 76, #B, #E, CF, #7, D6, #A, C3, 8E, 76, #B, 53#  
242# DATA 12, #1, 5C, #B, #B, #B, #6, #B, 22, 1E, #2, DD, 7E, #C, 3D, 3DB  
243# DATA 28, #4, DD, 77, #C, C9, DD, 7E, #D, DD, 77, #C, DD, 7E, #6, 67E  
244# DATA 3D, 28, #4, DD, 77, #6, C9, DD, 66, #B, DD, 6E, #7, 7E, B7, 65E  
245# DATA 2#, 28, 23, DD, 56, #E, DD, 5E, #F, DD, 72, #B, DD, 73, #1, 596  
246# DATA DD, 46, #A, DD, 4E, #9, #B, DD, 71, #9, DD, 7#, #A, 78, B1, 643  
247# DATA 28, #4, 7E, #7, 2#, #6, DD, 46, #B, C3, 33, 85, FE, #B, 3#, 566  
248# DATA #6, CD, 51, #1, 7E, 18, F6, 11, #9, #2, 7E, E6, #F, #3, 5F, 602  
249# DATA 3#, #1, 14, 1A, DD, 77, #6, 7E, CB, 6F, 2#, 1D, 23, DD, 75, 52B  
250# DATA #7, DD, 74, #B, DD, 46, #B, C3, #D, 85, #3, #4, #9, #6, #C, 4#5  
251# DATA 18, #1, #1, #3, #4, #9, #9, 12, 24, #1, #1, 23, CB, 77, 2#, 1F#  
252# DATA 17, 7E, E5, DD, E5, DD, 46, #B, DD, 4E, #3, CB, 77, 2#, #3, 7#5  
253# DATA DD, 4E, #4, CD, DB, #4, DD, E1, E1, 7E, 23, DD, 75, #7, DD, #D1  
254# DATA 74, #B, E6, 3F, CB, 3F, CB, 3F, CB, 3F, CB, 3F, 2#, 47, 7E, 6B9  
255# DATA E6, #F, 4F, DD, 86, #B, DD, 6E, #1, 78, #7, C4, 35, #3, DD, 75B  
256# DATA 7E, #2, DD, E5, CD, 58, #4, DD, E1, DD, 7E, #5, #3, 47, 4A, #1D  
257# DATA DD, E5, CD, 27, #4, DD, E1, DD, 46, #B, C3, 1E, 85, #1, #2, 7#F  
258# DATA #4, #B, 1#, 2#, 4#, #1, #2, #4, #B, 1#, 2#, #1, #2, #4, E5, 1A7  
259# DATA 11, 25, #3, 79, #3, 3#, #1, 14, 5F, 7#, 21, FF, #B, FE, #3, 4F2  
260# DATA 28, #5, 25, 3D, 2#, #1, 2C, 1A, A4, #7, 1A, EE, FF, DD, A6, 58#  
261# DATA #B, #4, DD, 77, #B, 1A, A5, 6F, 1A, EE, FF, DD, A6, #1, 85, 77#  
262# DATA DD, 77, #1, E1, C9, FD, 21, 1D, #2, 3E, 3#, FD, 77, #B, DD, 7#3  
263# DATA 21, F7, #1, CD, A1, #3, DD, 21, #3, #2, CD, A1, #3, DD, 21, 7FC  
264# DATA #F, #2, CD, A1, #3, FD, 4E, #B, 3E, #7, C3, 43, 85, 7E, E6, 7#1  
265# DATA #B, 2#, #5, 7E, 23, E6, 7F, C9, 11, #B, #B, 5E, #7, ED, 52, 5E1  
266# DATA 18, F2, DD, 86, #6, DD, 6E, #5, CD, 8E, #3, 4F, DD, 74, #6, 727  
267# DATA DD, 75, #5, FD, 7E, #B, 2#, #B, DD, AE, #B, FD, 77, #B, 3E, 642  
268# DATA #6, CD, 43, 85, DD, 6E, #3, DD, 66, #4, CD, 6E, #3, DD, 75, 76#  
269# DATA #3, DD, 74, #4, E6, #F, 26, #B, 6F, E5, D1, DD, 7E, #B, #7, 8B5  
270# DATA 28, #4, 19, 3D, 18, FA, CB, 3C, CB, 1D, CB, 3C, CB, 1D, CB, 63D  
271# DATA 3C, CB, 1D, CB, 3C, CB, 1D, 4D, DD, 7E, #1, CD, 43, 85, DD, 72E  
272# DATA 66, #B, DD, 6E, #7, CD, 8E, #3, DD, 74, #B, DD, 75, #7, 11, 661  
273# DATA #B, #B, 5F, 21, 1#, #B, #7, ED, 52, DD, 5E, #9, DD, 56, #A, 5#7  
274# DATA B7, 19, 4D, DD, 7E, #2, F5, CD, 43, 85, F1, 3C, 4C, C3, 43, 7#3  
275# DATA 85, 21, 4#, #4, 79, CB, 27, 85, 3#, #1, 24, 6F, 7#, 3C, 5E, 53#  
276# DATA 23, 56, E8, 3D, C8, CB, 3C, CB, 1D, 18, F8, EE, #E, 18, #E, 68A  
277# DATA 4D, #D, #E, #C, DA, #B, 2F, #B, #F, #A, F7, #9, #B, #9, E1, 4FE  
278# DATA #B, #1, #B, #9, #7, DD, 21, 1B, #2, 22, 1B, #2, 21, #B, #4, 51#  
279# DATA B7, 2#, #3, 21, AB, #4, 1E, #B, 79, FE, #7, 3#, #6, #7, DE, 59#  
280# DATA #7, 1C, 18, F6, F5, 85, 3#, #1, 24, 6F, 56, 7#, #7, 2#, 3E, 552  
281# DATA 21, #7, #4, F1, 85, 3#, #1, 24, 6F, 7E, DD, A6, #1, 28, #6, 5C6  
282# DATA DD, A6, #B, 2#, #A, 14, 7A, FE, #C, D8, 1C, DE, #C, 57, C9, 643  
283# DATA CD, 95, #4, 15, 7A, FE, #C, D8, C8, #C, 1D, 57, C9, #4, #5, 66F  
284# DATA #7, #9, #B, #C, #E, 1#, 11, 13, 15, 17, #1, #2, #4, #B, 1#, #4  
285# DATA 2#, 4#, F1, 7#, 3D, 2#, D1, 3D, 2#, D8, 18, CD, 19, #5, 2#, 55F

286# DATA FC, C9, 21, F7, #1, #5, 11, #C, #B, C4, CB, #4, E5, DD, E1, 833  
287# DATA C9, CD, CD, #4, 21, 61, 85, 11, #6, #B, 41, #5, C4, CB, #4, 65B  
288# DATA DD, 23, DD, 23, DD, 23, DD, E5, D1, #1, #6, #B, ED, #B, C9, #B#  
289# DATA DD, 75, #3, DD, 74, #4, 19, DD, 75, #5, DD, 74, #6, 19, DD, 667  
290# DATA 75, #7, DD, 74, #8, C9, CD, CD, #4, 11, #2, #B, 21, 18, 85, 58D  
291# DATA 18, E8, #B, #1, #B, #1, #F, #1, E5, CD, CD, #4, E1, DD, 75, 7C#  
292# DATA #9, DD, 74, #A, C9, C5, CD, CD, #4, C1, DD, 71, #B, C9, C5, #B#  
293# DATA CD, #D, #5, C1, 48, #6, #B, 21, 25, #2, #9, 3E, #B, 77, C9, 48D  
294# DATA #6, F4, ED, 79, #6, F6, ED, 78, F6, C#, ED, 79, #B, 3F, ED, #EF  
295# DATA 79, #6, F4, ED, 49, #6, F6, 4F, F6, #B, ED, 79, ED, 49, C9, 8CF  
296# DATA C1, #5, C3, #5, C5, #5, C7, #5, C9, #5, CB, #5, D4, #5, D6, #F1  
297# DATA #5, D8, #5, #B, #6, 33, #6, 35, #6, 37, #6, 4E, #6, 5#, #6, 643  
298# DATA 52, #6, #5, #6, #7, #6, #9, #6, A7, #6, A9, #6, B1, #6, CF, 86#  
299# DATA #6, D1, #6, D3, #6, D9, #6, DB, #6, DD, #6, FB, #6, FD, #6, ASD  
300# DATA #6, #7, #C, #7, #E, #7, 16, #7, 34, #7, 36, #7, 5F, #7, 65, 515  
301# DATA #7, 67, #7, #B, #7, 9F, #7, A1, #7, A3, #7, B2, #7, B4, #7, B7#  
302# DATA #6, #7, CB, #7, CD, #7, BF, #1, #B, #1, #F, #1, #F, #1, #B, #B#  
303# DATA #1, 9#, 91, 9#, #F, #E, #D, #E, #F, #B, #F, #1, #B, #1, #F, #21  
304# DATA 9#, #F, #3, 42, #E, #F, #3, 42, 9#, 9#, #F, #F, #E, #E, #F, #6AF  
305# DATA 9#, #3, 42, #F, #E, #3, 42, #F, 9#, 91, 9#, #F, #E, #D, #E, #6AF  
306# DATA #F, 9#, 92, 9#, #F, #E, #C, #E, #7, 76, #F, #7, #E, #3, 42, 65E  
307# DATA #D, #3, 42, #C, #3, 42, #B, #3, 42, #A, #3, 42, #9, #3, 42, 41#  
308# DATA #B, #3, 42, #7, #3, 42, #6, #3, 42, #5, #3, 42, #4, #3, 42, 3F7  
309# DATA #3, #3, 42, #2, #3, 42, #1, #3, 42, #B, #1, #B, #1, #F, #1, 3E7  
310# DATA 76, #F, #4, #E, #E, #D, #D, #C, #C, #B, #A, #9, #B, #7, #6, 78A  
311# DATA #5, #4, #3, #2, #1, #B, #1, #B, #1, #F, #1, 76, #F, #7, #E, 58B  
312# DATA #3, 42, #D, #3, 42, #C, #3, 42, #B, #3, 42, #A, #3, 42, #9, 41#  
313# DATA #3, 42, #B, #3, 42, #7, #3, 42, #6, #3, 42, #5, #3, 42, #4, 3F7  
314# DATA #3, 42, #3, #3, 42, #2, #3, 42, #1, #3, 42, #B, #1, #B, #1, 39C  
315# DATA 9#, 91, 9#, #F, #E, #D, #E, #F, #B, 76, #F, #4, #E, #E, #D, 732  
316# DATA #D, #C, #C, #B, #A, #9, #B, #7, #6, #5, #4, #3, #2, #1, #B, 7E7  
317# DATA #1, #B, #1, 9#, 91, 9#, #F, #E, #D, #E, #7, #1, #1, #2, #2, 67#  
318# DATA #3, #3, #4, #4, #5, #5, #6, #6, #7, #7, #8, #8, #9, #9, #A, 7DE  
319# DATA #A, #B, #B, #C, #C, #D, #D, #E, #E, #F, #1, #B, #1, #F, #1, 68F  
320# DATA #6, #A, #C, #E, #F, #1, #B, #1, #F, #1, #1, #1, #2, #2, #3, 65A  
321# DATA #3, #4, #4, #5, #5, #6, #6, #7, #7, #8, #8, #9, #9, #A, #A, 7E5  
322# DATA #B, #B, #C, #C, #D, #D, #E, #E, #F, #1, #B, #1, 9#, 91, 9#, 726  
323# DATA #F, #E, #D, #E, #F, #B, #6, #A, #C, #E, #F, #1, #B, #1, 9#, 69A  
324# DATA 91, 9#, 9F, #E, #D, #E, #7, #1, #1, #2, #2, #3, #3, #4, #4, 77A  
325# DATA #5, #5, #6, #6, #7, #7, #8, #8, #9, #9, #A, #A, #B, #B, #C, 7FC  
326# DATA #C, #D, #D, #E, #E, #F, #1, #B, #1, #F, #E, #F, #3, 42, 9#, 65A  
327# DATA #F, #F, #E, #E, #F, #F, 9#, 9#, #F, #F, #E, #3, 42, #F, 9#, 78#  
328# DATA #3, 42, #F, #E, #D, #E, #F, 9#, 91, 9#, #F, #E, #C, #E, #F, 7#3  
329# DATA 9#, 92, 9#, #F, #B, #6, #A, #C, #E, #F, #1, #B, #1, #F, 9#, 6A3  
330# DATA #F, #3, 42, #E, #F, #3, 42, 9#, 9#, #F, #F, #E, #E, #F, 9#, 6AF  
331# DATA #3, 42, #F, #E, #3, 42, #F, 9#, 91, 9#, #F, #E, #D, #E, #F, 6AE  
332# DATA #B, #2, 9#, #F, #E, #C, #E, #F, #B, #F, #E, #C, #A, #B, #6, 7C1  
333# DATA #5, #4, #3, #2, #1, #1, #B, #B, #1, #1, #1, #F, #1, #F, #E, 64#  
334# DATA #C, #A, #B, #6, #5, #4, #3, #2, #1, #1, #B, #B, #1, 9#, #1, 6C6  
335# DATA #F, #1, 76, #F, #4, #E, #D, #C, #B, #A, #9, #B, #7, #6, #5, #F#  
336# DATA #4, #3, #2, #1, #B, #B, #1, #7, #1, #F, #1, #3, #7, #5, 15, 547  
337# DATA #E, #7, 1F, #9, #A, 35, #B, 5C, #B, 53, #B, #B, 5C, #B, 53, 575  
338# DATA #A, #B, 13, #A, 53, #A, 53, #9, 53, #9, #B, 53, #A, 53, #A, 576  
339# DATA 53, #9, 53, #B, #B, 5C, #B, 53, #B, #B, 5C, #B, 53, #A, #B, 54#  
340# DATA 13, #A, 53, #A, 13, #A, 53, #A, #B, 13, #A, 53, #A, 13, #A, 5#B  
341# DATA 53, #A, #B, 5C, #B, 53, #B, #B, 5C, #B, 53, #A, #B, 13, #A, 5#B  
342# DATA 53, #A, 53, #9, 53, #9, #B, 53, #A, 53, #A, 53, #9, 53, #B, 6#6  
343# DATA #B, 5C, #B, 53, #B, #B, 5C, #B, 53, #A, #B, 15, #A, #B, 15, 43D  
344# DATA #A, #B, 5C, #B, 53, #B, #B, 5C, #B, 53, #A, #B, 13, #A, 53, 5#B



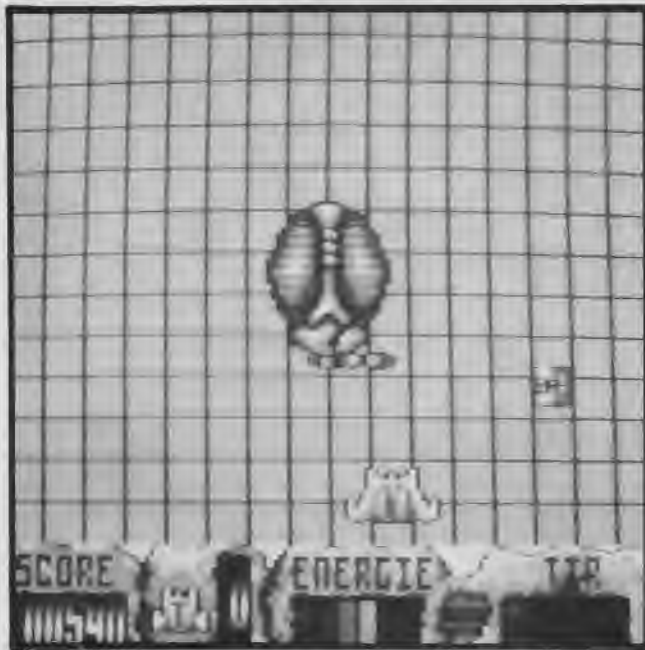
3450 DATA 8A, 53, 89, 53, 89, 00, 53, 8A, 53, 8A, 53, 89, 53, 88, 00, 583  
3460 DATA 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 8A, 00, 13, 8A, 53, 8A, 13, 516  
3470 DATA 8A, 53, 8A, 00, 13, 8A, 53, 8A, 13, 8A, 53, 8A, 00, 5C, 8B, 542  
3480 DATA 53, 8B, 00, 5C, 8B, 53, 8A, 00, 13, 8A, 53, 8A, 53, 89, 53, 54B  
3490 DATA 89, 00, 53, 8A, 53, 8A, 53, 89, 53, 88, 00, 5C, 8B, 53, 8B, 5BF  
3500 DATA 00, 5C, 8B, 53, 8A, 00, 15, 8A, 00, 15, 8A, 00, 5C, 8B, 53, 43C  
3510 DATA 0B, 00, 5C, 8B, 53, 8A, 00, 13, 8A, 53, 8A, 53, 89, 53, 89, 561  
3520 DATA 00, 53, 8A, 53, 8A, 53, 89, 53, 8B, 00, 5C, 8B, 53, 8B, 00, 536  
3530 DATA 5C, 8B, 53, 8A, 00, 13, 8A, 53, 8A, 13, 8A, 53, 8A, 00, 13, 4CB  
3540 DATA 8A, 53, 8A, 13, 8A, 53, 8A, 00, 5C, 8B, 53, 8B, 00, 5C, 8B, 58D  
3550 DATA 53, 8A, 00, 13, 8A, 53, 8A, 53, 89, 53, 89, 00, 53, 8A, 53, 53F  
3560 DATA 8A, 53, 89, 53, 8B, 00, 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 8A, 5CA  
3570 DATA 00, 15, 8A, 00, 15, 8A, 00, 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 434  
3580 DATA 87, 00, 13, 87, 53, 87, 53, 86, 53, 86, 00, 53, 87, 53, 87, 561  
3590 DATA 53, 86, 53, 85, 00, 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 87, 00, 52E  
3600 DATA 13, 87, 53, 87, 13, 87, 53, 87, 00, 13, 87, 53, 87, 13, 87, 4FB  
3610 DATA 53, 87, 00, 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 87, 00, 13, 87, 4F1  
3620 DATA 53, 87, 53, 86, 53, 86, 00, 53, 87, 53, 87, 53, 86, 53, 85, 5F1  
3630 DATA 00, 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 87, 00, 15, 87, 00, 15, 42E  
3640 DATA 87, 00, 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 8A, 00, 13, 8A, 53, 50F  
3650 DATA 8A, 53, 89, 53, 89, 00, 53, 8A, 53, 8A, 53, 89, 53, 8B, 00, 583  
3660 DATA 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 8A, 00, 13, 8A, 53, 8A, 13, 516  
3670 DATA 8A, 53, 8A, 00, 13, 8A, 53, 8A, 13, 8A, 53, 8A, 00, 5C, 8B, 542  
3680 DATA 53, 8B, 00, 5C, 8B, 53, 8A, 00, 13, 8A, 53, 8A, 53, 8B, 53, 54B  
3690 DATA 89, 00, 53, 8A, 53, 8A, 53, 89, 53, 8B, 00, 5C, 8B, 53, 8B, 5BF  
3700 DATA 00, 5C, 8B, 53, 8A, 00, 15, 8A, 00, 15, 8A, 00, 54, 8B, 53, 434  
3710 DATA 8B, 53, 8B, 00, 53, 8B, 54, 8B, 53, 8C, 00, 14, 8C, 53, 8C, 581  
3720 DATA 53, 8C, 00, 13, 8C, 53, 8C, 53, 8D, 53, 8C, 00, 55, 8B, 00, 4FC  
3730 DATA 15, 8B, 00, 73, 53, 8B, 53, 89, 53, 8B, 00, 13, 8B, 53, 8B, 524  
3740 DATA 53, 89, 53, 8B, 00, 54, 8B, 53, 8B, 53, 8B, 00, 53, 8B, 54, 581  
3750 DATA 8B, 53, 8C, 00, 14, 8C, 53, 8C, 53, 8C, 00, 13, 8C, 53, 8C, 546  
3760 DATA 53, 8D, 53, 8C, 00, 55, 8B, 00, 15, 8B, 00, 53, 8D, 53, 8D, 51F  
3770 DATA 53, 8D, 53, 8D, 00, 13, 8D, 53, 8C, 53, 8B, 53, 8C, 00, 54, 570  
3780 DATA 8B, 53, 8B, 53, 8B, 00, 53, 8B, 54, 8B, 53, 8C, 00, 14, 8C, 580  
3790 DATA 53, 8C, 53, 8C, 00, 13, 8C, 53, 8C, 53, 8D, 53, 8C, 00, 55, 550  
3800 DATA 8B, 00, 15, 8B, 00, 14, 8B, 53, 8B, 53, 8B, 00, 13, 8B, 53, 477  
3810 DATA 8B, 14, 8B, 00, 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 8A, 00, 13, 4C6  
3820 DATA 8A, 53, 8A, 53, 89, 53, 89, 00, 53, 8A, 53, 8A, 53, 89, 53, 60B  
3830 DATA 8B, 00, 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 8A, 00, 13, 8A, 53, 501  
3840 DATA 8A, 13, 8A, 53, 8A, 00, 13, 8A, 53, 8A, 13, 8A, 53, 8A, 00, 4FB  
3850 DATA 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 8A, 00, 13, 8A, 53, 8A, 53, 556  
3860 DATA 89, 53, 89, 00, 53, 8A, 53, 8A, 53, 89, 53, 8B, 00, 5C, 8B, 58D  
3870 DATA 53, 8B, 00, 5C, 8B, 53, 8A, 00, 55, 8B, 00, 15, 8B, 00, 85, 4A7  
3880 DATA 16, 55, 8B, 00, 03, 42, 83, 07, 85, FE, 96, 87, 1F, 89, 84, 591  
3890 DATA 35, 00, 75, 00, 75, 00, 75, 00, 75, 00, 75, 00, 75, 00, 75, 368  
3900 DATA 00, 75, 00, 75, 00, 75, 00, 75, 00, 75, 00, 75, 00, 75, 333  
3910 DATA 75, 00, 74, 54, C3, 00, 53, CB, 53, 8B, 53, 8B, 53, 8B, 00, 585  
3920 DATA 53, CB, 53, 8B, 53, 8B, 53, 87, 00, 53, CA, 53, 8A, 53, 8A, 68B  
3930 DATA 53, 89, 00, 53, CA, 53, 8A, 53, CA, 53, 89, 00, 53, CB, 53, 640  
3940 DATA 8B, 53, 8B, 53, 8B, 00, 53, CB, 53, 8B, 53, 8B, 53, 87, 00, 5FB  
3950 DATA 53, CA, 53, 8A, 53, 8A, 53, 89, 00, 53, CA, 53, 8A, 53, CA, 6CA  
3960 DATA 53, 89, 00, 53, CB, 53, 8B, 53, 8B, 53, 8B, 00, 53, CB, 53, 602  
3970 DATA 8B, 53, 8B, 53, 87, 00, 53, CA, 53, 8A, 53, 8A, 53, 89, 00, 5FB  
3980 DATA 53, CA, 53, 8A, 53, CA, 53, 89, 00, 53, CB, 53, 8B, 53, 8B, 6CD  
3990 DATA 53, 8B, 00, 53, CB, 53, 8B, 53, 8B, 50, 87, 50, 87, 00, 53, 586  
4000 DATA CA, 50, 89, 50, 89, 53, CA, 50, 89, 50, 89, 00, 53, CA, 53, 6BB  
4010 DATA CA, 53, C9, 53, C8, 00, 8E, 53, CB, 53, 8B, 53, 8B, 53, 8B, 744  
4020 DATA 00, 53, CB, 53, 8B, 53, 8B, 53, 87, 00, 53, CA, 53, 8A, 53, 601  
4030 DATA 8A, 53, 89, 00, 53, CA, 53, 8A, 53, CA, 53, 89, 00, 53, CB, 677

4040 DATA 53, 8B, 53, 8B, 53, 8B, 00, 53, CB, 53, 8B, 53, 8B, 53, 87, 64B  
4050 DATA 00, 53, CA, 53, 8A, 53, 8A, 53, 89, 00, 53, CA, 53, 8A, 53, 600  
4060 DATA CA, 53, 89, 00, 53, CB, 53, 8B, 53, 8B, 53, 8B, 00, 53, CB, 679  
4070 DATA 53, 8B, 53, 8B, 53, 87, 00, 53, CA, 53, 8A, 53, 8A, 53, 89, 649  
4080 DATA 00, 53, CA, 53, 8A, 53, CA, 53, 89, 00, 53, CB, 53, 8B, 53, 642  
4090 DATA 8B, 53, 8B, 00, 53, CB, 53, 8B, 53, 8B, 50, 87, 50, 87, 00, 5EE  
4100 DATA 53, CA, 50, 89, 50, 89, 53, CA, 50, 89, 50, 89, 00, 53, CA, 6BB  
4110 DATA 53, CA, 53, C9, 53, C8, 00, 53, C8, 53, 8B, 53, 8B, 53, 85, 6FD  
4120 DATA 00, 53, C8, 53, 8B, 53, 8B, 53, 84, 00, 53, C7, 53, 87, 53, 5EF  
4130 DATA 87, 53, 86, 00, 53, C7, 53, 87, 53, C7, 53, 86, 00, 53, CB, 662  
4140 DATA 53, 8B, 53, 8B, 53, 85, 00, 53, CB, 53, 8B, 53, 8B, 53, 84, 636  
4150 DATA 00, 53, C7, 53, 87, 53, 87, 53, 86, 00, 53, C7, 53, 87, 53, 5EE  
4160 DATA C7, 53, 86, 00, 53, CB, 53, 8B, 53, 8B, 53, 85, 00, 53, C8, 864  
4170 DATA 53, 8B, 53, 8B, 53, 84, 00, 53, C7, 53, 87, 53, 87, 53, 86, 634  
4180 DATA 00, 53, C7, 53, 87, 53, C7, 53, 86, 00, 53, CB, 53, 8B, 53, 630  
4190 DATA 8B, 53, 85, 00, 53, CB, 53, 8B, 53, 8B, 50, 84, 50, 84, 00, 5D9  
4200 DATA 53, C7, 50, 86, 50, 86, 53, C7, 50, 86, 50, 86, 00, 53, C7, 6A6  
4210 DATA 53, C7, 53, C6, 53, C5, 00, 53, CB, 53, 8B, 53, 8B, 53, 8B, 700  
4220 DATA 00, 53, CB, 53, 8B, 53, 8B, 53, 87, 00, 53, CA, 53, 8A, 53, 601  
4230 DATA 8A, 53, 89, 00, 53, CA, 53, 8A, 53, CA, 53, 89, 00, 53, CB, 677  
4240 DATA 53, 8B, 53, 8B, 53, 8B, 00, 53, CB, 53, 8B, 53, 8B, 53, 87, 64B  
4250 DATA 00, 53, CA, 53, 8A, 53, 8A, 53, 89, 00, 53, CA, 53, 8A, 53, 600  
4260 DATA CA, 53, 89, 00, 53, CB, 53, 8B, 53, 8B, 53, 8B, 00, 53, CB, 679  
4270 DATA 53, 8B, 53, 8B, 53, 87, 00, 53, CA, 53, 8A, 53, 8A, 53, 89, 649  
4280 DATA 00, 53, CA, 53, 8A, 53, CA, 53, 89, 00, 53, CB, 53, 8B, 53, 642  
4290 DATA 8B, 53, 8B, 00, 53, CB, 53, 8B, 53, 8B, 50, 87, 50, 87, 00, 5EE  
4300 DATA 53, CA, 50, 89, 50, 89, 53, CA, 50, 89, 50, 89, 00, 53, CA, 6BB  
4310 DATA 53, CA, 53, C9, 53, C8, 00, 53, C7, 53, 87, 53, 87, 53, 86, 6FB  
4320 DATA 00, 53, C7, 53, 87, 53, 87, 53, 86, 00, 53, CB, 53, 8B, 53, 5F0  
4330 DATA 8B, 53, 87, 00, 53, CB, 53, 8B, 53, 8B, 53, 87, 00, 53, CB, 62B  
4340 DATA 53, 8B, 53, 8B, 53, 8B, 00, 53, CB, 53, 8B, 53, 8B, 53, 8B, 64C  
4350 DATA 00, 53, CB, 53, 8B, 53, 8B, 53, 8B, 00, 53, CB, 53, 8B, 53, 601  
4360 DATA CB, 53, 87, 00, 53, C7, 53, 87, 53, 87, 53, 86, 00, 53, C7, 666  
4370 DATA 53, 87, 53, 87, 53, 86, 00, 53, CB, 53, 8B, 53, 8B, 53, 87, 638  
4380 DATA 00, 53, CB, 53, 8B, 53, 8B, 53, 87, 00, 53, C9, 53, 89, 53, 5FB  
4390 DATA 89, 53, 86, 00, 53, C9, 53, 89, 53, 89, 53, 86, 00, 53, C9, 62B  
4400 DATA 53, 89, 53, 89, 53, 86, 00, 53, C9, 53, 86, 53, C9, 53, 86, 67B  
4410 DATA 00, 53, C7, 53, 87, 53, 87, 53, 86, 00, 53, C7, 53, 87, 53, 5EE  
4420 DATA 87, 53, 86, 00, 53, CB, 53, 8B, 53, 8B, 53, 87, 00, 53, CB, 626  
4430 DATA 53, 8B, 53, 8B, 53, 87, 00, 53, CB, 53, 8B, 53, 8B, 53, 8B, 645  
4440 DATA 00, 53, CB, 53, 8B, 53, 8B, 53, 8B, 00, 53, CB, 53, CB, 53, 644  
4450 DATA CB, 53, CB, 00, 53, CB, 53, CB, 53, CB, 53, CB, 00, 53, CB, 77F  
4460 DATA 53, 8B, 53, 8B, 53, 8B, 00, 53, CB, 53, 8B, 53, 8B, 53, 87, 64B  
4470 DATA 00, 53, CA, 53, 8A, 53, 8A, 53, 89, 00, 53, CA, 53, 8A, 53, 600  
4480 DATA CA, 53, 89, 00, 53, CB, 53, 8B, 53, 8B, 53, 8B, 00, 53, CB, 679  
4490 DATA 53, 8B, 53, 8B, 53, 87, 00, 53, CA, 53, 8A, 53, 8A, 53, 89, 649  
4500 DATA 00, 53, CA, 53, 8A, 53, CA, 53, 89, 00, 53, CB, 53, 8B, 53, 642  
4510 DATA 8B, 53, 8B, 00, 53, CB, 53, 8B, 53, 8B, 53, 87, 00, 53, CA, 637  
4520 DATA 53, 8A, 53, 8A, 53, 89, 00, 53, CA, 53, 8A, 53, CA, 53, 89, 689  
4530 DATA 00, 53, CB, 53, 8B, 53, 8B, 53, 8B, 00, 53, CB, 53, 8B, 53, 604  
4540 DATA 8B, 50, 87, 50, 87, 00, 55, CB, 00, 14, CB, 53, 8B, 53, 8A, 5FB  
4550 DATA 00, 55, CB, 00, 03, 42, 83, 07, 85, 15, 8E, 87, 1D, 89, 84, 4CB  
4560 DATA 35, 00, 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 87, 00, 13, 87, 53, 49F  
4570 DATA 87, 53, 87, 53, 87, 00, 53, 87, 53, 87, 53, 87, 53, 87, 00, 5A3  
4580 DATA 5C, 8B, 53, 8B, 00, 5C, 8B, 53, 87, 00, 13, 87, 53, 87, 13, 504  
4590 DATA 87, 53, 87, 00, 13, 87, 53, 87, 13, 87, 53, 87, 00, 5C, 8B, 52D  
4600 DATA 53, 8B, 00, 5C, 8B, 53, 87, 00, 13, 87, 53, 87, 53, 87, 53, 53A  
4610 DATA 87, 00, 53, 87, 53, 87, 53, 87, 53, 87, 00, 5C, 8B, 53, 8B, 5AE  
4620 DATA 00, 5C, 8B, 53, 87, 00, 15, 87, 00, 15, 87, 00, 5C, 8B, 53, 42D



4630 DATA 88, 00, 5C, 88, 53, 87, 00, 13, 87, 53, 87, 53, 87, 53, 87, 56E  
 4640 DATA 00, 53, 87, 53, 87, 53, 87, 53, 87, 00, 5C, 88, 53, 88, 00, 52F  
 4650 DATA 5C, 88, 53, 87, 00, 13, 87, 53, 87, 13, 87, 53, 87, 00, 13, 489  
 4660 DATA 87, 53, 87, 13, 87, 53, 87, 00, 5C, 88, 53, 88, 00, 5C, 88, 578  
 4670 DATA 53, 87, 00, 13, 87, 53, 87, 53, 87, 53, 87, 00, 53, 87, 53, 52F  
 4680 DATA 87, 53, 87, 53, 87, 00, 5C, 88, 53, 88, 00, 5C, 88, 53, 87, 58B  
 4690 DATA 00, 15, 87, 00, 15, 87, 00, 87, 1F, 85, 56, 53, 84, 53, 84, 467  
 4700 DATA 53, 84, 53, 84, 00, 53, 84, 53, 84, 53, 84, 53, 82, 00, 53, 55B  
 4710 DATA 83, 53, 83, 53, 83, 53, 82, 00, 53, 83, 53, 82, 53, 83, 53, 50B  
 4720 DATA 82, 00, 53, 84, 53, 84, 53, 84, 53, 83, 00, 53, 84, 53, 84, 58B  
 4730 DATA 53, 84, 53, 82, 00, 53, 83, 53, 83, 53, 83, 53, 82, 00, 53, 556  
 4740 DATA 83, 53, 82, 53, 83, 53, 82, 00, 53, 84, 53, 84, 53, 84, 53, 5DB  
 4750 DATA 84, 00, 53, 84, 53, 84, 53, 84, 53, 82, 00, 53, 83, 53, 83, 58A  
 4760 DATA 53, 83, 53, 82, 00, 53, 83, 53, 82, 53, 83, 53, 82, 00, 53, 554  
 4770 DATA 84, 53, 84, 53, 84, 53, 83, 00, 53, 84, 53, 84, 53, 84, 53, 5E0  
 4780 DATA C3, 00, 15, C3, 00, 15, C3, 00, 53, 81, 53, 81, 53, 81, 53, 542  
 4790 DATA 88, 00, 53, 81, 53, 81, 53, 81, 53, 80, 00, 53, 80, 53, 80, 575  
 4800 DATA 53, 80, 53, 81, 00, 53, 80, 53, 81, 53, 80, 53, 81, 00, 53, 548  
 4810 DATA 81, 53, 81, 53, 81, 53, 80, 00, 53, 81, 53, 81, 53, 81, 53, 5CB  
 4820 DATA 80, 00, 53, 80, 53, 80, 53, 80, 53, 81, 00, 53, 80, 53, 81, 574  
 4830 DATA 53, 80, 53, 81, 00, 53, 81, 53, 81, 53, 81, 53, 80, 00, 53, 549  
 4840 DATA 81, 53, 81, 53, 81, 53, 80, 00, 53, 80, 53, 80, 53, 80, 53, 5C8  
 4850 DATA 81, 00, 53, 80, 53, 81, 53, 80, 53, 81, 00, 53, 81, 53, 81, 577  
 4860 DATA 53, 81, 53, 80, 00, 53, 81, 53, 81, 53, 81, 53, C8, 00, 13, 549  
 4870 DATA C0, 53, C1, 53, C2, 53, C1, 00, 53, C3, 53, C1, 53, D3, 53, 7A0  
 4880 DATA C1, 00, 53, 84, 53, 84, 53, 84, 53, 84, 00, 53, 84, 53, 84, 5CB  
 4890 DATA 53, 84, 53, 82, 00, 53, 83, 53, 83, 53, 83, 53, 82, 00, 53, 556  
 4900 DATA CA, 53, CB, 53, DC, 53, CB, 00, 53, 84, 53, 84, 53, 84, 53, 70D  
 4910 DATA 83, 00, 53, 84, 53, 84, 53, 84, 53, 82, 00, 53, 83, 53, 83, 589  
 4920 DATA 53, 83, 53, 82, 00, 53, DC, 53, CC, 53, CC, 53, CC, 00, 53, 68A  
 4930 DATA 84, 53, 84, 53, 84, 53, 84, 00, 53, 84, 53, 84, 53, 84, 53, 5E1  
 4940 DATA 82, 00, 53, 83, 53, 83, 53, 83, 53, 82, 00, 53, CA, 53, CB, 614  
 4950 DATA 53, DC, 53, CB, 00, 53, 84, 53, 84, 53, 84, 53, 83, 00, 53, 5FB  
 4960 DATA 84, 53, 84, 53, 84, 53, C3, 00, 53, C8, 53, C8, 53, C8, 53, 6EC  
 4970 DATA C8, 00, 53, C8, 53, C7, 53, D5, 53, C4, 00, 53, 80, 53, 80, 6E2  
 4980 DATA 53, 80, 53, 80, 00, 53, 80, 53, 80, 53, 80, 53, 80, 00, 53, 545  
 4990 DATA 81, 53, 81, 53, 81, 53, 81, 00, 53, 81, 53, 81, 53, 81, 53, 5CC  
 5000 DATA 81, 00, 53, 84, 53, 84, 53, 84, 53, 84, 00, 53, 84, 53, 84, 58B  
 5010 DATA 53, 84, 53, 84, 00, 53, 84, 53, 84, 53, 84, 53, 84, 00, 53, 55D  
 5020 DATA 84, 53, 84, 53, 84, 53, 84, 00, 53, 80, 53, 80, 53, 80, 53, 5D5  
 5030 DATA 80, 00, 53, 80, 53, 80, 53, 80, 53, 80, 00, 53, 81, 53, 81, 574  
 5040 DATA 53, 81, 53, 81, 00, 53, 81, 53, 81, 53, 81, 53, 81, 00, 53, 54B  
 5050 DATA 82, 53, 82, 53, 82, 53, 82, 00, 53, 82, 53, 82, 53, 82, 53, 5D3  
 5060 DATA 82, 00, 53, 82, 53, 82, 53, 82, 53, 82, 00, 53, 82, 53, 82, 58B  
 5070 DATA 53, 82, 53, 82, 00, 53, 80, 53, 80, 53, 80, 00, 53, 549  
 5080 DATA 80, 53, 80, 53, 80, 53, 80, 00, 53, 81, 53, 81, 53, 81, 53, 5C8  
 5090 DATA 81, 00, 53, 81, 53, 81, 53, 81, 00, 53, 84, 53, 84, 57F  
 5100 DATA 53, 84, 53, 84, 00, 53, 84, 53, 84, 53, 84, 53, 84, 00, 53, 55D  
 5110 DATA 84, 53, 84, 53, 84, 53, 84, 00, 53, 84, 53, 84, 53, 84, 53, 5E1  
 5120 DATA 84, 00, 53, 84, 53, 84, 53, 84, 00, 53, 84, 53, 84, 58E  
 5130 DATA 53, 84, 53, 82, 00, 53, 83, 53, 83, 53, 83, 53, 82, 00, 53, 556  
 5140 DATA 83, 53, 82, 53, 83, 53, 82, 00, 53, 84, 53, 84, 53, 84, 53, 5DB  
 5150 DATA 83, 00, 53, 84, 53, 84, 53, 84, 53, 82, 00, 53, 83, 53, 83, 589  
 5160 DATA 53, 83, 53, 82, 00, 53, 83, 53, 82, 53, 83, 53, 82, 00, 53, 554  
 5170 DATA 84, 53, 84, 53, 84, 53, 84, 00, 53, 84, 53, 84, 53, 84, 53, 5E1  
 5180 DATA 82, 00, 53, 83, 53, 83, 53, 83, 53, 82, 00, 53, 83, 53, 82, 584  
 5190 DATA 53, 83, 53, 82, 00, 53, 84, 53, 84, 53, 84, 53, 83, 00, 53, 559  
 5200 DATA 84, 53, 84, 53, 84, 53, C3, 00, 55, C4, 00, 15, C4, 00, 55, 58F  
 5210 DATA 84, 76, 00, 2C, 83, 07, 85, 15, 8E, 87, 1F, 89, 84, 35, 00, 4C0

5220 DATA 5C, 88, 53, 88, 00, 5C, 88, 53, 8A, 00, 13, 8A, 53, 8A, 53, 556  
 5230 DATA 89, 53, 89, 00, 53, 8A, 53, 8A, 53, 89, 53, 88, 00, 5C, 88, 58D  
 5240 DATA 53, 88, 00, 5C, 88, 53, 8A, 00, 13, 8A, 53, 8A, 00, 03, 42, 461  
 5250 DATA 21, 40, CD, 11, 00, CD, 01, 40, 00, ED, 80, 21, DE, 94, CD, 64A  
 5260 DATA 5F, 93, CD, 19, BD, 21, C2, 94, CD, 4F, 93, 21, EE, 94, CD, 82B  
 5270 DATA 5F, 93, CD, 75, 93, CD, 19, BD, 76, 76, 02, CD, 89, 80, 3A, 8EB  
 5280 DATA 0C, 80, FE, 85, 20, 03, CD, 75, 93, CD, 7E, 93, CD, 85, 93, 7FA  
 5290 DATA CD, 18, 8B, 30, E3, CD, 03, 80, 21, DE, 94, CD, 5F, 93, 21, 779  
 5300 DATA CF, 94, CD, 4F, 93, 06, BC, 7E, B7, F8, ED, 79, 23, 04, 7E, 80C  
 5310 DATA ED, 79, 23, 05, 18, F2, AF, 46, 48, F5, E5, CD, 32, BC, E1, 84B  
 5320 DATA F1, 23, 3C, FE, 10, 20, F1, 01, 00, 00, C3, 38, BC, 3E, 01, 566  
 5330 DATA DD, 21, DC, 94, C3, 00, 80, 3E, 09, FE, 09, 20, 2C, 21, 3A, 5A6  
 5340 DATA 95, 7E, 23, 87, F2, 91, 93, 21, 3A, 95, 7E, 22, 85, 93, D6, 7B1  
 5350 DATA 20, 26, 00, 6F, 11, FE, 94, 19, 7E, 26, 00, 6F, 11, 00, C0, 455  
 5360 DATA 76, 29, 07, 19, 22, 7F, 94, AF, 3C, 32, 7F, 93, C9, 3E, 10, 53A  
 5370 DATA 11, D0, 20, D5, 62, 6B, 23, ED, A0, ED, A0, ED, A0, ED, A0, 8FA  
 5380 DATA ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, BC8  
 5390 DATA A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, 87B  
 5400 DATA ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, BC8  
 5410 DATA A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, 87B  
 5420 DATA ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, BC8  
 5430 DATA A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, 87B  
 5440 DATA ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, BC8  
 5450 DATA A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, 87B  
 5460 DATA ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, BC8  
 5470 DATA A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, 87B  
 5480 DATA ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, BC8  
 5490 DATA A0, ED, A0, ED, A0, ED, A0, ED, A0, ED, A0, F5, 3A, 7F, 93, AA2  
 5500 DATA FE, 09, 28, 0F, 21, 00, 00, 1B, ED, A0, 01, 07, 00, 09, 22, 33A  
 5510 DATA 7F, 94, 18, 04, 3E, 3C, 1B, 12, F1, D1, F5, 7A, C6, 88, 57, 62C  
 5520 DATA E6, 38, 20, 14, 7A, D6, 40, 57, 7B, C6, 5C, 5F, 30, 0A, 14, 583  
 5530 DATA 7A, E6, 07, 20, 04, 7A, C6, 38, 57, F1, 30, C2, 8A, 93, 2A, 6C1  
 5540 DATA 7F, 94, 01, 7F, 00, B7, ED, 02, 42, 22, 7F, 94, C9, 01, 2E, 5A8  
 5550 DATA 02, 31, 06, 21, 07, 22, 0C, 0C, 00, 68, 80, 01, 28, 02, 2E, 1E9  
 5560 DATA 06, 19, 07, 1E, 0C, 30, 0D, 00, 80, 76, 00, 13, 03, 06, 0F, 1AE  
 5570 DATA 10, 19, 1A, 17, 01, 05, 0E, 0B, 11, 04, 07, 08, 2C, 1E, 76, 15D  
 5580 DATA 2C, 05, 21, 1F, 20, 2C, 2C, 1B, 2C, 1D, 2C, 22, 23, 24, 25, 207  
 5590 DATA 26, 27, 28, 29, 2A, 2B, 1A, 76, 2C, 04, 1C, 2C, 00, 01, 02, 1FE  
 5600 DATA 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 96  
 5610 DATA 12, 13, 14, 15, 16, 17, 18, 19, 2C, 76, 20, 0F, 43, 50, 43, 253  
 5620 DATA 20, 49, 4E, 46, 4F, 53, 20, 50, 52, 45, 53, 45, 4E, 54, 45, 425  
 5630 DATA 20, 3A, 20, 20, 41, 20, 58, 20, 59, 20, 53, 2E, 76, 20, 06, 309  
 5640 DATA 55, 4E, 20, 53, 48, 4F, 4F, 54, 27, 45, 4D, 27, 55, 50, 20, 3F5  
 5650 DATA 50, 52, 4F, 47, 52, 41, 4D, 4D, 45, 20, 50, 41, 52, 20, 46, 413  
 5660 DATA 41, 02, 42, 40, 45, 4E, 20, 46, 45, 53, 53, 41, 52, 44, 20, 3A0  
 5670 DATA 28, 46, 45, 46, 45, 53, 53, 45, 29, 2C, 20, 49, 4C, 4C, 55, 3D4  
 5680 DATA 53, 54, 52, 45, 20, 50, 41, 52, 20, 53, 54, 45, 50, 48, 41, 426  
 5690 DATA 4E, 45, 20, 53, 41, 49, 4E, 54, 20, 4D, 41, 52, 54, 49, 4E, 41D  
 5700 DATA 20, 28, 53, 54, 45, 46, 20, 4F, 55, 20, 53, 53, 4D, 29, 20, 39A  
 5710 DATA 45, 54, 20, 53, 4F, 4E, 4F, 52, 49, 53, 45, 20, 50, 41, 52, 42E  
 5720 DATA 20, 47, 52, 45, 47, 4F, 52, 59, 20, 43, 4C, 45, 4D, 45, 4E, 413  
 5730 DATA 54, 20, 28, 4D, 55, 53, 49, 43, 49, 45, 4E, 20, 4F, 45, 55, 402  
 5740 DATA 56, 52, 41, 4E, 54, 20, 45, 4E, 20, 46, 52, 45, 45, 20, 4C, 3EC  
 5750 DATA 41, 4E, 43, 45, 20, 50, 4F, 55, 52, 20, 55, 4E, 45, 20, 53, 3F8  
 5760 DATA 4F, 43, 49, 45, 54, 45, 20, 44, 27, 45, 44, 49, 54, 49, 4F, 402  
 5770 DATA 4E, 20, 46, 52, 41, 4E, 43, 41, 49, 53, 45, 20, 44, 4F, 4E, 3FB  
 5780 DATA 54, 20, 4A, 45, 20, 54, 41, 49, 52, 41, 49, 20, 4C, 45, 20, 3AE  
 5790 DATA 4E, 4F, 4D, 20, 21, 20, 29, 76, 20, 08, 43, 4F, 4D, 4D, 45, 383  
 5800 DATA 20, 56, 4F, 55, 53, 20, 4C, 27, 41, 56, 45, 5A, 20, 53, 41, 3EA



5810 DATA 4E,53,20,44,4F,55,54,45,20,52,45,4D,41,52,51,42A  
 5820 DATA 55,45,2C,20,4C,45,20,43,4F,50,59,52,49,47,4B,3FC  
 5830 DATA 54,20,4E,27,45,53,54,20,50,41,53,20,43,4F,52,3DD  
 5840 DATA 52,45,43,54,2E,03,42,20,20,45,48,20,02,42,49,31B  
 5850 DATA 45,4E,20,43,27,45,53,54,20,54,4F,55,54,20,53,3E8  
 5860 DATA 49,4D,50,4C,45,4D,45,4E,54,20,50,41,52,43,45,436  
 5870 DATA 20,51,55,27,20,41,58,59,53,20,41,20,45,54,45,381  
 5880 DATA 20,52,45,54,41,52,44,45,20,50,41,52,20,51,55,3F0  
 5890 DATA 45,4C,51,55,45,53,20,02,42,55,47,53,20,54,45,3DB  
 5900 DATA 4E,41,43,45,53,20,21,20,4A,27,41,49,4D,45,52,3AA  
 5910 DATA 41,49,20,44,27,41,49,4C,4C,45,55,52,53,20,56,3EC  
 5920 DATA 4F,55,53,20,50,52,45,56,45,4E,49,52,20,51,55,448  
 5930 DATA 27,41,56,45,43,20,55,4E,20,43,50,43,20,44,45,3A8  
 5940 DATA 20,36,34,48,2C,20,56,4F,55,53,20,4E,27,41,55,399  
 5950 DATA 52,45,5A,20,4D,41,4C,48,45,55,52,45,53,45,4D,449  
 5960 DATA 45,4E,54,20,50,41,53,20,44,45,20,02,42,52,55,39F  
 5970 DATA 49,54,41,47,45,53,2E,03,42,20,44,45,20,50,4C,395  
 5980 DATA 55,53,2C,20,4A,45,20,4E,27,41,49,20,50,41,53,3A6  
 5990 DATA 20,50,55,20,4D,45,54,54,52,45,20,55,4E,45,20,3DE  
 6000 DATA 4D,55,53,49,51,55,45,2E,03,42,76,20,00,50,4F,3D9  
 6010 DATA 55,52,51,55,4F,49,20,54,41,4E,54,20,44,45,20,405  
 6020 DATA 48,41,49,4E,45,20,3F,20,4F,55,49,2C,20,50,4F,38C  
 6030 DATA 55,52,51,55,4F,49,20,3F,20,28,54,45,52,29,76,416  
 6040 DATA 20,00,4C,45,53,20,02,42,52,55,49,54,41,47,45,381  
 6050 DATA 53,20,3A,20,4D,41,4E,51,55,45,20,44,45,20,4D,3AA  
 6060 DATA 45,4D,4F,49,52,45,20,21,76,20,04,4C,41,20,4D,396  
 6070 DATA 55,53,49,51,55,45,20,3A,20,4C,41,20,52,4F,55,3F9  
 6080 DATA 54,49,4E,45,20,44,45,20,4D,55,53,49,43,20,53,3ED  
 6090 DATA 54,55,44,49,4F,20,45,54,41,4E,54,20,31,30,20,3C2  
 6100 DATA 46,4F,49,53,20,54,52,4F,50,20,4C,45,4E,54,45,42E  
 6110 DATA 2C,20,4C,41,20,53,59,4E,43,48,52,4F,20,4E,45,3D2  
 6120 DATA 20,53,45,20,46,41,49,53,41,49,54,20,50,4C,55,3EA  
 6130 DATA 53,20,50,4F,55,52,20,4C,45,20,53,43,52,4F,4C,40D  
 6140 DATA 4C,49,4E,47,20,40,41,52,44,57,41,52,45,20,21,3D9  
 6150 DATA 03,42,76,20,00,43,45,50,45,4E,44,41,4E,54,20,395  
 6160 DATA 52,41,53,53,55,52,45,5A,20,56,4F,55,53,20,4A,456  
 6170 DATA 45,20,50,4F,53,53,45,44,45,20,4D,41,49,4E,54,411  
 6180 DATA 45,4E,41,4E,54,20,4D,45,53,20,50,52,4F,50,52,42E  
 6190 DATA 45,53,20,52,4F,55,54,49,4E,45,53,20,44,45,20,3FA

6200 DATA 4D,55,53,49,51,55,45,20,45,54,20,4A,27,41,49,3FD  
 6210 DATA 20,51,55,41,4E,44,20,4D,45,4D,45,20,50,55,20,3C2  
 6220 DATA 4C,45,53,20,55,54,49,4C,49,53,45,52,20,50,4F,43A  
 6230 DATA 55,52,20,4C,45,53,20,02,42,52,55,49,54,41,47,3DB  
 6240 DATA 45,53,20,21,76,20,00,45,54,20,4D,41,49,4E,54,3A9  
 6250 DATA 45,4E,41,4E,54,20,51,55,45,4C,51,55,45,53,20,42B  
 6260 DATA 49,4E,46,4F,53,20,53,55,50,50,4C,45,4D,45,4E,458  
 6270 DATA 54,41,49,52,45,53,2E,03,42,76,20,04,4A,45,20,384  
 6280 DATA 53,55,49,53,20,41,43,54,55,45,4C,4C,45,4D,45,445  
 6290 DATA 4E,54,20,45,4E,20,54,52,41,49,4E,20,44,45,20,3BC  
 6300 DATA 44,45,56,45,4C,4F,50,50,45,52,20,58,59,50,48,45F  
 6310 DATA 4F,45,53,20,46,41,4E,54,41,53,59,2C,20,55,4E,40C  
 6320 DATA 20,4A,45,55,20,44,27,48,45,52,4F,40,43,20,46,3AF  
 6330 DATA 41,4E,54,41,53,59,20,41,55,58,20,47,52,41,50,428  
 6340 DATA 48,49,53,4D,45,53,20,41,02,42,53,4F,4C,55,4D,3FE  
 6350 DATA 45,4E,54,20,53,55,02,42,4C,49,4D,45,53,20,28,385  
 6360 DATA 45,54,20,4A,45,20,4D,41,43,48,45,20,4D,45,53,3CB  
 6370 DATA 20,4D,4F,54,53,20,21,20,29,20,51,55,49,20,53,36F  
 6380 DATA 45,52,41,20,45,44,49,54,45,20,50,41,52,20,55,3DB  
 6390 DATA 4E,45,20,53,4F,43,49,45,54,45,20,46,52,41,4E,406  
 6400 DATA 43,41,49,53,45,20,4A,4F,4E,54,20,4A,45,20,54,3DD  
 6410 DATA 41,49,52,41,49,20,4C,45,20,4E,4F,4D,20,28,43,3AC  
 6420 DATA 45,20,4E,27,45,53,54,20,50,41,53,20,4C,41,20,397  
 6430 DATA 4D,45,4D,45,20,51,55,45,20,54,4F,55,54,20,41,3FC  
 6440 DATA 20,4C,27,48,45,55,52,45,29,2E,76,20,04,43,45,385  
 6450 DATA 20,4A,45,55,20,53,45,52,41,20,4C,45,20,46,52,388  
 6460 DATA 56,49,54,20,44,55,20,54,52,41,56,41,49,4C,20,3FE  
 6470 DATA 44,45,20,54,4F,55,54,45,20,55,4E,45,20,45,51,3FB  
 6480 DATA 55,49,50,45,20,45,54,20,4D,4F,4E,54,52,45,52,433  
 6490 DATA 41,20,28,3F,29,20,51,55,45,20,4C,45,53,20,4A,36A  
 6500 DATA 45,55,58,20,53,55,52,20,44,49,53,51,55,45,54,44B  
 6510 DATA 54,45,20,50,45,55,56,45,4E,54,20,53,55,52,43,43D  
 6520 DATA 4C,41,53,53,45,52,20,4C,45,53,20,4A,45,55,58,42A  
 6530 DATA 20,53,55,52,20,43,41,52,54,4F,55,43,48,45,20,3FB  
 6540 DATA 21,03,42,20,20,43,45,20,51,55,49,20,44,45,56,33C  
 6550 DATA 52,41,49,54,20,43,4F,4D,02,42,4C,45,52,20,4C,3C2  
 6560 DATA 45,53,20,55,54,49,4C,49,53,41,54,45,55,52,53,466  
 6570 DATA 20,45,54,20,46,41,49,52,45,20,52,45,46,4C,45,3CE  
 6580 DATA 43,48,49,52,20,43,45,52,54,41,49,4E,45,53,20,40A  
 6590 DATA 50,45,52,53,4F,4E,4E,45,53,2E,03,42,76,20,04,3CA  
 6600 DATA 56,4F,49,4C,41,2C,20,4A,45,20,43,52,4F,49,53,3F6  
 6610 DATA 20,51,55,45,20,4A,27,45,4E,20,41,49,20,41,53,38D  
 6620 DATA 53,45,5A,20,44,49,54,20,28,3F,03,42,29,20,45,34D  
 6630 DATA 54,20,53,55,52,20,43,45,20,4A,45,20,56,4F,55,3DF  
 6640 DATA 53,20,4C,41,49,53,53,45,20,45,4E,20,56,4F,55,401  
 6650 DATA 53,20,44,49,53,41,4E,54,20,3A,20,41,20,02,42,355  
 6660 DATA 49,45,4E,54,4F,54,20,4A,27,45,53,50,45,52,45,428  
 6670 DATA 20,21,76,20,19,00,FB,21,03,76,11,40,00,01,00,34F  
 6680 DATA 01,05,ED,00,C9,DD,21,F4,75,21,A4,9B,11,FD,26,837  
 6690 DATA DD,7E,00,FE,42,20,22,DD,7E,FF,FE,02,28,14,05,748  
 6700 DATA 4F,06,00,54,5D,18,DD,7E,FE,77,ED,08,DD,28,DD,77B  
 6710 DATA 2B,D1,18,0C,DD,7E,00,DD,2B,18,03,DD,7E,00,77,570  
 6720 DATA 2B,DD,2B,7C,BA,20,CC,7D,BB,20,C8,3A,00,A7,F5,748  
 6730 DATA 21,FF,AB,11,40,00,CD,CB,BC,F1,32,00,A7,C3,FE,7FB  
 6740 DATA 26,00,37,00,00,00,00,00,00,00,00,00,00,00,00,05D

A SUIVRE!

**SORACOM**  
editions

**Media Box Cassettes Vidéo (L)**

222x135 x348 mm  
Réf. 310.540.0

**155 F + port**  
pour 9 cassettes VHS, V2000, Betamax



**Media Box Mini Cassettes**

148x91x348  
Réf. 310.503.3

**95 F + port**  
pour 16 mini cassettes



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

182x178x348 mm  
Réf. 310.501.1.

**230 F + port**  
pour 70 disquettes, livré avec séparations et index



**Media Box Compact Disc**

148x135x348 mm  
Réf. 310.502.6

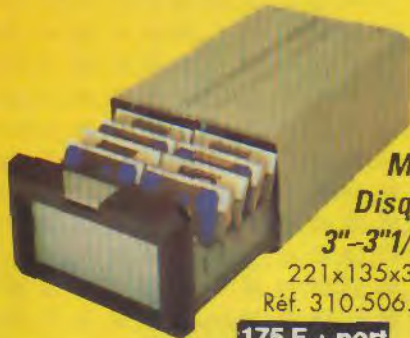
**125 F + port**  
pour 13 compacts discs simples



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

221x135x348 mm  
Réf. 310.506.4

**175 F + port**  
pour 150 disquettes



**Media-Box Data Cartridge**

222x135x348 mm  
Réf. 310.518.7

**155 F + port**  
pour 11 Data Cartridges type 3M



**Media Box Cassettes Vidéo VHS-C**

148x91x348 mm  
Réf. 310.505.7

**102 F + port**  
pour 8 cassettes



**Media Box Cassettes Vidéo 8**

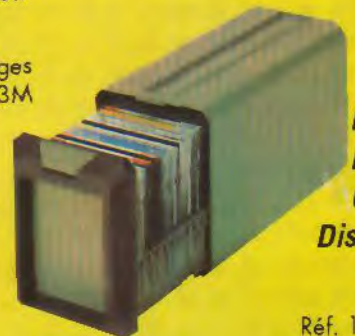
148x91x348 mm  
Réf. 310.531.6

**102 F + port**  
pour 12 cassettes



**Media Box Compact Disc Multi**

148x177 x348 mm  
Réf. 100.525.0  
pour 23 CD simples ou 11 CD doubles



**Media Box Photo**

222x135 x348 mm  
Réf. 310.501.9

**155 F + port**  
pour 140 tirages format maximum 13x18 plus



**Module serrure**

**60 F + port**  
(bien indiquer la boîte)



**Nouveau ! les mêmes en noir**



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

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

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

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

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

**Voir bon de commande page 35**